BlomURBEX Web Services

Blom

Audience: **BLOM partners and developers**

Abstract: This document describes the low level HTTP API for accessing the BlomURBEX services

Date	Blom Document
June 10, 2020	BlomURBEX Web Services API v7.4.5.docx

Notices

Blom expressly retains all intellectual and other property rights with respect to this document and all matters set forth herein.

Some technical assertions of capability included herein are estimates based on limited information gathered from past experience.

The terms, conditions, specifications, and procedures described herein are subject to change in the sole discretion of Blom and its affiliates. End-User is responsible for requesting and obtaining the latest release of these terms, conditions, specifications, and procedures prior to any purchase or deployment of the products described herein.

Confidentiality

This document and the information contained herein is the proprietary and confidential information of Blom. It is provided under contract agreement, and may not be reproduced or used for purposes outside the scope of such agreement.

Trademarks

Blom's logo is a registered trademark of Blom in the Kingdom of Norway and other countries. Other brands and their products are registered trademarks or trademarks of their respective holders and should be noted as such.

Copyrights

Revision History

Document Number	Rev.	Issue Date	Editor	Reason for Change
BUAPI_W_1010	1.0	28 January 2009	Alejandro Martinez	Original Document
BUAPI_W_2010	2.0	3 September 2009	Alejandro Martinez	New Data Availability services. Extended parameters for precise data retreiving
BUAPI_W_3010	3.0	15 February 2010		
BUAPI_W_5010	5.0	19 January 2011	Jorge Alonso	Dtm Layers option added on GetDTM
BUAPI_W_5020	5.0	7 March 2011	Jorge Alonso	POIs services added
BUAPI_W_5030	5.0	15 September 2011	Kuldeep Singh	CalculateVerticalArea service added
BUAPI_W_5040	5.0	20 october 2011	Jorge Alonso	GetRouting, GetTileLiDAR, GetBlomSTREETCredentials, Changes in Geocoding services, changes in POI services
BUAPI_W_5040	5.9	8 March 2012	Jorge Alonso	Vertical Image Services. GetWireframe
BUAPI_W_6010	6.0	May 2012	Jorge Alonso	GetUserAvailableProviders and Proxy services
BUAPI_W_6030	6.0	November 2012	Jose Aladro	GetUserAvailableData GetOrthoInformation now subject to user restrictions
BUAPI_W_6040	6.0	December 2012	Jose Aladro	Fixed ExtractAll service resulting file name
BUAPI_W_6150	6.0	July 2013	Jose Aladro	Text review
BUAPI_W_6210	6.2	September 2013	Jose Aladro	Added separate WMTSService Tile service with multiprojection support
BUAPI_W_6310	6.3	September 2013	Jose Aladro	Added separate WMSService Map service with multiprojection support
BUAPI_W_6311	6.3	September 2013	Jose Aladro	Fixed calculation of pixel offset in oblique tile
BUAPI_W_6311	6.3	October 2013	Jose Aladro	Amended GetRouting servlet parameters in examples.
BUAPI_W_6321	6.3	November 2013	Jose Aladro	Reformatted POIs services

				example responses.
BUAPI_W_645	6.4	March 2015	Alberto del Barrio	Added new parameter for GetUserAvailableData to support overlays, and support for overlays in GetOrthoInformation
BUAPI_W_65	6.7	January 2016	Alberto del Barrio	Added support for automatic login from authorized IPs
BUAPI_W_66	6.7	January 2017	Rafael de la Viuda	Added support for GetTile extra zoom
BUAPI_W_68	6.8	May 2017	Luis Rodriguez	Added Street services.
BUAPI_W_69	6.9	Jul 2017	Luis Rodriguez	Added Oblique Layer services.
BUAPI_W_697	6.9	Oct 2017	Luis Rodriguez	Updated Street Services
BUAPI_W_698	6.9	Nov 2017	Luis Rodriguez	Added WFS Lidar
BUAPI_W_700	7.0	Feb 2018	Luis Rodríguez	Added UserLayer service
BUAPI_W_702	7.0	Jun 2018	Luis Rodríguez	Added Lidar Services
BUAPI_W_702	7.3	May 2019	Luis Rodríguez	Added Extract Full Oblique service
BUAPI_W_740	7.4	July 2019	Luis Rodríguez	Added 3D Mesh services
BUAPI_w_746	7.4.5	June 2020	Luis Rodríguez	Added new parameters, Teg and Street projects services

Table of Contents

1 Scope

This document describes the low level http API for the *Blom Urban Explorer (BlomURBEX)* service.

This API provides a collection of web services allowing full access to the BlomURBEX functionality and imagery for third party developers to use the *BlomURBEX* services in their applications.

The document includes a description of the relevant concepts in the BlomURBEX tiling structure to permit developers make the needed calculations for locate any desired tile in the Blom Imagery Database.

For every service, the document provides a brief description, the list of parameters and examples of use.

1.1 Document organization

The document organizes the different API services into 5 main groups:

- **Vertical and orthorectified oblique images services:** this section describes the organization of images and tiles in the BlomURBEX database and the available functions for accessing them.
- **Oblique image services:** Oblique images have their own rules and structures. Functions available for accessing and working with them are covered in chapter 3.
- **Street image services:** the access to the Street panorama images and information are handled by their own services. Charter 4 covers all functions for the Street images.
- **Measurements on Ortho Images:** BlomURBEX provides many measurement functions. Chapter 5 covers the available functions for vertical and orthorectified oblique images.
- **Measurements on Oblique Images:** Chapter 6 covers the rest of measurements available in BlomURBEX, this time for pure oblique images.
- **3D services:** The access to buildings wireframe and textured models.
- **Lidar services:** Accessing to Lidar data.
- **Other Services:** Finally, chapter 7 covers general functions provided by the BlomURBEX API.

2 Vertical and orthorectified oblique images Services

2.1 Introduction

BlomURBEX serves vertical images as orthophotos, i.e., images with vertical sight in which it is not possible to appreciate the front of buildings, rectified to correct the distortion of the elevation model of area.

Orthorectified oblique images are images taken with an angle of approximately 45º, allowing the viewer to see the fronts of the buildings, transformed and rectified to correct the distortion of the elevation data, so that they represent ortho-images. These images are distorted but allow an easy overlap of maps while still offering the visualization of the facades of the buildings.

All images served by BlomURBEX are available in Spherical Mercator projection (EPSG:3785), and, optionally, also in a local projection. As an example, images from Spain are also available in the EPSG:32630 (UTM 30N WGS84) projection.

There are two types of services to retrieve the ortho imagery.

- A service to retrieve a tile
- A service to obtain an image of a precise extent and size.

BlomURBEX service can be accessed using the classic HTTP protocol or the secure HTTPS protocol.

2.2 Tiles

Tiles are 256x256 pixel images with a specific extent. The earth surface is considered as divided in grids, each one consisting on several squared sized tiles. Different zoom levels imply different grid sizes and so, different tiles.

Each individual tile has a unique code used to identify it uniquely. Given one pair of coordinates and one zoom level only one tile will be covering that point.

Tiles are very useful to allow great performance and to reduce the volume of information sent via the Internet, since a smart client application will be able to request just the tiles needed in a particular moment to represent a particular extent, and can also do some caching of these tiles. The use of file caching allows the user to pan through an area moving back and forward the same tiles without the need to request them several times. This provides fast response times, better user experience, lower bandwidth requirements and puts less stress on the servers (as they are already stored in the servers and there is no need for further processing the data).

These advantages of tiles are achieved with the BlomURBEX web services providing tile access functions. However, there are also image retrieving services. These services require lots of computing power on the server side, as it is the requested image must be

generated from available tiles in the server, cutting and scaling the available. It is not recommended to use image services for zooming and panning as a whole new image must be sent to the server and it cannot be easily cached on the client side to minimize the needed bandwidth.

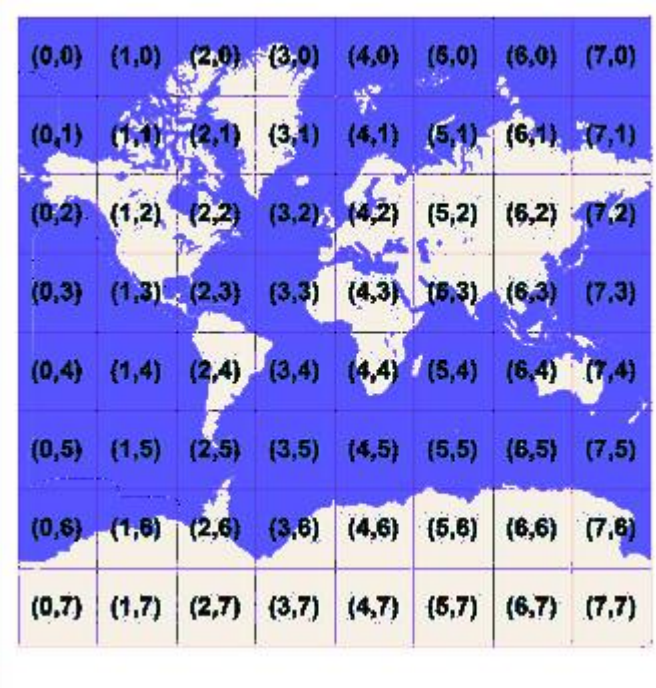
2.3 Tile Coordinates and QuadTree keys

To optimize the performance of images retrieval and display, ortho images are cut into tiles of 256 x 256 pixels each. This tile division is performed at different zoom levels.

All ortho or ortho-rectified images are available in Spherical Mercator Projection. This projection is used because it can be used almost world-wide (poles are not covered) and does not alter the aspect of features, though it scales up areas far from the equator.

For each zoom level, the whole world surface is divided in tiles. For every zoom level the earth surface is divided in ($2^{level} \times 2^{level}$) tiles, each one of them with 256 x 256 pixels. Each tile is given XY coordinates ranging from (0, 0) in the upper left to ($2^{level-1}$, $2^{level-1}$) in the lower right.

EXAMPLE: at level 3 the tile coordinates range from (0, 0) to (7, 7) as shown:



The relationship between World Coordinates (Spherical Mercator coordinates) and XY coordinates (pixel coordinates) is calculated using the formula:

$pixelX = floor\ (\ (x + 20037508.34)\ /\ (156543.0339 * 2^{-level})\ )$

pixelY=floor ((20037508.34 - y) / (156543.0339 * 2 $^{-level}$))

Given a pair of pixel XY coordinates, it's easy to determine the tile XY coordinates of the tile containing that pixel:

tileX = floor(pixelX / 256)

tileY = floor(pixelY / 256)

To optimize the indexing and storage of tiles, the two-dimensional tile XY coordinates are combined into one-dimensional strings called quadtree keys. Each quadtree key uniquely identifies a single tile at a particular zoom level.

To convert tile coordinates into a quadtree key, the bits of the Y and X coordinates are interleaved, and the result is interpreted as a base-4 number (with leading zeros maintained) and converted into a string. For instance, given tile XY coordinates of (7, 5) at level 3, the quadtree key is determined as follows:

tileX = 7 = 111 (binary)

tileY = 5 = 101 (binary)

quadtree key = 110111 (binary) = 313 (base-4) = "313"

The length of a quadtree key (the number of digits) equals the level of detail (zoom level) of the corresponding tile.

The quadtree key of any tile starts with the quadtree key of its parent tile (the containing tile at the previous level).

EXAMPLE: tile 2 is the parent of tiles 20 through 23, and tile 11 is the parent of tiles 110 through 113:

2.4 GetTile Service

2.4.1 Description

This service returns the requested tile.

The tile can be from any of the available views: ortho, north, east, west and south. The ortho view is the one generated from vertical images, which are properly rectified according to the underlying DTM; while for the rest of the views the corresponding oblique images were processed and rectified according to the DTM.

Tiles are returned as **jpg** images.

Besides the aerial images, the system is ready to deliver overlay data. This is rasterized vector data with transparency, which can be displayed on top of the images. This information is delivered as transparent **png** files and is grouped in layers. The layers can have information for one or all of the supported views, so that the information properly matches the different image views.

GetTile service can be configured to server an extra zoom level, resampling last real zoom level tiles. This feature must be enabled in the user profile settings.

2.4.2 Parameters

- **id [mandatory]**: Tile ID, it will be a sequence of digits, which only will be able to contain 0, 1, 2 and 3. This nomenclature corresponds to the way of codifying the Tiles for quadtrees.

- **Layer [mandatory]:** Name of the layer (orientation) to which the requested Tile belongs to. Supported values are:

  - `Ortho: For the ortho images.`
  - `North: For the orthorectified North view images.`
  - `South: For the orthorectified South view images.`
  - `East: For the orthorectified East view images.`
  - `West: For the orthorectified West view images.`

- **crs or srs [mandatory]:** Projection system used. It must follow OGC nomenclature. For example **EPSG:32630** for UTM 30 North WGS84 or **EPSG:3785** for Spherical Mercator projection.

- **Overlay [optional]**: If this parameter is present, instead of returning tiles from the aerial images, it will return tiles from the specified overlay layer. The system has a “generic” overlay layer available, with the streets, roads and main political divisions.

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom

- **time [optional]**: A date following the structure YYYYMMDD or YYYYMM or YYYY. The service will use the most current images available, whose date is previous to the requested date. It is also possible to use a range of dates with the structure YYYYMMDD-YYYYMMDD, in this case the images used will be the most current ones available in the specified date range.

  If the parameter is omitted the system will submit the most recent version of the tile.

- **BaseLayer [optional]**: If this parameter is present, instead of returning tiles from the aerial images, it will return tiles from the specified baselayer.

- **ExtraZoom [optional]**: This parameter allows disable extra zoom feature for current request. This behavior can be useful to confirm if current is real or resampled tile.

2.4.3 Results

The result of the request will be a JPG image. If the user is not authorized to obtain the requested data or if the requested tile is not available, then a transparent PNG file 256 x 256 pixels will be returned.

2.4.4 Example

```
http://www.blomurbex.com/v02/GetTile?usertoken=********&id=033111001210321200&
crs=epsg:3785&layer=ortho
```

In the example above, tile 033111001210321200 from the ortho image from the images collection with Spherical Mercator coordinate system is requested.

2.4.5 GetTile request with GetCapabilities:

To obtain the Xml file with the description of the GetTile WMTS Web Service, the request parameter must be set to "**GetCapabilities**". It will also be necessary to set the usertoken and service parameters.

Here is an example of a GetCapabilities request:

```
http://www.blomurbex.com/v02/GetTile?usertoken=********&service=WMTS&request=GetCapabilities
```

The result would be an Xml file with the capabilities of the WMS like the one shown below.

```
−<Capabilities xsi:schemaLocation="http://www.opengis.net/wmts/1.0 http://schemas.opengis.net/wmts/1.0/wmtsGetCapabilities_response.xsd" version="1.0.0">
    <!-- GetTile Capabilities -->
  −<ows:ServiceIdentification>
      <ows:Title>BlomURBEX Web Map Tile Service</ows:Title>
    −<ows:Abstract>
        BlomURBEX WMTS service, delivering all BlomURBEX orthogonal and ortho rectified data
      </ows:Abstract>
    −<ows:Keywords>
        <ows:Keyword>BlomURBEX</ows:Keyword>
        <ows:Keyword>Blom</ows:Keyword>
        <ows:Keyword>Images</ows:Keyword>
      </ows:Keywords>
      <ows:ServiceType>OGC WMTS</ows:ServiceType>
      <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
      <ows:Fees>Contact Blom Sales Representative</ows:Fees>
      <ows:AccessConstraints>Usertoken. Contact Blom Sales Representative.</ows:AccessConstraints>
    </ows:ServiceIdentification>
  +<ows:ServiceProvider></ows:ServiceProvider>
  −<ows:OperationsMetadata>
    −<ows:Operation name="GetCapabilities">
      −<ows:DCP>
        −<ows:HTTP>
          +<ows:Get xlink:href="http://prepro.blomnls.com/v02/GetTile?usertoken=tkbl2012"></ows:Get>
          +<ows:Post xlink:href="http://prepro.blomnls.com/v02/GetTile?usertoken=tkbl2012"></ows:Post>
          </ows:HTTP>
        </ows:DCP>
      </ows:Operation>
    −<ows:Operation name="GetTile">
      −<ows:DCP>
        −<ows:HTTP>
          +<ows:Get xlink:href="http://prepro.blomnls.com/v02/GetTile?usertoken=tkbl2012"></ows:Get>
          +<ows:Post xlink:href="http://prepro.blomnls.com/v02/GetTile?usertoken=tkbl2012"></ows:Post>
          </ows:HTTP>
        </ows:DCP>
      </ows:Operation>
    </ows:OperationsMetadata>
  −<Contents>
    −<Layer>
        <ows:Title>BlomURBEX ORTHO EPSG:3785</ows:Title>
      −<ows:WGS84BoundingBox>
          <ows:LowerCorner>-179.99000549316406 -85</ows:LowerCorner>
          <ows:UpperCorner>179.99000549316406 85</ows:UpperCorner>
        </ows:WGS84BoundingBox>
        <ows:Identifier>BlomURBEX:ORTHO</ows:Identifier>
      −<Style isDefault="true">
          <ows:Identifier>_null</ows:Identifier>
        </Style>
        <Format>image/jpeg</Format>
```

2.5 GetMap Service

2.5.1 Description

This service implements a WMS (Web Map Service), a service for publishing cartographic data on the Internet that follows the directives and OGC's regulation.

The service gets a request from a client specifying the extent and size of the mandatory image and returns the requested raster image. This raster image can be the ortho view or any of the ortho-rectified oblique views.

The service requires an http request with a few parameters, normalized by the OGC, which define the request. These parameters are: the service, the version, the type of request, the system of spatial reference, the coordinates of the area that define the spatial area of the request, the size of the image of response, the format of image, the layers, the transparency, etc.

Supported versions of the WMS protocol in BlomURBEX are: WMS 1.1.1. and WMS 1.3.0. Refer to the WMS specification to obtain more information about the obligatory and optional parameters of this type of requests:

http://www.opengeospatial.org/standards/wms

The GetMap service returns the image corresponding to an orthogonal view or any of the orthorectified oblique views, if the **request** parameter is set to **GetMap**

If the parameter request is set to **GetCapabilities**, the service will return the web descriptor xml file describing the WMS service.

2.5.2 Parameters

- **version [mandatory]**: This parameter is the WMS version that is used for this request. Only two values are accepted in the current API version:
  - 1.1.1
  - 1.3.0.

- **request [mandatory]:** The type of service GetMap. Possible values are:
  - GetMap. Used for retrieving an image/map.
  - GetCapabilities used to obtain the web services descriptor xml file.

- **layers [mandatory]:** Identifies which view the user is requesting. Possible values:
  - Ortho: For the ortho images.
  - North: For the orthorectified North view images.
  - South: For the orthorectified South view images.
  - East: For the orthorectified East view images.
  - West: For the orthorectified West view images.

- **styles [mandatory]:** Parameter mandatory for WMS Standard, it must be present but its value is ignored by Blom Urbex.

- **crs or srs [mandatory]:** Projection system used, It must follow OGC nomenclature. For example **EPSG:4326** for WGS84 LATLON or **EPSG:3785** for Spherical Mercator projection.

- **bbox [mandatory]:** requested extent, following the rules of the WMS Standard: Xmin, Ymin, Xmax, Ymax

- **width [mandatory]:** Width of the requested image in pixels.

- **height [mandatory]:** Height of the requested image in pixels..

- **format [mandatory]:** Image format. The only supported format is: **image/jpeg**

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

- **time [optional]** : A date following the structure YYYYMMDD or YYYYMM or YYYY. The service will use the most current images available, whose date is previous to the requested date. It is also possible to use a range of dates with the structure YYYYMMDD-YYYYMMDD, in this case the images used will be the most current ones available in the specified date range.

  If the parameter is omitted the system will submit the most recent version of the tile.

- **Service[mandatory]**:This parameter is mandatory in the request if the **request** parameter is set to **GetCapabilities**, and it will have to take the value: **WMS.**

- **BaseLayer [optional]**: If this parameter is present, instead of returning tiles from the aerial images, it will return tiles from the specified baselayer.

2.5.3 Result

The result of the request will be a JPG image. If the user is not authorized to obtain the requested data or if the requested tile is not available, an JPG file with the text "unauthorized" is returned.

2.5.4 Example

Example of a GetMap request:

```
http://www.blomurbex.com/v02/GetMap?usertoken=********&version=1.1.1&request=GetMap&srs=EPSG:3785&bbox=-523000,4960000,-521500,4961500&width=550&height=550&layers=ortho&format=image/jpeg&styles=
```

The result of the previous request is an image like the one shown below:

2.5.5 GetMap request with GetCapabilities:

To obtain the Xml file with the description of the GetMap Web Service (WMS), the request parameter must be set to "**GetCapabilities**". It will also be necessary to set the usertoken, service and version parameters.

Here is an example of a GetCapabilities request:

```
http://www.blomurbex.com/v02/GetMap?usertoken=********&service=WMS&version=1.3.0&request=GetCapabilities
```

The result would be an Xml file with the capabilities of the WMS like the one shown below.


```
−<WMS_Capabilities version="1.3.0" xsi:schemaLocation="http://www.opengis.net/wms http://schemas.opengis.net/wms/1.3.0/capabilities_1_3_0.xsd">
   <!-- GetMap Capabilities -->
  −<Service>
     <Name>WMS</Name>
     <Title>Blom</Title>
     <Abstract>Blom Map Image Service</Abstract>
    +<KeywordList></KeywordList>
     <OnlineResource xlink:href="http://www.blomurbex.com" xlink:type="simple"/>
    +<ContactInformation></ContactInformation>
    +<AccessConstraints></AccessConstraints>
   </Service>
  −<Capability>
    −<Request>
      +<GetCapabilities></GetCapabilities>
      +<GetMap></GetMap>
     </Request>
    +<Exception></Exception>
    −<Layer>
       <Title>BlomURBEX</Title>
      −<Layer>
         <Title>BlomURBEX Maps WGS84</Title>
         <Abstract>BlomURBEX Maps imagery, EPSG:4326 projection</Abstract>
        −<Layer queryable="0" opaque="0" cascaded="0" noSubsets="0">
           <Name>ortho_wgs84</Name>
           <Title>BlomURBEX ORTHO EPSG:4326 (WMS)</Title>
           <Abstract>BlomURBEX ORTHO imagery, EPSG:4326 projection</Abstract>
          −<KeywordList>
             <Keyword>Blom</Keyword>
             <Keyword>Images</Keyword>
           </KeywordList>
           <CRS>EPSG:4326</CRS>
          −<EX_GeographicBoundingBox>
             <westBoundLongitude>-179.99000549316406</westBoundLongitude>
             <southBoundLatitude>-85</southBoundLatitude>
             <eastBoundLongitude>179.99000549316406</eastBoundLongitude>
             <northBoundLatitude>85</northBoundLatitude>
           </EX_GeographicBoundingBox>
           <BoundingBox CRS="EPSG:4326" minx="-85" miny="-179.99000549316406" maxx="85" maxy="179.99000549316406" resx="1" resy="1"/>
         </Layer>
        −<Layer queryable="0" opaque="0" cascaded="0" noSubsets="0">
           <Name>north_wgs84</Name>
           <Title>BlomURBEX NORTH EPSG:4326 (WMS)</Title>
           <Abstract>BlomURBEX NORTH imagery, EPSG:4326 projection</Abstract>
          −<KeywordList>
             <Keyword>Blom</Keyword>
             <Keyword>Images</Keyword>
           </KeywordList>
           <CRS>EPSG:4326</CRS>
```

2.6 GetOrthoInformation Service

2.6.1 Description

This service is intended to provide copyright and date information about ortho images and overlays.

The service can be asked to share the information of the ortho images or overlays covering a certain ground extent or a single point. When requested, the service will publish the copyright, minimum date and maximum date when the images were taken for the ortho images covering that area/point meeting the criteria specified with the parameters of the request.

Finally, results will be filtered by user date restrictions.

If no information is available, this service will deliver an error notice saying that no data is available for the request.

2.6.2 Parameters

- **layers [mandatory]:** Identifies which view the user is requesting. Possible values:
    - Ortho: For the ortho images.
    - North: For the orthorectified North view images.
    - South: For the orthorectified South view images.
    - East: For the orthorectified East view images.
    - West: For the orthorectified West view images.

- **crs or srs [mandatory]:** Projection system used, It must follow OGC nomenclature. For example **EPSG:4326** for WGS84 LATLON or **EPSG:3785** for Spherical Mercator projection.

- **x [mandatory if requesting information in a point]:** X coordinate of the point of which the information is being requested, on the projection system indicated by crs/srs.

- **y [mandatory if requesting information in a point]:** Y coordinate of the point of which the information is being requested, on the projection system indicated by crs/srs.

- **zoom[mandatory if requesting information in a point]:** Zoom level of the ortho images of which the information is being requested

- **xmax [mandatory if requesting information in a extent]:** X coordinate of the highest point that defines the extent of which the information is being requested.

- **ymax [mandatory if requesting information in a extent]:** Y coordinate of the highest point that defines the extent of which the information is being requested.

- **xmin [mandatory if requesting information in a extent]:]:** X coordinate of the lowest point that defines the extent of which the information is being requested.

- **ymin [mandatory if requesting information in a extent]:]:** Y coordinate of the lowest point that defines the extent of which the information is being requested.

- **width [mandatory if requesting information in a extent]:** Width of the requested image in pixels. It will be used to calculate the best suited zoom for the image on that extent.

- **height [mandatory if requesting information in a extent]:** Height of the requested image in pixels. It will be used to calculate the best suited zoom for the image on that extent.

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

- **time [mandatory]** : A date following the structure YYYY to specify one year or YYYY-YYYY to specify a range of years. If a range is specified, the service will look for information of the ortho images taken in that range of years. If only a year is specified, the service will look for information of ortho images taken before that year.

- **baseLayer [optional]**: String indicating the baselayer preference for the request, formatted as *baselayer1,baselayer2,baselayer3*. If not set, default baselayer preference shall be *pictometry,countryortho,satellite*.

- **overlay[optional]:** If requesting information about an overlay image, this parameter would indicate the name of the overlay of which the information is being requested.

- **format [optional]:** Format in which the service will send the response. Allowed formats are: txt (Text), xml and js (Javascript). If no format is set, xml format will be default format.

- **function[optional]:** If the response format requested if Javascript (format=js), this parameter indicates the name of the Javascript function which will wrap the response. The response would be as follows:
  function nameOfFunction {
  return result;
  }

2.6.3 Result

The result of the request will be 3 Strings indicating the images copyright, the minimum date when the images were taken and the maximum date when the images were taken. Depending on the format chosen in the parameters of the request, the result will be presented on 3 different ways:

- **format=xml:** The result will be the following xml:

  <result>

  <copyright>copyright</copyright>

  <dateMin>date</dateMin>

  <dateMax>date</dateMax>

  </result>

- **format=txt:** The result would be a string made by concatenating the copyright, dateMin and dateMax strings separated by ",".

- **format=js:** If no function has been set on the request, the response would create a variable called Urbex.Var.GetOrthoInformation, which value will be set to an Array containing the three results:

  Urbex.Var.GetOrthoInformation = new Array(copyright, datemax, datemin);

  If, on the other hand, a function has been specified on the request, the response sent by the server would be as follows:

  Function nameOfFunction(){
  return new Array(copyright, datemax, datemin);
  }

  This format is for internal use of the Javascript API

2.6.4 Example

Example of a GetOrthoInformation request (requesting by point):

```
http://www.blomurbex.com/v02/GetOrthoInformation?usertoken=********&srs=EPSG:3785&x=1389525.984992505&y=5145333.536935514&layer=NORTH&zoom=17&time=&baselayer=&function=myfunction&format=js
```

The response to the previous request the following:

```
function myfunction() {
```

```
 return new Array ('Blom-Pictometry','2008-07-11
00:00:00','2008-09-17 00:00:00');
}
```

Another example, requesting by extent and with a different format:

http://www.blomurbex.com/v02/GetOrthoInformation?**usertoken**=********&**srs**=EPSG:3785&**xmin**=1389525.984992505&**ymin**=5145333.536935514&**xmax**=1389530&**ymax**=5145340&**height**=400&**width**=500&**layer**=NORTH&**zoom**=17&**time**=2009-2006&**format**=xml

The response to this request will be:

```
<result>
  <copyright>Blom-Pictometry</copyright>
  <dateMin>2008-07-11 00:00:00</dateMin>
  <dateMax>2008-09-17 00:00:00</dateMax>
</result>
```

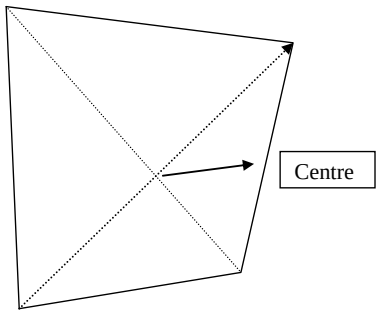
3 Oblique images services

3.1 Overview

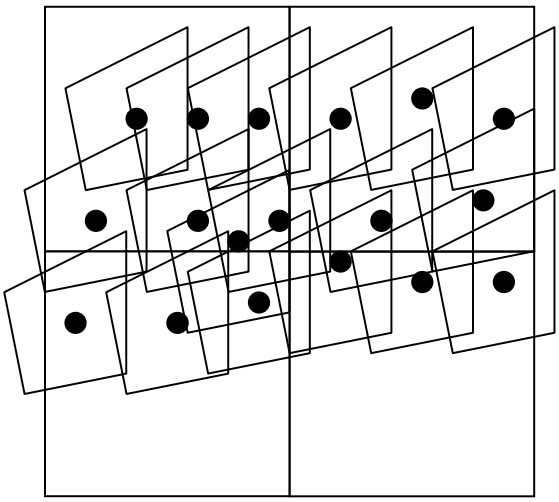
Oblique images are obtained with an inclination of approx. 45º allowing the users to see the facades of the buildings. These images are not ortho images and need to be treated specially.

When an area has oblique images available, it normally has 4 available views for every point: North, East, South and West. And more than one image of any view could cover one particular point, since the images have some overlap between then.

Every oblique picture has a centre. The coordinates of this point serve as an identifier to every image, and will also be used to select, among the different images covering a particular point, which one will be the best option for that point. The best image will be the one with its centre closer to the point as shown below.

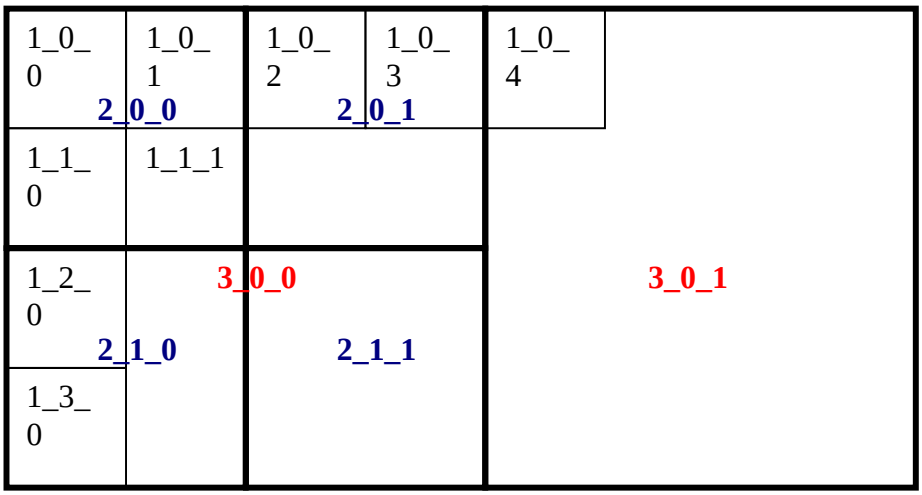


Example of coverage of a set of images:



For a better performance each particular oblique image is cut into tiles 256x256 pixels in size. These tiles are available at several resolutions: Maximum Image resolution, ½ resolution, ¼ resolution and 1/8 resolution. These different scales are called **zoom levels** and are referenced as: 1, 2, 3 and 4 in the oblique images handling services.

Next picture shows an example of how the oblique images are cut into tiles. The code used to identify the tiles has 3 parts: the **zoom level**, and a pair of **x**, **y** coordinates of the tiles referenced to the upper left corner of the image.



3.1.1 Locating a tile

To find the tile where a particular pixel (of the original oblique image) is located at a given zoom level, this formula can be used:

```
tileX = floor(pixelX / (256 * 2 level-1))

tileY = floor(pixelY / (256 * 2 level-1))
```

And the position of the original pixel within the tile:

```
Xoffset = (pixelX – tileX * 256 * 2 level-1) / (256 * 2 level-1)

Yoffset = (pixelY – tileY * 256 * 2 level-1) / (256 * 2 level-1)
```

All pixel coordinates are referred to the top-left corner of the images.

3.1.2 Image identifier

The identifier of any oblique image, follows this structure:

OXXXX.XXXXXX_YY.YYYYYY_YYMMDD

Where:

The first character (O) indicates the orientation of the image, being:

- N- North
- S-South
- E- East
- W-West

Characters from position 2 to 12 **(XXXX.XXXXXX)** represent the longitude of the photo center in LAT/LON WGS84.

The characters from position 14 to 22 **(YY.YYYYYY)** represent the latitude of the centre of the photo in LAT/LON WGS84.

The final 6 characters **(YYMMDD)** represent the date in which the photography was taken, (YY=year, MM=month, DD=day).

Example:

The image with this Id:

N-004.697670_40.655723_060830

has its center located at Longitude -4.697670, Latitude 40.655723 and was taken on August, 30; 2006.

3.2 BlomURBEX oblique services

To be able to request and manipulate oblique imaginary several services are available in BlomURBEX:

- **FindOblique:** This is the most basic service in the managing of oblique imagery. This service identifies the best oblique image for a given set of coordinates, i.e., the image which centre is closer to the supplied coordinates.

  The result of the **FindOblique** service is a plain text or XML file, identifying the best available oblique and the pixel in that image that represents the point of interest. It also provides the full size of the oblique image and the ground coordinates of the four corners of the oblique image.

- **FindObliques**: This service is similar to the previous one, but instead of returning information about the Best oblique covering the point, it will offer information about ALL the oblique images that cover the provided point.

- **FindObliquesByExtent:** This service returns all obliques which centres are located into an area passed to the service.

- **GetObliqueTile:** The aforementioned services make it possible to find the oblique image or images that cover a given coordinate. The **GetObliqueTile** service allows obtaining the tiles that compose this oblique image. Combining these services it is possible to identify the tile covering a given point for any desired zoom level.

  The client application must implement the logic to decide which tiles to request to show the desired area for the requests obtained via its user interface.

- **ExtractOblique:** This service allows recovering a portion of an oblique image without considering the existing tiles.

  This service will return a portion of an oblique image, centered in a precise point at a determined zoom level, and of a given size.

- **TransformPoints:** This service allows transforming world coordinates to pixel coordinates on the oblique images and vice versa.

- **ProjectCoordinates**: This service transforms a list of coordinates from one projection system to another one.

- **GetObliqueMetadata:** This service allows getting all available metadata for a particular Oblique Image. This information includes date and time of image capture, camera position and parameters, the original library where the image comes from, and others.

  All these functions are detailed in the following sections.

3.3 FindOblique Service

3.3.1 Description

This service will locate the best oblique image that covers a given point.

3.3.2 Parameters

The parameters needed in the request URL are the following:

- **x [mandatory]**: Coordinate x or longitude in the specified projection system.

- **y [mandatory]**: Coordinate y or latitude in the specified projection system.

- **crs or srs [mandatory]:** Projection system used, It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.

- **orient [Optional]**: Parameter that specifies which is the desired orientation of the searched image. The possible values for this parameter are:

  - **N** North Orientation.
  - **S** South Orientation.
  - **E** East Orientation.
  - **W** West Orientation.
  - ***** Signals that all orientations must be considered. It will return the best oblique for every orientation.

  If this parameter is not present, yaw must be used.

- **yaw [Optional]**: Parameter that specifies orientation angle of the oblique image in degrees between 0 and 360. It returns the oblique image with the nearest angle. If this parameter is not present, orient must be used.

- **format [mandatory]:** Response format. Possible values are:

  - **txt:** Response is a text string with comma-separated values (“,”).
  - **Xml:** Response is an XML file.
  - **Json:** Response in Json format.
  - **js:** Response is a JavaScript function that will return the value. This is for internal use of the JavaScript API. Requires the **function** parameter to specify the name of the function to return.

- **time [optional]** : A date following the structure YYYYMMDD or YYYYMM or YYYY. The service will use the most current images available, whose date is previous to the requested date. It is also possible to use a range of dates with the structure YYYYMMDD-YYYYMMDD, in this case the images used will be the most current ones available in the specified date range. If the parameter is omitted the system will submit the most recent version of the tile.

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom

- **layer [optional]:** Identifies which oblique layer the user is requesting. Possible values are obtained with the GetUserAvailableObliqueLayers service.

3.3.3 Error messages

If no images are found for the specified point an Error message is returned.

If the response format was specified as a text, the response message is:

```
ERROR, Error message text
```

If the response format is XML the error XML file is:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<result>
	<Error>
	ERROR, Error message text
	</Error>
</result>
```

3.3.4 Results

For each oblique image located the following information is provided:

- The oblique ID
- Position X, pixel coordinate in the image
- Position Y, pixel coordinate in the image
- The number of columns of the photography
- The number of rows.
- Upper-left corner x coordinate
- Upper-left corner y coordinate
- Upper-right corner x coordinate
- Upper-right corner y coordinate
- Lower-left corner x coordinate
- Lower-left corner y coordinate
- Lower-right corner x coordinate
- Lower-right corner y coordinate

If **Layer** parameter is present on the result will include the layer name.

In case of Json format, the output response will include the next parameters:

- Yaw: Oblique shot bearin angle in degrees
- copyright: Image copyright information
- tegid: TEG Model identifier for delegate client geo-referencing.
- epsg: EPSG code of image Externa Orientation projection
- has_teg: TEG Model availability
- oblparams: EO & CC params for delegate client geo-referencing [camera longitude, camera latitude, camera altitude, ground elevation, focal length, focal plane width, focal plane height, principal point x, principal point y, rotation matrix 9 params and internal affine transform 6 params]

3.3.5 Examples

3.3.5.1 FindOblique of a given orientation with plain text output

The http request for searching the best oblique image for only one orientation would be like this:

```
http://www.blomurbex.com/v02/FindOblique?usertoken=********&x=-
523000&y=4961500&crs=EPSG:3785&orient=N&format=txt
```

The response for this request is a comma-separated text string like the one shown below:

```
N-004.697670_40.655723_060830,1755.5,1973.5,4008,2672,-
523426.5274033113,4962294.095442287,-
522473.90464922023,4962036.324639748,-
523329.81442120374,4961408.951368517,-
522625.86237766274,4961300.148025963
```

The following table details the meaning of each field in the example. Data is provided as described in section 3.3.4

N-004.697670_40.655723_060830	Oblique image Id
1775.5	Pixel coordinate X in the image
1973.5	Pixel coordinate Y in the image
4008	Number of columns of the oblique image
2672	Number of rows of the oblique image
-523426.5274033113	Upper-Left corner, X coordinate
4962294.095442287	Upper-Left corner, Y coordinate
-522473.90464922023	Upper-Right corner, X coordinate
4962036.324639748	Upper-Right corner, Y coordinate
-523329.81442120374	Lower-Left corner, X coordinate
4961408.951368517	Lower-Left corner, Y coordinate

-522625.86237766274	Lower-Right corner, X coordinate
4961300.148025963	Lower-Right corner, Y coordinate

For details on the oblique image coding refer to section 3.1.2.

The coordinates of the four corners of the image are provided in the reference system specified in the request.

3.3.5.2 FindOblique of all orientations with Xml output

If the parameter **orient** in the request is set to * (**orient**=*), the service will search all available orientations in the images database for the best oblique covering the provided coordinates. The response will contain the data specified in section 3.3.4.

The request will be like this:

```
http://www.blomurbex.com/v02/FindOblique?usertoken=********&x=-
523000&y=4961500&crs=EPSG:3785&orient=*&format=xml
```

The result of this request will be a XML file like the one shown below:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<result>
    <metadata>
       <getIdOblique>N-004.697670_40.655723_060830</getIdOblique>
       <PositionX>1755.5</PositionX>
       <PositionY>1973.5</PositionY>
       <ImageCols>4008</ImageCols>
       <ImageRows>2672</ImageRows>
       <ulx>-523426.5274033113</ulx>
       <uly>4962294.095442287</uly>
       <urx>-522473.90464922023</urx>
       <ury>4962036.324639748</ury>
       <llx>-523329.81442120374</llx>
       <lly>4961408.951368517</lly>
       <lrx>-522625.86237766274</lrx>
       <lry>4961300.148025963</lry>
    </metadata>
    <metadata>
       <getIdOblique>S-004.700164_40.654708_060830</getIdOblique>
       <PositionX>972.5</PositionX>
       <PositionY>1126.5</PositionY>
       <ImageCols>4008</ImageCols>
       <ImageRows>2672</ImageRows>
       <ulx>-522667.7192075137</ulx>
       <uly>4960984.548107109</uly>
```

```
      <urx>-523674.94480635994</urx>
      <ury>4961048.957347591</ury>
      <llx>-522878.9537855673</llx>
      <lly>4961914.679733773</lly>
      <lrx>-523584.0503938487</lrx>
      <lry>4961912.587994628</lry>
    </metadata>
    <metadata>
      <getIdOblique>E-004.699949_40.654537_060830</getIdOblique>
      <PositionX>2152.5</PositionX>
      <PositionY>754.5</PositionY>
      <ImageCols>4008</ImageCols>
      <ImageRows>2672</ImageRows>
      <ulx>-522560.24126329226</ulx>
      <uly>4962178.212536203</uly>
      <urx>-522473.9035137614</urx>
      <ury>4960969.043738725</ury>
      <llx>-523695.6549849898</llx>
      <lly>4961898.854822863</lly>
      <lrx>-523616.3722529034</lrx>
      <lry>4961082.257706512</lry>
    </metadata>
    <metadata>
      <getIdOblique>W-004.698844_40.652942_060830</getIdOblique>
      <PositionX>3058.5</PositionX>
      <PositionY>1561.5</PositionY>
      <ImageCols>4008</ImageCols>
      <ImageRows>2672</ImageRows>
      <ulx>-523509.75205875415</ulx>
      <uly>4960773.752004366</uly>
      <urx>-523558.5320926406</urx>
      <ury>4961725.361645324</ury>
      <llx>-522717.0973721561</llx>
      <lly>4960961.399391948</lly>
      <lrx>-522747.85638348374</lrx>
      <lry>4961664.72391461</lry>
    </metadata>
  </result>
```

3.4 FindObliques Service

3.4.1 Description

This service returns information of al oblique images covering a given coordinates.

3.4.2 Parameters

Parameters of the service are described below:

- **x [mandatory]**: Coordinate x or longitude in the specified projection system.

- **y [mandatory]**: Coordinate y or latitude in the specified projection system.

- **crs or srs [mandatory]:** Projection system used, It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.

- **orient [mandatory]**: Parameter that specifies which is the desired orientation of the searched image. The possible values for this parameter are:

  - **N** North Orientation.
  - **S** South Orientation.
  - **E** East Orientation.
  - **W** West Orientation.
  - ***** Signals that all orientations must be considered. It will return the best oblique for every orientation.

  If this parameter is not present, yaw must be used.

- **yaw [Optional]**: Parameter that specifies orientation angle of the oblique image in degrees between 0 and 360. It returns the oblique image with the nearest angle. If this parameter is not present, orient must be used.

- **format [mandatory]:** Response format. Possible values are:

  - **txt:** Response is a text string with comma-separated values (“,”).
  - **Xml:** Response is an XML file.
  - **Json:** Response in Json format.
  - **js:** Response is a JavaScript function that will return the value. This is for internal use of the JavaScript API. Requires the **function** parameter to specify the name of the function to return.

- **time [optional]** : A date following the structure YYYYMMDD or YYYYMM or YYYY. The service will use the most current images available, whose date is previous to the requested date. It is also possible to use a range of dates with the structure YYYYMMDD-YYYYMMDD, in this case the images used will be the most current ones available in the specified date range. If the parameter is omitted the system will submit the most recent version of the tile.

- **layer [optional]:** Identifies which oblique layer the user is requesting. Possible values are obtained with the GetUserAvailableObliqueLayers service.

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom

3.4.3 Error messages

If no images are found for the specified point an Error message is returned.

If the response format was specified as a text, the response message is:

```
ERROR, Error message text
```

If the response format is XML the error XML file is:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<result>
	<Error>
	ERROR, Error message text
	</Error>
</result>
```

3.4.4 Results

For each oblique image located the following information is provided:

- The oblique ID
- Position X, pixel coordinate in the image
- Position Y, pixel coordinate in the image
- The number of columns of the photography
- The number of rows.
- Upper-left corner x coordinate
- Upper-left corner y coordinate
- Upper-right corner x coordinate
- Upper-right corner y coordinate
- Lower-left corner x coordinate
- Lower-left corner y coordinate
- Lower-right corner x coordinate
- Lower-right corner y coordinate

If Layer parameter is present, the result will include the layer name at the end of the list.

In case of Json format, the output response will include the next parameters:

- Yaw: Oblique shot bearin angle in degrees
- copyright: Image copyright information
- tegid: TEG Model identifier for delegate client geo-referencing.
- epsg: EPSG code of image Externa Orientation projection
- has_teg: TEG Model availability
- oblparams: EO & CC params for delegate client geo-referencing [camera longitude, camera latitude, camera altitude, ground elevation, focal length, focal plane width, focal plane height, principal point x, principal point y, rotation matrix 9 params and internal affine transform 6 params]

3.4.5 Examples

3.4.5.1 FindObliques in a given orientation with XML output

Following is an example of a request for searching all oblique images covering a given point, in a given orientation:

```
http://www.blomurbex.com/v02/FindObliques?usertoken=********&x=-
523000&y=4961500&crs=EPSG:3785&orient=N&format=xml
```

The result of the request would be like this:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<result>
<metadata>
	<getIdOblique>N-004.697670_40.655723_060830</getIdOblique>
	<PositionX>1755.5</PositionX>
	<PositionY>1973.5</PositionY>
	<ImageCols>4008</ImageCols>
	<ImageRows>2672</ImageRows>
	<ulx>-523426.5274033113</ulx>
	<uly>4962294.095442287</uly>
	<urx>-522473.90464922023</urx>
	<ury>4962036.324639748</ury>
	<llx>-523329.81442120374</llx>
	<lly>4961408.951368517</lly>
	<lrx>-522625.86237766274</lrx>
	<lry>4961300.148025963</lry>
</metadata>
<metadata>
	<getIdOblique>N-004.698549_40.653011_060830</getIdOblique>
	<PositionX>2209.5</PositionX>
	<PositionY>589.5</PositionY>
	<ImageCols>4008</ImageCols>
	<ImageRows>2672</ImageRows>
	<ulx>-523574.4616112104</ulx>
	<uly>4961819.393668511</uly>
```

```
        <urx>-522578.85866889096</urx>
        <ury>4961701.5386721995</ury>
        <llx>-523413.3038055277</llx>
        <lly>4960965.542452324</lly>
        <lrx>-522667.1530365837</lrx>
        <lry>4960928.2444666</lry>
</metadata>
<metadata>
        <getIdOblique>N-004.696866_40.657295_060830</getIdOblique>
        <PositionX>1133.5</PositionX>
        <PositionY>2631.5</PositionY>
        <ImageCols>4008</ImageCols>
        <ImageRows>2672</ImageRows>
        <ulx>-523425.14571692346</ulx>
        <uly>4962492.926705989</uly>
        <urx>-522340.4629477161</urx>
        <ury>4962564.940425159</ury>
        <llx>-523207.1889110433</llx>
        <lly>4961479.661125997</lly>
        <lrx>-522465.6373291253</lrx>
        <lry>4961538.232155216</lry>
</metadata>
</result>
```

3.4.5.2 FindObliques in all orientations with plain text output

To find oblique images in all orientations the parameter **orient** must be set to * (**orient**=*).

A request to the service will be like this:

http://www.blomurbex.com/v02/FindObliques?**usertoken**=********&**x**=-523000&**y**=4961500&**crs**=EPSG:3785&**orient**=*&**format**=txt

The result of this request will be a set of text strings, one per oblique, with the format described in section 3.3.5.1:

```
N-004.697670_40.655723_060830,1755.5,1973.5,4008,2672,-
523426.5274033113,4962294.095442287,-522473.90464922023,4962036.324639748,-
523329.81442120374,4961408.951368517,-522625.86237766274,4961300.148025963

N-004.698549_40.653011_060830,2209.5,589.5,4008,2672,-
523574.4616112104,4961819.393668511,-522578.85866889096,4961701.5386721995,-
523413.3038055277,4960965.542452324,-522667.1530365837,4960928.2444666
```

E-004.699949_40.654537_060830,2152.5,754.5,4008,2672,-
522560.24126329226,4962178.212536203,-522473.9035137614,4960969.043738725,-
523695.6549849898,4961898.854822863,-523616.3722529034,4961082.257706512

S-004.700164_40.654708_060830,972.5,1126.5,4008,2672,-
522667.7192075137,4960984.548107109,-523674.94480635994,4961048.957347591,-
522878.9537855673,4961914.679733773,-523584.0503938487,4961912.587994628

W-004.698910_40.656425_060830,509.5,1425.5,4008,2672,-
523531.20712062466,4961264.224428313,-523578.1495146209,4962220.344932538,-
522697.5112866124,4961469.359169706,-522770.2052072819,4962161.49470831

E-004.695852_40.654117_060830,1841.5,1947.5,4008,2672,-
522068.35557704,4962078.24306289,-522092.99386427744,4960901.043569278,-
523205.5714945019,4961857.430561183,-523167.013484177,4961062.175241458

N-004.696866_40.657295_060830,1133.5,2631.5,4008,2672,-
523425.14571692346,4962492.926705989,-522340.4629477161,4962564.940425159,-
523207.1889110433,4961479.661125997,-522465.6373291253,4961538.232155216

E-004.704493_40.654483_060830,2111.5,178.5,4008,2672,-
523081.07440313866,4962136.697168496,-522978.0803189505,4960919.838329274,-
524171.6384148154,4961894.440223184,-524151.37341396126,4961057.608761

E-004.704481_40.651329_060830,660.5,158.5,4008,2672,-
523118.3537977922,4961677.138792106,-522931.09006870526,4960468.237515023,-
524184.3825595521,4961412.610506133,-524134.08911492454,4960579.399779418

3.5 FindObliquesByExtent Service

3.5.1 Description

This service will locate all the oblique photos whose centers are located inside of a given extent.

3.5.2 Parameters

The parameters needed in the request URL are the following:

- **bbox [mandatory]**: Bounding Box defined by four coordinates that specify the area in the specified projection system. Coordinates are provided in the following order: minimum X coordinate, minimum Y coordinate, maximum X coordinate, maximum Y coordinate.

- **crs or srs [mandatory]:** Projection system used, It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.

- **orient [mandatory]**: Parameter that specifies which is the desired orientation of the searched image. The possible values for this parameter are:

  - **N** North Orientation.
  - **S** South Orientation.
  - **E** East Orientation.
  - **W** West Orientation.
  - ***** Signals that all orientations must be considered. It will return the best oblique for every orientation.

- **format [mandatory]:** Response format. Possible values are:

  - **txt:** Response is a text string with comma-separated values (“,”).
  - **Xml:** Response is an XML file.
  - **js:** Response is a JavaScript function that will return the value. This is for internal use of the JavaScript API. Requires the **function** parameter to specify the name of the function to return.

- **time [optional]** : A date following the structure YYYYMMDD or YYYYMM or YYYY. The service will use the most current images available, whose date is previous to the requested date. It is also possible to use a range of dates with the structure YYYYMMDD-YYYYMMDD, in this case the images used will be the most current ones available in the specified date range. If the parameter is omitted the system will submit the most recent version of the tile.

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom

- **layer [optional]:** Identifies which oblique layer the user is requesting. Possible values are obtained with the GetUserAvailableObliqueLayers service.

3.5.3 Error messages

If no images are found for the specified point an Error message is returned.

If the response format was specified as a text, the response message is:

```
ERROR, Error message text
```

If the response format is XML the error XML file is:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<result>
       <Error>
       ERROR, Error message text
       </Error>
</result>
```

3.6 GetObliqueTile Service

3.6.1 Description

This service returns a tile of a oblique image specified in the request by the image ID and the tile ID.

The desired tile to cover a given point at a desired **Zoom Level** can be calculated from the data provided in the **FindOblique** or **FindObliques** services as follows:

$$X = PositionX / (256 * 2^{(Zoom\_level - 1)})$$

$$Y = PositionY / (256 * 2^{(Zoom\_level - 1)})$$

In addition to aerial images, BlomURBEX can deliver overlay data. This is rasterized vector data with transparency, which can be displayed on top of the images. This information is delivered as transparent **png** files and grouped in layers. Layers can have information for one or all of the supported views, so that the information properly matches the different image views.

3.6.2 Parameters

The parameters needed in the request URL are the following:

- **id [mandatory]**: This is the identifier of the oblique image the tile belongs to.

- **Tile [mandatory]:** Tile coding inside the BlomURBEX structure, as follows.

  - **T_Y_X**, where:

    - T zoom level: 1,2,3 or 4
    - Y Row occupied by the tile in the provided zoom level
    - X Column occupied by the tile in the provided zoom level

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

- **Overlay [optional]**: If this parameter is present, the service will return tiles from the specified overlay layer instead of aerial images. BlomURBEX provides a "**generic**" overlay layer, with streets, roads and main administrative divisions.

3.6.3 Results

The result of the request will be a JPG image. If the user is not authorized to get the requested data or if the requested tile is not available, a transparent PNG file 256 x 256 pixels will be returned.

Overlays are returned as PNG images.

3.6.4 Example

A request of the service can be like this:

```
http://www.blomurbex.com/v02/GetObliqueTile?usertoken=********&id=N-
004.697670_40.655723_060830&Tile=2_2_2
```

The result of the service is the requested tile in JPG format.

3.7 GetObliqueHRTile Service

3.7.1 Description

This service is a clone of GetObliqueTile only valid for Oblique high-resolution images for the oblique layer: obliquehr.

3.8 ExtractOblique Service

3.8.1 Description

This service returns a portion of an oblique image. The image will be centered at a given point and zoom level.

If an oblique image Id is not specified, the system will automatically select the best oblique for the point and extract the data from this image.

3.8.2 Parameters

Parameters of the service are described below:

- **x [mandatory]**: Coordinate x or longitude in the specified projection system.

- **y [mandatory]**: Coordinate y or latitude in the specified projection system.

  The point defined by this (x,y) pair will be used as the centre of the image portion returned by the service

- **crs or srs [mandatory]:** Projection system used, It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.

- **width [mandatory]:** Returned image width in pixels.

- **height [mandatory]:** Returned image height in pixels.

- **zoom [mandatory]:** Desired zoom level for the oblique image.

  - Zoom=1 for level 1:1
  - Zoom=2 for level 1:2
  - Zoom=3 for level 1:4
  - Zoom=4 for level 1:8

- **time [optional]** : A date following the structure YYYYMMDD or YYYYMM or YYYY. The service will use the most current images available, whose date is previous to the requested date. It is also possible to use a range of dates with the structure YYYYMMDD-YYYYMMDD, in this case the images used will be the most current ones available in the specified date range.

  If the parameter is omitted the system will submit the most recent version of the tile or an error file if none is available.

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

- **idobliq [optional]**: Id of the oblique image from which to extract the image. If this parameter is not provided, the **FindOblique** service will be called internally to obtain the best oblique image for the given point.

- **orient [optional]**: If the **idobliq** parameter is not provided, this parameter will specify the orientation of the image that must be searched and will then become **mandatory**. Possible values for this parameter are:

    - **N** North Orientation
    - **S** South Orientation.
    - **E** East Orientation.
    - **W** West Orientation.

3.8.3 Results

The result of the request will be a JPG image. If the user is not authorized to get the requested data or if the requested image is not available, an JPG file with the text "unauthorized" is returned.

3.8.4 Examples

Example of an ExtractOblique request with no Oblique Id provided:

```
http://www.blomurbex.com/v02/ExtractOblique?usertoken=********&x=-
523000&y=4961500&crs=EPSG:3785&width=300&height=300&orient=N&zoom=1
```

The result of this request would be a portion of the best oblique image in North orientation, zoom level 1, with a size of 300x300 pixels, and centered in the given coordinates.

The same request providing an Oblique ID would be:

```
http://www.blomurbex.com/v02/ExtractOblique?usertoken=********&x=-
523000&y=4961500&crs=EPSG:3785&width=300&height=300&orient=N&zoom=1&idObliq=N-
004.697670_40.655723_060830
```

The result of this request would be the same image as above.

3.9 ExtractObliqueHR Service

3.9.1 Description

This service is a clone of ExtractObliques service only valid for Oblique high-resolution images for the oblique layer: obliquehr.

3.10 TransformPoints Service

3.10.1 Description

This service will be used to transform world coordinates to pixel coordinates in a given oblique image, and vice versa. This service needs the **ID** of an oblique image, against which all conversions are carried out.

3.10.2 Parameters

The parameters for this service are:

- **points [mandatory]**: This parameter is a list of coordinates separated by commas. The following rules must be observed to provide the list:
    - The number of coordinates must be an even number.
    - Odd coordinates, represent X coordinates.
    - Even coordinates, represent Y coordinates.
- **Crs or srs [mandatory]:** Projection system of the world coordinates used or requested. It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.
- **format [mandatory]:** Response format. Possible values are:
    - **txt:** The result of the service will be the list of transformed coordinates.
    - **Xml:** The result is an XML file, with the list of transformed coordinates.
    - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.
- **id [mandatory]**: This is the ID of the Oblique image used.
- **type [mandatory]**: This parameter defines how the conversion is done according to the following values:

- type = **toOblique**, The service will convert world coordinates to pixel coordinates of the oblique image.
- type = **toWorld**, The service will convert pixel coordinates of the oblique image to world coordinates.

- **dtm [mandatory]:** Indicates if the BlomURBEX digital terrain model of the area must be used in the conversion or not. Possible values for this parameter are:

  - **yes:** The Digital Terrain Model of the Area will be used. This slows down the transformation process, but results are more precise.
  - **no:** The Digital Terrain Model of the Area will not be used.

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

3.10.3 Examples

3.10.3.1 Transformation of pixel coordinates to world coordinates with Xml output

The request to the service would be made as follows:

```
http://www.blomurbex.com/v02/TransformPoints?usertoken=********&points=1183.0,723.0,1461.0,663.0&id=N-004.697670_40.655723_060830&type=toWorld&srs=EPSG:32630&dtm=YES&format=xml
```

The result of the service is a XML file with two couples of coordinates:

```
<result>
	<Point>
		<X>356352.0981982595</X>
		<Y>4502076.786308548</Y>
	</Point>
	<Point>
		<X>356398.5064077923</X>
		<Y>4502079.851212771</Y>
	</Point>
	</result>
```

3.10.3.2 Transformation of world to pixel coordinates with plain text output

The request to the service would be made as follows:

```
http://www.blomurbex.com/v02/TransformPoints?usertoken=********&points=356360,
4502070&id=N-
004.697670_40.655723_060830&type=toOblique&dtm=YES&crs=epsg:32630&format=txt
```

The result is a text string containing as many coordinates as values in the **points** parameter (in this example, there are only two). The response for this request should be:

```
1213.5,734.5
```

3.11 ExtractFullOblique Service

3.11.1 Description

This service returns a full oblique image. The image will be returned at a given zoom level.

3.11.2 Parameters

Parameters of the service are described below:

- **idobliq [mandatory]**: Id of the oblique image from which to extract the image.

- **zoom [mandatory]:** Desired zoom level for the oblique image.

    - Zoom=1 for level 1:1
    - Zoom=2 for level 1:2
    - Zoom=3 for level 1:4
    - Zoom=4 for level 1:8

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

3.11.3 Results

The result of the request will be a JPG image. If the user is not authorized to get the requested data or if the requested image is not available, an JPG file with the text "unauthorized" is returned.

3.11.4 Examples

Example of an ExtractFullOblique request with Oblique Id provided:

```
http://www.blomurbex.com/v02/ExtractFullOblique?usertoken=********&idobliq=N-003.702380_40.415195_090718&zoom=1
```

The result of this request would be complete oblique image in North orientation, zoom level 1.

3.12 GetObliqueMetadata Service

3.12.1 Description

The service **GetObliqueMetadata** returns all available metadata of a given oblique image.

The result of the service will be all the information for the oblique image that is stored in the BlomURBEX database.

3.12.2 Parameters

Parameters of the service are described below:

- **id [mandatory]**: ID of the oblique image.

- **format [mandatory]:** Response format. Possible values are:

  - **txt:**. The information is returned in the same order that it is stored in the database. Records are separated by "#".
  - **Xml:** Information is provided in an XML file.
  - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The function parameter is required to specify the name of the function to return.

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

3.12.3 Example

Below is an example of request for this service:

```
http://www.blomurbex.com/v02/GetObliqueMetadata?usertoken=********&format=xml&
id=N-004.697670_40.655723_060830
```

Metadata information is enclosed between tags identifying the name of the field.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<result>
	<metadata>
		<getIdOblique>N-004.697670_40.655723_060830</getIdOblique>
		<actual>true</actual>
		<id_dtm>blom.flt</id_dtm>
```

```
<imagename>ESAVLA013011NeighObliq98N_060830</imagename>
<imagepath>N:/Final_Delivery/Spain/ESAVLA06-093-LIB/ESAVLA06-093-WHS/ESAVLA011011 cluster/ESAVLA013011/</imagepath>
<imageext>psi</imageext>
<shotlevel>N</shotlevel>
<shottype>Ob</shottype>
<shotorient>N</shotorient>
<shotdate>2006-08-30 16:14:11</shotdate>
<gsd>0.135</gsd>
<imagerows>2672.0</imagerows>
<imagecols>4008.0</imagecols>
<ulx>-4.7020204968</ulx>
<uly>40.659821061</uly>
<lrx>-4.6948280005</lrx>
<lry>40.6530474199</lry>
<llx>-4.7011517093</llx>
<lly>40.6537889361</lly>
<flags>0.0</flags>
<cameralat>0.7094105642</cameralat>
<cameralon>-0.0819786207</cameralon>
<bearing>1.6234421273</bearing>
<roll>0.8179436368</roll>
<declin>0.1149655419</declin>
<alt>2080.3223750276</alt>
<elevation>1130.703</elevation>
<fpx>36.072</fpx>
<fpy>24.048</fpy>
<focallen>84.6807</focallen>
<k1>-6.3508536E-10</k1>
<k2>7.36459128E-17</k2>
<k3>2.29424674E-33</k3>
<ppx>0.001912841</ppx>
<ppy>0.0018005655</ppy>
<camera>[1]</camera>
<a1>583370.063953259</a1>
<a2>73047.7327615835</a2>
<a3>-7.69771121933379</a3>
<b1>84056.6404667805</b1>
<b2>623997.416267498</b2>
<b3>62.6751005781103</b3>
<c1>-674709.767016702</c1>
<c2>-2.50244507801924E7</c2>
<c3>-2583.16841779741</c3>
<ai1>-1.21494062710781E-4</ai1>
<ai2>0.00106550867121769</ai2>
<ai3>2.62143532254733E-5</ai3>
```

```
            <bi1>4.88555767639287E-4</bi1>
            <bi2>-0.00422524728528373</bi2>
            <bi3>-1.03972531537599E-4</bi3>
            <ci1>-4.70115167336928</ci1>
            <ci2>40.6537889259874</ci2>
            <ci3>1.0</ci3>
            <midpoint_x>-4.69767093658447</midpoint_x>
            <midpoint_y>40.6557235717773</midpoint_y>
            <int00>0.0</int00>
            <int01>111.111111111111</int01>
            <int10>111.111111111111</int10>
            <int11>0.0</int11>
            <int20>2004.0</int20>
            <int21>1336.0</int21>
            <ext0>-1.53312006368964E-6</ext0>
            <ext1>6.91416319484025E-6</ext1>
            <ext2>0.00228995240256543</ext2>
            <ext3>-1.37139511566242E-7</ext3>
            <ext4>1.37011347899792E-5</ext4>
            <ext5>-0.0198176351911305</ext5>
            <ext6>-4.95452177762923E-4</ext6>
            <ext7>-7.89273109805588E-4</ext7>
            <ext8>1.03204983974748</ext8>
        </metadata>
</result>
```

3.13 GetTeg Service

3.13.1 Description

The service **GetTeg** returns a TEG (Tiled Grid Elevation) file based on the Urbex ortho Qtree structure. This information could be used by client application on a delegate geo-referencing system.

3.13.2 Parameters

Parameters of the service are described below:

- **id [mandatory]**: Tile ID, it will be a sequence of digits, which only will be able to contain 0, 1, 2 and 3. This nomenclature corresponds to the way of codifying the Tiles for quadtrees.

- **tegid [mandatory]:** TEG model identifier that contains the requested tile id.

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

3.13.3 Example

Below is an example of request for this service:

```
http://www.blomurbex.com/v02/GetTeg?usertoken=********&tegid=nooslo18_4326l&id
= 120021311223021
```

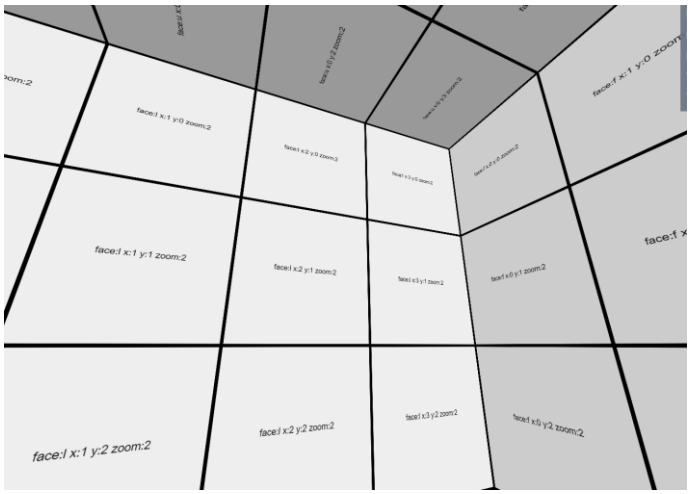
The response will be a Teg file with gzip compression with the name of the requested id.

4 Street Images services

4.1 Overview

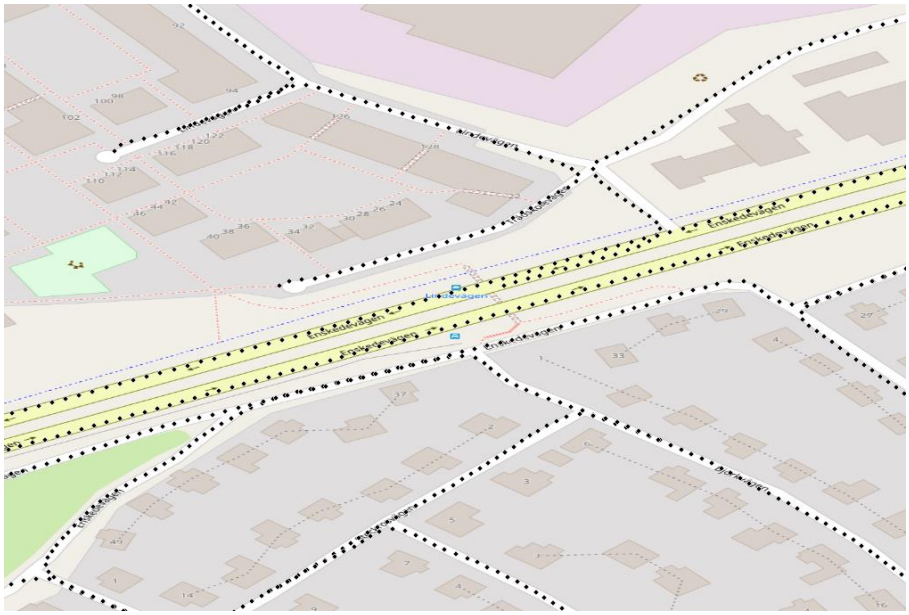
Street images are 360º degrees images obtained at ground level allowing the users to see the facades of the buildings. These images need to be treated specially.

The images are projected in a cubic projection, and each file is composed by a determinate number of frames obtained by determinate number of cameras.



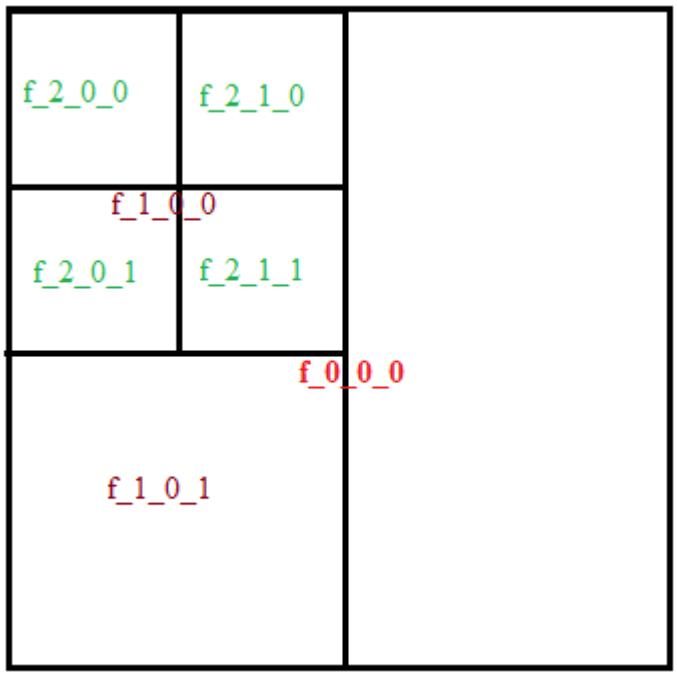
Every street picture has a centroid. The coordinates of this point serve as an identifier to every image. This point will be the location of the shoot.

Example of coverage of a set of images:



The cubic image is composed by 6 faces: front (f), back (b), right (r), left (l), up (u) & down (d). The front face will be centered and perpendicular to the car direction driving (direction yaw angle). Each face is composed by ties of 256x256. These tiles are available at several resolutions: Maximum Image resolution, ½ resolution, ¼ resolution and 1/8 resolution. These different scales are called **zoom levels** and are referenced as: 0,1, 2, 3 and so on if is needed. The minimum resolution is 0 and each face is composed by only one tile of 256x256.

Next picture shows an example of how the street image faces are cut into tiles. The code used to identify the tiles has 3 parts: the **zoom level**, and a pair of **x**, **y** coordinates of the tiles referenced to the upper left corner of the image.



4.1.1 Locating a tile

To find the tile where a particular pixel (of the original street image face) is located at a given zoom level, this formula can be used:

```
tileX = floor(pixelX / (256 * 2 ^level))

tileY = floor(pixelY / (256 * 2 ^level))
```

And the position of the original pixel within the tile:

```
Xoffset = (pixelX – tileX * 256 * 2 ^level) / (256 * 2 ^level)

Yoffset = (pixelY – tileY * 256 * 2 ^level) / (256 * 2 ^level)
```

All pixel coordinates are referred to the top-left corner of the images.

4.1.2 Image identifier

The identifier of any Street image, follows this structure:

CXXXX.XXXXXX_YY.YYYYYY_YYMMDD

Where:

The first character (C) indicates ‘Cubic’ projection.

Characters from position 2 to 12 **(XXXX.XXXXXX)** represent the longitude of the photo center in LAT/LON WGS84.

The characters from position 14 to 22 **(YY.YYYYYY)** represent the latitude of the centre of the photo in LAT/LON WGS84.

The final 6 characters **(YYMMDD)** represent the date in which the photography was taken, (YY=year, MM=month, DD=day).

Example:

The image with this Id:

C+018.075481_59.285323_160807

has its center located at Longitude **18.075481**, Latitude **59.285323** and was taken on August, 7; 2016

4.2 WFSStreetImages service

4.2.1 Description

The WFSStreetImages service is a web service based in the WFS standard protocol for serving vector information.

BlomURBEX™ offers the WFS service to allow users to access to vector information related with the Street Image centroids and metadata information following OGC's regulation and directives.

The service uses the http protocol with the parameters normalized by OGC, which define version, type of request, coordinate system, bounding box, etc.

The service supports several input projection systems, but vector information will provided on EPSG:4326 projection.

The WFS supported versions are 1.0.0 & 1.1.0.

4.2.2 Connection

To connect to the BlomURBEX™ WFSStreetImages service from a GIS client, it is necessary to setup the following URL:

```
http://www.blomurbex.com/v02/WFSProxyBlomStreet?REQUEST=GetCapabilities&SERVICE=WFS&ServiceName=WFS&version=1.1.0&userToken=****************
```

In this request, `usertokenstring` must be replaced by the unique platform access token provided by Blom.

GetCapabilities returns information about available features, capabilities and spatial operators.

4.2.3 SSL service (WFSProxyBlomStreetSSL)

In order to support SSL web protocol (HTTPS) it has been added a new clone service on this URL:

```
https://www.blomurbex.com/v02/WFSProxyBlomStreetSSL?REQUEST=GetCapabilities&SERVICE=WFS&ServiceName=WFS&version=1.1.0&userToken=****************
```

The response of this service will be equal to standard HTTP service, but with HTTPS links support.

4.2.4 Parameters

Parameters defined by OGC standard are:

- **Version [mandatory]**: This parameter is the WFS version that is used for this request. Only two values are accepted in the current API version:
  - **1.0.0**
  - **1.1.0**
- **Request [mandatory]:** Possible values are:
  - GetCapabilites: used to obtain the web services descriptor xml file
  - DescribeFeatureType: returns the feature descriptor xml file.
  - GetFeature: used to retrieving a feature query.
- **Service [mandatory]:** The service name: WFS
- **Typename:** mandatory for GetFeature and DescribeFeatureType requests. It is the name of the feature layer provided by the service. Street images will be provided by the name of 'panorama'.
- **Usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested from Blom.
- **Bbox [optional]:** Optional for GetFeature request. It defines the bounding box extent of the query.
- **SrsName [optional]:** it defines the projection system of the input information (i.e. bbox).

4.2.5 Result

The result of the GetCapabilities & DescribeFeatureType will be a Xml file according the OGC standard definition.

The output of GetFeature request is a Gml file with next parameters:

- **Geometry**: location of Street image centroid (x,y,z). Provided in EPSG:4326 projection
- **Pk_panorama:** Street image identifier.
- **Url:** This is the link url for the load of the single Street image in a web browser.
- **Direction_yaw:** This is the car direction angle to the North in degrees.

- **Tile_size:** this is the tile size (x & y value) of all tiles in each cubic face
- **Zoom_levels:** Number of available zoom levels inside the cubic image.
- **Min_zoom_level_size:** Number of tiles per cubic face on the minimum zoom level.
- **Source_name:** Name of the source file name. Normally, this is the name of the panorama (equirectangular) image.
- **Shot_date:** Date of the image capture. Format: YYYY-MM-DD hh:mm:ss
- **Srid:** EPSG code of the projection system used for the EO parameters calculation.
- **Camera_height:** Relative elevation between the camera system and the ground in meters.

4.2.6 Example

```
http://www.blomurbex.com/v02/WFSProxyBlomStreet?ServiceName=WFS&version=1.1.0&
userToken=********************&SERVICE=WFS&VERSION=1.0.0&REQUEST=GetFeature&TYP
ENAME=panorama&SRSNAME=EPSG:4326&BBOX=18.07535121,59.2847811,18.0766629616,59.
28556428
```

The result of the previous url:


```
▼<wfs:FeatureCollection xmlns:ms="http://mapserver.gis.umn.edu/mapserver" xmlns:wfs="http://www.opengis.net/wfs" xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http:/
 xsi:schemaLocation="http://www.opengis.net/wfs http://schemas.opengis.net/wfs/1.0.0/WFS-basic.xsd http://mapserver.gis.umn.edu/mapserver http://www.blomurbex.com/v02/WF
 SERVICE=WFS&VERSION=1.0.0&REQUEST=DescribeFeatureType&TYPENAME=panorama&OUTPUTFORMAT=XMLSCHEMA">
 ▼<gml:boundedBy>
  ▼<gml:Box srsName="EPSG:4326">
     <gml:coordinates>18.075358,59.284807 18.076620,59.285559</gml:coordinates>
   </gml:Box>
 </gml:boundedBy>
 ▼<gml:featureMember>
  ▼<ms:panorama fid="panorama.C+018.075481_59.285323_160807">
   ▼<gml:boundedBy>
    ▼<gml:Box srsName="EPSG:4326">
       <gml:coordinates>18.075481,59.285323 18.075481,59.285323</gml:coordinates>
     </gml:Box>
   </gml:boundedBy>
   ▼<ms:geometry>
    ▼<gml:Point srsName="EPSG:4326">
       <gml:coordinates>18.075481,59.285323</gml:coordinates>
     </gml:Point>
   </ms:geometry>
   <ms:pk_panorama>C+018.075481_59.285323_160807</ms:pk_panorama>
   ▼<ms:url>
     http://172.16.1.201/blomstreet.html?USERTOKEN=484441ce101a3bbc562b7f0ca5eb65b6&ID=C+018.075481_59.285323_160807
   </ms:url>
   <ms:direction_yaw>65.432200</ms:direction_yaw>
   <ms:tile_size>256</ms:tile_size>
   <ms:zoom_levels>4</ms:zoom_levels>
   <ms:min_zoom_level_size>1</ms:min_zoom_level_size>
   <ms:source_name>M01_160807_IMG111324.jpg</ms:source_name>
   <ms:shot_date>2016-08-07 12:16:05</ms:shot_date>
   <ms:srid>3011</ms:srid>
   <ms:camera_height>3.25</ms:camera_height>
  </ms:panorama>
 </gml:featureMember>
 ▼<gml:featureMember>
  ▼<ms:panorama fid="panorama.C+018.075538_59.285335_160807">
   ▼<gml:boundedBy>
    ▼<gml:Box srsName="EPSG:4326">
       <gml:coordinates>18.075538,59.285335 18.075538,59.285335</gml:coordinates>
     </gml:Box>
   </gml:boundedBy>
   ▼<ms:geometry>
    ▼<gml:Point srsName="EPSG:4326">
       <gml:coordinates>18.075538,59.285335</gml:coordinates>
     </gml:Point>
   </ms:geometry>
   <ms:pk_panorama>C+018.075538_59.285335_160807</ms:pk_panorama>
   ▼<ms:url>
     http://172.16.1.201/blomstreet.html?USERTOKEN=484441ce101a3bbc562b7f0ca5eb65b6&ID=C+018.075538_59.285335_160807
   </ms:url>
   <ms:direction_yaw>65.263500</ms:direction_yaw>
   <ms:tile_size>256</ms:tile_size>
   <ms:zoom_levels>4</ms:zoom_levels>
   <ms:min_zoom_level_size>1</ms:min_zoom_level_size>
   <ms:source_name>M01_160807_IMG111330.jpg</ms:source_name>
   <ms:shot_date>2016-08-07 12:16:05</ms:shot_date>
   <ms:srid>3011</ms:srid>
   <ms:camera_height>3.25</ms:camera_height>
  </ms:panorama>
 </gml:featureMember>
▼<gml:featureMember>
```

4.3 FindStreetImagesByExtent

4.3.1 Description

This service will locate all the Street photos whose centers are located inside of a given extent.

4.3.2 Parameters

The parameters needed in the request URL are the following:

- **bbox [mandatory]**: Bounding Box defined by four coordinates that specify the area in the specified projection system. Coordinates are provided in the following order: minimum X coordinate, minimum Y coordinate, maximum X coordinate, maximum Y coordinate.

- **crs or srs [mandatory]:** Projection system used, It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.

- **format [mandatory]:** Response format. Possible values are:
  - **txt:** Response is a text string with comma-separated values (“,”).
  - **Xml:** Response is an XML file.
  - **js:** Response is a JavaScript function that will return the value. This is for internal use of the JavaScript API. Requires the **function** parameter to specify the name of the function to return.
  - **Json:** Response is Json format.

- **time [optional]** : A date following the structure YYYYMMDD or YYYYMM or YYYY. The service will use the most current images available, whose date is previous to the requested date. It is also possible to use a range of dates with the structure YYYYMMDD-YYYYMMDD, in this case the images used will be the most current ones available in the specified date range.

  If the parameter is omitted the system will submit the most recent version of the tile.

- **metadata [optional]:** This parameter allows to obtain all metadata info for all images inside the searching area.

- **usertoken [mandatory]**: User’s Token for authentication purposes. This token is unique for each client and must be requested to Blom

4.3.3 Error messages

If no images are found for the specified extent an Error message is returned.

If the response format was specified as a text, the response message is:

```
ERROR, Error message text
```

If the response format is XML the error XML file is:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<result>
      <Error>
      ERROR, Error message text
      </Error>
</result>
```

4.3.4 Results

For each Street image located the following information is provided:

- Id (Street image identifier)
- Xcp: longitude location of the image centroid (EPSG:4326)
- Ycp: latitude location of the image centroid (EPSG:4326)
- Zcp: Altitude to the sea level of the image centroid in meters
- Srid: Local projection system (EPSG code) for the EO calculation
- Camera_height: Camera system elevation to the ground.
- Xcp_local: X location in local projection system (Srid)
- Ycp_local: Y location in local projection system (Srid)

4.3.5 Example

```
http://www.blomurbex.com/v02/FindStreetImagesByExtent?srs=EPSG:4326&bbox=10.218658438246603,60.153809133223405,10.220031729266088,60.15449257649393&usertoken=**********&format=xml
```

The resoult of previous url:

```
<result>
  <metadata>
    <Id>C+018.079269_59.282349_160807</Id>
    <xcp>18.0792685782187</xcp>
    <ycp>59.2823489312905</ycp>
    <zcp>22.906</zcp>
    <srid>3011</srid>
    <camera_height>3.25</camera_height>
    <xcp_local>154518.63</xcp_local>
    <ycp_local>6574124.758</ycp_local>
  </metadata>
  <metadata>
    <Id>C+018.079323_59.282366_160807</Id>
    <xcp>18.0793234553512</xcp>
    <ycp>59.2823656573122</ycp>
    <zcp>22.94</zcp>
    <srid>3011</srid>
    <camera_height>3.25</camera_height>
    <xcp_local>154521.756</xcp_local>
```

```
  <ycp_local>6574126.625</ycp_local>
</metadata>
<metadata>
  <Id>C+018.079377_59.282384_160807</Id>
  <xcp>18.0793768286367</xcp>
  <ycp>59.2823844309067</ycp>
  <zcp>23.017</zcp>
  <srid>3011</srid>
  <camera_height>3.25</camera_height>
  <xcp_local>154524.796</xcp_local>
  <ycp_local>6574128.72</ycp_local>
</metadata>
```

4.4 FindStreetImages

4.4.1 Description

This service will locate all the Street photos whose centers are located inside of a given extent defined with a ratio from a location defined by a coordinate or a Street Image identifier.

4.4.2 Parameters

The parameters needed in the request URL are the following:

- **id [optional]**: ID of the Street image.
- **x [optional]**: Coordinate x or longitude in the specified projection system.
- **y [optional]**: Coordinate y or latitude in the specified projection system.

  The **x,y** location or **id** value must be used.
- **ratio[mandatory]:** searching ratio in meters.
- **crs or srs [mandatory]:** Projection system used, It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.
- **format [mandatory]:** Response format. Possible values are:
  - **txt:** Response is a text string with comma-separated values (“,”).
  - **Xml:** Response is an XML file.
  - **js:** Response is a JavaScript function that will return the value. This is for internal use of the JavaScript API. Requires the **function** parameter to specify the name of the function to return.
  - **Json:** Response is Json format.
- **time [optional]** : A date following the structure YYYYMMDD or YYYYMM or YYYY. The service will use the most current images available, whose date is previous to the requested date. It is also possible to use a range of dates with the structure YYYYMMDD-YYYYMMDD, in this case the images used will be the most current ones available in the specified date range.

  If the parameter is omitted the system will submit the most recent version of the tile.
- **Layer [optional]**: This parameter restricts the search to the selected Street layer.

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom

4.4.3 Error messages

If no images are found for the specified extent an Error message is returned.

If the response format was specified as a text, the response message is:

```
ERROR, Error message text
```

If the response format is XML the error XML file is:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<result>
	<Error>
	ERROR, Error message text
	</Error>
</result>
```

4.4.4 Results

The image selected by the identifier or the nearest image to the x,y location will be provided with all metadata. The other images will be provided with a limited number of parameters:

Main image:

- Id (Street image identifier)
- IdSet: Set of images identifier
- Imagename: source image name
- Shotdate: Image capture date
- Xcp: longitude location of the image centroid (EPSG:4326)
- Ycp: latitude location of the image centroid (EPSG:4326)
- Zcp: Altitude to the sea level of the image centroid in meters
- Direction_yaw: Car direction to the North in degrees
- Min_zoom_level_size: Number of tiles of the minimum zoom levels per cubic face
- Zoom_levels: Number of zoom levels per cubic face
- Tile_size: pixel tile size (x & y axis)
- Id_dtm: Available Dtm for the street image
- Id_dsm: Available Dsm for the street image
- Omega: EO angle of the main frame
- Phi: EO angle of the main frame
- Kappa: EO angle of the main frame
- Ext0,1,2,3,4,5,6,7,8: Rotation matrix parameters
- Xcp_local: X location in local projection system (Srid)
- Ycp_local: Y location in local projection system (Srid)
- Zcp_local: Altitude to the sea level of the image centroid in meters
- Equi_img_width: Source equirectangular image with (pixels)
- Column_center: column pixel center of source equirectangular image.
- Row_center: Row pixel center of source equirectangular image.
- Pixel_angle: Number of pixels per degree.

- Srid: Local projection system (EPSG code) for the EO calculation
- Camera_height: Camera system elevation to the ground.
- Street_type: Street layer name
- Depthmapavailable: Depthmap file availability for measurements.
- Copyright: image copyright.

Images on the searching area:
- Id (Street image identifier)
- Xcp: longitude location of the image centroid (EPSG:4326)
- Ycp: latitude location of the image centroid (EPSG:4326)
- Zcp: Altitude to the sea level of the image centroid in meters
- Srid: Local projection system (EPSG code) for the EO calculation
- Camera_height: Camera system elevation to the ground.
- Xcp_local: X location in local projection system (Srid)
- Ycp_local: Y location in local projection system (Srid)

4.4.5 Example

http://www.blomurbex.com/v02/FindStreetImages?srs=EPSG:4326&x=18.07963810&y=59.28996795&ratio=50&usertoken=**********&format=xml

The resoult of previous url:

```
<result>
  <metadata>
    <Id>C+018.079638_59.289968_160822</Id>
    <IdSet>-2147483580</IdSet>
    <imagename>M01_160822_IMG71958.jpg</imagename>
    <shotdate>2016-08-22 13:16:18</shotdate>
    <xcp>18.079638135368</xcp>
    <ycp>59.2899679325398</ycp>
    <zcp>43.26</zcp>
    <direction_yaw>28.1005</direction_yaw>
    <tile_size>256</tile_size>
    <zoom_levels>4</zoom_levels>
    <min_zoom_level_size>1</min_zoom_level_size>
    <id_dtm>sestoc16_4326-street.flt</id_dtm>
    <id_dsm>null</id_dsm>
    <omega>1.63453</omega>
    <phi>0.684249</phi>
    <kappa>26.3995</kappa>
    <ext0>0.895651410322364</ext0>
    <ext1>-0.444596375489939</ext1>
    <ext2>0.0119421140829742</ext2>
    <ext3>0.444752279617201</ext3>
    <ext4>0.895199365372074</ext4>
    <ext5>-0.0285220267993026</ext5>
    <ext6>0.0019902167883174</ext6>
    <ext7>0.0308570759898991</ext7>
    <ext8>0.999521825623878</ext8>
    <local_xcp>154538.682</local_xcp>
    <local_ycp>6574973.54</local_ycp>
    <local_zcp>43.26</local_zcp>
    <equi_img_width>8000</equi_img_width>
    <column_center>4000</column_center>
    <row_center>2000</row_center>
```

```
  <pixel_angle>22.22222222</pixel_angle>
  <srid>3011</srid>
  <camera_height>3.25</camera_height>
  <street_type>StreetST</street_type>
</metadata>
<hotspots>
  <metadata>
    <Id>C+018.079338_59.290191_160807</Id>
    <xcp>18.07933813542</xcp>
    <ycp>59.2901912303305</ycp>
    <zcp>43.53</zcp>
    <srid>3011</srid>
    <camera_height>3.25</camera_height>
    <xcp_local>154521.555</xcp_local>
    <ycp_local>6574998.395</ycp_local>
  </metadata>
  <metadata>
    <Id>C+018.079374_59.290162_160807</Id>
    <xcp>18.0793735648733</xcp>
    <ycp>59.2901622680161</ycp>
    <zcp>43.521</zcp>
    <srid>3011</srid>
    <camera_height>3.25</camera_height>
    <xcp_local>154523.578</xcp_local>
    <ycp_local>6574995.171</ycp_local>
  </metadata>
```

4.5 FindNearestStreetImage

4.5.1 Description

This service will obtain all metadata information of the nearest Street image to the location provided.

4.5.2 Parameters

The parameters needed in the request URL are the following:

- **x [mandatory]**: Coordinate x or longitude in the specified projection system.
- **y [mandatory]**: Coordinate y or latitude in the specified projection system.
- **ratio[mandatory]:** searching ratio in meters.
- **crs or srs [mandatory]:** Projection system used, It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.
- **format [mandatory]:** Response format. Possible values are:
  - **txt:** Response is a text string with comma-separated values (“,”).
  - **Xml:** Response is an XML file.
  - **js:** Response is a JavaScript function that will return the value. This is for internal use of the JavaScript API. Requires the **function** parameter to specify the name of the function to return.
  - **Json:** Response is Json format.
- **time [optional]** : A date following the structure YYYYMMDD or YYYYMM or YYYY. The service will use the most current images available, whose date is previous to the requested date. It is also possible to use a range of dates with the structure YYYYMMDD-YYYYMMDD, in this case the images used will be the most current ones available in the specified date range.

  If the parameter is omitted the system will submit the most recent version of the tile.
- **Layer [optional]**: This parameter restricts the search to the selected Street layer.

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom

4.5.3 Error messages

If no images are found for the specified extent an Error message is returned.

If the response format was specified as a text, the response message is:

```
ERROR, Error message text
```

If the response format is XML the error XML file is:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<result>
	<Error>
	ERROR, Error message text
	</Error>
</result>
```

4.5.4 Results

For each Street image located the following information is provided:

- Id (Street image identifier)
- IdSet: Set of images identifier
- Imagename: source image name
- Shotdate: Image capture date
- Xcp: longitude location of the image centroid (EPSG:4326)
- Ycp: latitude location of the image centroid (EPSG:4326)
- Zcp: Altitude to the sea level of the image centroid in meters
- Direction_yaw: Car direction to the North in degrees
- Min_zoom_level_size: Number of tiles of the minimum zoom levels per cubic face
- Zoom_levels: Number of zoom levels per cubic face
- Tile_size: pixel tile size (x & y axis)
- Id_dtm: Available Dtm for the street image
- Id_dsm: Available Dsm for the street image
- Omega: EO angle of the main frame
- Phi: EO angle of the main frame
- Kappa: EO angle of the main frame
- Ext0,1,2,3,4,5,6,7,8: Rotation matrix parameters
- Xcp_local: X location in local projection system (Srid)
- Ycp_local: Y location in local projection system (Srid)
- Zcp_local: Altitude to the sea level of the image centroid in meters
- Equi_img_width: Source equirectangular image with (pixels)
- Column_center: column pixel center of source equirectangular image.
- Row_center: Row pixel center of source equirectangular image.
- Pixel_angle: Number of pixels per degree.
- Srid: Local projection system (EPSG code) for the EO calculation
- Camera_height: Camera system elevation to the ground.

4.5.5 Example

http://www.blomurbex.com/v02/FindNearestStreetImage?srs=EPSG:4326&x=18.0796381
0&y=59.28996795&ratio=50&usertoken=**********&format=xml

The resoult of previous url:

```
<result>
  <metadata>
    <IdPanorama>C+018.079638_59.289968_160822</IdPanorama>
    <IdSet>-2147483580</IdSet>
    <imagename>M01_160822_IMG71958.jpg</imagename>
    <shotdate>2016-08-22 13:16:18</shotdate>
    <xcp>18.079638135368</xcp>
    <ycp>59.2899679325398</ycp>
    <zcp>43.26</zcp>
    <direction_yaw>28.1005</direction_yaw>
    <tile_size>256</tile_size>
    <zoom_levels>4</zoom_levels>
    <min_zoom_level_size>1</min_zoom_level_size>
    <id_dtm>sestoc16_4326-street.flt</id_dtm>
    <id_dsm>null</id_dsm>
    <omega>1.63453</omega>
    <phi>0.684249</phi>
    <kappa>26.3995</kappa>
    <ext0>0.895651410322364</ext0>
    <ext1>-0.444596375489939</ext1>
    <ext2>0.0119421140829742</ext2>
    <ext3>0.444752279617201</ext3>
    <ext4>0.895199365372074</ext4>
    <ext5>-0.0285220267993026</ext5>
    <ext6>0.0019902167883174</ext6>
    <ext7>0.0308570759898991</ext7>
    <ext8>0.999521825623878</ext8>
    <local_xcp>154538.682</local_xcp>
    <local_ycp>6574973.54</local_ycp>
    <local_zcp>43.26</local_zcp>
    <equi_img_width>8000</equi_img_width>
    <column_center>4000</column_center>
    <row_center>2000</row_center>
    <pixel_angle>22.22222222</pixel_angle>
    <srid>3011</srid>
    <camera_height>3.25</camera_height>
    <street_type>StreetST</street_type>
  </metadata>
</result>
```

4.6 GetStreetImageTile

4.6.1 Description

This service returns a tile of a Street image specified in the request by the image ID and the tile ID.

The desired tile to cover a given point at a desired **Zoom Level** in a cubic face can be calculated from the data provided in the **WFSStreetImage** or **FindStreetImageByExtent** services as follows:

$$X = PositionX / (256 * 2^{(Zoom\_level)})$$

$$Y = PositionY / (256 * 2^{(Zoom\_level)})$$

4.6.2 Parameters

The parameters needed in the request URL are the following:

- **id [mandatory]**: This is the identifier of the street image the tile belongs to.
- **Tile [mandatory]:** Tile coding inside the BlomURBEX structure, as follows.
  - **F_T_Y_X**, where:
    - F Cubic face: f, b, r, l, u, d
    - T zoom level: 0,1,2,3…
    - Y Row occupied by the tile in the provided zoom level
    - X Column occupied by the tile in the provided zoom level
- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

4.6.3 Results

The result of the request will be a JPG image. If the user is not authorized to get the requested data or if the requested tile is not available, a transparent PNG file 256 x 256 pixels will be returned.

4.6.4 Example

A request of the service can be like this:

```
http://www.blomurbex.com/v02/GetStreetTile?usertoken=**********************&
id=C+018.080492_59.282739_160807&tile=f_2_1_2
```

The result of the service is the requested tile in JPG format.

4.7 GetStreetImageMetadata

4.7.1 Description

The service **GetStreetImageMetadata** returns all available metadata of a given street image.

The result of the service will be all the information for the Street image that is stored in the BlomURBEX database.

4.7.2 Parameters

Parameters of the service are described below:

- **id [mandatory]**: ID of the Street image.

- **format [mandatory]:** Response format. Possible values are:

  - **txt:**. The information is returned in the same order that it is stored in the database. Records are separated by "#".
  - **Xml:** Information is provided in an XML file.
  - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The function parameter is required to specify the name of the function to return.
  - **Json:** Json format.

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

4.7.3 Results

For each Street image located the following information is provided:

- Id (Street image identifier)
- IdSet: Set of images identifier
- Imagename: source image name
- Shotdate: Image capture date
- Xcp: longitude location of the image centroid (EPSG:4326)
- Ycp: latitude location of the image centroid (EPSG:4326)
- Zcp: Altitude to the sea level of the image centroid in meters
- Direction_yaw: Car direction to the North in degrees
- Min_zoom_level_size: Number of tiles of the minimum zoom levels per cubic face
- Zoom_levels: Number of zoom levels per cubic face
- Tile_size: pixel tile size (x & y axis)
- Id_dtm: Available Dtm for the street image

- Id_dsm: Available Dsm for the street image
- Omega: EO angle of the main frame
- Phi: EO angle of the main frame
- Kappa: EO angle of the main frame
- Ext0,1,2,3,4,5,6,7,8: Rotation matrix parameters
- Xcp_local: X location in local projection system (Srid)
- Ycp_local: Y location in local projection system (Srid)
- Zcp_local: Altitude to the sea level of the image centroid in meters
- Equi_img_width: Source equirectangular image with (pixels)
- Column_center: column pixel center of source equirectangular image.
- Row_center: Row pixel center of source equirectangular image.
- Pixel_angle: Number of pixels per degree.
- Srid: Local projection system (EPSG code) for the EO calculation
- Camera_height: Camera system elevation to the ground.

4.7.4 Example

Below is an example of request for this service:

```
http://www.blomurbex.com/v02/GetStreetImageMetadata?usertoken=****************
&id=C+010.260788_60.082537_160708&format=json
```

This is the result of previous url:


```
{
  "idPanorama": "C+010.260788_60.082537_160708",
  "imageName": "M03_160708_IMG55800.jpg",
  "shotDate": "2016-07-08 10:52:05",
  "xcp": 10.2607883928267,
  "ycp": 60.0825374424905,
  "zcp": 150.4587,
  "direction_yaw": 107.432,
  "tile_size": 256,
  "zoom_levels": 4,
  "min_zoom_level_size": 1,
  "id_set": -2147483617,
  "dtmName": "null",
  "dsmName": "null",
  "omega": 4.21005,
  "phi": -4.34762,
  "kappa": -52.9319,
  "ext0": 0.317209037853662,
  "ext1": -0.163217251090218,
  "ext2": 0.934204771584103,
  "ext3": 0.948355063189615,
  "ext4": 0.0535037635797599,
  "ext5": -0.3126659901643,
  "ext6": 0.0010490121900986,
  "ext7": 0.985138303097263,
  "ext8": 0.171759783837984,
  "local_xcp": 86685.2742,
  "local_ycp": 1232009.9831,
  "local_zcp": 150.4587,
  "equi_img_width": 8000,
  "column_center": 4000,
  "row_center": 2000,
  "pixel_angle": 22.0,
  "srid": "5110",
  "camera_height": 3.25
}
```

4.8 TransformPointsStreet

4.8.1 Description

This service will be used to transform world coordinates to pixel coordinates in a given Street image, and vice versa. This service needs the **ID** of a Street image, against which all conversions are carried out.

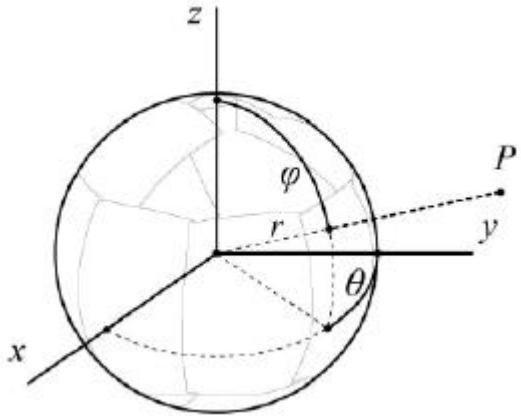
4.8.2 Parameters

The parameters for this service are:

- **points [mandatory]**: This parameter is a list of coordinates separated by commas. The following rules must be observed to provide the list:
    - The number of coordinates must be an even number in 2D input values.
    - Odd coordinates, represent X coordinates in 2D input values.
    - Even coordinates, represent Y coordinates in 2D input values.
    - in 3D input values, the order is x,y,z
- **Crs or srs [mandatory]:** Projection system of the world coordinates used or requested. It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.
- **format [mandatory]:** Response format. Possible values are:
    - **txt:** The result of the service will be the list of transformed coordinates.
    - **Xml:** The result is an XML file, with the list of transformed coordinates.
    - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.
    - **Json**: The result is an json file, with the list of transformed coordinates.
- **id [mandatory]**: This is the ID of the Oblique image used.
- **type [mandatory]**: This parameter defines how the conversion is done according to the following values:
    - type = **toStreet2D**, The service will convert world 2D coordinates to point coordinates of the street image at ground level.
    - type = **toStreet3D**, The service will convert world 3D coordinates to point coordinates of the Street image.

- type = **toWorld**, The service will convert point coordinates of the Street image at a identified distance to world coordinates.
- type = **toWorldDepthMap**, The service will convert point coordinates of the Street image at a identified distance to world coordinates using a Depth map image of the street image. This is the most accurate option.

The point Street values are defined by two values in a spherical coordinate system: Theta & Phi



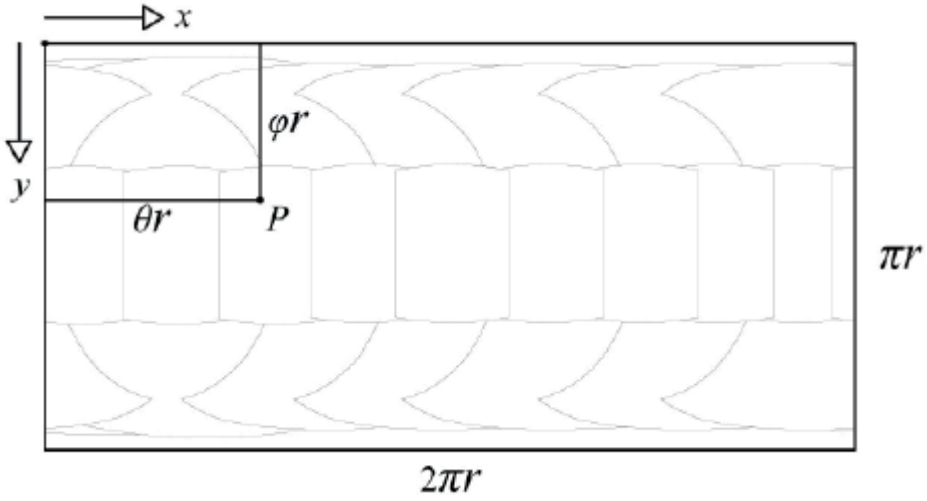
The Theta value is the azimuthal angle (longitude) measured from the x-axis in the xy plane by the pixel angle factor:

Theta (px) = Theta (rad) * pixel_angle

The Phi value is the polar angle or zenith angle or colatitudes and measured from the positive z-axis by the pixel angle.

Phi (px) = Phi (rad) * pixel_angle

The pixel coordinates are result of the equirectangular projection system:



- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

4.8.3 Examples

4.8.3.1 Transformation of pixel coordinates & distance to world coordinates with Xml output

The request to the service would be made as follows:

```
http://www.blomurbex.com/v02/TransformPointsStreet?usertoken=*****************
&points=1393,1654,22.7,1393,1654,20.7&id=C+018.079480_59.289828_160822&type=to
World&srs=EPSG:3011&format=xml
```

The result of the service is a XML file with two couples of coordinates:

```
<result>
	<Point>
		<X>154513.22949788778</X>
		<Y>6574972.091435375</Y>
		<Z>50.19940081274179</Z>
	</Point>
	<Point>
		<X>154514.6772954306</X>
		<Y>6574970.8465512</Y>
		<Z>49.60429941955603</Z>
	</Point>
</result>
```

4.8.3.2 Transformation of 2D world to pixel coordinates with plain text output

The request to the service would be made as follows:

```
http://www.blomurbex.com/v02/TransformPointsStreet?usertoken=*****************
&points=18.07963810,59.28996795,18.07973810,59.28986795&id=C+018.079480_59.289
828_160822&type=toStreet2D&srs=EPSG:3011&format=xml
```

The result is a text string containing as many coordinates as values in the **points** parameter (in this example, there are only two). The response for this request should be:

```
3166.2221705364805,2226.9722500327634,4123.528453949508,2237.770
322191771,
```

4.8.3.3 Transformation of 3D world to pixel coordinates with plain text output

The request to the service would be made as follows:

```
http://www.blomurbex.com/v02/TransformPointsStreet?usertoken=****************
&points=18.07963810,59.28996795,50.5&id=C+018.079480_59.289828_160822&type=toS
treet3D&srs=EPSG:3011&format=txt
```

The result is a text string containing as many coordinates as values in the **points** parameter (in this example, there are only two). The response for this request should be:

```
3143.6613428076703,1524.2795542874042
```

5 Vertical images services

5.1 Overview

Vertical images are orthogonal images reprojected to Spherical Mercator and north-oriented that are used as a source for mosaic creation.

Their treatment will be very similar to the oblique images explained in last chapter, although Vertical Image interaction will be easier since there's no need for a DTM transformation to get World coordinates value from pixel coordinates inside the image and the other way around. This enables measurements, transformations and other operations in vertical images to be immediate and with no need whatsoever of contacting the server to do DTM operations.

As in oblique images, each particular vertical image is cut into tiles 256x256 pixels in size. These tiles are available at several resolutions: Maximum Image resolution, ½ resolution, ¼ resolution and 1/8 resolution. These different scales are called **zoom levels** and are referenced as: 1, 2, 3 and 4 in the vertical images handling services.

Everything explained in oblique images related to tiling and locating tiles can be applied in Vertical images as well.

5.1.1 Image identifier

The identifier of any vertical image will have the same structure as oblique images:

VXXXX.XXXXXX_YY.YYYYYY_YYMMDD

Where:

The first character (V) indicates that the image is a vertical image.

Characters from position 2 to 12 **(XXXX.XXXXXX)** represent the longitude of the photo center in LAT/LON WGS84.

Characters from position 14 to 22 **(YY.YYYYYY)** represent the latitude of the centre of the photo in LAT/LON WGS84.

The final 6 characters **(YYMMDD)** represent the date in which the photography was taken, (YY=year, MM=month, DD=day).

Example:

The image with this Id:

V-004.697670_40.655723_060830

has its center located at Longitude -4.697670, Latitude 40.655723 and was taken on August, 30; 2006.

5.2 ExtractVertical Service

5.2.1 Description

This service returns a portion of a vertical image. The image will be centered at a given point and zoom level.

If a vertical image Id is not specified, the system will automatically select the best vertical for the point (using the same procedure as in 'FindVertical' service) and extract the data from this image.

5.2.2 Parameters

Parameters of the service are described below:

- **x [mandatory]**: Coordinate x or longitude in the specified projection system.

- **y [mandatory]**: Coordinate y or latitude in the specified projection system.

  The point defined by this (x,y) pair will be used as the centre of the image portion returned by the service

- **crs or srs [mandatory]:** Projection system used, It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.

- **width [mandatory]:** Returned image width in pixels.

- **height [mandatory]:** Returned image height in pixels.

- **zoom [mandatory]:** Desired zoom level for the oblique image.

  - Zoom=1 for level 1:1
  - Zoom=2 for level 1:2
  - Zoom=3 for level 1:4
  - Zoom=4 for level 1:8

- **time [optional]** : A date following the structure YYYYMMDD or YYYYMM or YYYY. The service will use the most current images available, whose date is previous to the requested date. It is also possible to use a range of dates with the structure YYYYMMDD-YYYYMMDD, in this case the images used will be the most current ones available in the specified date range.

  If the parameter is omitted the system will submit the most recent version of the tile or an error file if none is available.

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

- **idvertical [optional]**: Id of the oblique image from which to extract the image. If this parameter is not provided, the **FindVertical** service will be called internally to obtain the best oblique image for the given point.

5.2.3 Results

The result of the request will be a JPG image. If the user is not authorized to get the requested data or if the requested image is not available, an JPG file with the text "unauthorized" is returned.

5.2.4 Examples

Example of an ExtractOblique request with no Vertical Id provided:

```
http://www.blomurbex.com/v02/ExtractVertical?usertoken=********&x=-
87.765998&y=30.4028127&crs=EPSG:4326&width=500&height=500&zoom=1
```

The result of this request would be a portion of the best oblique image in North orientation, zoom level 1, with a size of 300x300 pixels, and centered in the given coordinates.

The same request providing an Oblique ID would be:

```
www.blomurbex.com/v02/ExtractVertical?usertoken=********&x=-
87.765998&y=30.4028127&crs=EPSG:4326&width=500&height=500&zoom=1&idvertical=V-
087.766006_30.402841_120207
```

The result of this request would be the same image as above.

5.3 BlomURBEX oblique services

To be able to request and manipulate oblique imaginary several services are available in BlomURBEX:

- **FindVertical:** This is the most basic service in the managing of vertical imagery. This service identifies the best vertical image for a given set of coordinates, i.e., the image which center is closer to the supplied coordinates.

  The result of the **FindOblique** service is a plain text or XML file, identifying the best available vertical and the pixel in that image that represents the point of interest. It also provides the full size of the oblique image and the ground coordinates of the four corners of the oblique image.

- **ExtractVertical:** This service allows recovering a portion of a vertical image without considering the existing tiles.

  This service will return a portion of a vertical image, centered in a precise point at a determined zoom level, and of a given size.

  All these functions are detailed in the following sections.

5.4 FindVertical Service

5.4.1 Description

This service will locate the best vertical image that covers a given point.

5.4.2 Parameters

The parameters needed in the request URL are the following:

- **x [mandatory]**: Coordinate x or longitude in the specified projection system.

- **y [mandatory]**: Coordinate y or latitude in the specified projection system.

- **crs or srs [mandatory]:** Projection system used, It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.

- **format [mandatory]:** Response format. Possible values are:

  - **txt:** Response is a text string with comma-separated values (“,”).
  - **Xml:** Response is an XML file.
  - **js:** Response is a JavaScript function that will return the value. This is for internal use of the JavaScript API. Requires the **function** parameter to specify the name of the function to return.

- **time [optional]** : A date following the structure YYYYMMDD or YYYYMM or YYYY. The service will use the most current images available, whose date is previous to the requested date. It is also possible to use a range of dates with the structure YYYYMMDD-YYYYMMDD, in this case the images used will be the most current ones available in the specified date range.

  If the parameter is omitted the system will submit the most recent version of the tile.

- **usertoken [mandatory]**: User’s Token for authentication purposes. This token is unique for each client and must be requested to Blom

5.4.3 Error messages

If no images are found for the specified point an Error message is returned.

If the response format was specified as a text, the response message is:

```
ERROR NUMBER, Error message text
```

If the response format is XML the error XML file is:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<result>
        <Error>
        ERROR NUMBER, Error message text
        </Error>
</result>
```

5.4.4 Results

For each oblique image located the following information is provided:

- The oblique ID
- Position X, pixel coordinate in the image
- Position Y, pixel coordinate in the image
- The number of columns of the photography
- The number of rows.
- Upper-left corner x coordinate
- Upper-left corner y coordinate
- Upper-right corner x coordinate
- Upper-right corner y coordinate
- Lower-left corner x coordinate
- Lower-left corner y coordinate
- Lower-right corner x coordinate
- Lower-right corner y coordinate

5.4.5 Examples

5.4.5.1 FindVertical of a given orientation with plain text output

The http request for searching the best oblique image for only one orientation would be like this:

```
http://www.blomurbex.com/v02/FindVertical?usertoken=********&x=-
81.477472&y=28.480029&crs=EPSG:4326&format=txt
```

The response for this request is a comma-separated text string like the one shown below:

```
V-081.477737_28.479913_111231,2166,3379,4200,6053,-
81.479450603,28.482748899,-81.475614151,28.482384655,-
81.476499331,28.477876229,-81.47934026,28.47804254
```

The following table details the meaning of each field in the example. Data is provided as described in section 3.3.4

V-081.477737_28.479913_111231	Oblique image Id
2166	Pixel coordinate X in the image
3379	Pixel coordinate Y in the image
4200	Number of columns of the oblique image
6053	Number of rows of the oblique image
-81.479450603	Upper-Left corner, X coordinate
28.482748899	Upper-Left corner, Y coordinate
-81.475614151	Upper-Right corner, X coordinate
28.482384655	Upper-Right corner, Y coordinate
-81.476499331	Lower-Left corner, X coordinate
28.477876229	Lower-Left corner, Y coordinate
-81.47934026	Lower-Right corner, X coordinate
28.47804254	Lower-Right corner, Y coordinate

The coordinates of the four corners of the image are provided in the reference system specified in the request.

6 Measurements on Ortho Images

6.1 CalculateElevationInOrtho Service

6.1.1 Description

This service calculates the elevation from sea level of a given point of an ortho image. The elevation is measured in **meters**.

6.1.2 Parameters

This service has the following parameters:

- **x [mandatory]:** World coordinate X in any valid coordinate system.
- **y [mandatory]:** World coordinate Y in any valid coordinate system.
- **crs or srs [mandatory]:** Projection system used, It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.
- **format [mandatory]:** Response format. Possible values are:
  - **txt:** Response is a text string with the point Elevation in meters.
  - **Xml:** The result is an XML file Xml with information of the point Elevation.
  - **js:** Result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.
- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

6.1.3 Example

Here is an example of a request to this service:

```
http://www.blomurbex.com/v02/CalculateElevationInOrtho?usertoken=********&format=xml&x=356678&y=4501818&srs=EPSG:32630
```

The response for this request should be:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
```

```
-<result>
      <ElevationInOrtho> 1122.7246755304689</ElevationInOrtho>
 </result>
```

6.2 CalculateGroundLength Service

6.2.1 Description

This service calculates the length between 2 points, but, as opposite to CalculateLength (See section 7.3 for further information), it takes into account the ground's surface shape. Therefore, the ground length between two points will always be greater than or equals to the linear length between those points. The difference between these two measures would depend on how rough the ground is. If the ground is smooth, both measures would be almost the same value. Otherwise, ground length may be significantly higher than linear length.

In order to calculate the ground length, the service will divide the imaginary straight line between the points given into 10 or less divisions, given that the divisions are greater than 10 metres. For each division, the service will calculate the height of its edge points and then it will calculate the length between those points using their height and the linear length between them. The result will be the addition of all these partial lengths.

6.2.2 Parameters

This service has the following parameters:

- **crs or srs [mandatory]:** Projection system used, It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.
- **points[mandatory]:** coordinates of the points between which the length is being measured, separated by commas. Xmin, Ymin, Xmax, Ymax. Point coordinates may be either oblique coordinates or world coordinates on the same projection crs/srs indicates.
- **format [mandatory]:** Response format. Possible values are:
    - **txt:** Response is a text string with the ground length in meters, the number of edge points of the divisions and the coordinates of all the edge points, separated by semicolons (";").
    - **Xml:** The result is an XML file with the measure in meters, the number of edge points of the divisions and the coordinates of all the edge points
    - **js:** Result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.
- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

- **id [mandatory if measuring in an oblique image]:** Id of the oblique image in which the measure is being taken. If no ID or an empty one is provided, the service would interpret that the measure is being taken in an ortho image.

6.2.3 Example

Here is an example of a request to this service:

```
http://www.blomurbex.com/v02/CalculateGroundLength?usertoken=********&points=-522975.29191485746,4960687.497070187,-522793.75397271273,4960272.065057527&srs=EPSG:3785&id=&format=xml
```

The response for this request should be

```
<result>
<length>343.9890713024825</length>
<numPoints>11</numPoints>
      <point>-522975.29191485746,4960687.497070187</point>
      <point>-522957.138120643,4960645.953868921</point>
      <point>-522938.9843264285,4960604.410667655</point>
      <point>-522920.830532214,4960562.867466389</point>
      <point>-522902.67673799954,4960521.324265123</point>
      <point>-522884.52294378506,4960479.781063857</point>
      <point>-522866.36914957064,4960438.237862591</point>
      <point>-522848.21535535617,4960396.694661325</point>
      <point>-522830.0615611417,4960355.151460059</point>
      <point>-522811.9077669272,4960313.608258793</point>
      <point>-522793.75397271273,4960272.065057527</point>
</result>
```

7 Measurements on Oblique Images

7.1 CalculateArea Service

7.1.1 Description

This service will calculate the area in **square meters** defined by a collection of points comprised of pixel coordinates in a given oblique image. The area calculated does not consider buildings.

7.1.2 Parameters

Parameters of the service are described below

- **points [mandatory]**: This parameter is a list of coordinates separated by commas. The following rules must be observed to provide the list:
    - The number of coordinates must be an even number
    - There must be at least 6 values (i.e., 3 points)
    - Odd coordinates, represent X coordinates.
    - Even coordinates, represent Y coordinates.
- **id [mandatory]**: This is the ID of the Oblique image used.
- **dtm [mandatory]:** Indicates if the BlomURBEX digital terrain model of the area must be used in the conversion or not. Possible values for this parameter are:
    - **yes:** The Digital Terrain Model of the Area will be used. This slows down the transformation process, but results are more precise.
    - **no:** The Digital Terrain Model of the Area will not be used.
- **format [mandatory]:** Response format. Possible values are:
    - **txt:** The result is text string containing the area of the surface defined by the list of coordinates.
    - **Xml:** The result is an XML file containing the area of the surface defined by the list of coordinates.
    - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.
- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom

7.1.3 Example

Here is an example call for this service:

```
http://www.blomurbex.com/v02/CalculateArea?usertoken=********&format=xml&id=N-004.705806_40.658386_060830&points=100,1000,200,2000,105,600&dtm=yes
```

As the response is requested in XML format, it should be a file like this:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<result>
      <Area>1208.6383056640625</Area>
</result>
```

7.2 CalculateHeight Service

7.2.1 Description

This service calculates the height, **in meters**, between two given points. Each point is defined by pixel coordinates within a given oblique image. The first point has to be a point on the terrain, and the second point will be considered as located on the vertical of that point.

7.2.2 Parameters

Parameters of the service are described below:

- **x1 [mandatory]:** Starting Pixel X coordinate. For a valid measurement it must be located on the ground level of the image.
- **y1 [mandatory]:** Starting Pixel Y coordinate. For a valid measurement it must be located on the ground level of the image.
- **x2 [mandatory]:** Ending Pixel X coordinate.
- **y2 [mandatory]:** Ending Pixel Y coordinate.
- **id [mandatory]**: Id of the Oblique Image.
- **dtm [mandatory]:** Indicates if the BlomURBEX digital terrain model of the area must be used in the conversion or not. Possible values for this parameter are:
    - **yes:** The Digital Terrain Model of the Area will be used. This slows down the transformation process, but results are more precise.
    - **no:** The Digital Terrain Model of the Area will not be used.
- **format [mandatory]:** Response format. Possible values are:
    - **txt:** Result is a text string containing the calculated height.
    - **Xml:** Result is a XML file containing the calculated height.
    - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.
- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

7.2.3 Example

A request to the service could be:

```
http://www.blomurbex.com/v02/CalculateHeight?usertoken=********&format=xml&id=
N-004.705806_40.658386_060830&x1=10&x2=20&y1=100&y2=200&dtm=yes
```

The service response should be an XML file like this:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<result>
      <Height>20.988949809925465</Height>
</result>
```

7.3 CalculateLength Service

7.3.1 Description

This service calculates the total length, in **meters,** of the line defined by a list of points passed to the service. Calculations are done considering the terrain only, but not the buildings.

7.3.2 Parameters

Parameters of the service are described below

- **points [mandatory]**: This parameter is a list of coordinates separated by commas. The following rules must be observed to provide the list:
    - The number of coordinates must be an even number
    - Odd coordinates, represent X coordinates.
    - Even coordinates, represent Y coordinates.
- **id [mandatory]**: This is the ID of the Oblique image used.
- **dtm [mandatory]:** Indicates if the BlomURBEX digital terrain model of the area must be used in the conversion or not. Possible values for this parameter are:
    - **yes:** The Digital Terrain Model of the Area will be used. This slows down the transformation process, but results are more precise.
    - **no:** The Digital Terrain Model of the Area will not be used.
- **format [mandatory]:** Response format. Possible values are:
    - **txt:** The result is text string containing the length of the polyline defined by the list of coordinates.
    - **Xml:** The result is an XML file containing the length of the polyline defined by the list of coordinates.
    - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.
- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom:

7.3.3 Example

A request to the service could be:

```
http://www.blomurbex.com/v02/CalculateLength?usertoken=********&format=xml&id=
N-004.705806_40.658386_060830&points=10,20,20,60,80,90,150,160&dtm=yes
```

And the service's response in this case:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<result>
      <Length>50.878919097178304</Length>
</result>
```

7.4 CalculateBearing Service

7.4.1 Description

This service calculates the Azimuth angle defined by two points. Each point is defined by its pixel coordinates referred to a given oblique image.

First point defines the vector origin and the second defines its end. Azimuth angle is calculated as the angle of this vector with respect to the geographical North.

7.4.2 Parameters

Parameters of the service are described below:

- **x1 [mandatory]:** Starting Pixel X coordinate.
- **y1 [mandatory]:** Starting Pixel Y coordinate.
- **x2 [mandatory]:** Ending Pixel X coordinate.
- **y2 [mandatory]:** Ending Pixel Y coordinate.
- **id [mandatory]**: Id of the Oblique Image.
- **dtm [mandatory]:** Indicates if the BlomURBEX digital terrain model of the area must be used in the conversion or not. Possible values for this parameter are:
    - **yes:** The Digital Terrain Model of the Area will be used. This slows down the transformation process, but results are more precise.
    - **no:** The Digital Terrain Model of the Area will not be used.
- **format [mandatory]:** Response format. Possible values are:
    - **txt:** Result is a text string containing the calculated bearing angle.
    - **Xml:** Result is a XML file containing the calculated bearing angle.
    - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.
- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

7.4.3 Example

A request to the service could be:

```
http://www.blomurbex.com/v02/CalculateBearing?usertoken=********&format=xml&id
=N-004.705806_40.658386_060830&x1=100&x2=1000&y1=200&y2=2000&dtm=yes
```

And the response should be:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<result>
      <Bearing> 156.6780120996657</Bearing>
</result>
```

7.5 CalculateElevationInOblique Service

7.5.1 Description

This service calculates the elevation (height from sea level) in a given point of a given oblique image. This calculation considers only the terrain height, but not the building heights. The returned value is in **meters**.

7.5.2 Parameters

Parameters of the service are described below:

- **x [mandatory]:** Pixel X coordinate.
- **y [mandatory]:** Pixel Y coordinate.
- **id [mandatory]**: Id of the Oblique Image.
- **format [mandatory]:** Response format. Possible values are:

  - **txt:** Result is a text string containing the elevation of the point.
  - **Xml:** Result is a XML file containing the elevation of the point.
  - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

7.5.3 Example

A request to the service could be:

```
http://www.blomurbex.com/v02/CalculateElevationInOblique?usertoken=********&format=xml&id=N-004.705806_40.658386_060830&x=100&y=1000
```

The service's response should be:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<result>
      <ElevationInOblique>1060.088049900917</ElevationInOblique>
</result>
```

7.6 CalculateHeightMeasuringAngle Service

7.6.1 Description

This service will calculate the angle (in radians) between the world's vertical axis and the image's horizontal axis in a certain oblique image. This is used by the Javascript API's height measure control, so that the height measure tool can only move up and down along the world's vertical axis and the height measure would me more accurate.

7.6.2 Parameters

Parameters of the service are described below:

- **x [mandatory]:** Pixel X coordinate of the point in which the measure starts.
- **y [mandatory]:** Pixel Y coordinate of the point in which the measure starts.
- **id [mandatory]**: Id of the Oblique Image.
- **format [mandatory]:** Response format. Possible values are:
    - **txt:** Result is a text string containing the calculated angle.
    - **Xml:** Result is a XML file containing the calculated angle.
    - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.
- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

7.6.3 Example

A request to the service could be:

```
http://www.blomurbex.com/v02/CalculateHeightMeasuringAngle?usertoken=********&
id=N+012.480965_41.896465_080810&x=3876.5&y=1948.5&function=myfunction&format=
js
```

The service response should be:

```
function myfunction() {
  return -1.4999928942249878;
}
```

7.7 CalculateGroundLength service

See section 0 for information regarding this service.

7.8 CalculateVerticalArea Service

7.8.1 Description

This service will calculate the vertical area in **square meters** defined by a collection of points comprised of pixel coordinates in a given oblique image. The vertical area calculated considers first two points as baseline and rest points to be above the baseline.

7.8.2 Parameters

Parameters of the service are described below

- **points [mandatory]**: This parameter is a list of coordinates separated by commas. The following rules must be observed to provide the list:
    - The number of coordinates must be an even number
    - There must be at least 6 values (i.e., 3 points)
    - Odd coordinates, represent X coordinates.
    - Even coordinates, represent Y coordinates.
- **id [mandatory]**: This is the ID of the Oblique image used.
- **dtm [mandatory]:** Indicates if the BlomURBEX digital terrain model of the area must be used in the conversion or not. Possible values for this parameter are:
    - **yes:** The Digital Terrain Model of the Area will be used. This slows down the transformation process, but results are more precise.
    - **no:** The Digital Terrain Model of the Area will not be used.
- **format [mandatory]:** Response format. Possible values are:
    - **txt:** The result is text string containing the vertical area of the surface defined by the list of coordinates.
    - **Xml:** The result is an XML file containing the vertical area of the surface defined by the list of coordinates.
    - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.
- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom

7.8.3 Example

Here is an example call for this service:

```
http://www.blomurbex.com/v02/CalculateVerticalArea?usertoken=********&id=N-004.697670_40.655723_060830&function=FRequestGetVerticalAreaSrc1870&points=1847.5,2088.5,1969,2089.5,1959.5,2014.5,1835,2013.5,1847.5,2088.5&format=XML&dtm=yes
```

As the response is requested in XML format, it should be a file like this:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
-<result>
      <verticalarea> 237.105</verticalarea>
</result>
```

8 3D Services

8.1 GetDTM Service

8.1.1 Description

This service returns the BD3 DTM file for a tile specified.

8.1.2 Parameters

The parameters for this service are:

- **Crs or srs [mandatory]:** Projection system. It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.
- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.
- **id [mandatory]:** Quadkey for the tile of whose DTM file will be returned by the service.
- **dtmLayer [optional]:** Preference list of dtmLayers, separated by commas, where the tile requested will be searched. The service will start the search in the first layer (if the user is allowed to check that layer) and if the tile is not found in that layer, it will continue with following layers. If omitted, default preference for each user will be applied. Example: dtmLayer=NASA,someOtherLayer.

8.1.3 Examples

```
http://www.blomurbex.com/v02/GetDTM?usertoken=********&id=12021000113100012&crs=EPSG:3785&dtmLayer=NASA
```

Above example should return the .bd3 file for the tile **12021000113100012.**

8.2 Get3DTile Service

8.2.1 Description

This service returns the 3D tile information requested.

8.2.2 Parameters

The parameters for this service are:

- **id [mandatory]**: Tile ID, it will be a sequence of digits, which only will be able to contain 0, 1, 2 and 3. This nomenclature corresponds to the way of codifying the Tiles for quadtrees.

- **Layer [mandatory]:** Code for the layer requested. Layers are different blocks of data.

- **crs or srs [mandatory]:** Projection system used. It must follow OGC nomenclature. For example **EPSG:32630** for UTM 30 North WGS84 or **EPSG:3785** for Spherical Mercator projection.

- **type mandatory]**: The type of tile requested. The values supported are:

    - IDX: Request for indexes file (.bi3).
    - MDL: Request for model file (.bp3).
    - TEX: Request for texture file (images, .bt3).

- **sid [mandatory]:** Identifier of the resource, for model an texture. A tile id can have several resources.

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom

- **time [optional]** : A date following the structure YYYYMMDD or YYYYMM or YYYY. The service will use the most current images available, whose date is previous to the requested date. It is also possible to use a range of dates with the structure YYYYMMDD-YYYYMMDD, in this case the images used will be the most current ones available in the specified date range.

  If the parameter is omitted the system will submit the most recent version of the tile the user is allowed to retrieve.

8.2.3 Results

The result of the request will be either a .bi3 file, a .bp3 file and a .bt3 file, depending on the parameters of the request. If the user is not authorized to obtain the requested data or if the requested tile is not available, then an empty file will be returned.

8.2.4 Example

```
http://www.blomurbex.com/v02/GetTile3D?usertoken=********&id=120221202330002001&crs=epsg:3785&type=mdl&layer=blom_lod1_10&sid=0
```

This request will return a .bp3 model file.

9 LiDAR Services

9.1 Introduction

BlomURBEX serves Lidar cloud points under de 3D Tile standard format defined by the Cesium project in ECEF projection system (EPSG:4978) in 3D tiled system: regular Octree.

The transfer format of the Web Services will based on the open 3D Tiles Cesium specs:

https://github.com/AnalyticalGraphicsInc/3d-tiles

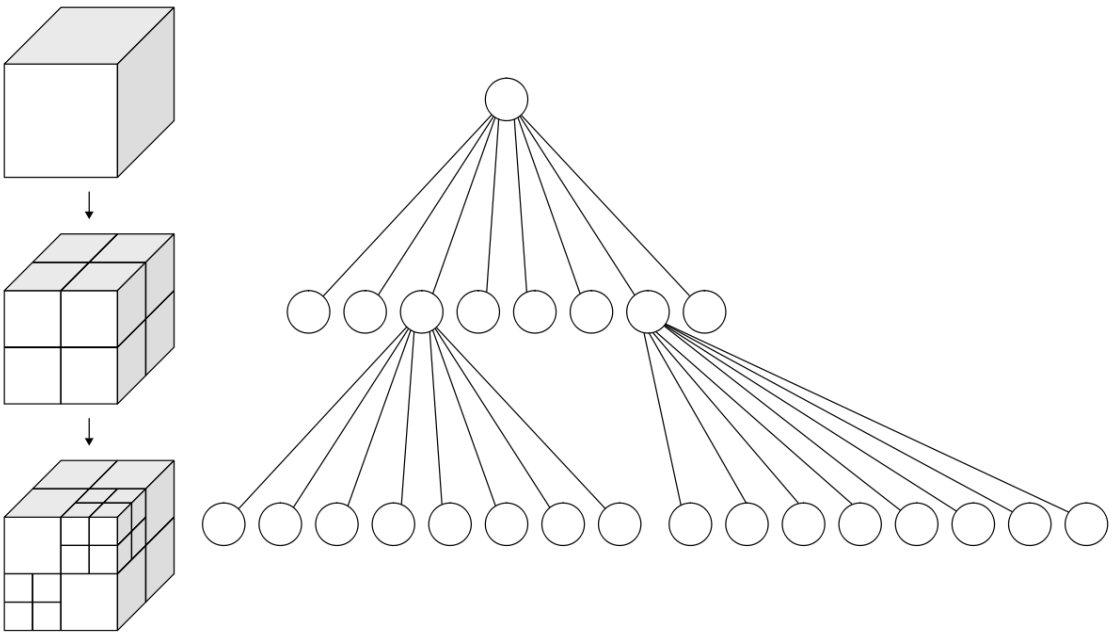
The format is composed by a Json file with the index and location (RTC anchor) info of each 3D tile and set of 3D tiles, and a binary file with the geographic and feature information of the Lidar cloud of points.

The binary format is based on the GLTF standard.

https://github.com/AnalyticalGraphicsInc/3d-tiles/blob/master/TileFormats/PointCloud/README.md

9.1.1 Octree

An octree is a tree data structure in which each internal node has exactly eight children. Octrees are most often used to partition a three-dimensional space by recursively subdividing it into eight octants. Octrees are the three-dimensional analog of quadtrees.



This octree will have global extension where the first level of the Octree, cube 0, will contain the whole world.

The maximum level of resolution of the Octree will be a cube with a size of 1 m^3.

According this cube minimum size it will be necessary to implement 25 octree levels (0 to 24) to contain the whole world, where level 0 is the biggest cube and level 24 is smallest and detailed cube.

Earth Radius WGS84: 6378137 meters

Earth Diameter WGS84: 12756274 meters

Octree Level 0 size: 2^{24} = 16777216 meters

Level	Edge Size
0	16777216
1	8388608
2	4194304
3	2097152
4	1048576
5	524288
6	262144
7	131072
8	65536
9	32768
10	16384
11	8192
12	4096
13	2048
14	1024
15	512
16	256
17	128
18	64
19	32
20	16
21	8
22	4
23	2
24	1

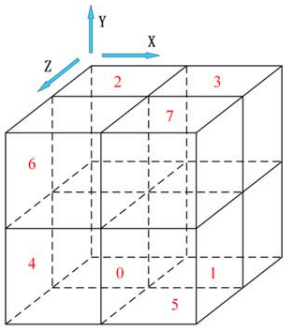
9.1.2 Octree Naming

The Octree naming system will use the existing QTree Urbex definition.

Each cube will be split in 8 children cubes.

The children cubes of the Z bottom level will use the QTree naming system: 0,1,2,3

The children cubes of the Z top leve will use the ‘extended’ naming system: 4,5,6,7



Figure

X range: (XMIN~XMAX)
Y range: (YMIN~YMAX)
Z range: (ZMIN~ZMAX)
The midpoints of the range in all directions are: XMID, YMID, ZMID.

Information of parent node

Number	Range
0	(XMIN~XMID, YMIN~YMID, ZMIN~ZMID)
1	(XMID~XMAX, YMIN~YMID, ZMIN~ZMID)
2	(XMIN~XMID, YMID~YMAX, ZMIN~ZMID)
3	(XMID~XMAX, YMID~YMAX, ZMIN~ZMID)
4	(XMIN~XMID, YMIN~YMID, ZMID~ZMAX)
5	(XMID~XMAX, YMIN~YMID, ZMID~ZMAX)
6	(XMIN~XMID, YMID~YMAX, ZMID~ZMAX)
7	(XMID~XMAX, YMID~YMAX, ZMID~ZMAX)

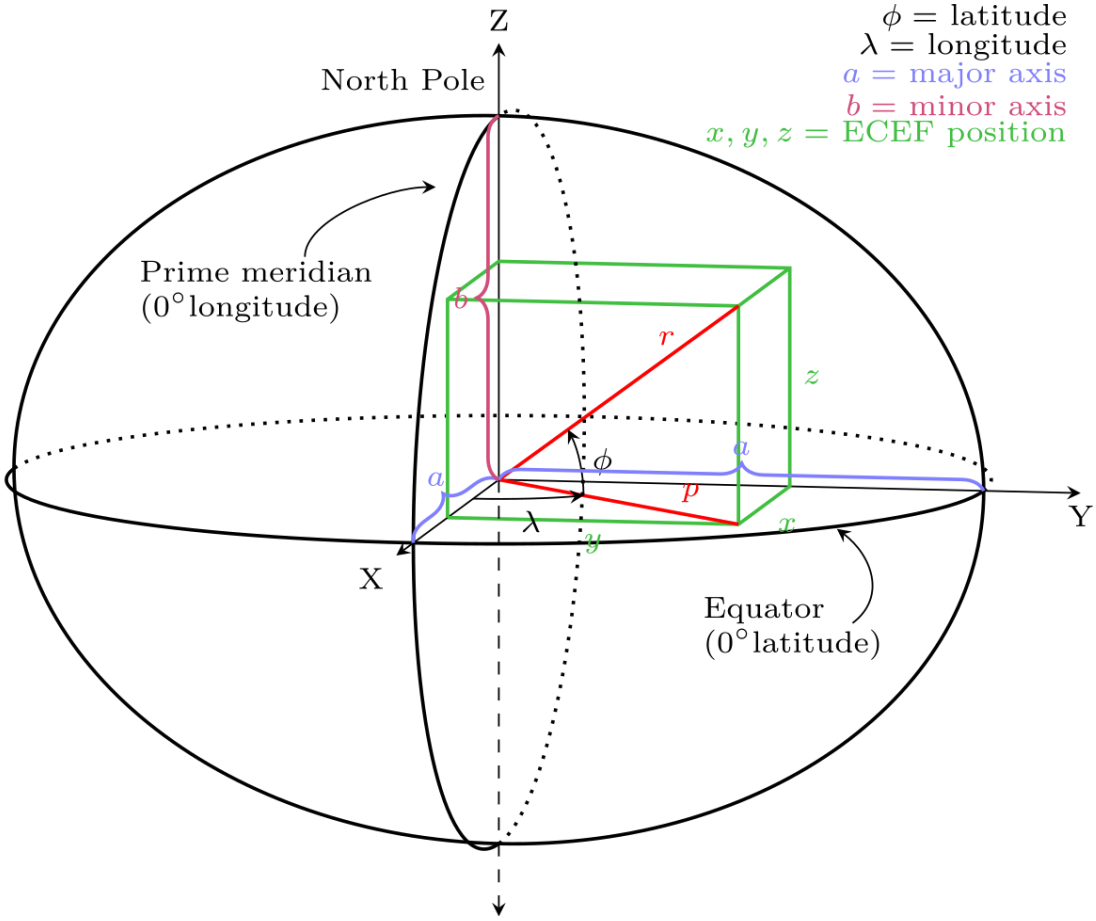
Information of child nodes

9.1.3 Projection System

The projection system for each 3D tile of the Octree will be the ECEF system.

This system is a geographic coordinate system and Cartesian coordinate system, and is sometimes known as a "conventional terrestrial" system. It represents positions as an X, Y, and Z coordinate. The point (0,0,0) is defined as the center of mass of the earth, hence the name "earth-centered." Its axes are aligned with the international reference pole (IRP) and international reference meridian (IRM) that are fixed with respect to the surface of the earth, hence the description "earth-fixed." This term can cause confusion since the earth does not rotate about the z-axis (unlike an inertial system such as ECI), and is therefore alternatively called ECR.

The z-axis extends through True north, which does not coincide with the instantaneous earth rotational axis. The slight "wobbling" of the rotational axis is known as polar motion. The x-axis intersects the sphere of the earth at 0° latitude (the equator) and 0° longitude (prime meridian in Greenwich). This means that ECEF rotates with the earth, and therefore coordinates of a point fixed on the surface of the earth do not change. Conversion from a WGS84 datum to ECEF can be used as an intermediate step in converting velocities to the north east down coordinate system



9.2 WFSLidarProjectsInfo service

9.2.1 Description

The WFS Lidar Projects Info service is a web service based in the WFS standard protocol for serving vector information.

BlomURBEX™ offers the WFS service to allow users to access to vector information related with the Lidar Projects coverage and metadata information following OGC's regulation and directives.

The service uses the http protocol with the parameters normalized by OGC, which define version, type of request, coordinate system, bounding box, etc.

The service supports several input projection systems, but vector information will provided on EPSG:4326 projection.

The WFS supported versions are 1.0.0 & 1.1.0.

9.2.2 Connection

To connect to the BlomURBEX™ WFS Lidar Projects Info service from a GIS client, it is necessary to setup the following URL:

```
http://www.blomurbex.com/v02/WFSLidarProjectsInfo?REQUEST=GetCapabilities&SERV
ICE=WFS&ServiceName=WFS&version=1.1.0&userToken=****************
```

In this request, `usertokenstring` must be replaced by the unique platform access token provided by Blom.

GetCapabilities returns information about available features, capabilities and spatial operators.

9.2.3 SSL service (WFSLidarProjectsInfoSSL)

In order to support SSL web protocol (HTTPS) it has been added a new clone service on this URL:

```
https://www.blomurbex.com/v02/WFSLidarProjectsInfoSSL?REQUEST=GetCapabilities&
SERVICE=WFS&ServiceName=WFS&version=1.1.0&userToken=****************
```

The response of this service will be equal to standard HTTP service, but with HTTPS links support.

9.2.4 Parameters

Parameters defined by OGC standard are:

- **Version [mandatory]**: This parameter is the WFS version that is used for this request. Only two values are accepted in the current API version:
    - **1.0.0**
    - **1.1.0**
- **Request [mandatory]:** Possible values are:
    - GetCapabilites: used to obtain the web services descriptor xml file
    - DescribeFeatureType: returns the feature descriptor xml file.
    - GetFeature: used to retrieving a feature query.
- **Service [mandatory]:** The service name: WFS
- **Typename:** mandatory for GetFeature and DescribeFeatureType requests. It is the name of the feature layer provided by the service. Street images will be provided by the name of ‘panorama’.
- **Usertoken [mandatory]**: User’s Token for authentication purposes. This token is unique for each client and must be requested from Blom.
- **Bbox [optional]:** Optional for GetFeature request. It defines the bounding box extent of the query.
- **SrsName [optional]:** it defines the projection system of the input information (i.e. bbox).

9.2.5 Result

The result of the GetCapabilities & DescribeFeatureType will be a Xml file according the OGC standard definition.

There are available two types of features: lidarblock & lidarproject.

- Lidarblock: it returns the available blocks polygons of Lidar projects.
- Lidarproject: It returns the Lidar project extents.

The output of GetFeature request is a Gml file with next parameters:

- **Geometry**: Polygon of the requested feature. Provided in EPSG:4326 projection
- **Name (lidarproject):** project name
- **code (lidarproject):** project code.
- **Init_date (lidarproject):** project starting capture date.
- **End_date (lidarproject):** project ending capture date.
- **Scanner_description (lidarproject):** scanner type.
- **Block_id (lidarblock):** Lidar block identifier.
- **Project_name (lidarblock):** Lidar project name
- **Extended_properties:** Additional parameters in Json format.(optional)

9.2.6 Example

```
http://www.blomurbex.com/v02/WFSLidarProjectsInfo?SERVICE=WFS&REQUEST=GetFeatu
re&ServiceName=WFS&version=1.1.0&userToken=**********************&SRS=urn:ogc
:def:crs:EPSG::4326&TYPENAME=lidarblock&BBOX=17.94770968534563,59.395481283060
01,17.95191002466295,59.39665835598541,urn:x-ogc:def:crs:EPSG:6.9:4326
```

The result of the previous url:


```
<?xml version="1.0" encoding="UTF-8"?>
- <wfs:FeatureCollection xsi:schemaLocation="http://mapserver.gis.umn.edu/mapserver https://www.blomurbex.com/v02/WFSLidarProjectsInfoSSL?
SERVICE=WFS&VERSION=1.1.0&REQUEST=DescribeFeatureType&TYPENAME=lidarblock&OUTPUTFORMAT=text/xml;%20subtype=gml/3.1.1 http://www.opengis.net/w
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:gml="http://www.opengis.net/gml" xmlns:ms="http://mapserver.gis.umn.edu/mapserver">
  - <gml:boundedBy>
    - <gml:Envelope srsName="EPSG:4326">
        <gml:lowerCorner>60.039975 11.111993</gml:lowerCorner>
        <gml:upperCorner>60.050903 11.120633</gml:upperCorner>
      </gml:Envelope>
    </gml:boundedBy>
  - <gml:featureMember>
    - <ms:lidarblock gml:id="lidarblock.-2147483517">
      - <gml:boundedBy>
        - <gml:Envelope srsName="EPSG:4326">
            <gml:lowerCorner>60.049374 11.117810</gml:lowerCorner>
            <gml:upperCorner>60.050903 11.120633</gml:upperCorner>
          </gml:Envelope>
        </gml:boundedBy>
      - <ms:geometry>
        - <gml:Polygon srsName="EPSG:4326">
          - <gml:exterior>
            - <gml:LinearRing>
                <gml:posList srsDimension="2">60.050688 11.117810 60.050903 11.120287 60.050293 11.120498 60.049904 11.120606 60.049596 11.120633
                60.049374 11.120603 60.049460 11.118096 60.049611 11.118116 60.049789 11.118100 60.050100 11.118014 60.050688 11.117810
                </gml:posList>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </ms:geometry>
        <ms:block_id>50376_000010</ms:block_id>
        <ms:project_name>50376_TrondheimsveienLindeberg</ms:project_name>
        <ms:pk_lidar_block>-2147483517</ms:pk_lidar_block>
        <ms:extended_properties>{"Las_Path": "ftp://terratec:Z3chPN18@ftp.terratec.no/Customer/BlomGeomatics/Trondheimsveien/LAS/50376_000010.las"}
          </ms:extended_properties>
      </ms:lidarblock>
    </gml:featureMember>
  - <gml:featureMember>
```

9.3 GetLidarTile Service

9.3.1 Description

This service returns the Lidar tile information requested, in pnts or Json format.

The load of the Lidar model must be handled by the first request of the main Json file of the project. An Alias has been generated for this purpose.

The HTTPS support is available under GetLidarTileSSL service alias.

9.3.2 Parameters

The parameters for this service are:

- **id [mandatory]**: Tile ID, it will be a sequence of digits, which only will be able to contain 0, 1, 2, 4, 5, 6, and 7. This nomenclature corresponds to the way of codifying the Tiles for octrees.
- **layer [mandatory]:** Code for the layer requested. Layers are different data sources of data.
- **project [mandatory]:** Code for the project requested. Projects are different blocks of data.
- **type [mandatory]:** Request format type. Allowed values are '**json**', '**pnts**' and '**root**'. The Json and root format will be valid for the Json files of the 3D Tile format. Root type will obtain the main Json file of the project, necessary for the accessing of the next json files. Pnts format will return the binary cloud points.
- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom

9.3.3 Results

The result of the request will be json & pnts files. If the user is not authorized to obtain the requested data or if the requested tile is not available, then an empty file will be returned.

9.3.4 Example

```
http://www.blomurbex.com/v02/GetLidarTile?usertoken=********&layer=blom_lidar_
st&project=nooslocdmb_171127&id=7414315757650644067 1&type=pnts
```

This request will return a .pnts model file with gzip compression.

http://www.blomurbex.com/v02/GetLidarTile?usertoken=******&layer=blom_lidar_st&project=nooslocdmb_171127&id=74143157576506442&type=json**

This request will return a Json model file.


```
{
	"asset" :
	{
		"version" : "0.0"
	},
	"geometricError" : 100.0,
	"refine" : "ADD",
	"root" :
	{
		"boundingVolume" :
		{
			"box" :
			[
				638.0,
				-114.0,
				-366.0,
				64.0,
				0.0,
				0.0,
				0.0,
				64.0,
				0.0,
				0.0,
				0.0,
				64.0
			]
		},
		"children" :
		[
			{
				"boundingVolume" :
				{
					"box" :
					[
						670.0,
						-146.0,
						-398.0,
						32.0,
						0.0,
						0.0,
						0.0,
						32.0,
						0.0,
						0.0,
						0.0,
						32.0
					]
				},
				"children" :
				[
					{
						"boundingVolume" :
						{
							"box" :
							[
								686.0,
								-162.0,
								-382.0,
								16.0,
								0.0,
								0.0,
								0.0,
								16.0,
								0.0,
								0.0,
								0.0,
								16.0
							]
						},
```


…

It has been added an ‘Alias’ for the accessing of the project Json file:

http://www.blomurbex.com/v02/GetLidarTile/tileset.json?usertoken=******&layer=blom_lidar_st&project=nooslocdmb_171127&id=74143157576506442&type=root**

9.4 GetDtm3DTile Service

9.4.1 Description

This service returns the Dtm layer in 3D format for the Lidar services based on the TMS system (https://en.wikipedia.org/wiki/Tile_Map_Service) in a geodetic model.

It returns the Dtm3D tile information requested, in terrain or Json format.

The load of the Dtm3D model must be handled by the first request of the main Json file of the project. An Alias has been generated for this purpose.

9.4.2 Parameters

The parameters for this service are:

- **x [optional]:** X axis value on TMS schema.
- **y [optional]: Y** axis value on TMS schema.
- **z [optional]:** Z axis value on TMS schema.

The absence of X,Y or Z values will handled as a Json request.

- **layer [mandatory]:** Code for the layer requested. Layers are different data sources of data.
- **time [optional]**: A date following the structure YYYYMMDD or YYYYMM or YYYY. The service will use the most current images available, whose date is previous to the requested date. It is also possible to use a range of dates with the structure YYYYMMDD-YYYYMMDD, in this case the images used will be the most current ones available in the specified date range.

  If the parameter is omitted the system will submit the most recent version of the tile.

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom

9.4.3 Results

The result of the request will be json & terrain files. If the user is not authorized to obtain the requested data or if the requested tile is not available, then an empty file will be returned.

9.4.4 Example

http://www.blomurbex.com/v02/GetDtm3DTile?**usertoken**=********&z=0&x=1&y=0&layer=srtm15&time=2014

This request will return a .terrain model file.

http://www.blomurbex.com/v02/GetDtm3DTile?usertoken=********&layer=srtm15&time=2014

This request will return the Json model file.

```
{
  "tilejson": "2.1.0",
  "format": "heightmap-1.0",
  "version": "1.0.0",
  "scheme": "tms",
  "tiles": ["GetDtm3DTile?z={z}&x={x}&y={y}&v={version}"]
}
```

9.5 FindLidarProjectByExtent

9.5.1 Description

This service will locate all the Lidar projects whose extents intersects a given extent.

9.5.2 Parameters

The parameters needed in the request URL are the following:

- **bbox [mandatory]**: Bounding Box defined by four coordinates that specify the area in the specified projection system. Coordinates are provided in the following order: minimum X coordinate, minimum Y coordinate, maximum X coordinate, maximum Y coordinate.

- **crs or srs [mandatory]:** Projection system used, It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.

- **format [mandatory]:** Response format. Possible values are:
  - **txt:** Response is a text string with comma-separated values (“,”).
  - **Xml:** Response is an XML file.
  - **js:** Response is a JavaScript function that will return the value. This is for internal use of the JavaScript API. Requires the **function** parameter to specify the name of the function to return.
  - **Json:** Response is Json format.

- **time [optional]** : A date following the structure YYYYMMDD or YYYYMM or YYYY. The service will use the most current images available, whose date is previous to the requested date. It is also possible to use a range of dates with the structure YYYYMMDD-YYYYMMDD, in this case the images used will be the most current ones available in the specified date range.

  If the parameter is omitted the system will submit the most recent version of the tile.

- **layer [optional]:** This parameter indicates the Lidar layer where searching will be performed.

- **usertoken [mandatory]**: User’s Token for authentication purposes. This token is unique for each client and must be requested to Blom

9.5.3 Error messages

If no projects are found for the specified extent an Error message is returned.

If the response format was specified as a text, the response message is:

```
ERROR, Error message text
```

If the response format is XML the error XML file is:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<result>
      <Error>
      ERROR, Error message text
      </Error>
</result>
```

9.5.4 Results

For each Lidar Project located the following information is provided:

- Code: Lidar project code
- Name: Lidar project name
- Init_date: Starting capture date
- End_date: Ending capture date
- Layer: Lidar project layer
- Xcp: X location of the project centroid (EPSG:4978)
- Ycp: Y location of the project centroid (EPSG:4978)
- Zcp: Z location of the project centroid (EPSG:4978)
- Layer: Lidar layer name
- Dtm_3d_layer: Dtm 3D layer name to be used for this proyect.
- Copyright: Lidar project copyright
- Bbox: Project bbox geometry in WKT format (EPSG:4326)

9.5.5 Example

```
http://www.blomurbex.com/v02/FindLidarProjectByExtent?srs=EPSG:4326&bbox=10.218658438246603,60.153809133223405,10.220031729266088,60.15449257649393&usertoken=**********&format=xml&layer=blom_lidar_st
```

The result of previous url:

```
<metadata>
    <code>nooslocdmb_171127</code>
    <name>50376_TrondheimsveienLindeberg</name>
    <init_date>2017/Nov/27</init_date>
    <end_date>2017/Nov/27</end_date>
    <layer>blom_lidar_st</layer>
    <xcp>3132530.0</xcp>
    <ycp>615686.0</ycp>
    <zcp>5503334.0</zcp>
    <dtm3dlayer>norway10m</dtm3dlayer>
    <copyright>Blom</copyright>
    <bbox>POLYGON((11.1117114795147 60.0325040500015,11.1210453023079
    60.0325040500015,11.1210453023079
    60.0587856614726,11.1117114795147
    60.0587856614726,11.1117114795147 60.0325040500015))</bbox>
  </metadata>
```

10 3D Mesh Services

10.1 Introduction

BlomURBEX serves 3D Textured Mesh under de 3D Tile standard format defined by the Cesium project in ECEF projection system (EPSG:4978) in 2D tiled system: Mercator Quadtree (EPSG:3857).

The transfer format of the Web Services will based on the open 3D Tiles Cesium specs:

https://github.com/AnalyticalGraphicsInc/3d-tiles

The format is composed by a Json file with the index and location (RTC anchor) info of each 3D tile and set of 3D tiles, and a binary file with the geographic and feature information of the Lidar cloud of points.

The binary format is based on the GLTF standard.

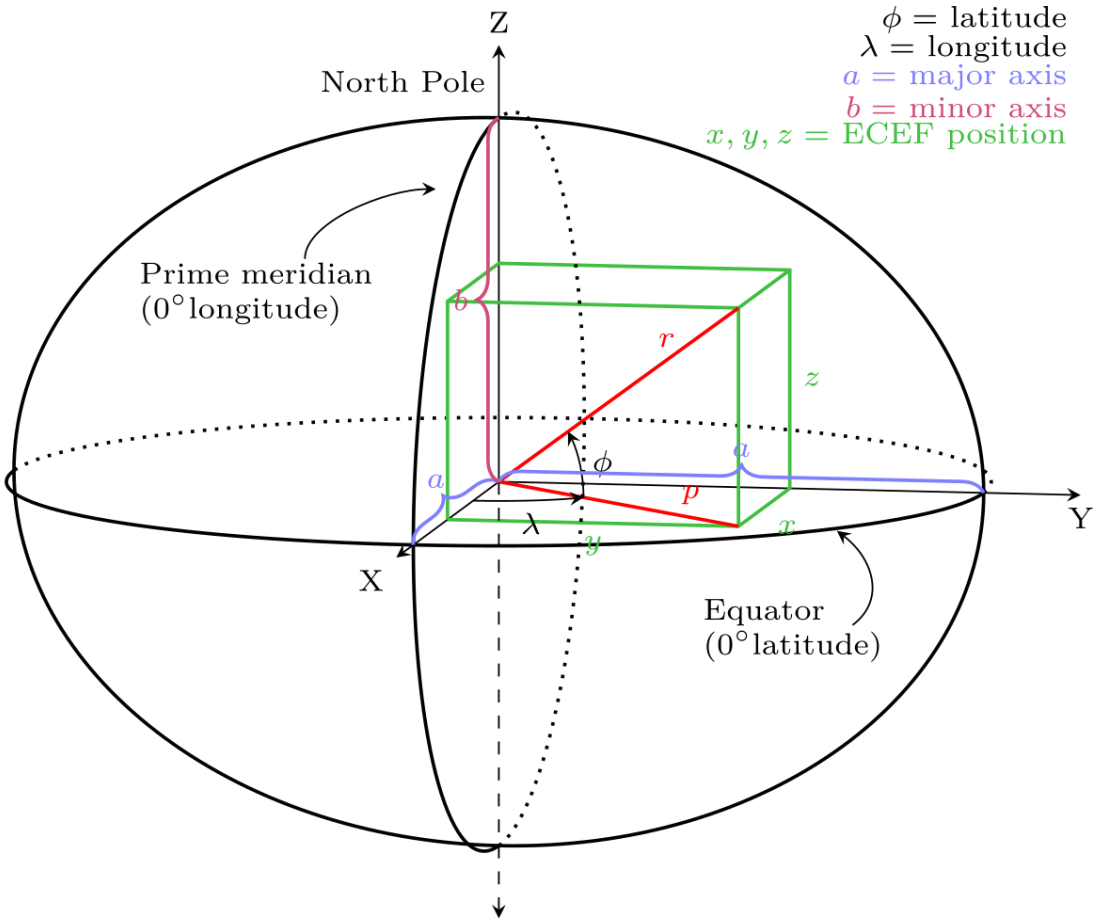
https://github.com/AnalyticalGraphicsInc/3d-tiles/blob/master/TileFormats/PointCloud/README.md

10.1.1 Projection System

The projection system for each 3D tile of the Octree will be the ECEF system.

This system is a geographic coordinate system and Cartesian coordinate system, and is sometimes known as a "conventional terrestrial" system. It represents positions as an X, Y, and Z coordinate. The point (0,0,0) is defined as the center of mass of the earth, hence the name "earth-centered." Its axes are aligned with the international reference pole (IRP) and international reference meridian (IRM) that are fixed with respect to the surface of the earth, hence the description "earth-fixed." This term can cause confusion since the earth does not rotate about the z-axis (unlike an inertial system such as ECI), and is therefore alternatively called ECR.

The z-axis extends through True north, which does not coincide with the instantaneous earth rotational axis. The slight "wobbling" of the rotational axis is known as polar motion. The x-axis intersects the sphere of the earth at 0° latitude (the equator) and 0° longitude (prime meridian in Greenwich). This means that ECEF rotates with the earth, and therefore coordinates of a point fixed on the surface of the earth do not change. Conversion from a WGS84 datum to ECEF can be used as an intermediate step in converting velocities to the north east down coordinate system



10.2 WFS3DMeshProjectsInfo service

10.2.1 Description

The WFS 3D Mesh Projects Info service is a web service based in the WFS standard protocol for serving vector information.

BlomURBEX™ offers the WFS service to allow users to access to vector information related with the Lidar Projects coverage and metadata information following OGC's regulation and directives.

The service uses the http protocol with the parameters normalized by OGC, which define version, type of request, coordinate system, bounding box, etc.

The service supports several input projection systems, but vector information will provided on EPSG:4326 projection.

The WFS supported versions are 1.0.0 & 1.1.0.

10.2.2 Connection

To connect to the BlomURBEX™ WFS 3D Mesh Projects Info service from a GIS client, it is necessary to setup the following URL:

```
http://www.blomurbex.com/v02/WFS3DMeshProjectsInfo?REQUEST=GetCapabilities&SERVICE=WFS&ServiceName=WFS&version=1.1.0&userToken=****************
```

In this request, `usertokenstring` must be replaced by the unique platform access token provided by Blom.

GetCapabilities returns information about available features, capabilities and spatial operators.

10.2.3 SSL service (WFS3DMeshProjectsInfoSSL)

In order to support SSL web protocol (HTTPS) it has been added a new clone service on this URL:

```
https://www.blomurbex.com/v02/WFS3DMeshProjectsInfoSSL?REQUEST=GetCapabilities&SERVICE=WFS&ServiceName=WFS&version=1.1.0&userToken=****************
```

The response of this service will be equal to standard HTTP service, but with HTTPS links support.

10.2.4 Parameters

Parameters defined by OGC standard are:

- **Version [mandatory]**: This parameter is the WFS version that is used for this request. Only two values are accepted in the current API version:
  - **1.0.0**
  - **1.1.0**
- **Request [mandatory]:** Possible values are:
  - GetCapabilites: used to obtain the web services descriptor xml file
  - DescribeFeatureType: returns the feature descriptor xml file.
  - GetFeature: used to retrieving a feature query.
- **Service [mandatory]:** The service name: WFS
- **Typename:** mandatory for GetFeature and DescribeFeatureType requests. It is the name of the feature layer provided by the service. Street images will be provided by the name of ‘panorama’.
- **Usertoken [mandatory]**: User’s Token for authentication purposes. This token is unique for each client and must be requested from Blom.
- **Bbox [optional]:** Optional for GetFeature request. It defines the bounding box extent of the query.
- **SrsName [optional]:** it defines the projection system of the input information (i.e. bbox).

10.2.5 Result

The result of the GetCapabilities & DescribeFeatureType will be a Xml file according the OGC standard definition.

There is available one type of features: mesh3dproject. This is the project extent

The output of GetFeature request is a Gml file with next parameters:

- **Geometry**: Polygon of the requested feature. Provided in EPSG:4326 projection
- **Name:** project name

- **code:** project code.
- **Init_date:** project starting capture date.
- **End_date:** project ending capture date.
- **Scanner_description:** scanner type.
- **Extended_properties:** Additional parameters in Json format.(optional)

10.2.6 Example

```
http://www.blomurbex.com/v02/WFS3DMeshProjectsInfo?SERVICE=WFS&REQUEST=GetFeature&ServiceName=WFS&version=1.1.0&userToken=**********&SRS=urn:ogc:def:crs:EPSG::4326&TYPENAME=lidarblock&BBOX=17.94770968534563,59.39548128306001,17.95191002466295,59.39665835598541,urn:x-ogc:def:crs:EPSG:6.9:4326
```

The result of the previous url:


```
<wfs:FeatureCollection xmlns:ms="http://mapserver.gis.umn.edu/mapserver" xmlns:gml="http://www.opengis.net/gml" xmlns:wfs="http://www.opengis.net/wfs"
 xmlns:ogc="http://www.opengis.net/ogc" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://mapserver.gis.umn.edu/mapserver
 http://www.blomurbex.com/v02/WFSLidarProjectsInfo?
 SERVICE=WFS&VERSION=1.1.0&REQUEST=DescribeFeatureType&TYPENAME=mesh3dproject&OUTPUTFORMAT=text/xml;%20subtype=gml/3.1.1 http://www.opengis.net/wfs
 http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
  <gml:boundedBy>
    <gml:Envelope srsName="EPSG:4326">
      <gml:lowerCorner>59.917494 10.675705</gml:lowerCorner>
      <gml:upperCorner>59.924569 10.693519</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
  <gml:featureMember>
    <ms:mesh3dproject gml:id="mesh3dproject.noosloctae3m_180629">
      <gml:boundedBy>
        <gml:Envelope srsName="EPSG:4326">
          <gml:lowerCorner>59.917494 10.675705</gml:lowerCorner>
          <gml:upperCorner>59.924569 10.693519</gml:upperCorner>
        </gml:Envelope>
      </gml:boundedBy>
      <ms:geometry>
        <gml:Polygon srsName="EPSG:4326">
          <gml:exterior>
            <gml:LinearRing>
              <gml:posList srsDimension="2">
                59.924569 10.676050 59.924346 10.693519 59.917494 10.693170 59.917717 10.675705 59.924569 10.676050
              </gml:posList>
            </gml:LinearRing>
          </gml:exterior>
        </gml:Polygon>
      </ms:geometry>
      <ms:name>Oslo2018Demo3DMesh</ms:name>
      <ms:code>noosloctae3m_180629</ms:code>
      <ms:init_date>2018-06-27</ms:init_date>
      <ms:end_date>2018-06-29</ms:end_date>
      <ms:scanner_description>Aerial</ms:scanner_description>
      <ms:extended_properties>
        {"PERIMETER": "3.482 km", "ENCLOSED_A": "0.746 sq km"}
      </ms:extended_properties>
      <ms:data_source_name>blom3dmesh</ms:data_source_name>
    </ms:mesh3dproject>
  </gml:featureMember>
</wfs:FeatureCollection>
```

10.3 Get3DMeshTile Service

10.3.1 Description

This service returns the Lidar tile information requested, in b3dm or Json format.

The load of the 3D Mesh model must be handled by the first request of the main Json file of the project. An Alias has been generated for this purpose.

The HTTPS support is available under Get3DMeshTileSSL service alias.

10.3.2 Parameters

The parameters for this service are:

- **id [mandatory]**: Tile ID, it will be a sequence of digits, which only will be able to contain 0, 1, 2, and 3. This nomenclature corresponds to the way of codifying the Tiles for quadtree.
- **layer [mandatory]:** Code for the layer requested. Layers are different data sources of data.
- **project [mandatory]:** Code for the project requested. Projects are different blocks of data.
- **type [mandatory]:** Request format type. Allowed values are '**json**', '**b3dm**' and '**root**'. The Json and root format will be valid for the Json files of the 3D Tile format. Root type will obtain the main Json file of the project, necessary for the accessing of the next json files. B3dm format will return the binary 3d textured mesh.
- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom

10.3.3 Results

The result of the request will be json & b3dm files. If the user is not authorized to obtain the requested data or if the requested tile is not available, then an empty file will be returned.

10.3.4 Example

```
http://www.blomurbex.com/v02/Get3DMeshTile?usertoken=********&layer=blom3dmesh
&project=nooslocdmb3m_171127&id=7414315757650644067 1&type=b3dm
```

This request will return a *.b3dm model file with gzip compression.

http://www.blomurbex.com/v02/Get3DMeshTile?usertoken=******&layer=blom3dmesh&project=nooslocdmb3m_171127&id=74143157576506442&type=json**

This request will return a Json model file.


```
{
	"asset" :
	{
		"version" : "0.0"
	},
	"geometricError" : 100.0,
	"refine" : "ADD",
	"root" :
	{
		"boundingVolume" :
		{
			"box" :
			[
				638.0,
				-114.0,
				-366.0,
				64.0,
				0.0,
				0.0,
				0.0,
				64.0,
				0.0,
				0.0,
				0.0,
				64.0
			]
		},
		"children" :
		[
			{
				"boundingVolume" :
				{
					"box" :
					[
						670.0,
						-146.0,
						-398.0,
						32.0,
						0.0,
						0.0,
						0.0,
						32.0,
						0.0,
						0.0,
						0.0,
						32.0
					]
				},
				"children" :
				[
					{
						"boundingVolume" :
						{
							"box" :
							[
								686.0,
								-162.0,
								-382.0,
								16.0,
								0.0,
								0.0,
								0.0,
								16.0,
								0.0,
								0.0,
								0.0,
								16.0
							]
						},
```


…

It has been added an ‘Alias’ for the accessing of the project Json file:

http://www.blomurbex.com/v02/Get3DMeshTile/tileset.json?usertoken=******&layer=blom3dmesh&project=nooslocdmb3m_171127&id=74143157576506442&type=root**

10.4 Find3DMeshProjectByExtent

10.4.1 Description

This service will locate all the 3D Mesh projects whose extents intersects a given extent.

10.4.2 Parameters

The parameters needed in the request URL are the following:

- **bbox [mandatory]**: Bounding Box defined by four coordinates that specify the area in the specified projection system. Coordinates are provided in the following order: minimum X coordinate, minimum Y coordinate, maximum X coordinate, maximum Y coordinate.

- **crs or srs [mandatory]:** Projection system used, It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.

- **format [mandatory]:** Response format. Possible values are:
  - **txt:** Response is a text string with comma-separated values (“,”).
  - **Xml:** Response is an XML file.
  - **js:** Response is a JavaScript function that will return the value. This is for internal use of the JavaScript API. Requires the **function** parameter to specify the name of the function to return.
  - **Json:** Response is Json format.

- **time [optional]** : A date following the structure YYYYMMDD or YYYYMM or YYYY. The service will use the most current images available, whose date is previous to the requested date. It is also possible to use a range of dates with the structure YYYYMMDD-YYYYMMDD, in this case the images used will be the most current ones available in the specified date range.

  If the parameter is omitted the system will submit the most recent version of the tile.

- **layer [optional]:** This parameter indicates the 3D Mesh layer where searching will be performed.

- **usertoken [mandatory]**: User’s Token for authentication purposes. This token is unique for each client and must be requested to Blom

10.4.3 Error messages

If no projects are found for the specified extent an Error message is returned.

If the response format was specified as a text, the response message is:

```
ERROR, Error message text
```

If the response format is XML the error XML file is:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<result>
	<Error>
	ERROR, Error message text
	</Error>
</result>
```

10.4.4 Results

For each Lidar Project located the following information is provided:

- Code: 3D Mesh project code
- Name: 3D Mesh project name
- Init_date: Starting capture date
- End_date: Ending capture date
- Layer: 3D Mesh project layer
- Xcp: X location of the project centroid (EPSG:4978)
- Ycp: Y location of the project centroid (EPSG:4978)
- Zcp: Z location of the project centroid (EPSG:4978)

10.4.5 Example

http://www.blomurbex.com/v02/Find3DMeshProjectByExtent?srs=EPSG:4326&bbox=10.218658438246603,60.153809133223405,10.220031729266088,60.15449257649393&usertoken=**********&format=xml&layer=blom3dmesh

The result of previous url:

```
<result>
	<metadata>
		<code>nooslocdmb3m_171127</code>
		<name>50376_TrondheimsveienLindeberg</name>
		<init_date>2017/Nov/27</init_date>
		<end_date>2017/Nov/27</end_date>
		<layer>blom3dmesh</layer>
		<xcp>3132530.0</xcp>
		<ycp>615686.0</ycp>
		<zcp>5503334.0</zcp>
	</metadata>
</result>
```

11 Geocoding Services

BlomURBEX provides services to convert a street address into its coordinates (geocoding) and the other way around (reverse geocoding).

There are two possible providers of geocoding/reverse geocoding data: The free one (openStreetMaps) or the charging one (deCarta). Every BlomURBEX usertoken will be able to use the free provider, but only users who subscribe to the paying geocoding service will be entitled to use the charging provider.

11.1 Geocoding Service

11.1.1 Description

This service returns the coordinates for a given street address.

11.1.2 Parameters

The parameters for this service are:

- **Crs or srs [mandatory]:** Projection system. It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.
- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.
- **format [mandatory]:** Response format. Possible values are:
    - **txt:** Result is a text string containing the calculated angle.
    - **Xml:** Result is a XML file containing the calculated angle.
    - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.
- **service [optional]:** The service provider the geocoding data will be retrieved of. Possible values are 'free' or 'provider1'. If omitted, the free provider will be used.

The service will need to receive some parameters depending on which kind of provider was chosen.

For the free provider, the parameters needed will be:

- **address [mandatory for free provider]:** The address to geocode, in the format: *Number Street, municipality, country subdivision, country*. Depending on the country, the order of the first two parameters will be *street number* or *number street*. It is important not to include any extra commas so the parsing of the address is not interfered. Any parameter can be omitted.

For charging provider, the parameters needed for geocoding an address are:

- **country [mandatory]:** ISO 3-letter country code for the country where the address is located.
- **lang [mandatory]:** ISO 2-letter code describing the locale in which the address is written. Supporter locales at the moment are Spanish (es), English (en), French (fr), German (de) and Italian (it). When trying to geocode in a country whose language is not supported, use 'en'.
- **street[optional]:** Street name.
- **number [optional]:** Number of the address to geocode.
- **municipality [optional]:** Municipality/City
- **countrySubdiv [optional]:** Country subdivision. Province, state …
- **pCode [optional]:** Zip/Postal code of the address.

11.1.3 Examples

11.1.3.1 Geocoding with free provider and javascript format

```
http://www.blomurbex.com/v02/GetGeocoding?usertoken=********&format=js&functio
n=Fgeocode1236&address=castellana%20100%2C%20madrid%2C%20spain&service=free
```

Above example should deliver this response, consisting of a javascript function returning a JSON object.:

```
function Fgeocode1236 () { return {'name': 'castellana 100, madrid,
spain',
'candidates': [
{coordX:-410340.3872960271,coordY:4935660.475226232,address:"Paseo de
la Castellana, Fuencarral-El Pardo, Madrid, El Goloso, España"},
{coordX:-410937.1077883844,coordY:4930356.869533497,address:"Paseo de
la Castellana, Chamberí, Chamartín, Madrid, 28010, El Goloso,
España"},
{coordX:-410438.64166397037,coordY:4935158.70956552,address:"Paseo de
la Castellana, Tetuán, Madrid, El Goloso, España"},
{coordX:-410339.58412839344,coordY:4935749.741013277,address:"Paseo de
la Castellana, Fuencarral-El Pardo, Madrid, El Goloso, España"},
{coordX:-410323.4204250248,coordY:4935788.523867752,address:"Paseo de
la Castellana, Fuencarral-El Pardo, Madrid, El Goloso, España"},
{coordX:-410626.86667917395,coordY:4928694.83920398,address:"Paseo de
la Castellana, Salamanca, Madrid, El Goloso, España"},
{coordX:-410821.2584325327,coordY:4929457.501184746,address:"Paseo de
la Castellana, Salamanca, Chamberí, Madrid, 28010, El Goloso,
España"},
{coordX:-410788.0453905335,coordY:4929591.434330969,address:"Paseo de
la Castellana, Salamanca, Chamberí, Madrid, 28010, El Goloso,
España"},
```

{coordX:-410717.8295296114,coordY:4928690.607039906,address:"Paseo de la Castellana, Salamanca, Chamberí, Madrid, 28010, El Goloso, España"},
{coordX:-410433.7530865259,coordY:4935298.349667073,address:"Paseo de la Castellana, Fuencarral-El Pardo, Madrid, El Goloso, España"}]};}

11.1.3.2 Geocoding with charging provider in XML format

http://www.blomurbex.com/v02/GetGeocoding?**usertoken**=********&**format**=xml&**address**=&**service**=provider1&**country**=ESP&**num**=&**street**=castellana&**municipality**=madrid&**pcode**=&**lang**=es&**countrySubdiv**=

Above example should return a response like the following:

```
<result>
<name> castellana, madrid, ESP</name>
<candidates>
        <candidate>
                <coordX>-410412.03073969623</coordX>
                <coordY>4935011.8443109775</coordY>
                <address>
                        Paseo de la Castellana, Madrid, Comunidad de Madrid, ES
                </address>
        </candidate>
      <candidate>
                <coordX>-402280.47589572</coordX>
                <coordY>4924831.452938068</coordY>
                <address>
                        Calle de la Castellana, Madrid, Comunidad de Madrid, ES
                </address>
      </candidate>
      <candidate>
                <coordX>-417771.91887349356</coordX>
                <coordY>4918818.032154235</coordY>
                <address>
                        Calle del Castellano, Madrid, Comunidad de Madrid, ES
                </address>
      </candidate>
      <candidate>
                <coordX>-410116.47749164014</coordX>
                <coordY>4936814.885798859</coordY>
                <address>
                        Avenida del Llano Castellano, Madrid, Comunidad de Madrid,
      ES
                </address>
</candidate>
</candidates>
</result>
```

- **Ubiest: Request Parameters involved are:**
  location: address to geocode [textual address (**required**)]
  idcust: customer id (**required**)
  callback: function to call for jsonp

location is made of the following comma separated attributes:
amc1,amd1,amd2,amd3,amd4,addr,num
where:
amc1 = country code
amd1 = country name
amd2 = region
amd3 = province
amd4 = city (required)
addr = address
num = number

Request:

```
http://www.blomurbex.com/v02/GetGeocoding?usertoken=********&format=js&function=Fgeocode298&srs=EPSG:3785&address=%2Citalia%2C%2C%2Ctreviso%2C%2C&service=ubiest
```

Response: (json response)

function Fgeocode535 () { return {'name': ',italia,,,treviso,,',

'candidates': [

{coordX:1362776.668039336,coordY:5727034.757096411,address:"31100 Treviso, Veneto, Italia"}]};}

- **oym (on your map):** The OYM platform provides a complete free form geocoding API. Request parameters involved are:

  f: value is 'g' denoting gecoding function
  free: a free form address
  country: *char[2]* an ISO 3166 country code
  state : a state (actually, it means the first administrative level for a given country)
  city: a city, a suburb, a settlement or a national POI name
  street : a street name (can contain a street number, a street type or directional)
  streetnum : a street number (alphanumeric format supported)

  Request:

```
http://www.blomurbex.com/v02/GetGeocoding?usertoken=token&format=js&function=Fgeocode536&srs=EPSG:3785&address=regent%20street%20london&service=oym
```

Response:

```
function Fgeocode730 () { return {'name': 'regent street london',
'candidates': [
{coordX:-31459.667334614664,coordY:6708186.48494219,address:"Regent
St, Chiswick, London, W4 3, United Kingdom"},
{coordX:-15812.377069730544,coordY:6712976.095084815,address:"Regent
St, Fitzrovia, London, W1W 8, United Kingdom"},
{coordX:-24107.682884663704,coordY:6715367.214501576,address:"Regent
St, Kensal Green, London, NW10 5, United Kingdom"},
{coordX:-15092.696561752033,coordY:6712003.209633373,address:"Regent
St, Mayfair, London, W1J 0, United Kingdom"},{coordX:-
14858.36903363219,coordY:6711695.931472021,address:"Regent St, St
James's, London, SW1Y 4, United Kingdom"} ]};}
```

11.2 Reverse Geocoding Service

11.2.1 Description

This service returns the address for a given point defined by it coordinates.

11.2.2 Parameters

The parameters for this service are:

- **Crs or srs [mandatory]:** Projection system. It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.
- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.
- **format [mandatory]:** Response format. Possible values are:
    - **txt:** Result is a text string containing the calculated angle.
    - **Xml:** Result is a XML file containing the calculated angle.
    - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.
- **service [optional]:** The service provider the geocoding data will be retrieved of. Possible values are 'free' or 'provider1'. If omitted, the free provider will be used.
- **X [mandatory]:** X coordinate of the point to reverse geocode, in the coordinate system defined by crs/srs.
- **Y [mandatory]:** Y coordinate of the point to reverse geocode, in the coordinate system defined by crs/srs.
- **zoom [optional]:** Zoom level. It will define the accuracy of the reverse geocoding response.

11.2.3 Examples

11.2.3.1 Charging provider with javascript response

```
http://www.blomurbex.com/v02/GetReverseGeocoding?usertoken=token&srs=EPSG:3785
&x=-
410882.22687&y=4930737.91741&function=FreverseGeocode1432&format=js&service=pr
ovider1&zoom=17
```

Above request should return this response:

```
function FreverseGeocode1432 () { return "Paseo de la Castellana, (96
- 102), 28046, Madrid, Madrid, Comunidad de Madrid, ES";}
```

- **Ubiest:** Request Parameters involved are:

  location: point to geocode [latitude and longitude (**required**)]
  idcust: customer id (**required**)
  callback: function to call for jsonp

  Sample Request:

```
http://www.blomurbex.com/v02/GetReverseGeocoding?usertoken=********&srs=EPSG:3
785&x=-
523013.79723&y=4962130.04475&function=FreverseGeocode308&format=js&service=pro
vider2&zoom=17
```

Above request should return this response:

Response:

```
function FreverseGeocode308 () { return "Avenida De Madrid, 48, 05001
Ávila, Castilla y León, España";}
```

- **oym:** The OYM platform provides a simple reverse geocoding API.
  nearest mode: it returns the nearest street from point1 (x1,y1) if radius (r) is equal to 0.

  The request parameters are:
  f=rv (denoting reverse geocoding)
  x1 =point1's longitude
  y1=point1's latitude
  r=0 (radius set to 0)
  key= apiKey

  Request:

```
http://www.blomurbex.com/v02/GetReverseGeocoding?usertoken=********&srs=EPSG:3
785&x=-
522665.05329&y=4962194.5385&function=FreverseGeocode304&format=js&service=prov
ider5&zoom=17
```

  Response:

```
function FreverseGeocode305 () { return "Plaza San Andr?, ?vila,
05005, Spain";}
```

11.2.3.2 Free provider with XML response

```
http://www.blomurbex.com/v02/GetReverseGeocoding?usertoken=********&srs=EPSG:3
785&x=-
410882.22687&y=4930737.91741&function=FreverseGeocode1432&format=js&service=pr
ovider1&zoom=17
```

Above should return a response like this:

<result>
Paseo de la Castellana, Chamberí, Chamartín, Madrid, 28010, El Goloso, España
</result>

12 Routing Service

12.1.1 Description

This service returns list of coordinates, along with other routing information like duration, distance, route instructions between two addresses or two points taken as origin and destination.

12.1.2 Parameters

The parameters for this service are:

- **Crs or srs [mandatory]:** Projection system. It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.
- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.
- **format [mandatory]:** Response format. Possible values are:
    - **txt:** Result is a text string containing the required route information.
    - **Xml:** Result is a XML file containing the route information.
    - **js:** The result is a JavaScript function that will return the result. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.
- **provider [mandatory]:** The service provider the routing data will be retrieved of. Possible values are 'provider2 or 'Ubiest' and 'provider5' or 'oym'.

The service will need to receive some parameters depending on which kind of provider was chosen.

a. ubiest provider:

Ubiest requires following additional parameters for point to point routing:

- **origin [required]**: This can be a point in the coordinates indicated by crs parameter.
- **destination [required]**: This can be a point in the coordinates indicated by the crs parameter.
- **mode**: specifies what mode of transport to use when calculating directions. Valid values are specified in Travel Modes (drivng (default) or walking)

For textual address the origin & destination are composed of the following sub-params:

Param-name	Param-value
originCountry/ destCountry	country name

originCity/destCity	**city (required)**
originAddress/destAddress	address
originNumber/destNumber	number
mode	walking or **car(default-value)**

b. oym (on your map) provider:

There are 2 ways to define the start and end of the route:

- using coordinates (when you know where you are)
- using addresses. User is allowed to provide freeform addresses as origin and destination.

The parameters required for oym routing are:

- **origin:** point coordinates in longitude/latitude or spherical mercator**(Required)**
- **destination:** point coordinates in longitude/latitude or spherical mercator**(Required)**
- **mode:** walking/**car(default-value)**
- **originAddress/destinationAddress:** containing free-form text addresses.**(Required if no points are provided)**

12.1.3 Response

The response of the request will contain all the steps to take in the route, with start and end points and directions/guidelines.

It will also contain an overview polyline to be drawn on top of a map. For every point, you can find X and Y coordinates in the EPSG given.

12.1.4 Examples

12.1.4.1 Sample routing request to ubiest(point to point, js format) would be:

```
http://www.blomurbex.com/v02/GetRouting?usertoken=********&format=js&function=FgetRoute311&provider=ubiest&origin=-523043.65545%2C4962276.94717&destination=-521900.68301%2C4962096.60355&srs=EPSG:3785
```

- Response received from above request:

```
function FgetRoute311 () { return
{"route":[{"description":"","startAddress":"CALLE
CASTILLA, ÁVILA","endAddress":"CALLE DE SAN PEDRO
BAUTISTA, ÁVILA","duration":"  3 min","distance":"1.3
km","step":[{"description":"direction south go ahead to 11
```

m ","startPoint":"-
523040.63874469866,4962273.982448238","endPoint":"-
523041.86325909744,4962260.775435424","duration":"1
min","distance":"11.0 m"},{"description":"turn left go
ahead to 113 m ","startPoint":"-
523041.86325909744,4962260.775435424","endPoint":"-
522947.51999065006,4962197.528765484","duration":"1
min","distance":"113.0 m"},{"description":"turn right go
ahead to 169 m ","startPoint":"-
522947.51999065006,4962197.528765484","endPoint":"-
523172.0514035802,4962215.431493705","duration":"1
min","distance":"169.0 m"},{"description":"keep left go
ahead to 23 m ","startPoint":"-
523172.0514035802,4962215.431493705","endPoint":"-
523187.5804725458,4962190.338334681","duration":"1
min","distance":"23.0 m"},{"description":"at the
roundabout take the fourth exit go ahead to 31 m
","startPoint":"-
523187.5804725458,4962190.338334681","endPoint":"-
523181.569220043,4962176.030860525","duration":"1
min","distance":"31.0 m"},{"description":"go ahead to 628
m ","startPoint":"-
523181.569220043,4962176.030860525","endPoint":"-
522368.2690203074,4962013.587466612","duration":" 1
min","distance":"628.0 m"},{"description":"at the
roundabout take the second exit go ahead to 25 m
","startPoint":"-
522368.2690203074,4962013.587466612","endPoint":"-
522344.3353297868,4962012.413540423","duration":"1
min","distance":"25.0 m"},{"description":"go ahead to 342
m ","startPoint":"-
522344.3353297868,4962012.413540423","endPoint":"-
521902.50827082834,4962098.404006274","duration":"1
min","distance":"342.0 m"},{"description":"you have
arrived ","startPoint":"-
521902.50827082834,4962098.404006274","endPoint":"-
521902.50827082834,4962098.404006274","duration":"1
min","distance":"0.0 m"}]}],"overviewPolyline":["-
523041.30666164344,4962274.862916396,3","-
523042.4198565514,4962260.188457491,0","-
522946.68509446917,4962252.851236289,1","-
522947.7982893772,4962197.088534877,1","-
523056.8913903545,4962200.02340598,0","-
523132.5886440939,4962205.893150821,0","-
523172.66366077954,4962214.697774693,1","-
523188.24838949065,4962198.555970316,0","-

```
523188.24838949065,4962189.751360972,0","-
523196.0407538462,4962188.283926848,0","-
523199.38033856987,4962185.349059265,0","-
523200.4935334778,4962180.946759542,0","-
523199.38033856987,4962175.077029655,1","-
523192.70116912224,4962170.6747345505,0","-
523188.24838949065,4962170.6747345505,0","-
523181.569220043,4962175.077029655,0","-
523161.53171170014,4962170.6747345505,0","-
523050.212220907,4962135.4564449815,0","-
523010.1372042213,4962128.119317272,0","-
522931.10036575806,4962128.119317272,0","-
522896.5913236122,4962122.24961906,1","-
522681.7447063812,4962045.943862605,0","-
522638.3301049718,4962032.737157446,1","-
522571.5384104959,4962020.997878926,0","-
522482.4828178612,4962020.997878926,0","-
522368.9369372521,4962012.193429279,0","-
522367.8237423442,4962007.791207422,0","-
522362.2577678045,4962003.388987548,0","-
522354.46540344896,4962001.92158136,1","-
522345.5598441855,4962006.32380058,0","-
522344.4466492776,4962012.193429279,0","-
522328.86192056653,4962031.269746863,0","-
522215.3160399574,4962051.813515063,0","-
522205.29728578596,4962051.813515063,0","-
522180.8069978115,4962056.215756713,0","-
522031.6388801485,4962078.22699467,0","-
521934.7909231584,4962095.836020669,0","-
521902.5082708282,4962097.303440934,3"]};}
```

12.1.4.2 Sample routing request to ubiest(point to point, xml format) would be:

```
http://www.blomurbex.com/v02/GetRouting?usertoken=********&format=xml&
provider=ubiest&origin=-523043.65545%2C4962296.05642&destination=-
521783.63881%2C4962155.12565&srs=EPSG:3785&mode=car
```

Xml Response received from above request:

```
<Routing>
<route>
      <end-address>AVENIDA DE SANTA CRUZ DE TENERIFE, ÁVILA</end-
address>
      <start-address>CALLE CASTILLA, ÁVILA</start-address>
      <step>
            <end-point>-
523041.86325909744,4962260.775435424</end-point>
```

```
                  <start-point>-
523037.07652099326,4962293.059275984</start-point>
                  <description>direction south go ahead to 25 m
</description>
                  <duration>1 min</duration>
                  <distance>25.0 m</distance>
            </step>
            <step>
                  <end-point>-
522947.51999065006,4962197.528765484</end-point>
                  <start-point>-
523041.86325909744,4962260.775435424</start-point>
                  <description>turn left go ahead to 113 m
</description>
                  <duration>1 min</duration>
                  <distance>113.0 m</distance>
            </step>
            <step>
                  <end-point>-523172.0514035802,4962215.431493705</end-
point>
                  <start-point>-
522947.51999065006,4962197.528765484</start-point>
                  <description>turn right go ahead to 169 m
</description>
                  <duration>1 min</duration>
                  <distance>169.0 m</distance>
            </step>
            <step>
                  <end-point>-523187.5804725458,4962190.338334681</end-
point>
                  <start-point>-
523172.0514035802,4962215.431493705</start-point>
                  <description>keep left go ahead to 23 m
</description>
                  <duration>1 min</duration>
                  <distance>23.0 m</distance>
            </step>
            <step>
                  <end-point>-523181.569220043,4962176.030860525</end-
point>
                  <start-point>-
523187.5804725458,4962190.338334681</start-point>
                  <description>at the roundabout take the fourth exit
go ahead to 31 m
                  </description>
                  <duration>1 min</duration>
                  <distance>31.0 m</distance>
            </step>
            <step>
                  <end-point>-522368.2690203074,4962013.587466612</end-
point>
                  <start-point>-
523181.569220043,4962176.030860525</start-point>
                  <description>go ahead to 628 m </description>
                  <duration> 1 min</duration>
                  <distance>628.0 m</distance>
            </step>
```

```
            <step>
                  <end-point>-522344.3353297868,4962012.413540423</end-
point>
                  <start-point>-
522368.2690203074,4962013.587466612</start-point>
                  <description>at the roundabout take the second exit
go ahead to 25 m
                  </description>
                  <duration>1 min</duration>
                  <distance>25.0 m</distance>
            </step>
            <step>
                  <end-point>-
521783.06245720707,4962150.937802497</end-point>
                  <start-point>-
522344.3353297868,4962012.413540423</start-point>
                  <description>go ahead to 440 m </description>
                  <duration>1 min</duration>
                  <distance>440.0 m</distance>
            </step>
            <step>
                  <end-point>-
521783.06245720707,4962150.937802497</end-point>
                  <start-point>-
521783.06245720707,4962150.937802497</start-point>
                  <description>you have arrived </description>
                  <duration>1 min</duration>
                  <distance>0.0 m</distance>
            </step>
            <description></description>
            <duration> 3 min</duration>
            <distance>1.5 km</distance>
      </route>
      <overviewPolyline>
            <point>-523037.9670769197,4962293.939745858,3</point>
            <point>-523042.4198565514,4962260.188457491,1</point>
            <point>-522946.68509446917,4962252.851236289,1</point>
            <point>-522947.7982893772,4962197.088534877,1</point>
            <point>-523056.8913903545,4962200.02340598,0</point>
            <point>-523132.5886440939,4962205.893150821,0</point>
            <point>-523172.66366077954,4962214.697774693,1</point>
            <point>-523188.24838949065,4962198.555970316,0</point>
            <point>-523188.24838949065,4962189.751360972,0</point>
            <point>-523196.0407538462,4962188.283926848,0</point>
            <point>-523199.38033856987,4962185.349059265,0</point>
            <point>-523200.4935334778,4962180.946759542,0</point>
            <point>-523199.38033856987,4962175.077029655,1</point>
            <point>-523192.70116912224,4962170.6747345505,0</point>
            <point>-523188.24838949065,4962170.6747345505,0</point>
            <point>-523181.569220043,4962175.077029655,0</point>
            <point>-523161.53171170014,4962170.6747345505,0</point>
            <point>-523050.212220907,4962135.4564449815,0</point>
            <point>-523010.1372042213,4962128.119317272,0</point>
            <point>-522931.10036575806,4962128.119317272,0</point>
            <point>-522896.5913236122,4962122.24961906,1</point>
            <point>-522681.7447063812,4962045.943862605,0</point>
            <point>-522638.3301049718,4962032.737157446,1</point>
```

```
            <point>-522571.5384104959,4962020.997878926,0</point>
            <point>-522482.4828178612,4962020.997878926,0</point>
            <point>-522368.9369372521,4962012.193429279,0</point>
            <point>-522367.8237423442,4962007.791207422,0</point>
            <point>-522362.2577678045,4962003.388987548,0</point>
            <point>-522354.46540344896,4962001.92158136,1</point>
            <point>-522345.5598441855,4962006.32380058,0</point>
            <point>-522344.4466492776,4962012.193429279,0</point>
            <point>-522328.86192056653,4962031.269746863,0</point>
            <point>-522215.3160399574,4962051.813515063,0</point>
            <point>-522205.29728578596,4962051.813515063,0</point>
            <point>-522180.8069978115,4962056.215756713,0</point>
            <point>-522031.6388801485,4962078.22699467,0</point>
            <point>-521934.7909231584,4962095.836020669,0</point>
            <point>-521890.26312684105,4962100.238282116,0</point>
            <point>-521864.65964395856,4962111.977655661,0</point>
            <point>-521832.3769916286,4962123.717043285,0</point>
            <point>-521783.39641567954,4962150.130716907,3</point>
        </overviewPolyline>
</Routing>
```

12.1.4.3 Sample url request to ubiest having textual address,js format would be:

http://www.blomurbex.com/v02/GetRouting?**usertoken**=********&**format**=js&**function**=FgetRoute311&**originCountry**=italia&**originCity**=Quinto+di+treviso&**destinationCountry**=italia&**destinationCity**=treviso&**idcust**=XX.YYYY.ZZZ&**provider**=ubiest

- Response received from above request:

```
function FgetRoute298 () { return
{"route":[{"description":"","startAddress":"VIA VITTORIO
EMANUELE, QUINTO DI TREVISO","endAddress":"PIAZZA DEL
DUOMO, TREVISO","duration":"  14 min","distance":"7.5
km","step":[{"description":"direction north go ahead to
4.0 Km.
","startPoint":"1354041.3054334307,5723225.998942291","end
Point":"1359255.7886809146,5725348.398259548","duration":"
7 min","distance":"4.0 km"},{"description":"at the
roundabout take the third exit go ahead to 133 m
","startPoint":"1359255.7886809146,5725348.398259548","end
Point":"1359388.3701944493,5725391.399263688","duration":"
1 min","distance":"133.0 m"},{"description":"go ahead to
1.4 Km.
","startPoint":"1359388.3701944493,5725391.399263688","end
Point":"1360579.0991277196,5726817.39814835","duration":"
2 min","distance":"1.4 km"},{"description":"turn right go
```

ahead to 418 m
","startPoint":"1360579.0991277196,5726817.39814835","endP
oint":"1361117.7741436684,5727021.608768155","duration":"1
min","distance":"418.0 m"},{"description":"go ahead to 967
m
","startPoint":"1361117.7741436684,5727021.608768155","end
Point":"1362125.7721328014,5726585.715553219","duration":"
2 min","distance":"967.0 m"},{"description":"turn left go
ahead to 552 m
","startPoint":"1362125.7721328014,5726585.715553219","end
Point":"1362791.017409782,5727019.299027505","duration":"
1 min","distance":"552.0 m"},{"description":"turn left go
ahead to 14 m
","startPoint":"1362791.017409782,5727019.299027505","endP
oint":"1362776.6571954698,5727034.830053719","duration":"1
min","distance":"14.0 m"},{"description":"you have arrived
","startPoint":"1362776.6571954698,5727034.830053719","end
Point":"1362776.6571954698,5727034.830053719","duration":"
1 min","distance":"0.0
m"}]}],"overviewPolyline":["1354046.8714079703,5723240.329
177036,3","1354062.4561366814,5723292.873568051,0","135407
8.0408653924,5723332.680130929,0","1354096.9651788273,5723
375.671418353,0","1354121.4554668018,5723421.8474763725,1"
,"1354150.398534408,5723464.839193538,0","1354173.77562747
45,5723493.500453425,0","1354208.2846696207,5723531.715609
859,1","1354263.944415017,5723579.484785625,0","1354309.58
54062424,5723608.146413876,0","1354614.600811016,5723761.0
09986584,1","1354878.4280041961,5723889.990663713,0","1354
987.5211051735,5723937.761758171,1","1355622.0422026953,57
24159.104501426,0","1355677.7019480919,5724181.398319603,0
","1355786.7950490692,5724216.431575033,0","1356473.636307
2637,5724455.298347719,0","1356582.7294082409,5724496.7025
721595,1","1356647.2947129013,5724525.367147822,0","135682
2.0663134467,5724616.1389118405,0","1357089.2330913504,572
4761.057151555,0","1357135.9872774836,5724791.3151031425,0
","1357323.0040220164,5724885.274659724,1","1357365.305428
5177,5724902.792652631,0","1357632.4722064217,5724984.0128
87504,0","1357668.0944434756,5724998.345946911,0","1357908
.544543589,5725119.381589284,0","1357968.6570686174,572514
6.455576195,1","1358175.711321493,5725213.344602093,0","13
58246.9557956005,5725232.45584449,0","1358307.0683206287,5
725246.789303163,0","1358518.575353136,5725280.234129652,1
","1358619.876089758,5725291.382433019,0","1358672.1962504
308,5725291.382433019,0","1358767.931012513,5725285.011972
245,0","1358814.6851986465,5725285.011972245,0","1358862.5
525796874,5725289.789817397,0","1358932.683858887,5725299.

345515384,1","1358979.4380450202,5725308.901223607,0","135
9135.2853321307,5725372.606206746,0","1359149.756865934,57
25375.7914678445,1","1359182.0395182641,5725377.384098824,
0","1359212.0957807782,5725371.013576619,0","1359255.51038
21876,5725347.1241588835,0","1359281.11386507,5725323.2348
0514,0","1359317.8492970318,5725310.493842639,0","1359354.
5847289935,5725323.23480514,2","1359362.3770933494,5725328
.012670769,0","1359380.1882118762,5725347.1241588835,0","1
359389.0937711396,5725375.7914678445,0","1359387.980576231
6,5725390.125156879,0","1359395.772940587,5725404.45886895
3,0","1359405.7916947587,5725415.6073276065,0","1359432.50
8372549,5725437.904286713,0","1359439.1875419964,5725450.6
45431229,1","1359445.866711444,5725549.389918428,0","13594
55.8854656154,5725582.835879704,1","1359504.8660415646,572
5638.579427236,0","1359577.2237105803,5725708.657524185,1"
,"1359606.1667781863,5725729.36252185,0","1359635.10984579
26,5725745.289475838,0","1359902.2766236966,5725872.706131
752,0","1359967.9551232646,5725910.931483514,1","1359994.6
71801055,5725931.636950798,0","1360011.369724674,5725947.5
64266034,0","1360078.16141915,5726031.979511709,0","136010
2.6517071244,5726060.649022201,0","1360134.9343594543,5726
090.911383269,0","1360247.3670451557,5726173.735212375,0",
"1360415.4594762535,5726305.936379565,0","1360441.06295913
6,5726329.828365864,0","1360464.4400522024,5726356.9060277
24,1","1360490.043535085,5726391.947829827,0","1360505.628
263796,5726419.025680304,0","1360524.552577231,5726462.031
847122,0","1360533.4581364945,5726493.888400705,1","136054
0.1373059421,5726528.93074109,0","1360579.0991277196,57268
17.238859519,1","1360601.3630258783,5726839.539323363,0","
1360654.796381459,5726900.069434929,0","1360684.8526439734
,5726927.148828393,0","1360718.248491211,5726946.263743876
5,2","1360765.0026773445,5726965.37870034,0","1360790.6061
602267,5726970.157445856,0","1361051.093768683,5727008.387
502196,0","1361116.772268251,5727021.13089073,0","1361293.
7702586125,5727043.431864489,0","1361899.348288528,5727132
.6363172745,0","1361913.819822331,5727127.857484682,2","13
61920.4989917784,5727123.0786546515,0","1361930.51774595,5
727113.5210022675,0","1361939.4233052135,5727099.184542904
,0","1361948.328864477,5727037.060152012,0","1361966.13998
30037,5726895.290726973,0","1361973.9323473594,5726855.468
2602575,1","1361995.08305061,5726804.49576236,0","1362125.
326854838,5726584.680202779,2","1362158.7227020762,5726597
.4229858145,0","1362196.5713289457,5726621.315753089,0","1
362235.5331507237,5726645.208584382,0","1362326.815133174,
5726712.108852573,0","1362435.9082341515,5726780.602504276
,0","1362578.3971823668,5726857.061155509,0","1362670.7923

```
597253,5726917.591386015,0","1362711.9805713189,5726941.48
5011165,1","1362755.3951727282,5726989.272453536,0","13627
89.9042148741,5727017.945041887,1","1362776.5458759787,572
7033.874297479,3"]};}
```

12.1.4.4 Sample url request to ubiest having textual address,xml format would be:

http://www.blomurbex.com/v02/GetRouting?**usertoken**=********&**format**=xml&**originCountry**=italia&**originCity**=Quinto+di+treviso&**destinationCountry**=italia&**destinationCity**=treviso&**idcust**=XX.YYYY.ZZZ&**provider**=ubiest&**mode**=car

Response received from this request:

```
<Routing>
      <route>
            <end-address>PIAZZA DEL DUOMO, TREVISO</end-address>
            <start-address>VIA VITTORIO EMANUELE, QUINTO DI TREVISO
            </start-address>
            <step>
                  <end-point>1359255.7886809146,5725348.398259548</end-
point>
                  <start-
point>1354041.3054334307,5723225.998942291</start-point>
                  <description>direction north go ahead to 4.0 Km.
</description>
                  <duration> 7 min</duration>
                  <distance>4.0 km</distance>
            </step>
            <step>
                  <end-point>1359388.3701944493,5725391.399263688</end-
point>
                  <start-
point>1359255.7886809146,5725348.398259548</start-point>
                  <description>at the roundabout take the third exit go
ahead to 133 m
                  </description>
                  <duration>1 min</duration>
                  <distance>133.0 m</distance>
            </step>
            <step>
                  <end-point>1360579.0991277196,5726817.39814835</end-
point>
                  <start-
point>1359388.3701944493,5725391.399263688</start-point>
                  <description>go ahead to 1.4 Km. </description>
                  <duration> 2 min</duration>
                  <distance>1.4 km</distance>
            </step>
            <step>
                  <end-point>1361117.7741436684,5727021.608768155</end-
point>
                  <start-
point>1360579.0991277196,5726817.39814835</start-point>
                  <description>turn right go ahead to 418 m
</description>
```

```
                <duration>1 min</duration>
                <distance>418.0 m</distance>
            </step>
            <step>
                <end-point>1362125.7721328014,5726585.715553219</end-
point>
                <start-
point>1361117.7741436684,5727021.608768155</start-point>
                <description>go ahead to 967 m </description>
                <duration> 2 min</duration>
                <distance>967.0 m</distance>
            </step>
            <step>
                <end-point>1362791.017409782,5727019.299027505</end-
point>
                <start-
point>1362125.7721328014,5726585.715553219</start-point>
                <description>turn left go ahead to 552 m
</description>
                <duration> 1 min</duration>
                <distance>552.0 m</distance>
            </step>
            <step>
                <end-point>1362776.6571954698,5727034.830053719</end-
point>
                <start-
point>1362791.017409782,5727019.299027505</start-point>
                <description>turn left go ahead to 14 m
</description>
                <duration>1 min</duration>
                <distance>14.0 m</distance>
            </step>
            <step>
                <end-point>1362776.6571954698,5727034.830053719</end-
point>
                <start-
point>1362776.6571954698,5727034.830053719</start-point>
                <description>you have arrived </description>
                <duration>1 min</duration>
                <distance>0.0 m</distance>
            </step>
            <description></description>
            <duration> 14 min</duration>
            <distance>7.5 km</distance>
      </route>
      <overviewPolyline>
            <point>1354046.8714079703,5723240.329177036,3</point>
            <point>1354062.4561366814,5723292.873568051,0</point>
            <point>1354078.0408653924,5723332.680130929,0</point>
            <point>1354096.9651788273,5723375.671418353,0</point>
            <point>1354121.4554668018,5723421.8474763725,1</point>
            <point>1354150.398534408,5723464.839193538,0</point>
            <point>1354173.7756274745,5723493.500453425,0</point>
            <point>1354208.2846696207,5723531.715609859,1</point>
            <point>1354263.944415017,5723579.484785625,0</point>
            <point>1354309.5854062424,5723608.146413876,0</point>
            <point>1354614.600811016,5723761.009986584,1</point>
```

```
<point>1354878.4280041961,5723889.990663713,0</point>
<point>1354987.5211051735,5723937.761758171,1</point>
<point>1355622.0422026953,5724159.104501426,0</point>
<point>1355677.7019480919,5724181.398319603,0</point>
<point>1355786.7950490692,5724216.431575033,0</point>
<point>1356473.6363072637,5724455.298347719,0</point>
<point>1356582.7294082409,5724496.7025721595,1</point>
<point>1356647.2947129013,5724525.367147822,0</point>
<point>1356822.0663134467,5724616.1389118405,0</point>
<point>1357089.2330913504,5724761.057151555,0</point>
<point>1357135.9872774836,5724791.3151031425,0</point>
<point>1357323.0040220164,5724885.274659724,1</point>
<point>1357365.3054285177,5724902.792652631,0</point>
<point>1357632.4722064217,5724984.012887504,0</point>
<point>1357668.0944434756,5724998.345946911,0</point>
<point>1357908.544543589,5725119.381589284,0</point>
<point>1357968.6570686174,5725146.455576195,1</point>
<point>1358175.711321493,5725213.344602093,0</point>
<point>1358246.9557956005,5725232.45584449,0</point>
<point>1358307.0683206287,5725246.789303163,0</point>
<point>1358518.575353136,5725280.234129652,1</point>
<point>1358619.876089758,5725291.382433019,0</point>
<point>1358672.1962504308,5725291.382433019,0</point>
<point>1358767.931012513,5725285.011972245,0</point>
<point>1358814.6851986465,5725285.011972245,0</point>
<point>1358862.5525796874,5725289.789817397,0</point>
<point>1358932.683858887,5725299.345515384,1</point>
<point>1358979.4380450202,5725308.901223607,0</point>
<point>1359135.2853321307,5725372.606206746,0</point>
<point>1359149.756865934,5725375.7914678445,1</point>
<point>1359182.0395182641,5725377.384098824,0</point>
<point>1359212.0957807782,5725371.013576619,0</point>
<point>1359255.5103821876,5725347.1241588835,0</point>
<point>1359281.11386507,5725323.23480514,0</point>
<point>1359317.8492970318,5725310.493842639,0</point>
<point>1359354.5847289935,5725323.23480514,2</point>
<point>1359362.3770933494,5725328.012670769,0</point>
<point>1359380.1882118762,5725347.1241588835,0</point>
<point>1359389.0937711396,5725375.7914678445,0</point>
<point>1359387.9805762316,5725390.125156879,0</point>
<point>1359395.772940587,5725404.458868953,0</point>
<point>1359405.7916947587,5725415.6073276065,0</point>
<point>1359432.508372549,5725437.904286713,0</point>
<point>1359439.1875419964,5725450.645431229,1</point>
<point>1359445.866711444,5725549.389918428,0</point>
<point>1359455.8854656154,5725582.835879704,1</point>
<point>1359504.8660415646,5725638.579427236,0</point>
<point>1359577.2237105803,5725708.657524185,1</point>
<point>1359606.1667781863,5725729.36252185,0</point>
<point>1359635.1098457926,5725745.289475838,0</point>
<point>1359902.2766236966,5725872.706131752,0</point>
<point>1359967.9551232646,5725910.931483514,1</point>
<point>1359994.671801055,5725931.636950798,0</point>
<point>1360011.369724674,5725947.564266034,0</point>
<point>1360078.16141915,5726031.979511709,0</point>
<point>1360102.6517071244,5726060.649022201,0</point>
<point>1360134.9343594543,5726090.911383269,0</point>
```

```
            <point>1360247.3670451557,5726173.735212375,0</point>
            <point>1360415.4594762535,5726305.936379565,0</point>
            <point>1360441.062959136,5726329.828365864,0</point>
            <point>1360464.4400522024,5726356.906027724,1</point>
            <point>1360490.043535085,5726391.947829827,0</point>
            <point>1360505.628263796,5726419.025680304,0</point>
            <point>1360524.552577231,5726462.031847122,0</point>
            <point>1360533.4581364945,5726493.888400705,1</point>
            <point>1360540.1373059421,5726528.93074109,0</point>
            <point>1360579.0991277196,5726817.238859519,1</point>
            <point>1360601.3630258783,5726839.539323363,0</point>
            <point>1360654.796381459,5726900.069434929,0</point>
            <point>1360684.8526439734,5726927.148828393,0</point>
            <point>1360718.248491211,5726946.2637438765,2</point>
            <point>1360765.0026773445,5726965.37870034,0</point>
            <point>1360790.6061602267,5726970.157445856,0</point>
            <point>1361051.093768683,5727008.387502196,0</point>
            <point>1361116.772268251,5727021.13089073,0</point>
            <point>1361293.7702586125,5727043.431864489,0</point>
            <point>1361899.348288528,5727132.6363172745,0</point>
            <point>1361913.819822331,5727127.857484682,2</point>
            <point>1361920.4989917784,5727123.0786546515,0</point>
            <point>1361930.51774595,5727113.5210022675,0</point>
            <point>1361939.4233052135,5727099.184542904,0</point>
            <point>1361948.328864477,5727037.060152012,0</point>
            <point>1361966.1399830037,5726895.290726973,0</point>
            <point>1361973.9323473594,5726855.4682602575,1</point>
            <point>1361995.08305061,5726804.49576236,0</point>
            <point>1362125.326854838,5726584.680202779,2</point>
            <point>1362158.7227020762,5726597.4229858145,0</point>
            <point>1362196.5713289457,5726621.315753089,0</point>
            <point>1362235.5331507237,5726645.208584382,0</point>
            <point>1362326.815133174,5726712.108852573,0</point>
            <point>1362435.9082341515,5726780.602504276,0</point>
            <point>1362578.3971823668,5726857.061155509,0</point>
            <point>1362670.7923597253,5726917.591386015,0</point>
            <point>1362711.9805713189,5726941.485011165,1</point>
            <point>1362755.3951727282,5726989.272453536,0</point>
            <point>1362789.9042148741,5727017.945041887,1</point>
            <point>1362776.5458759787,5727033.874297479,3</point>
      </overviewPolyline>
</Routing>
```


Sample routing request to oym (point to point,js format) would be:

```
http://www.blomurbex.com/v02/GetRouting?usertoken=********&format=js&function=
FgetRoute309&provider=oym&origin=-523018.57455%2C4962266.19821&destination=-
521858.88151%2C4962120.49012&srs=EPSG:3785
```

Response received:
In json format:


```
json response=function FgetRoute309 () { return
{"route":[{"description":"","startAddress":"","endAddress":"","duratio
```

n":"2.63","distance":"1.47KM","step":[{"description":"Head east on Calle Castilla. Distance: 0.05. Time: 0.28.","startPoint":"-523063.5705598021,4962259.8949685395","endPoint":"-523063.5705598021,4962259.8949685395","duration":"0.28","distance":"0.05"},{"description":"Turn left on Calle Castilla. Distance: 0.22. Time: 0.38.","startPoint":"-523037.9670769197,4962261.068924405","endPoint":"-523037.9670769197,4962261.068924405","duration":"0.38","distance":"0.22"},{"description":"Turn right on Calle Prado Sancho. Distance: 0.44. Time: 0.77.","startPoint":"-522995.6656704182,4962544.876886573","endPoint":"-522995.6656704182,4962544.876886573","duration":"0.77","distance":"0.44"},{"description":"At the roundabout, go right. Distance: 0.21. Time: 0.35.","startPoint":"-522435.72863172815,4962431.294015465","endPoint":"-522435.72863172815,4962431.294015465","duration":"0.35","distance":"0.21"},{"description":"Turn left on Calle Virgen de Covadonga. Distance: 0.25. Time: 0.38.","startPoint":"-522496.95435166435,4962160.842949297","endPoint":"-522496.95435166435,4962160.842949297","duration":"0.38","distance":"0.25"},{"description":"Turn right on Calle David Herrero. Distance: 0.04. Time: 0.07.","startPoint":"-522174.12782836385,4962106.254709301","endPoint":"-522174.12782836385,4962106.254709301","duration":"0.07","distance":"0.04"},{"description":"Turn left on Calle de San Pedro Bautista. Distance: 0.24. Time: 0.37.","startPoint":"-522176.3542181797,4962057.389688156","endPoint":"-522176.3542181797,4962057.389688156","duration":"0.37","distance":"0.24"},{"description":"Turn slightly right on Avenida de Santa Cruz de Tenerife. Distance: 0.02. Time: 0.03.","startPoint":"-521860.20686432684,4962113.004851516","endPoint":"-521860.20686432684,4962113.004851516","duration":"0.03","distance":"0.02"},{"description":"Arrival on Avenida de Santa Cruz de Tenerife. Distance: 0.0. Time: 0.0.","startPoint":"-521847.96172033955,4962117.407120685","endPoint":"-521847.96172033955,4962117.407120685","duration":"0.0","distance":"0.0"}]}],"overviewPolyline":["-523063.5705598021,4962259.8949685395,-1","-523054.55368104787,4962261.215668898,15","-523037.9670769197,4962261.215668898,11","-523021.1578338099,4962375.8237899635,3","-523018.931443994,4962392.406126023,11","-523018.931443994,4962397.982493133,15","-523007.79949491477,4962454.920318077,3","-522995.6656704182,4962544.876886573,-13","-522912.1760523233,4962528.294292219,3","-522886.46124995,4962522.864599288,11","-522828.68643422826,4962509.217006031,3","-522722.9329179748,4962484.563337413,-13","-522623.7472516779,4962461.230458236,3","-522583.67223499226,4962454.920318077,-5","-522454.65294516296,4962428.946063039,11","-522444.6341909915,4962426.891605129,17","-522435.72863172815,4962431.440762511,-13","-522431.16453260556,4962428.946063039,15","-522441.29460626777,4962374.649820334,15","-522441.29460626777,4962369.953943222,13","-522446.8605808074,4962341.044999379,11","-

522457.8812103959,4962286.749244232,3","-
522466.78676965943,4962245.5140205845,11","-
522472.35274419904,4962231.133093763,17","-
522478.03003822954,4962214.11079952,15","-
522480.2564280454,4962207.654074964,3","-
522488.0487924009,4962186.523006194,7","-
522496.8430321736,4962160.989692288,11","-
522476.9168433216,4962156.293917671,15","-
522464.5603798435,4962152.918831057,15","-
522454.65294516296,4962148.663288723,15","-
522427.82494788175,4962134.282504178,11","-
522417.80619371036,4962129.586742376,17","-
522410.1251488456,4962126.211664966,15","-
522397.88000485837,4962124.010528152,17","-
522386.6367362882,4962121.956134245,15","-
522372.2765219759,4962120.63545267,3","-
522364.4841576203,4962120.63545267,17","-
522327.7487256585,4962117.260378347,15","-
522174.12782836385,4962106.254709301,-13","-
522175.24102327175,4962082.92273184,15","-
522176.3542181797,4962057.389688156,11","-
522139.6187862179,4962064.139796678,15","-
522101.7701593483,4962067.221369503,-45","-
522093.9777949928,4962069.715976792,17","-
522027.07478102593,4962079.547670519,3","-
522020.5069310692,4962080.72160476,17","-
521988.11295924825,4962087.618471262,15","-
521948.0379425628,4962093.928374656,11","-
521930.3381435266,4962097.303440934,-13","-
521914.6420953247,4962097.303440934,15","-
521899.16868610453,4962098.624119357,15","-
521885.69902771845,4962101.999187254,3","-
521876.793468455,4962104.053576974,17","-
521867.99922868237,4962108.309099918,15","-
521860.20686432684,4962113.004851516,3","-
521851.30130506336,4962116.379924372,17","-
521847.8504008488,4962117.260378347,-1"]};}

12.1.4.5 oym(point to point,xml format)

Request:

http://www.blomurbex.com/v02/GetRouting?**usertoken**=********&**format**=xml&**provider**=oym&**origin**=-523018.57455%2C4962266.19821&**destination**=-521858.88151%2C4962120.49012&**srs**=EPSG:3785&**mode**=car

Response:

```
            <Routing>
      <route>
            <end-address></end-address>
            <start-address></start-address>
            <step>
                  <end-point>-
523017.9295685768,4962259.8949685395</end-point>
                  <start-point>-
523017.9295685768,4962259.8949685395</start-point>
```

```
                <description>Head east on Calle de Santander.
Distance: 0.07. Time:
                      0.13.</description>
                <duration>0.13</duration>
                <distance>0.07</distance>
            </step>
            <step>
                <end-point>-
522942.23231483746,4962254.318680093</end-point>
                <start-point>-
522942.23231483746,4962254.318680093</start-point>
                <description>Turn right on Travesía Ajates. Distance:
0.04. Time:
                      0.08.</description>
                <duration>0.08</duration>
                <distance>0.04</distance>
            </step>
            <step>
                <end-point>-522943.3455097454,4962196.501560764</end-
point>
                <start-point>-
522943.3455097454,4962196.501560764</start-point>
                <description>Turn right on Calle Ajates. Distance:
0.17. Time: 0.28.
                </description>
                <duration>0.28</duration>
                <distance>0.17</distance>
            </step>
            <step>
                <end-point>-523168.2108811478,4962216.458700416</end-
point>
                <start-point>-
523168.2108811478,4962216.458700416</start-point>
                <description>Turn left on Calle de la Encarnación.
Distance: 0.02.
                      Time: 0.05.</description>
                <duration>0.05</duration>
                <distance>0.02</distance>
            </step>
            <step>
                <end-point>-523183.7956098588,4962190.925308425</end-
point>
                <start-point>-
523183.7956098588,4962190.925308425</start-point>
                <description>At the roundabout, go left. Distance:
0.67. Time: 0.93.
                </description>
                <duration>0.93</duration>
                <distance>0.67</distance>
            </step>
            <step>
                <end-point>-522364.4841576203,4962013.954318576</end-
point>
                <start-point>-
522364.4841576203,4962013.954318576</start-point>
                <description>At the roundabout, go left. Distance:
0.14. Time: 0.23.
```

```
                </description>
                <duration>0.23</duration>
                <distance>0.14</distance>
            </step>
            <step>
                <end-point>-522197.5049214304,4962052.840704605</end-
point>
                <start-point>-
522197.5049214304,4962052.840704605</start-point>
                <description>Turn slightly left on Calle de San Pedro
Bautista.
                        Distance: 0.26. Time: 0.38.</description>
                <duration>0.38</duration>
                <distance>0.26</distance>
            </step>
            <step>
                <end-point>-
521860.20686432684,4962113.004851516</end-point>
                <start-point>-
521860.20686432684,4962113.004851516</start-point>
                <description>Turn slightly right on Avenida de Santa
Cruz de
                        Tenerife. Distance: 0.02. Time:
0.03.</description>
                <duration>0.03</duration>
                <distance>0.02</distance>
            </step>
            <step>
                <end-point>-
521856.86727960303,4962114.032047479</end-point>
                <start-point>-
521856.86727960303,4962114.032047479</start-point>
                <description>Arrival on Avenida de Santa Cruz de
Tenerife. Distance:
                        0.0. Time: 0.0.</description>
                <duration>0.0</duration>
                <distance>0.0</distance>
            </step>
            <description></description>
            <duration>2.11</duration>
            <distance>1.39KM</distance>
        </route>
        <overviewPolyline>
            <point>-523017.8182490861,4962259.8949685395,-1</point>
            <point>-522942.12099534675,4962254.465424482,3</point>
            <point>-522942.12099534675,4962233.187510588,15</point>
            <point>-522943.3455097454,4962213.230336831,11</point>
            <point>-522943.3455097454,4962196.648304289,11</point>
            <point>-522960.0434333643,4962198.8494574325,15</point>
            <point>-522977.7432324005,4962199.583175257,17</point>
            <point>-523052.327291232,4962200.903867482,3</point>
            <point>-523130.2509347874,4962207.654074964,11</point>
            <point>-523155.96573716047,4962211.909642942,15</point>
            <point>-523168.09956165694,4962216.6054442385,3</point>
            <point>-523184.9088047667,4962199.583175257,17</point>
            <point>-523183.7956098588,4962190.778564989,11</point>
            <point>-523187.13519458263,4962190.778564989,17</point>
```

```
<point>-523192.70116912224,4962189.898104398,17</point>
<point>-523194.92755893816,4962186.523006194,15</point>
<point>-523197.1539487541,4962179.626070013,17</point>
<point>-523196.0407538462,4962176.250975352,17</point>
<point>-523192.70116912224,4962172.8758818535,17</point>
<point>-523188.24838949065,4962171.9954228755,15</point>
<point>-523180.4560251351,4962172.8758818535,17</point>
<point>-523177.0051209205,4962176.250975352,15</point>
<point>-523157.0789320685,4962170.82147769,11</point>
<point>-523047.87451160024,4962136.336900677,11</point>
<point>-523006.68630000675,4962128.706287283,11</point>
<point>-522987.87330606265,4962126.211664966,15</point>
<point>-522967.72447822907,4962126.211664966,3</point>
<point>-522935.441825899,4962128.706287283,11</point>
<point>-522927.6494615436,4962128.706287283,17</point>
<point>-522912.1760523233,4962126.211664966,15</point>
<point>-522892.0272244896,4962124.010528152,3</point>
<point>-522861.9709619755,4962113.004851516,15</point>
<point>-522834.252408768,4962102.879639939,15</point>
<point>-522796.4037818983,4962089.672857968,11</point>
<point>-522749.6495957651,4962073.9714848045,-13</point>
<point>-522731.72715774743,4962066.3409200255,11</point>
<point>-522718.480138343,4962061.64519081,11</point>
<point>-522677.29192674946,4962047.264534103,3</point>
<point>-522634.990520248,4962034.057827162,15</point>
<point>-522599.36828319414,4962027.307739383,3</point>
<point>-522584.7854299003,4962023.785956299,17</point>
<point>-522567.0856308641,4962023.052251652,11</point>
<point>-522490.2751822167,4962023.052251652,-13</point>
<point>-522479.14323313744,4962023.052251652,17</point>
<point>-522398.99319976626,4962016.155430639,3</point>
<point>-522364.4841576203,4962014.101059365,11</point>
<point>-522362.2577678045,4962006.030319234,17</point>
<point>-522355.4672788661,4962002.655284427,17</point>
<point>-522349.9013043264,4962002.655284427,11</point>
<point>-522342.1089399709,4962007.350985347,15</point>
<point>-522339.8825501551,4962011.606466249,7</point>
<point>-522333.31470019824,4962020.410915372,17</point>
<point>-522328.86192056653,4962027.307739383,15</point>
<point>-522324.4091409348,4962031.856711069,11</point>
<point>-522296.57926823647,4962037.432872794,15</point>
<point>-522210.86326032557,4962051.813515063,-13</point>
<point>-522197.3936019396,4962052.693963234,15</point>
<point>-522191.8276274,4962055.335308225,17</point>
<point>-522176.3542181797,4962057.389688156,11</point>
<point>-522139.6187862179,4962064.139796678,15</point>
<point>-522101.7701593483,4962067.221369503,3</point>
<point>-522093.9777949928,4962069.715976792,17</point>
<point>-522027.07478102593,4962079.547670519,-45</point>
<point>-522020.5069310692,4962080.72160476,17</point>
<point>-521988.11295924825,4962087.618471262,15</point>
<point>-521948.0379425628,4962093.928374656,11</point>
<point>-521930.3381435266,4962097.303440934,-13</point>
<point>-521914.6420953247,4962097.303440934,15</point>
<point>-521899.16868610453,4962098.624119357,15</point>
<point>-521885.69902771845,4962101.999187254,3</point>
<point>-521876.793468455,4962104.053576974,17</point>
```

```
            <point>-521867.99922868237,4962108.309099918,15</point>
            <point>-521860.20686432684,4962113.004851516,3</point>
            <point>-521856.86727960303,4962114.325532062,-1</point>
        </overviewPolyline>
</Routing>
```

12.1.4.6 Sample of oym textual address (js format) routing:

Request:

http://www.blomurbex.com/v02/GetRouting?**usertoken**=********&**format**=js&**function**=FgetRoute298&**srs**=EPSG:3785&**provider**=oym&**f**=r&**ref**=1&**originAddress**=quinto%20di%20treviso&**destinationAddress**=treviso

Response:


```
function FgetRoute298 () { return
{"route":[{"description":"","startAddress":"","endAddress":"","d
uration":"9.9","distance":"7.57KM","step":[{"description":"Head
north on Via Vittorio Emanuele on Sr515. Distance: 0.32. Time:
0.43.","startPoint":"1354047.9846028783,5723243.354451757","endP
oint":"1354047.9846028783,5723243.354451757","duration":"0.43","
distance":"0.32"},{"description":"Keep right on Via Vittorio
Emanuele on Sr515. Distance: 1.76. Time:
2.37.","startPoint":"1354289.5478978995,5723597.318676826","endP
oint":"1354289.5478978995,5723597.318676826","duration":"2.37","
distance":"1.76"},{"description":"At the roundabout, go right.
Distance: 1.98. Time:
2.37.","startPoint":"1356616.1252554792,5724522.341438265","endP
oint":"1356616.1252554792,5724522.341438265","duration":"2.37","
distance":"1.98"},{"description":"At the roundabout, go left.
ontinue on Sr515. Distance: 1.52. Time:
1.78.","startPoint":"1359256.6235770956,5725348.398259548","endP
oint":"1359256.6235770956,5725348.398259548","duration":"1.78","
distance":"1.52"},{"description":"Turn slightly right on Via
Carlo Cattaneo. Distance: 1.4. Time:
1.93.","startPoint":"1360579.0991277196,5726816.601704226","endP
oint":"1360579.0991277196,5726816.601704226","duration":"1.93","
distance":"1.4"},{"description":"Turn left on Porta Calvi.
Distance: 0.56. Time:
0.97.","startPoint":"1362126.440049746,5726586.273049667","endPo
int":"1362126.440049746,5726586.273049667","duration":"0.97","di
stance":"0.56"},{"description":"Turn left on Piazza del Duomo.
Distance: 0.03. Time:
0.05.","startPoint":"1362791.017409782,5727019.21938129","endPoi
```

nt":"1362791.017409782,5727019.21938129","duration":"0.05","dist
ance":"0.03"},{"description":"Arrival on Piazza del Duomo.
Distance: 0.0. Time:
0.0.","startPoint":"1362778.7722657947,5727034.830053719","endPo
int":"1362778.7722657947,5727034.830053719","duration":"0.0","di
stance":"0.0"}]}],"overviewPolyline":["1354047.9846028783,572324
3.195226749,-
1","1354054.663772326,5723262.302248355,11","1354063.5693315894,
5723292.555116262,15","1354079.1540603004,5723334.590850413,11",
"1354098.0783737353,5723376.945237439,11","1354122.5686617098,57
23421.529019981,11","1354151.4004098251,5723464.839193538,15","1
354174.7775028918,5723493.818912386,3","1354190.362231603,572351
0.378793999,-
13","1354209.2865450378,5723531.5563797,11","1354237.2277372268,
5723557.033241154,15","1354265.0576099253,5723579.484785625,3","
1354276.1895590045,5723587.127877489,17","1354289.4365784088,572
3597.318676826,11","1354310.6986011504,5723609.738729259,3","135
4321.8305502296,5723616.107993626,15","1354376.2657812277,572364
5.406668282,11","1354419.680382637,5723665.31071411,-
13","1354444.2819901025,5723678.5270250775,3","1354470.998667892
8,5723690.7879575845,11","1354584.433229011,5723746.36044737,-
13","1354615.714005924,5723762.124626422,11","1354783.695117531,
5723843.175239362,-
13","1354829.3361087563,5723865.786740175,11","1354879.541199104
,5723889.990663713,3","1354912.8257268511,5723905.755096602,15",
"1354988.5229805906,5723939.194894955,11","1355067.6711385446,57
23968.176154935,-
45","1355257.9161483103,5724033.463953724,3","1355384.8203678147
,5724078.051017757,-
13","1355450.4988673825,5724100.663113977,3","1355623.044078112,
5724160.537673773,-
13","1355679.8170184167,5724181.716803124,11","1355787.796924486
3,5724218.660968672,3","1355905.906904218,5724259.904851492,-
45","1356086.1331598123,5724321.850655443,-
45","1356268.6971247133,5724385.54858886,-
13","1356432.4480956702,5724442.080885774,3","1356454.600674338,
5724450.043209906,11","1356483.655061435,5724459.9165016925,11",
"1356504.6944451951,5724467.878841741,-
13","1356584.8444785662,5724500.206015351,11","1356616.125255479
2,5724522.341438265,-
13","1356626.0326901595,5724518.837986378,17","1356633.825054515
4,5724518.837986378,15","1356642.7306137786,5724523.615421113,17
","1356647.1833934104,5724534.603530715,11","1356670.560486477,5
724537.947740675,15","1356687.3697295866,5724544.795412594,15","
1356788.6704662086,5724597.825235937,11","1356823.1795083545,572
4615.661163294,-13","1357044.7052950333,5724737.169367553,-

29","1357090.3462862584,5724762.649672762,-
5","1357138.1023478087,5724791.95211375,3","1357170.4963196297,5
724808.51440559,3","1357201.5544575609,5724825.076728183,3","135
7282.9290053307,5724866.164161248,3","1357324.1172169242,5724886
.230185722,15","1357366.3073039348,5724904.066689823,11","135742
8.6462187793,5724925.247584768,11","1357522.1545910456,5724952.0
02471336,-13","1357556.6636331913,5724962.194830187,-
5","1357599.0763591838,5724974.457527367,15","1357633.5854013297
,5724985.605448524,11","1357674.773612923,5725003.442151361,15",
"1357843.867919438,5725087.848461079,-
109","1357909.546419006,5725121.45195009,11","1357969.6589440345
,5725147.729648194,11","1358032.1091783694,5725170.344456429,11"
,"1358120.051576096,5725197.100078251,-
45","1358176.824516401,5725214.937204061,11","1358214.5618237797
,5725225.9261654,15","1358248.0689905086,5725233.570668227,3","1
358308.070196046,5725247.10782473,3","1358318.0889502175,5725248
.063389493,15","1358399.4634979872,5725261.600567994,-
13","1358441.653584998,5725267.174706287,11","1358479.6135313585
,5725273.863676841,3","1358519.5772285534,5725280.234129652,3","
1358574.123779042,5725288.356463585,15","1358620.989284666,57252
92.656525721,11","1358673.1981258478,5725292.656525721,-
13","1358722.2900212877,5725288.356463585,11","1358768.932887930
2,5725285.011972245,3","1358815.798393554,5725286.286064037,15",
"1358879.1391838156,5725292.656525721,11","1358933.6857343041,57
25299.345515384,11","1358980.4399204375,5725310.493842639,11","1
359033.984595509,5725332.949801275,-
13","1359070.720027471,5725347.442684033,-
21","1359136.287207548,5725374.039574099,-
13","1359150.8700608418,5725377.065572605,17","1359164.228399737
,5725378.339677546,15","1359183.152713172,5725378.339677546,15",
"1359195.2865376684,5725377.065572605,3","1359213.2089756862,572
5370.695050632,15","1359237.6992636607,5725358.431808635,15","13
59256.6235770956,5725348.238996954,3","1359282.1157404871,572532
2.757018716,-
5","1359318.851172449,5725311.767938071,3","1359355.5866044108,5
725322.757018716,15","1359363.4902882571,5725329.605293216,11","
1359381.1900872933,5725348.238996954,15","1359390.0956465567,572
5376.26925711,3","1359389.0937711396,5725391.87705379,11","13593
96.7748160043,5725403.821814595,17","1359406.7935701758,5725417.
359229517,15","1359422.3782988868,5725428.507704292,15","1359433
.510247966,5725437.426494147,17","1359440.1894174137,5725451.760
282241,3","1359440.1894174137,5725461.953212237,17","1359444.753
5165362,5725528.68533859,11","1359446.979906352,5725550.82331419
7,15","1359452.4345614007,5725569.935279782,15","1359456.8873410
325,5725584.269280848,-
13","1359484.717213731,5725616.600501172,11","1359505.9792364724

,5725640.01283734,11","1359527.129939723,5725661.195480161,3","1
359578.2255859973,5725709.1353312805,3","1359607.2799730944,5725
729.203252453,15","1359636.2230407007,5725746.8821728,-
13","1359701.7902207777,5725779.213982653,-
45","1359769.8064296527,5725810.271742886,3","1359802.0890819826
,5725824.606130243,11","1359844.279168993,5725843.718682563,3","
1359903.2784991136,5725872.706131752,3","1359912.295377868,57258
77.325019718,15","1359969.0683181726,5725912.524210062,-
13","1359995.673676472,5725932.911134972,-
21","1360002.3528459198,5725939.282058561,11","1360012.482919582
,5725949.475545774,17","1360034.6354982497,5725976.233505149,3",
"1360054.6730065926,5726001.717350626,11","1360079.2746140577,57
26033.094435396,3","1360104.7667774493,5726062.082500147,15","13
60136.0475543623,5726090.911383269,-
13","1360216.1975877336,5726150.799305762,-
13","1360248.4802400637,5726175.487263222,11","1360271.857333130
2,5726194.600567547,11","1360289.668451657,5726208.935572676,3",
"1360338.5377081153,5726246.843808389,3","1360367.5920952123,572
6268.824287781,3","1360415.4594762535,5726307.847736113,15","136
0442.064834553,5726331.26188708,11","1360465.4419276197,5726358.
020992033,15","1360483.2530461466,5726382.709523179,-
13","1360491.156729993,5726393.699923544,15","1360506.6301392133
,5726420.459215859,15","1360525.6657721389,5726463.783954615,11"
,"1360532.3449415863,5726485.924251641,3","1360534.5713314023,57
26494.844099071,17","1360541.1391813592,5726530.364294312,15","1
360546.8164753895,5726568.273896433,11","1360551.2692550214,5726
607.298655195,15","1360552.2711304384,5726616.218624033,-
13","1360571.306763364,5726759.735777915,3","1360579.0991277196,
5726816.601704226,11","1360602.3649012954,5726841.291505031,15",
"1360655.7982568764,5726901.184467154,11","1360685.9658388812,57
26926.989537601,15","1360719.2503666284,5726947.856655349,11","1
360766.1158722523,5726965.697283293,15","1360791.7193551348,5726
971.272486842,-
45","1360887.4541172169,5726985.608740559,3","1360941.889348215,
5726993.732627893,3","1361052.0956441003,5727010.299009315,11","
1361117.7741436684,5727021.290183202,-
13","1361140.1493613177,5727023.838863143,11","1361162.301939985
7,5727026.865421511,11","1361265.9403859142,5727041.679643104,3"
,"1361294.7721340295,5727044.7062075315,11","1361313.807766955,5
727048.051358886,11","1361332.7320803902,5727051.555804503,3","1
361383.827726664,5727059.201872458,-
13","1361425.0159382576,5727063.8213750245,-
61","1361467.42866425,5727070.670986679,11","1361476.2229040226,
5727071.467453491,15","1361508.6168758434,5727077.042723177,3","
1361640.9757503967,5727095.998666175,-
109","1361686.6167416219,5727102.848302546,11","1361711.21834908

73,5727104.919123886,-
29","1361785.8024079187,5727116.069708566,3","1361832.5565940517
,5727122.760066072,11","1361880.423975093,5727129.609723002,-
13","1361900.4614834357,5727132.6363172745,3","1361914.933017239
,5727129.609723002,17","1361922.6140621037,5727124.990186356,15"
,"1361931.519621367,5727113.839590512,15","1361940.4251806305,57
27099.503130636,3","1361949.442059385,5727037.060152012,15","136
1958.3476186483,5726963.626494289,3","1361967.2531779117,5726896
.883629338,15","1361974.9342227764,5726856.583286906,11","136199
2.856660794,5726812.460195949,11","1362031.818482572,5726745.240
599796,11","1362090.8178126924,5726645.049298627,3","1362126.328
7302554,5726586.113764959,-
13","1362159.8358969842,5726598.37869528,15","1362197.573204363,
5726620.837897118,11","1362214.3824474728,5726631.987876491,15",
"1362236.5350261407,5726645.049298627,3","1362327.8170085913,572
6713.064574329,3","1362350.1922262406,5726726.603976839,11","136
2437.0214290593,5726782.195386155,-
13","1362520.3997276635,5726827.751928384,3","1362579.5103772746
,5726859.131919765,3","1362598.323371219,5726871.078646018,15","
1362650.7548513825,5726904.529564667,11","1362671.7942351424,572
6919.02500172,-
45","1362695.1713282089,5726933.361171431,15","1362712.982446736
,5726942.281466438,11","1362756.3970481453,5726990.387496916,3",
"1362790.906090291,5727019.21938129,3","1362778.7722657947,57270
34.989346436,-1"]};}

12.1.4.7 oym textual address xml format Request:

http://www.blomurbex.com/v02/GetRouting?**usertoken**=********&**format**=xml&**srs**=EPSG:3785&**provider**=oym&**f**=r&**ref**=1&**originAddress**=quinto%20di%20treviso&**destinationAddress**=treviso

Response:

```
<Routing>
      <route>
            <end-address></end-address>
            <start-address></start-address>
            <step>
                  <end-point>1354047.9846028783,5723243.354451757</end-
point>
                  <start-
point>1354047.9846028783,5723243.354451757</start-point>
                  <description>Head north on Via Vittorio Emanuele on
Sr515. Distance:
                        0.32. Time: 0.43.</description>
                  <duration>0.43</duration>
                  <distance>0.32</distance>
            </step>
```

```
            <step>
                  <end-point>1354289.5478978995,5723597.318676826</end-
point>
                  <start-
point>1354289.5478978995,5723597.318676826</start-point>
                  <description>Keep right on Via Vittorio Emanuele on
Sr515. Distance:
                        1.76. Time: 2.37.</description>
                  <duration>2.37</duration>
                  <distance>1.76</distance>
            </step>
            <step>
                  <end-point>1356616.1252554792,5724522.341438265</end-
point>
                  <start-
point>1356616.1252554792,5724522.341438265</start-point>
                  <description>At the roundabout, go right. Distance:
1.98. Time: 2.37.
                  </description>
                  <duration>2.37</duration>
                  <distance>1.98</distance>
            </step>
            <step>
                  <end-point>1359256.6235770956,5725348.398259548</end-
point>
                  <start-
point>1359256.6235770956,5725348.398259548</start-point>
                  <description>At the roundabout, go left. ontinue on
Sr515. Distance:
                        1.52. Time: 1.78.</description>
                  <duration>1.78</duration>
                  <distance>1.52</distance>
            </step>
            <step>
                  <end-point>1360579.0991277196,5726816.601704226</end-
point>
                  <start-
point>1360579.0991277196,5726816.601704226</start-point>
                  <description>Turn slightly right on Via Carlo
Cattaneo. Distance:
                        1.4. Time: 1.93.</description>
                  <duration>1.93</duration>
                  <distance>1.4</distance>
            </step>
            <step>
                  <end-point>1362126.440049746,5726586.273049667</end-
point>
                  <start-
point>1362126.440049746,5726586.273049667</start-point>
                  <description>Turn left on Porta Calvi. Distance:
0.56. Time: 0.97.
                  </description>
                  <duration>0.97</duration>
                  <distance>0.56</distance>
            </step>
            <step>
```

```
                  <end-point>1362791.017409782,5727019.21938129</end-
point>
                  <start-
point>1362791.017409782,5727019.21938129</start-point>
                  <description>Turn left on Piazza del Duomo. Distance:
0.03. Time:
                        0.05.</description>
                  <duration>0.05</duration>
                  <distance>0.03</distance>
            </step>
            <step>
                  <end-point>1362778.7722657947,5727034.830053719</end-
point>
                  <start-
point>1362778.7722657947,5727034.830053719</start-point>
                  <description>Arrival on Piazza del Duomo. Distance:
0.0. Time: 0.0.
                  </description>
                  <duration>0.0</duration>
                  <distance>0.0</distance>
            </step>
            <description></description>
            <duration>9.9</duration>
            <distance>7.57KM</distance>
      </route>
      <overviewPolyline>
            <point>1354047.9846028783,5723243.195226749,-1</point>
            <point>1354054.663772326,5723262.302248355,11</point>
            <point>1354063.5693315894,5723292.555116262,15</point>
            <point>1354079.1540603004,5723334.590850413,11</point>
            <point>1354098.0783737353,5723376.945237439,11</point>
            <point>1354122.5686617098,5723421.529019981,11</point>
            <point>1354151.4004098251,5723464.839193538,15</point>
            <point>1354174.7775028918,5723493.818912386,3</point>
            <point>1354190.362231603,5723510.378793999,-13</point>
            <point>1354209.2865450378,5723531.5563797,11</point>
            <point>1354237.2277372268,5723557.033241154,15</point>
            <point>1354265.0576099253,5723579.484785625,3</point>
            <point>1354276.1895590045,5723587.127877489,17</point>
            <point>1354289.4365784088,5723597.318676826,11</point>
            <point>1354310.6986011504,5723609.738729259,3</point>
            <point>1354321.8305502296,5723616.107993626,15</point>
            <point>1354376.2657812277,5723645.406668282,11</point>
            <point>1354419.680382637,5723665.31071411,-13</point>
            <point>1354444.2819901025,5723678.5270250775,3</point>
            <point>1354470.9986678928,5723690.7879575845,11</point>
            <point>1354584.433229011,5723746.36044737,-13</point>
            <point>1354615.714005924,5723762.124626422,11</point>
            <point>1354783.695117531,5723843.175239362,-13</point>
            <point>1354829.3361087563,5723865.786740175,11</point>
            <point>1354879.541199104,5723889.990663713,3</point>
            <point>1354912.8257268511,5723905.755096602,15</point>
            <point>1354988.5229805906,5723939.194894955,11</point>
            <point>1355067.6711385446,5723968.176154935,-45</point>
            <point>1355257.9161483103,5724033.463953724,3</point>
            <point>1355384.8203678147,5724078.051017757,-13</point>
            <point>1355450.4988673825,5724100.663113977,3</point>
```

```
<point>1355623.044078112,5724160.537673773,-13</point>
<point>1355679.8170184167,5724181.716803124,11</point>
<point>1355787.7969244863,5724218.660968672,3</point>
<point>1355905.906904218,5724259.904851492,-45</point>
<point>1356086.1331598123,5724321.850655443,-45</point>
<point>1356268.6971247133,5724385.54858886,-13</point>
<point>1356432.4480956702,5724442.080885774,3</point>
<point>1356454.600674338,5724450.043209906,11</point>
<point>1356483.655061435,5724459.9165016925,11</point>
<point>1356504.6944451951,5724467.878841741,-13</point>
<point>1356584.8444785662,5724500.206015351,11</point>
<point>1356616.1252554792,5724522.341438265,-13</point>
<point>1356626.0326901595,5724518.837986378,17</point>
<point>1356633.8250545154,5724518.837986378,15</point>
<point>1356642.7306137786,5724523.615421113,17</point>
<point>1356647.1833934104,5724534.603530715,11</point>
<point>1356670.560486477,5724537.947740675,15</point>
<point>1356687.3697295866,5724544.795412594,15</point>
<point>1356788.6704662086,5724597.825235937,11</point>
<point>1356823.1795083545,5724615.661163294,-13</point>
<point>1357044.7052950333,5724737.169367553,-29</point>
<point>1357090.3462862584,5724762.649672762,-5</point>
<point>1357138.1023478087,5724791.95211375,3</point>
<point>1357170.4963196297,5724808.51440559,3</point>
<point>1357201.5544575609,5724825.076728183,3</point>
<point>1357282.9290053307,5724866.164161248,3</point>
<point>1357324.1172169242,5724886.230185722,15</point>
<point>1357366.3073039348,5724904.066689823,11</point>
<point>1357428.6462187793,5724925.247584768,11</point>
<point>1357522.1545910456,5724952.002471336,-13</point>
<point>1357556.6636331913,5724962.194830187,-5</point>
<point>1357599.0763591838,5724974.457527367,15</point>
<point>1357633.5854013297,5724985.605448524,11</point>
<point>1357674.773612923,5725003.442151361,15</point>
<point>1357843.867919438,5725087.848461079,-109</point>
<point>1357909.546419006,5725121.45195009,11</point>
<point>1357969.6589440345,5725147.729648194,11</point>
<point>1358032.1091783694,5725170.344456429,11</point>
<point>1358120.051576096,5725197.100078251,-45</point>
<point>1358176.824516401,5725214.937204061,11</point>
<point>1358214.5618237797,5725225.9261654,15</point>
<point>1358248.0689905086,5725233.570668227,3</point>
<point>1358308.070196046,5725247.10782473,3</point>
<point>1358318.0889502175,5725248.063389493,15</point>
<point>1358399.4634979872,5725261.600567994,-13</point>
<point>1358441.653584998,5725267.174706287,11</point>
<point>1358479.6135313585,5725273.863676841,3</point>
<point>1358519.5772285534,5725280.234129652,3</point>
<point>1358574.123779042,5725288.356463585,15</point>
<point>1358620.989284666,5725292.656525721,11</point>
<point>1358673.1981258478,5725292.656525721,-13</point>
<point>1358722.2900212877,5725288.356463585,11</point>
<point>1358768.9328879302,5725285.011972245,3</point>
<point>1358815.798393554,5725286.286064037,15</point>
<point>1358879.1391838156,5725292.656525721,11</point>
<point>1358933.6857343041,5725299.345515384,11</point>
<point>1358980.4399204375,5725310.493842639,11</point>
```

<point>1359033.984595509,5725332.949801275,-13</point>
<point>1359070.720027471,5725347.442684033,-21</point>
<point>1359136.287207548,5725374.039574099,-13</point>
<point>1359150.8700608418,5725377.065572605,17</point>
<point>1359164.228399737,5725378.339677546,15</point>
<point>1359183.152713172,5725378.339677546,15</point>
<point>1359195.2865376684,5725377.065572605,3</point>
<point>1359213.2089756862,5725370.695050632,15</point>
<point>1359237.6992636607,5725358.431808635,15</point>
<point>1359256.6235770956,5725348.238996954,3</point>
<point>1359282.1157404871,5725322.757018716,-5</point>
<point>1359318.851172449,5725311.767938071,3</point>
<point>1359355.5866044108,5725322.757018716,15</point>
<point>1359363.4902882571,5725329.605293216,11</point>
<point>1359381.1900872933,5725348.238996954,15</point>
<point>1359390.0956465567,5725376.26925711,3</point>
<point>1359389.0937711396,5725391.87705379,11</point>
<point>1359396.7748160043,5725403.821814595,17</point>
<point>1359406.7935701758,5725417.359229517,15</point>
<point>1359422.3782988868,5725428.507704292,15</point>
<point>1359433.510247966,5725437.426494147,17</point>
<point>1359440.1894174137,5725451.760282241,3</point>
<point>1359440.1894174137,5725461.953212237,17</point>
<point>1359444.7535165362,5725528.68533859,11</point>
<point>1359446.979906352,5725550.823314197,15</point>
<point>1359452.4345614007,5725569.935279782,15</point>
<point>1359456.8873410325,5725584.269280848,-13</point>
<point>1359484.717213731,5725616.600501172,11</point>
<point>1359505.9792364724,5725640.01283734,11</point>
<point>1359527.129939723,5725661.195480161,3</point>
<point>1359578.2255859973,5725709.1353312805,3</point>
<point>1359607.2799730944,5725729.203252453,15</point>
<point>1359636.2230407007,5725746.8821728,-13</point>
<point>1359701.7902207777,5725779.213982653,-45</point>
<point>1359769.8064296527,5725810.271742886,3</point>
<point>1359802.0890819826,5725824.606130243,11</point>
<point>1359844.279168993,5725843.718682563,3</point>
<point>1359903.2784991136,5725872.706131752,3</point>
<point>1359912.295377868,5725877.325019718,15</point>
<point>1359969.0683181726,5725912.524210062,-13</point>
<point>1359995.673676472,5725932.911134972,-21</point>
<point>1360002.3528459198,5725939.282058561,11</point>
<point>1360012.482919582,5725949.475545774,17</point>
<point>1360034.6354982497,5725976.233505149,3</point>
<point>1360054.6730065926,5726001.717350626,11</point>
<point>1360079.2746140577,5726033.094435396,3</point>
<point>1360104.7667774493,5726062.082500147,15</point>
<point>1360136.0475543623,5726090.911383269,-13</point>
<point>1360216.1975877336,5726150.799305762,-13</point>
<point>1360248.4802400637,5726175.487263222,11</point>
<point>1360271.8573331302,5726194.600567547,11</point>
<point>1360289.668451657,5726208.935572676,3</point>
<point>1360338.5377081153,5726246.843808389,3</point>
<point>1360367.5920952123,5726268.824287781,3</point>
<point>1360415.4594762535,5726307.847736113,15</point>
<point>1360442.064834553,5726331.26188708,11</point>
<point>1360465.4419276197,5726358.020992033,15</point>

```
<point>1360483.2530461466,5726382.709523179,-13</point>
<point>1360491.156729993,5726393.699923544,15</point>
<point>1360506.6301392133,5726420.459215859,15</point>
<point>1360525.6657721389,5726463.783954615,11</point>
<point>1360532.3449415863,5726485.924251641,3</point>
<point>1360534.5713314023,5726494.844099071,17</point>
<point>1360541.1391813592,5726530.364294312,15</point>
<point>1360546.8164753895,5726568.273896433,11</point>
<point>1360551.2692550214,5726607.298655195,15</point>
<point>1360552.2711304384,5726616.218624033,-13</point>
<point>1360571.306763364,5726759.735777915,3</point>
<point>1360579.0991277196,5726816.601704226,11</point>
<point>1360602.3649012954,5726841.291505031,15</point>
<point>1360655.7982568764,5726901.184467154,11</point>
<point>1360685.9658388812,5726926.989537601,15</point>
<point>1360719.2503666284,5726947.856655349,11</point>
<point>1360766.1158722523,5726965.697283293,15</point>
<point>1360791.7193551348,5726971.272486842,-45</point>
<point>1360887.4541172169,5726985.608740559,3</point>
<point>1360941.889348215,5726993.732627893,3</point>
<point>1361052.0956441003,5727010.299009315,11</point>
<point>1361117.7741436684,5727021.290183202,-13</point>
<point>1361140.1493613177,5727023.838863143,11</point>
<point>1361162.3019399857,5727026.865421511,11</point>
<point>1361265.9403859142,5727041.679643104,3</point>
<point>1361294.7721340295,5727044.7062075315,11</point>
<point>1361313.807766955,5727048.051358886,11</point>
<point>1361332.7320803902,5727051.555804503,3</point>
<point>1361383.827726664,5727059.201872458,-13</point>
<point>1361425.0159382576,5727063.8213750245,-61</point>
<point>1361467.42866425,5727070.670986679,11</point>
<point>1361476.2229040226,5727071.467453491,15</point>
<point>1361508.6168758434,5727077.042723177,3</point>
<point>1361640.9757503967,5727095.998666175,-109</point>
<point>1361686.6167416219,5727102.848302546,11</point>
<point>1361711.2183490873,5727104.919123886,-29</point>
<point>1361785.8024079187,5727116.069708566,3</point>
<point>1361832.5565940517,5727122.760066072,11</point>
<point>1361880.423975093,5727129.609723002,-13</point>
<point>1361900.4614834357,5727132.6363172745,3</point>
<point>1361914.933017239,5727129.609723002,17</point>
<point>1361922.6140621037,5727124.990186356,15</point>
<point>1361931.519621367,5727113.839590512,15</point>
<point>1361940.4251806305,5727099.503130636,3</point>
<point>1361949.442059385,5727037.060152012,15</point>
<point>1361958.3476186483,5726963.626494289,3</point>
<point>1361967.2531779117,5726896.883629338,15</point>
<point>1361974.9342227764,5726856.583286906,11</point>
<point>1361992.856660794,5726812.460195949,11</point>
<point>1362031.818482572,5726745.240599796,11</point>
<point>1362090.8178126924,5726645.049298627,3</point>
<point>1362126.3287302554,5726586.113764959,-13</point>
<point>1362159.8358969842,5726598.37869528,15</point>
<point>1362197.573204363,5726620.837897118,11</point>
<point>1362214.3824474728,5726631.987876491,15</point>
<point>1362236.5350261407,5726645.049298627,3</point>
<point>1362327.8170085913,5726713.064574329,3</point>
```

```
            <point>1362350.1922262406,5726726.603976839,11</point>
            <point>1362437.0214290593,5726782.195386155,-13</point>
            <point>1362520.3997276635,5726827.751928384,3</point>
            <point>1362579.5103772746,5726859.131919765,3</point>
            <point>1362598.323371219,5726871.078646018,15</point>
            <point>1362650.7548513825,5726904.529564667,11</point>
            <point>1362671.7942351424,5726919.02500172,-45</point>
            <point>1362695.1713282089,5726933.361171431,15</point>
            <point>1362712.982446736,5726942.281466438,11</point>
            <point>1362756.3970481453,5726990.387496916,3</point>
            <point>1362790.906090291,5727019.21938129,3</point>
            <point>1362778.7722657947,5727034.989346436,-1</point>
        </overviewPolyline>
</Routing>
```

13 POI Services

BlomURBEX provides a fully functional POI platform to access all its POIs and all information needed to process and represent them

13.1 GetPOICategories services

13.1.1 Description

This service returns a description of all POI categories in the platform.

13.1.2 Parameters

The parameters for this service are:

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.
- **format [optional]:** Response format. Possible values are:
    - **Xml:** Result is a XML file containing all the categories (default format if parameter omitted)
    - **js:** The result is a JavaScript function that will return an array with all the categories. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.
- **lang [optional]:** 3-letter iso code of the language in which the delivered information will be presented. If omitted, information will be sent in default language.
- **providers [optional]:** Name of the POI providers the information is desired from. If omitted, all allowed providers' info will be returned. Only information of allowed providers will be returned, whether this parameter is specified on the request or not. For example, if any given user have permission for only checking out information from 'provider1' and he requests for 'provider2' he will not receive any information. If, on the other hand, he does not specify any provider on his request, he will get only information from 'provider1'.

13.1.3 Examples

13.1.3.1 Getting all categories from all providers in XML format

```
http://www.blomurbex.com/v02/GetPOICategories?usertoken=********&format=xml
```

Above example should deliver a response like the following

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<categories>
       <category description="" icon="" id="-32756" name="AEROPUERTOS"
       parentCategoryId="null" version="-1">
```

```
                <category description="" icon="" id="-32721" name="AEROPUERTOS"
                parentCategoryId="-32756" version="-1" />
        </category>
        <category description="" icon="" id="-32755" name="ALOJAMIENTOS"
        parentCategoryId="null" version="-1">
                <category description="" icon="" id="-32720" name="PENSIONES"
                parentCategoryId="-32755" version="-1" />
                <category description="" icon="" id="-32719" name="MOTELES"
                parentCategoryId="-32755" version="-1" />
                <category description="" icon="" id="-32718" name="HOSTALES"
                parentCategoryId="-32755" version="-1" />
                <category description="" icon="" id="-32717"
                name="CASAS RURALES" parentCategoryId="-32755" version="-1" />
                <category description="" icon="" id="-32716" name="CAMPINGS"
                parentCategoryId="-32755" version="-1" />
                <category description="" icon="" id="-32715"
                name="CABA&#209;AS Y REFUGIOS DE MONTA&#209;A"
                parentCategoryId="-32755" version="-1" />
                <category description="" icon="" id="-32714" name="BALNEARIOS"
                parentCategoryId="-32755" version="-1" />
                <category description="" icon="" id="-32713"
                name="APARTHOTELES" parentCategoryId="-32755" version="-1" />
                <category description="" icon="" id="-32712"
                name="APARTAMENTOS" parentCategoryId="-32755" version="-1" />
                <category description="" icon="" id="-32711" name="ALBERGUES"
                parentCategoryId="-32755" version="-1" />
        </category>
        <category description="" icon="" id="-32754"
        name="ALQUILER DE VEHICULOS" parentCategoryId="null"
        version="-1">
                <category description="" icon="" id="-32710"
                name="MOTOCICLETAS ALQUILER" parentCategoryId="-32754"
                version="-1" />
                <category description="" icon="" id="-32709"
                name="AVIS ALQUILER DE AUTOMOVILES" parentCategoryId="-32754"
                version="-1" />
                <category description="" icon="" id="-32708"
                name="EUROPCAR ALQUILER DE AUTOMOVILES"
                parentCategoryId="-32754" version="-1" />
                <category description="" icon="" id="-32707"
                name="AUTOMOVILES ALQUILER" parentCategoryId="-32754"
                version="-1" />
        </category>
</categories>
```

13.2 GetPOIs Service

13.2.1 Description

This service returns information about the nearest POIs to a given geographical point or all the POIs belonging to a certain geographical bounding box. It will deliver a maximum number of pois of 2000. In case of bounding box request, the POIs will be sorted in terms of distance to the centre of the box.

13.2.2 Parameters

The parameters for this service are:

- **Crs or srs [mandatory]:** Projection system. It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.
- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.
- **format [optional]:** Response format. Possible values are:
    - **Xml:** Result is a XML file containing all the information about the POIs requested. This is the default format if 'format' parameter is not given in the request.
    - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.
- **X [optional]:** X coordinate of the point from which the nearest POIs will be calculated, in the coordinate system defined by crs/srs.
- **Y [optional]:** coordinate of the point from which the nearest POIs will be calculated, in the coordinate system defined by crs/srs
- **bbox[optional]:** String defining the bounding box in the format xmin,ymin,xmax,ymax. **IMPORTANT NOTICE:** Either a x,y pair or a bbox parameter should be provided on the request. If not, a parameters missing error will be delivered.
- **lang [optional]:** 3-letter iso code of the language in which the delivered information will be presented. If omitted, information will be sent in default language.
- **providers [optional]:** Name of the POI providers the information is desired from. If omitted, all allowed providers' info will be returned. Only information of allowed providers will be returned, whether this parameter is specified on the request or not. For example, if any given user have permission for only checking out information from 'provider1' and he requests for 'provider2' he will not receive any information. If, on the other hand, he does not specify any provider on his request, he will get only information from 'provider1'

- **number[optional]:** Number of pois to be delivered. If not present, default number of POIs defined by the platform will be delivered. If too high, max number defined by the platform will be delivered.

- **Categories[optional]:** List of category ids the POIs delivered will belong to. If not present, all categories will be taken into account.

13.2.3 Examples

13.2.3.1 Asking for 5 nearest POIs belonging to certain categories

```
http://www.blomurbex.com/v02/GetPOIs?usertoken=********&format=xml&x=-
3&y=40&srs=EPSG:4326&number=5&categories=-32527,-32688
```

Above request should return this response:

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<pois>
	<poi email="" id="-2147006372" latitude="40.00296"
	longitude="-3.00001" phone="969325125" subtitle="" text=""
	title="DESTO. TARANCON" url="" version="-1">
		<categories>
			<category id="-32527" />
		</categories>
	</poi>
	<poi email="" id="-2146908985" latitude="40.00296"
	longitude="-3.00001" phone="969320900" subtitle="" text=""
	title="PUESTO P. DE TARANCON" url="" version="-1">
		<categories>
			<category id="-32527" />
		</categories>
	</poi>
	<poi email="" id="-2146762746" latitude="40.00296"
	longitude="-3.00001" phone="969320900" subtitle="" text=""
	title="COMPA&#209;IA DE TARANCON" url="" version="-1">
		<categories>
			<category id="-32527" />
		</categories>
	</poi>
	<poi email="" id="-2146716294" latitude="40.00296"
	longitude="-3.00001" phone="969322235" subtitle="" text=""
	title="SERVICIO DE SALUD DE CASTILLA LA MANCHA" url=""
	version="-1">
		<categories>
			<category id="-32688" />
		</categories>
	</poi>
	<poi email="" id="-2146863033" latitude="40.01154"
	longitude="-3.00435" phone="969324962" subtitle="" text=""
	title="ASOCIACION PRO SALUD MENTAL NUEVA LUZ" url=""
	version="-1">
		<categories>
			<category id="-32688" />
		</categories>
	</poi>
</pois>
```

13.2.3.2 Example using bounding box

http://www.blomurbex.com/v02/GetPOIs?**usertoken**=********&**format**=xml&**bbox**=-3.5,35,-3,40&**srs**=EPSG:4326&**number**=5

```
<?xml version="1.0" encoding="UTF-8"?>
<pois>
	<poi title="PUESTO DE PEDRO MARTINEZ" text="" version="-1" url="" subtitle="" phone="958687005" longitude="-3.23163" latitude="37.50058" id="1d4125ec-972c-4bc4-a951-49df43da19a4" email=""><categories><category id="00796f04-a7ce-4f4d-af67-d5f0b6a89a7d"/></categories><icons><icon url="http://192.168.40.37/v02/GetPOIInfo?usertoken=********&mediaId=a0c28ff9-9e4f-4192-a910-f26b508c397e&reqtype=icon" id="a0c28ff9-9e4f-4192-a910-f26b508c397e"/></icons></poi>

	<poi title="JUAN ANTONIO FERNANDEZ VACA" text="" version="-1" url="" subtitle="" phone="958687166" longitude="-3.2312" latitude="37.50172" id="483f7e26-051e-4ccb-b3d9-83c9f2ebec13" email=""><categories><category id="b55a8f17-4133-4d3e-862f-f87913e71ef3"/></categories><icons><icon url="http://192.168.40.37/v02/GetPOIInfo?usertoken=********&mediaId=a0c28ff9-9e4f-4192-a910-f26b508c397e&reqtype=icon" id="a0c28ff9-9e4f-4192-a910-f26b508c397e"/></icons></poi>

	<poi title="CENTRO SALUD PEDRO MARTINEZ" text="" version="-1" url="" subtitle="" phone="958687247" longitude="-3.23098" latitude="37.50036" id="95be974f-05a8-4ba6-a878-32693bf6bbf8" email=""><categories><category id="d6a3c6b2-44ed-4ff0-a6fe-9488dd82c438"/></categories><icons><icon url="http://192.168.40.37/v02/GetPOIInfo?usertoken=********&mediaId=a0c28ff9-9e4f-4192-a910-f26b508c397e&reqtype=icon" id="a0c28ff9-9e4f-4192-a910-f26b508c397e"/></icons></poi>

	<poi title="BENJAMIN LINDEZ LINDEZ" text="" version="-1" url="" subtitle="" phone="958687015" longitude="-3.23088" latitude="37.49909" id="c24ffb82-6eba-4c07-adb7-297fc515d947" email=""><categories><category id="a94311ff-59d5-4837-8f09-ccdd9dee81c5"/></categories><icons><icon url="http://192.168.40.37/v02/GetPOIInfo?usertoken=********&mediaId=a0c28ff9-9e4f-4192-a910-f26b508c397e&reqtype=icon" id="a0c28ff9-9e4f-4192-a910-f26b508c397e"/></icons></poi>

	<poi title="CAJA GRANADA" text="" version="-1" url="" subtitle="" phone="" longitude="-3.2306591" latitude="37.5016808" id="4d9473b0-c854-4ce3-9d20-a234b194d927" email=""><categories><category id="e789942a-3eee-4ba2-ad15-95ac383897ae"/></categories><icons><icon url="http://192.168.40.37/v02/GetPOIInfo?usertoken=********&mediaId=a0c28ff9-9e4f-4192-a910-f26b508c397e&reqtype=icon" id="a0c28ff9-9e4f-4192-a910-f26b508c397e"/></icons></poi>
</pois>
```

13.3 GetPOIInfo Service

13.3.1 Description

This service returns all information regarding a certain POI, incluiding media items such as images, icons, videos or audios.

13.3.2 Parameters

The parameters for this service are:

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.
- **format [optional]:** Response format. Possible values are:
    - **Xml:** Result is a XML file containing all the information about the POIs requested. This is the default format if 'format' parameter is not given in the request.
    - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.
- **lang [optional]:** 3-letter iso code of the language in which the delivered information will be presented. If omitted, information will be sent in default language.
- **providers [optional]:** Name of the POI providers the information is desired from. If omitted, all allowed providers' info will be returned. Only information of allowed providers will be returned, whether this parameter is specified on the request or not. For example, if any given user have permission for only checking out information from 'provider1' and he requests for 'provider2' he will not receive any information. If, on the other hand, he does not specify any provider on his request, he will get only information from 'provider1'
- **id[mandatory]:** Id of the poi whose data will be retrieved.
- **mediaId[mandatory/optional]:** Id of the media item to be returned (if a media request if performed, see next parameter for details).
- **reqtype[optional]:** Type of request. Available types are:
    - audio: The audio file referenced by the parameter *mediaId* will be returned.
    - video: The video file referenced by the parameter *mediaId* will be returned.
    - image: The image file referenced by the parameter *mediaId* will be returned.
    - icon: The icon file referenced by the parameter *mediaId* will be returned.
    - extra: The extra POI parameter referenced by the parameter *mediaId* will be returned.
    - info (default if parameter not present): Delivers the POI info in the same format as GetPOIs service

13.3.3 Examples

13.3.3.1 Asking for the info of a POI

http://www.blomurbex.com/v02/GetPOIInfo?**usertoken**=********&**id**=-2146716294

Above request should return this response:

```
<?xml version="1.0" encoding="UTF-8"?>
<pois>
	<poi email="" id="-2146716294" latitude="40.00296" longitude="-3.00001"
	phone="969322235" subtitle="" text="" title="SERVICIO DE SALUD DE CASTILLA LA
	MANCHA" url="" version="-1">
		<categories>
			<category id="-32688" />
		</categories>
	</poi>
</pois>
```

14 Wireframe Services

14.1 GetWireframe Service

14.1.1 Description

This service delivers the wireframe file for a building containing/next to a given geo-referenced point. It will search for a building containing the given point. If found, the wireframe file of the building will be delivered in the requested format. If not, the wireframe file of the nearest building within a given search radius will be delivered.

14.1.2 Parameters

The parameters for this service are:

- **x [mandatory]**: X coordinate for the point to search the building at.
- **y [mandatory]**: Y coordinate for the point to search the building at.
- **crs or srs [mandatory]:** Point projection system. It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.
- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.
- **radius [optional]:** Search radius, in decimal meters. If not supplied, a default value of 10 meters will be set.
- **format [optional]:** Format used for the response file. Possible values are:
  - **dwg:** A dwg file will be delivered. This will be the default format if no format has been set in the request.
  - **3ds:** A dwg file will be delivered.

14.1.3 Examples

14.1.3.1 Search for a building within a 50 meters radius and deliver its 3ds format wireframe

In this example, the system is queried with a Lon/Lat point value to search for a 3ds wireframe file of a building containing the point or the nearest building to the point within a 50 meters radius if no building contains it.

```
http://www.blomurbex.com/v02/GetWireframe?usertoken=********&x=-
87.7710231809672&y=30.3954927639296&crs=EPSG:4326&format=dwg&radius=50
```

The result should be a dwg file with the wireframe information of the building.

15 Proxy Services

15.1 GenericProxy

15.1.1 Description

This service shall be used as a generic proxy for connecting to any external URL and converting its response into a JSONP response, so that any external URL can be contacted using Javascript API through this service, avoiding thus the cross-domain restriction of web browsers. An example of use of this service is the WFS server connection in BlomWEB

15.1.2 Parameters

The parameters for this service are:

- **url[mandatory]:** URL to connect to
- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.
- **function [mandatory]:** Function inside which the result of the request is going to be wrapped (JSONP format).
- **user[optional]**: If the remote part supports/requires HTTP basic authentication, then user/password can be provided to the proxy service to use them.
- **password [optional]**: If the remote part supports/requires HTTP basic authentication, then user/password can be provided to the proxy service to use them.

15.1.3 Examples

15.1.3.1 Asking for a WFS GetCapabilities document through GenericProxy

```
http://www.blomurbex.com/v02/GenericProxy?usertoken=********&function=FWFSLayerGetCapabilities433&url=http%3A%2F%2Fdemo.cubewerx.com%2Fdemo%2Fcubeserv%2Fcubeserv.cgi%3Fdatastore%3DFoundation%26service%3DWFS%26request%3DGetCapabilities0
```

The result will be the response of the URL requested, inside a Javascript function as a string to be returned when that function is called.

16 Other Services

16.1 ProjectCoordinates Service

16.1.1 Description

This service converts points between different coordinate systems.

16.1.2 Parameters

The parameters for this service are:

- **points [mandatory]**: This parameter is a list of coordinates separated by commas. The following rules must be observed to provide the list:

    - The number of coordinates must be an even number.
    - Odd coordinates, represent X coordinates.
    - Even coordinates, represent Y coordinates.

  The list will be converted from the reference system provided as crs1 to the reference system provided as crs2 (see below).

- **Crs1 or srs1 [mandatory]:** Origin projection system. It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.

- **Crs2 or srs2 [mandatory]:** Target projection system. It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.

- **format [mandatory]:** Format used for the response. Possible values are:

    - **txt:** The service response will be the list of transformed coordinates separated by commas (“,”).
    - **Xml:** The service response is an XML file, with the list of coordinates transformed to the new projection system.
    - **js:** The service response is a JavaScript function that will return the values. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.

- **usertoken [mandatory]**: User’s Token for authentication purposes. This token is unique for each client and must be requested to Blom.

16.1.3 Examples

16.1.3.1 ProjectCoordinates with plain text output

In this example the service is called to transform a coordinate in EPSG:32630 reference system to geographical coordinates WGS84 (EPSG:4326). Results are returned as plain text.

```
http://www.blomurbex.com/v02/ProjectCoordinates?usertoken=********&points=355919,4612881&crs1=EPSG:32630&crs2=EPSG:4326&format=txt
```

The result is a text string with an even number of coordinates, where odd coordinates correspond to X coordinates and even coordinates correspond to Y coordinates in the target projection system. Values are separated by commas.

In the example above, the result should be:

-4.730382771703441,41.65466037756979

16.1.3.2 ProjectCoordinates with Xml output

In this example the service is called to transform a coordinate in EPSG:32630 reference system to geographical coordinates WGS84 (EPSG:4326). Results are returned as an XML file.

```
http://www.blomurbex.com/v02/ProjectCoordinates?usertoken=********&points=355919,4612881&crs1=EPSG:32630&crs2=EPSG:4326&format=xml
```

The result is an XML file with point tags including X and Y coordinates in the target projection system.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<result>
      <Point>
            <X>-4.730382771703441</X>
            <Y>41.65466037756979</Y>
      </Point>
</result>
```

16.2 ExtractAll Service

16.2.1 Description

This service receives a coordinate and returns a zip file with 5 files, one for each of the views availables of that point in the BlomURBEX database. The coordinate is considered as the center of the views to locate. Images are returned as JPG files.

If any image doesn't exist, it won't be included in the zip file.

16.2.2 Parameters

Parameters of the service are described below:

- **x [mandatory]**: Coordinate x or longitude in the specified projection system.
- **y [mandatory]**: Coordinate y or latitude in the specified projection system.

  This point will be considered the center of the image to be returned by the service
- **crs or srs [mandatory]:** Projection system used, It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.
- **width [mandatory]:** Width in pixels of the desired image.
- **height [mandatory]:** Height in pixels of the desired image.
- **zoom [mandatory]:** Desired zoom level for the oblique image.
  - Zoom=1 for level 1:1
  - Zoom=2 for level 1:2
  - Zoom=3 for level 1:4
  - Zoom=4 for level 1:8
- **time [optional]** : A date following the structure YYYYMMDD or YYYYMM or YYYY. The service will use the most current images available, whose date is previous to the requested date. It is also possible to use a range of dates with the structure YYYYMMDD-YYYYMMDD, in this case the images used will be the most current ones available in the specified date range.

  If the parameter is omitted the system will submit the most recent version of the tile or an error XML file if none is available.
- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

16.2.3 Example

Below is a request example for this service:

```
http://www.blomurbex.com/v02/ExtractAll?usertoken=********&x=356426&y=4501040&
crs=EPSG:32630&width=950&height=500&zoom=1
```

The service returns a zip file called **BlomURBEXExtractAll.zip**

16.3 GetDynamicToken Service

16.3.1 Description

This service delivers a dynamic user token for the given user and password. This token will only be valid for a certain amount of time which will be specified in the request. This service is currently unavailable.

16.3.2 Parameters

The parameters for this service are:

- **user [mandatory if no userToken provided]**: Web user id for whom the dynamic token will be generated.
- **password[mandatory if no userToken provided]:** Password for the user id.
- **userToken[mandatory if no user and password provided]:** User token of the user for whom the dynamic token will be generated.
- **Seconds:** Number of second until the dynamic token expires.

16.3.3 Examples

This service is currently unavailable.

16.4 GetZones Service

16.4.1 Description

This service returns the loaded zones in a particular region.

16.4.2 Parameters

The parameters for this service are:

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.
- **format [mandatory]:** Response format. Possible values are:
  - **txt:** The result is a text string describing each of the loaded zones that comply with the requests. Records are separated by "#". Fields in each record are separated by ",".
  - **Xml:** The result is an XML file describing each of the loaded zones that comply with the requests.
  - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.
- **crs [mandatory]:** Coordinate reference system used, It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.

  This Coordinate reference system will be used for specifying the region of interest in the input parameters and the extensions of the loaded layers in the response.
- **points [mandatory]**: This parameter is a list of coordinates separated by commas. It defines the region where the service has to check for loaded Zones. The following rules must be observed to provide the list:
  - The number of coordinates must be an even number
  - Odd coordinates, represent X coordinates
  - Even coordinates, represent Y coordinates
- **types [optional]**: This parameter is a list of types of zones that want to be retrieved. The different types will be comma separated. Possible values for types are:
  - **ortho** – orthogonal data
  - **rect_north**, **rect_south**, **rect_east**, **rect_west** – ortho rectified oblique data. There is a record for every orientation, since they have different coverage.
  - **north**, **south**, **east**, **west** – natural oblique data. There is a record for every orientation, since they have different coverage.
  - **dtm** – Dtm data

- **3d** – 3D model

The output of the service will be a list of zones that meet all criteria. For every zone there will be a record with information of the zone and the coordinates of its extent:

- Layer Code. The identifier of the layer
- Layer name. Long text description (common to the different data types from the same original dataset, p.e. “Paris 2008”)
- Type of data. This identifies the Data Model of the dataset:
- Date of the data. When the data was captured
- Storage year. Signals under which **year** the data is stored in the system, it will normally match the year that the data was captured.
- Projection for which the data is available
- Original Data Resolution
- LayerName. Name of the layer group to which this zone belongs, for example for ortho images: satellite, countryortho or pictometry, for 3D layers: LOD1, LOD2,…
- Overlay: true (t) or false (f). Signals if this dataset is overlay data to be displayed on top of other layers
- Maximum and Minimum Zoom levels covered by this dataset
- Status. Whether the data has been checked and approved for general delivery or if it’s still pending approval.
- Extent. Geographical coverage of the layer in WKT

16.4.3 Examples

16.4.3.1 GetZones with plain text output

In this example the service is called to search Loaded zones of type north or south in a particular area defined in the EPSG:3785 reference system. Results are returned as plain text.

```
http://www.blomurbex.com/v02/GetZones?usertoken=********&format=txt&CRS=EPSG:3785&POINTS=0,3000000,1,-3000000,2,-4000000&types=north,south
```

The result is a text string with all the data of the zones available. There are two records separated by the "#" character. The order of the diffent parameters is the one explained above.

```
2, Paris 2008, north, 2009-01-01, 2007, EPSG:3785, 1.0, countryortho, f, 20, 7, 0, -1.1131949E7, 2273031.0, -1.1131949E7, 1.5538711E7, 1.1131949E7, 1.5538711E7, 1.1131949E7, 2273031.0, -1.1131949E7, 2273031.0#3, Paris 2008, south, 2009-01-01, 2007, EPSG:3785, 1.0, countryortho, f, 20, 7, 1, -1.1131949E7, 2273031.0, -1.1131949E7, 1.5538711E7, 1.1131949E7, 1.5538711E7, 1.1131949E7, 2273031.0, -1.1131949E7, 2273031.0
```

16.4.3.2 *GetZones with Xml output*

In this example the service is called to search Loaded zones of type north or south in a particular area defined in the EPSG:3785 reference system and with status=1 (i.e. pending approval). Results are returned as an XML file.

```
http://www.blomurbex.com/v02/GetZones?usertoken=********&format=xml&CRS=EPSG:3785&POINTS=0,3000000,1,-3000000,2,-4000000&types=north,south&status=0
```

The result is an XML file with the data of the loaded zones.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<result>
      <Zone>
            <Idlayer>2</Idlayer>
            <Name> Paris 2008</Name>
            <Type>north</Type>
            <Date>2009-01-01</Date>
            <Storageyear>2007</Storageyear>
            <CRS>EPSG:3785</CRS>
            <Resolution>1.0</Resolution>
            <Layername>countryortho</Layername>
            <Overlay>f</Overlay>
            <Maxzoomlevel>20</Maxzoomlevel>
            <Minzoomlevel>7</Minzoomlevel>
            <Status>0</Status>
            <Extent>-1.1131949E7, 2273031.0, -1.1131949E7, 1.5538711E7, 1.1131949E7, 1.5538711E7, 1.1131949E7, 2273031.0, -1.1131949E7, 2273031.0</Extent>
      </Zone>
</result>
```

16.5 GetUserExtents Service

16.5.1 Description

This service returns all available extents for a given usertoken.

16.5.2 Parameters

The parameters for this service are:

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.
- **user[mandatory]**: Usertoken of which we want to query for the available Extents. Can be the same as usertoken.
- **format [mandatory]:** Response format. Possible values are:
  - **txt:** The result is a text string describing each of the available extents that comply with the requests. Records are separated by "#". Fields in each record are separated by ":".
  - **Xml:** The result is an XML file describing each of the extents that comply with the requests.
  - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.
- **crs [mandatory]:** Coordinate reference system used, It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.
- **types [mandatory]**: This parameter indicates which type of view is requested. Possible values are 'ortho' for ortho views or 'oblique' for oblique views.

16.5.3 Examples

16.5.3.1 GetExtents with plain text output

In this example the service is called to search all ortho extents for a user.

```
http://www.blomurbex.com/v02/GetUserExtents?usertoken=********&format=txt&types=ortho&user=********&srs=EPSG:3785
```

The result is a text string with all the data of the extents available. There are two records separated by the "#" character. Every record is a description of an extent, with its parameters separated by colons (:). The parameters for every extent are: Name,

minimum zoom, maximum zoom, X and Y coordinates for all edge point defining the extent and the WKT format for the extent, with coordinates in geographic projection.

```
world top: 13: 1: 180.0: 89.0: 180.0: -89.0: -180.0: -89.0: -180.0:
89.0: 180.0: 89.0: POLYGON((20037508.3427892
30240971.9583852,20037508.3427892 -30240971.9583853,-20037508.3427892
-30240971.9583853,-20037508.3427892 30240971.9583852,20037508.3427892
30240971.9583852))#Demo Europe: 20: 0: -157.5: 14.604846954345703: -
100.0: 89.0: 100.0: 89.0: 160.3125: 22.593725204467773: -157.5:
14.604846954345703: POLYGON((-17532819.7999406 1643701.8562442,-
11131949.0793274 30240971.9583852,11131949.0793274
30240971.9583852,17845905.8677967 2582960.05981233,-17532819.7999406
1643701.8562442
```

16.5.3.2 GetExtent with Xml output

Same example as above, just changing the output to XML to see how the result is given.

http://www.blomurbex.com/v02/GetUserExtents?**usertoken**=********&**format**=xml&**types**=ortho&**user**=********&**srs**=EPSG:3785

The result is an XML file with the data of the extents

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
- <result>
- <Region>
    <Name>world top</Name>
      <Maxzoomlevel>13</Maxzoomlevel>
      <Minzoomlevel>1</Minzoomlevel>
      <Extent>180.0, 89.0, 180.0, -89.0, -180.0, -89.0, -180.0, 89.0,
      180.0, 89.0</Extent>
    <WKT>POLYGON((20037508.3427892
      30240971.9583852,20037508.3427892 -30240971.9583853,-
      20037508.3427892 -30240971.9583853,-20037508.3427892
      30240971.9583852,20037508.3427892
      30240971.9583852))</WKT>
  </Region>
- <Region>
    <Name>Demo Europe</Name>
    <Maxzoomlevel>20</Maxzoomlevel>
    <Minzoomlevel>0</Minzoomlevel>
      <Extent>-157.5, 14.604846954345703, -100.0, 89.0, 100.0,
      89.0, 160.3125, 22.593725204467773, -157.5,
      14.604846954345703</Extent>
    <WKT>POLYGON((-17532819.7999406 1643701.8562442,-
      11131949.0793274 30240971.9583852,11131949.0793274
      30240971.9583852,17845905.8677967 2582960.05981233,-
      17532819.7999406 1643701.8562442))</WKT>
  </Region>
</result>
```

16.6 GetUserAvailableData Service

16.6.1 Description

This service returns all available datalayers in a certain geographic point at a certain zoom level for a given usertoken.

The service will check for all datalayers available for the user in the given geographic parameters, and it will intersect those layer's geometries with the extents allowed for the user, so it will only deliver layers on geographic regions allowed for the user.

Finally, results will be filtered by user date restrictions and allowed baselayers and overlays.

16.6.2 Parameters

The parameters for this service are:

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.
- **user [mandatory]**: Usertoken of which we want to query for the available datalayers. Can be the same as usertoken.
- **format [mandatory]:** Response format. Possible values are:
    - **txt:** The result is a text string describing each of the datalayers that comply with the requests. Records are separated by "#". Fields in each record are separated by ":".
    - **Xml:** The result is an XML file describing each of the datalayers that comply with the requests.
    - **Json:** The result is a JSON file describint each of the datalayers that comply with the request.
    - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.
- **crs [mandatory]:** Coordinate reference system used, It must follow OGC nomenclature. For example **EPSG:32630** for WGS84 UTM 30 North or **EPSG:3785** for Spherical Mercator projection.
- **types [mandatory]**: This parameter indicates which type of view is requested. Possible values are:
    - **'raster'** for all image layers
    - '**ortho'** for ortho views
    - **'oblique'** for oblique views.

- **'3d'** for 3D data
- **'lidar'** for lidar data.
- **'dtm'** for dtm data.
- **Leave it empty** for all types of data

- **layers [optional]:** A list of data sources names to which the layers retrieved should belong.
- **x [mandatory]:** X coordinates of the point requested, in the projection indicated by crs.
- **y [mandatory]:** Y coordinates of the point requested, in the projection indicated by crs.
- **zoom [mandatory]:** Zoom level requested.
- **yearMode [optional]:** If present and set to true (yearMode=true), the year of the data layer will be added to the response. Else the year will not be added.
- **deliverExtents [optional]:** If present and set to false, the extents will not be added to the response. Else the extents will be added.
- **includeOverlays [optional]:** if present and set to true overlays' data will be included in the response.

16.6.3 Examples

16.6.3.1 GetUserAvailableData with plain text output

In this example the service is called to search all ortho layers for a user.

```
http://www.blomurbex.com/v02/GetUserAvailableData?usertoken=********&format=txt&types=ortho&user=********&srs=EPSG:3785&x=-523121&y=4960371&zoom=17
```

The result is a text string with all the data of the zones available. There are several records separated by the "#" character. Every record is a description of a datalayer previously intersected with the user permitted extents. Datalayers text representation consists of some datalayer parameters separated by colons (:). The parameters for every datalayer are: Name, orientation, number of the dataset the datalayer belongs to and the WKT format for the datalayer's extent(the result of intersecting its original extent with all the extent the user is allowed to see), with coordinates in geographic projection.

```
generic:ortho:1000:POLYGON((-9.44824218750001 41.6729118196021,-
9.66796875 43.8345267822368,-1.71386718749999
43.5803908556078,3.69140625 42.4558876419717,4.57031250000001
39.4022443402928,-4.65820312500001 35.924644531441,-7.95410156249998
35.9957853864203,-7.20703125 41.5085772974393,-9.44824218750001
41.6729118196021))# generic:rect_west:1000:POLYGON((-9.44824218750001
41.6729118196021,-9.66796875 43.8345267822368,-1.71386718749999
```

```
43.5803908556078,3.69140625 42.4558876419717,4.57031250000001
39.4022443402928,-4.65820312500001 35.924644531441,-7.95410156249998
35.9957853864203,-7.20703125 41.5085772974393,-9.44824218750001
41.6729118196021))# generic:rect_south:1000:POLYGON((-9.44824218750001
41.6729118196021,-9.66796875 43.8345267822368,-1.71386718749999
43.5803908556078,3.69140625 42.4558876419717,4.57031250000001
39.4022443402928,-4.65820312500001 35.924644531441,-7.95410156249998
35.9957853864203,-7.20703125 41.5085772974393,-9.44824218750001
41.6729118196021))# generic:rect_north:1000:POLYGON((-9.44824218750001
41.6729118196021,-9.66796875 43.8345267822368,-1.71386718749999
43.5803908556078,3.69140625 42.4558876419717,4.57031250000001
39.4022443402928,-4.65820312500001 35.924644531441,-7.95410156249998
35.9957853864203,-7.20703125 41.5085772974393,-9.44824218750001
41.6729118196021))# generic:rect_east:1000:POLYGON((-9.44824218750001
41.6729118196021,-9.66796875 43.8345267822368,-1.71386718749999
43.5803908556078,3.69140625 42.4558876419717,4.57031250000001
39.4022443402928,-4.65820312500001 35.924644531441,-7.95410156249998
35.9957853864203,-7.20703125 41.5085772974393,-9.44824218750001
41.6729118196021))# countryortho:ortho:1001:POLYGON((-9.44824218750001
41.6729118196021,-9.66796875 43.8345267822368,-1.71386718749999
43.5803908556078,3.69140625 42.4558876419717,4.57031250000001
39.4022443402928,-4.65820312500001 35.924644531441,-7.95410156249998
35.9957853864203,-7.20703125 41.5085772974393,-9.44824218750001
41.6729118196021))
```

16.6.3.2 GetUserAvailableData with Xml output

Same example as above, just changing the output to XML to see how the result is given

```
http://www.blomurbex.com/v02/GetUserAvailableData?usertoken=********&format=xml&types=ortho&user=********&srs=EPSG:3785&x=-523121&y=4960371&zoom=17
```

The result is an XML file with the datalayers description

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
- <result>
- <Datalayer>
  <layerName>generic</layerName>
  <dataType>ortho</dataType>
  <dataSet>1000</dataSet>
  <Extent>POLYGON((-9.44824218750001 41.6729118196021,-
    9.66796875 43.8345267822368,-1.71386718749999
    43.5803908556078,3.69140625
    42.4558876419717,4.57031250000001 39.4022443402928,-
    4.65820312500001 35.924644531441,-7.95410156249998
    35.9957853864203,-7.20703125 41.5085772974393,-
    9.44824218750001 41.6729118196021))</Extent>
  </Datalayer>
- <Datalayer>
  <layerName>generic</layerName>
  <dataType>rect_west</dataType>
  <dataSet>1000</dataSet>
```

```
<Extent>POLYGON((-9.44824218750001 41.6729118196021,-
9.66796875 43.8345267822368,-1.71386718749999
43.5803908556078,3.69140625
42.4558876419717,4.57031250000001 39.4022443402928,-
4.65820312500001 35.924644531441,-7.95410156249998
35.9957853864203,-7.20703125 41.5085772974393,-
9.44824218750001 41.6729118196021))</Extent>
</Datalayer>
- <Datalayer>
<layerName>generic</layerName>
<dataType>rect_south</dataType>
<dataSet>1000</dataSet>
<Extent>POLYGON((-9.44824218750001 41.6729118196021,-
9.66796875 43.8345267822368,-1.71386718749999
43.5803908556078,3.69140625
42.4558876419717,4.57031250000001 39.4022443402928,-
4.65820312500001 35.924644531441,-7.95410156249998
35.9957853864203,-7.20703125 41.5085772974393,-
9.44824218750001 41.6729118196021))</Extent>
</Datalayer>
- <Datalayer>
<layerName>generic</layerName>
<dataType>rect_north</dataType>
<dataSet>1000</dataSet>
<Extent>POLYGON((-9.44824218750001 41.6729118196021,-
9.66796875 43.8345267822368,-1.71386718749999
43.5803908556078,3.69140625
42.4558876419717,4.57031250000001 39.4022443402928,-
4.65820312500001 35.924644531441,-7.95410156249998
35.9957853864203,-7.20703125 41.5085772974393,-
9.44824218750001 41.6729118196021))</Extent>
</Datalayer>
- <Datalayer>
<layerName>generic</layerName>
<dataType>rect_east</dataType>
<dataSet>1000</dataSet>
<Extent>POLYGON((-9.44824218750001 41.6729118196021,-
9.66796875 43.8345267822368,-1.71386718749999
43.5803908556078,3.69140625
42.4558876419717,4.57031250000001 39.4022443402928,-
4.65820312500001 35.924644531441,-7.95410156249998
35.9957853864203,-7.20703125 41.5085772974393,-
9.44824218750001 41.6729118196021))</Extent>
</Datalayer>
- <Datalayer>
<layerName>countryortho</layerName>
<dataType>ortho</dataType>
<dataSet>1001</dataSet>
<Extent>POLYGON((-9.44824218750001 41.6729118196021,-
9.66796875 43.8345267822368,-1.71386718749999
43.5803908556078,3.69140625
42.4558876419717,4.57031250000001 39.4022443402928,-
```

4.65820312500001 35.924644531441,-7.95410156249998 35.9957853864203,-7.20703125 41.5085772974393,-9.44824218750001 41.6729118196021))</Extent>
</Datalayer>
</result>

16.6.3.3 GetUserAvailableData with plain text output and year

In this example the service is called to search all ortho layers for a user.

```
http://www.blomurbex.com/v02/GetUserAvailableData?usertoken=********&format=txt&types=ortho&user=********&srs=EPSG:3785&x=-523121&y=4960371&zoom=17&yearMode=true
```

The response will contain also the year separated by a hash sign (#):

pictometry:rect_east:232:POLYGON((-4.67579531687559 40.6899996455666,-4.67581110727782 40.6890106710943,-4.67215936605774 40.6894933875079,-4.67224257547568 40.6883100523355,-4.66735741392812 40.6887791465314,-4.66742061920496 40.6880385960567,-4.66505677862728 40.6882121721641,-4.66506338600255 40.6884415902363,-4.65820037227021 40.6893142673888,-4.65824301554454 40.6879755102142,-4.65264342705494 40.6884573898021,-4.6527130275063 40.6871995509892,-4.64772948421333 40.688202277051,-4.64806203406339 40.6849037068045,-4.64568477797336 40.6854564969124,-4.64498960697358 40.6770768191362,-4.64526023528605 40.6770874892213,-4.64439766372156 40.670688216812,-4.64787495176884 40.6707040591118,-4.6478255995254 40.6697145673539,-4.64675609433485 40.6699159272626,-4.646909772248 40.6654832043002,-4.64686687109888 40.6654899044154,-4.64693828743235 40.6646607054141,-4.64703134263651 40.6619765985571,-4.64716851603316 40.6619875728588,-4.64754182021761 40.6576532218482,-4.64853933068846 40.6577650139523,-4.64745759104793 40.6539448128128,-4.64718626887093 40.6539785486603,-4.64669584285558 40.651704915602,-4.64561586323296 40.6518875751129,-4.64525451257397 40.6434399808014,-4.65027931440219 40.6439095530007,-4.65024310630966 40.6430465200551,-4.65251999664949 40.6432875393602,-4.65269139774343 40.637909549721,-4.65456064445089 40.6382387087547,-4.65452815099364 40.6359577518404,-4.6587486898611 40.6363984192667,-4.65880346151178 40.6353218221508,-4.66547504692479 40.6362981126816,-4.66546893242521 40.6364189608725,-4.66879775419012 40.6366798909032,-4.66870879439938 40.6356688813568,-4.67260700076607 40.6360393434404,-4.67256039135498 40.6353832638066,-4.67738816430874 40.6359048850632,-4.67737084072209 40.635465260036,-4.68257457313044 40.6360582784405,-4.68256761353665 40.6357129515289,-4.68637598189475 40.636147175266,-4.68640642490526 40.6351404882208,-4.69012250627996 40.6356721154836,-4.69002621642752 40.6349088798693,-4.69603670180617 40.6356027404297,-4.69604171263818 40.6353708901753,-4.70056001846562 40.6359712958103,-4.70050573320525 40.6350677036786,-4.70774123241089 40.6359638835205,-4.70774879738381 40.6362006890697,-4.71007056547493 40.6364632082686,-4.71002904204073 40.6357655801783,-4.71727182717823 40.6364993193793,-4.71740206613589 40.6396977687883,-4.71777705290562 40.639749326437,-4.71773408578012 40.6435339132862,-4.71807484854701 40.6435679488141,-4.71839549913602 40.6482699156219,-4.7139978161555 40.648799963895,-4.71440488816333 40.6536930033183,-4.71383162326076 40.6538012566258,-4.71412503554181 40.6569761722767,-4.71402728731763 40.656993312851,-4.71437207006158 40.6607579997432,-

4.71411104334712 40.6608085722452,-4.71409474492282 40.6643772606154,-
4.71399986477988 40.664393729885,-4.71406487516193 40.6656167688859,-
4.71465318381937 40.6656921811978,-4.71421871777702 40.6710931373734,-
4.71385362794985 40.6711399950317,-4.71337414718161 40.6747088965522,-
4.71163784539944 40.6748789715084,-4.71163473964005 40.6753979572834,-
4.71142149959852 40.6754311707561,-4.7115682995746 40.6764310192683,-
4.71202542668312 40.6764786822936,-4.71214934447062 40.6793744906751,-
4.71855218461981 40.6800590312548,-4.71903423485119 40.6844919310492,-
4.71739413814456 40.6846958219013,-4.71742200562142 40.6883456611584,-
4.71021532189405 40.6891103122157,-4.71009069399946 40.6881995180348,-
4.70443457008907 40.6886930208348,-4.70437644558404 40.6875551655926,-
4.69938484350401 40.6877749777008,-4.69939876405859 40.6876498905171,-
4.69907512611041 40.6876635917822,-4.69803157098543 40.6878382589391,-
4.69804518373139 40.6877071945601,-4.69745069025445 40.687732362538,-
4.69743993431653 40.6879011285053,-4.69277750409561 40.6882523663567,-
4.69276800931491 40.6886046252503,-4.68556215102867 40.6893954556246,-
4.68559242646727 40.6885890668484,-4.68165059295674 40.6890162296662,-
4.6816692523426 40.6883614316197,-4.67579531687559
40.6899996455666)):2006# pictometry:rect_north:232:POLYGON((-
4.67335771725396 40.6934877106087,-4.67359172495589 40.6910286111162,-
4.66839358170432 40.6926458233638,-4.66856611277464 40.6907070928138,-
4.65901589347845 40.6928323480966,-4.65928350570079 40.6901390877583,-
4.65292218915091 40.6912037609562,-4.65317322984182 40.6898040553774,-
4.65087863356181 40.69043255497,-4.65091332875449 40.6879864425201,-
4.64942554948901 40.6884156597905,-4.6500144800265 40.6818275724574,-
4.64965620452 40.6818482395401,-4.65015501673011 40.6786275515399,-
4.64855837258086 40.6787346329938,-4.64985972962175 40.6723346022912,-
4.65022224787485 40.6723166749858,-4.65029508833696 40.6716844231281,-
4.64935934809931 40.6718087659305,-4.64990027727152 40.6682265366038,-
4.64852737761391 40.6683073066051,-4.64941110633762 40.664298602687,-
4.64898937408593 40.6643351324793,-4.64959151881689 40.6612542555189,-
4.64878791210878 40.6613195961317,-4.65014587817344 40.6545075850187,-
4.65176496787833 40.6544837780414,-4.65245446249733 40.650700796023,-
4.65373786595941 40.6507610178955,-4.65110887632338 40.6505325096664,-
4.65166531951064 40.6479322482036,-4.65085281068242 40.647824674702,-
4.65159331201219 40.6446598712426,-4.65088906909478 40.6445940863088,-
4.65155303748483 40.6413882143579,-4.6514836704012 40.6413886021167,-
4.65157660928596 40.6404926373567,-4.65064752307524 40.6403523248571,-
4.65196460424257 40.6351584468425,-4.65567786113512 40.6355395738765,-
4.65871138396431 40.6354782520712,-4.65883383855887 40.6358744544616,-
4.66084440092835 40.6356862820976,-4.66086578616628 40.6355995042,-
4.66140211726955 40.6356340843624,-4.66229766579901 40.6355502682687,-
4.66236100465371 40.6347348099687,-4.66870538230441 40.6346204871319,-
4.66893447253679 40.6353400801481,-4.67073487790381 40.6353174941963,-
4.67078795226853 40.6350448248121,-4.67722618860637 40.6352748780964,-
4.67726366906813 40.635457202646,-4.68291343261668 40.6355142943936,-
4.68291699134128 40.6355264418283,-4.68508612261473 40.6355375159864,-
4.68509297414465 40.635508746368,-4.68549145480663 40.6355395853461,-
4.68644432250844 40.6355444500621,-4.68646614903613 40.6350876552388,-
4.69283205457502 40.6348879309426,-4.6929475672436 40.6352064079151,-
4.69882887390289 40.6345995963531,-4.69939116856901 40.6356873437035,-
4.70160101994885 40.6353093004959,-4.70158949264565 40.6350486056742,-
4.70446921556621 40.6348160349384,-4.70453159990631 40.634530702301,-
4.70642386734025 40.6346581743339,-4.70800169253235 40.6345307468085,-
4.70811403971831 40.6347720322935,-4.71088580209792 40.6349587512419,-
4.71092904803986 40.6352007864763,-4.7169110715953 40.6350156316425,-
4.71719856593768 40.6358080322787,-4.71818064432012 40.6358424572688,-

4.72325379412108 40.6351833146313,-4.72670529990928 40.6411781913553,-
4.72275066693761 40.6415113178724,-4.72290536207138 40.6421112304521,-
4.72175231320397 40.6420222819863,-4.72268824460169 40.6458042859326,-
4.72088619264416 40.6456529194448,-4.72158614105211 40.6497855804234,-
4.72099133891576 40.6497149849289,-4.7215536981672 40.6530307325349,-
4.72074424244004 40.6529556822028,-4.72122155613687 40.6561246143574,-
4.72173788002985 40.6588272182939,-4.72022710530923 40.658721721418,-
4.72084412810058 40.6609405728185,-4.71887999189766 40.6613212447093,-
4.71968326367066 40.6649893870119,-4.71936471295641 40.6650432298965,-
4.72029677094489 40.6686755909365,-4.71924190076444 40.6688442858299,-
4.72009708724385 40.6728527503124,-4.71960037086791 40.6729108933943,-
4.72064942215721 40.677418665333,-4.72022517640011 40.6774665119858,-
4.72062060514453 40.6789613698512,-4.7194569736777 40.6791086940756,-
4.72046924999851 40.682901622463,-4.71917525645535 40.6831286881569,-
4.7197478077976 40.6857766739037,-4.71844094052444 40.6859597004975,-
4.71960669457915 40.6905252608859,-4.70925761522122 40.6931996686593,-
4.70960501146055 40.6913795910361,-4.70638395831901 40.6924676436986,-
4.70649356173922 40.6908059116958,-4.70011736948281 40.6929221499292,-
4.70040204088348 40.6910092877029,-4.69569135736786 40.6923416597189,-
4.69580438928556 40.6915298408922,-4.69093846729107 40.6926953054288,-
4.69108049823425 40.6917235032841,-4.68531201435471 40.6927348185581,-
4.68559411384834 40.6913240419967,-4.68222404499938 40.6920749621907,-
4.68232955466607 40.6911848552087,-4.67335771725396
40.6934877106087)):2006# pictometry:rect_south:232:POLYGON((-
4.65697715834193 40.6889713727925,-4.65687027502525 40.6884544552447,-
4.65306841466636 40.688495356806,-4.65244858197266 40.6823560794312,-
4.65262282763118 40.6823402293099,-4.65228948700479 40.6797624856587,-
4.65271252274714 40.6797178845811,-4.65234804083617 40.6767594458927,-
4.65226099303947 40.6767729764832,-4.65223546820238 40.6767322693672,-
4.64804547639527 40.6773750664954,-4.64531466449182 40.672825373957,-
4.64858976488928 40.6724519033697,-4.64851466513193 40.6722236080737,-
4.64814337292952 40.6722355748803,-4.64586548732112 40.6667591617429,-
4.64697688081729 40.6668083567984,-4.64519433401512 40.6630669889866,-
4.64752939989231 40.6631889425843,-4.64631100021046 40.6599962591157,-
4.64750319358115 40.6601116651727,-4.64727206146743 40.6594557352678,-
4.64977404246501 40.6596290686755,-4.64880691658954 40.656134422524,-
4.64954696756711 40.656139751718,-4.64841234410792 40.6522242810897,-
4.64673691593466 40.6482501421404,-4.64875163911237 40.6482131690337,-
4.64720764254612 40.6432676337452,-4.6492860089491 40.6433610273422,-
4.64853732599667 40.6400707999588,-4.64914370695101 40.6401081679248,-
4.64837142216298 40.6370141215811,-4.65281568804504 40.6371637961378,-
4.65254337264308 40.6361744397194,-4.65797179940229 40.6363743461134,-
4.66251400212761 40.6363858369262,-4.66248714189811 40.6362667735039,-
4.66371312952766 40.6363888704651,-4.66516312923649 40.6363925386578,-
4.66515585414423 40.6365325524136,-4.66838409010181 40.6368540546791,-
4.67105133124886 40.6370364525502,-4.6705655398764 40.6355253628454,-
4.68001800836745 40.6360728456773,-4.67991807032827 40.6370027951289,-
4.680806986954 40.6370485894719,-4.68058104260854 40.6363733415228,-
4.68966812931334 40.6364219259874,-4.68962622485788 40.636904795525,-
4.69039025991319 40.6369526781161,-4.69026107467903 40.6365589117567,-
4.69934579852228 40.6365871957866,-4.69933559390388 40.6367036751414,-
4.70148889786249 40.6367358389058,-4.7051780298892 40.6367310409927,-
4.70513789130954 40.6365692435516,-4.70706734163465 40.6367285838419,-
4.70926206761614 40.6367257294836,-4.7092426290302 40.636908226158,-
4.71035746613407 40.637000293036,-4.71020530568077 40.6363469636675,-
4.71925049925802 40.6371573883557,-4.71826531242102 40.6426967776887,-
4.71815534503267 40.6426919314275,-4.71799470316655 40.643874965451,-

4.71966974769754 40.6440070045908,-4.7188430304293 40.6483456909443,-
4.71896173788614 40.6483502121944,-4.7186084612724 40.6508842568714,-
4.71911341301859 40.6509061931036,-4.71845002244208 40.6556032934941,-
4.71862425776362 40.6556205847054,-4.7182827716096 40.6581852333603,-
4.71949727917843 40.6583548885561,-4.71848844275789 40.6639866294253,-
4.71801869905503 40.6639599100724,-4.71755078026171 40.6670032473497,-
4.72259981318103 40.6665622657337,-4.72200709511138 40.6693561970835,-
4.72326910315958 40.6697469638038,-4.72281884073889 40.6705968231457,-
4.72439225941433 40.6706971511376,-4.72284164141731 40.6748635036518,-
4.72332326925838 40.6748722120209,-4.72158877108805 40.6810783921308,-
4.72287093399101 40.6810674331967,-4.72085079820951 40.6886418824958,-
4.71673285300343 40.6886035603073,-4.71671440356484 40.6886578599402,-
4.71492900477727 40.6885867734371,-4.71425560979172 40.6885805067268,-
4.71425161181847 40.6885598027138,-4.7100250598275 40.6883915205334,-
4.70999675559379 40.6881439236712,-4.70678009105613 40.6880409110312,-
4.7067634180035 40.6881040218301,-4.70198044455295 40.6880409972382,-
4.70196890546984 40.6880859564004,-4.69492816031406 40.6881439344648,-
4.69488461330157 40.687898597688,-4.69261273352003 40.6878690671049,-
4.69248943978516 40.6883294866596,-4.68561393583339 40.688619569428,-
4.68557896345097 40.688401043779,-4.68133675236216 40.6884022568539,-
4.68130299264039 40.6880890975312,-4.67801941471199 40.6881775873062,-
4.67798027224077 40.6883285493584,-4.67119840781801 40.6883528682654,-
4.67113884490794 40.6880006356977,-4.66908468530046 40.6878999573324,-
4.66904890145657 40.6880229157549,-4.66369336372782 40.6882015072495,-
4.66360728122651 40.6885873303053,-4.65697715834193
40.6889713727925)):2006# pictometry:rect_west:232:POLYGON((-
4.67458532954229 40.6912054618086,-4.66558908271489 40.689420550686,-
4.66554424568356 40.6903179087664,-4.65988815659536 40.6894198258605,-
4.65987564521463 40.6899744313118,-4.65027683135593 40.6885164277113,-
4.65032774025259 40.6851545390579,-4.64725073254556 40.6847633955047,-
4.64729281193433 40.6806365465605,-4.64618593988778 40.6806204197753,-
4.64528865637717 40.6766803400181,-4.64656001733955 40.6764203637617,-
4.64613906664089 40.6764206767676,-4.64541484890223 40.6724614287374,-
4.64626839989211 40.6722706191602,-4.64566024340597 40.6692623442977,-
4.64660549889052 40.6690669804498,-4.6465632182343 40.66887621529,-
4.64742389616385 40.6686773723763,-4.64707988639491 40.6663133033542,-
4.64668949519057 40.6662823722016,-4.64640613438792 40.6621122063162,-
4.64739846390994 40.6619692426124,-4.64710938472802 40.6585585205412,-
4.64735967952784 40.6585220502899,-4.64727633395606 40.6556225819013,-
4.64727286144551 40.6556222248181,-4.64727566988051 40.6555994797019,-
4.6472510493644 40.6547429685941,-4.64738362156535 40.6547251943224,-
4.64782019797992 40.651189423882,-4.6479991643023 40.6511654497682,-
4.64781048281747 40.6470600313723,-4.6483941253476 40.6469564824612,-
4.64816909690576 40.6441831561752,-4.65200422781176 40.6436386696037,-
4.65205066938235 40.6428427122593,-4.65334802867382 40.6426378488436,-
4.65331512164658 40.6389175234811,-4.65730719833589 40.6382230631289,-
4.65784146547477 40.6353777466079,-4.6700557716958 40.6339651442715,-
4.669782528909 40.6348191729459,-4.67329534908357 40.6343123186642,-
4.67328320336199 40.6351619849902,-4.67805417545618 40.6346015487364,-
4.67794673698979 40.6352750851762,-4.68746033777206 40.6338686721308,-
4.68733720477031 40.6350150066562,-4.69340908578225 40.6341291451053,-
4.69339463724708 40.6350512356918,-4.69854739901413 40.6344085156304,-
4.69848376590059 40.6350209320969,-4.7038530705835 40.6342532888018,-
4.70387860471893 40.6347106953495,-4.70932148809494 40.6339285918259,-
4.70932173580346 40.635045902296,-4.71184227472975 40.634774082467,-
4.71173998466639 40.6355911652214,-4.71754452626115 40.6350197115014,-
4.71744936581647 40.6356729573475,-4.72658751465117 40.6345213248905,-

4.72585690484589 40.6433742095475,-4.72555550703412 40.6433318663675,-
4.7244352714452 40.6505865772143,-4.72086509322343 40.6500523438278,-
4.72073385546757 40.6508869233233,-4.71784998428971 40.6504314172362,-
4.71800355681632 40.6531220678068,-4.71785110997725 40.6531085296624,-
4.71806703476028 40.656051687257,-4.71677885673799 40.6559882034444,-
4.71706892593261 40.6597341454468,-4.71764675956439 40.6636198107226,-
4.71624524292139 40.6635245298788,-4.71641687845544 40.6643783629009,-
4.71656669556394 40.6643552124318,-4.71718548087377 40.6679203947846,-
4.71754208024167 40.6678685315239,-4.71793026343358 40.6744528610816,-
4.71617960699909 40.6743169342984,-4.71621055933587 40.6745393074166,-
4.71506136203358 40.6744660153655,-4.7157022186974 40.6779372210645,-
4.71672140914793 40.677806145969,-4.7166080232962 40.6782102013848,-
4.72051250564821 40.6776344635616,-4.72044537969861 40.6779790389203,-
4.72617596858789 40.6770414653541,-4.72514755328502 40.6856856573792,-
4.72227460911508 40.685390836192,-4.72169820181195 40.6897035011639,-
4.71748729977396 40.6891799195966,-4.71712563448531 40.6909142338015,-
4.71261022088726 40.6900537669549,-4.71259296148482 40.6901679658721,-
4.70877898598406 40.689623657202,-4.70877103772743 40.6896726609122,-
4.70443473176566 40.6891876447499,-4.70402050700059 40.6908657360409,-
4.70010587741441 40.6901464789521,-4.70007105317714 40.6906930962106,-
4.69260947001567 40.6894932972546,-4.6926029240455 40.6896479904627,-
4.6892104107236 40.6891811341882,-4.68920440989168 40.6904039739769,-
4.68399880755591 40.6896098408872,-4.68399734481354 40.6897823774788,-
4.67831509821051 40.689006651188,-4.67828244635276 40.6902578443927,-
4.67454327826081 40.6896256102019,-4.67458532954229
40.6912054618086),(-4.65070161048869 40.6766709983507,-
4.65163475489319 40.6764598964022,-4.65108903168222 40.6764629889399,-
4.65107946638264 40.67641700324,-4.65063520713068 40.6764173335774,-
4.65070161048869 40.6766709983507)):2006#
pictometry:ortho:232:POLYGON((-4.70160620320633 40.6887267644342,-
4.70160767462052 40.6884622860588,-4.70157657541499 40.6884617884654,-
4.7015713202117 40.6887523655435,-4.696998329971 40.6886393758664,-
4.69700137318423 40.6884298416927,-4.69646487885323 40.6883896349081,-
4.6964540378423 40.6885070755819,-4.694200656252 40.6883730082755,-
4.69419122634909 40.6885732919605,-4.69206490179737 40.6884968001667,-
4.69206497598043 40.688528148645,-4.68759896232474 40.688478023541,-
4.68759716672534 40.6883213186858,-4.68701754595538 40.6883244366629,-
4.6870158383304 40.6883840120512,-4.68453124928683 40.688318728344,-
4.68452858748503 40.6883457550719,-4.68385646906047 40.6883009981866,-
4.68273649345977 40.6882715703183,-4.6827371736252 40.6882264634388,-
4.68219240480812 40.6881901868623,-4.68216797169721 40.6884083745694,-
4.67980950998453 40.6882261036053,-4.67980552620544 40.6883313203154,-
4.67731299466295 40.6882474130128,-4.67731290380531 40.6882836465896,-
4.6743742787748 40.6882525503431,-4.67437352300664 40.6882658229073,-
4.67380151331849 40.6882464893943,-4.67321181232228 40.6882402492353,-
4.67321169206381 40.6882265538632,-4.67212227075456 40.6881897322123,-
4.67210065128568 40.6886480605107,-4.66792591685846 40.6885025428035,-
4.66793072238297 40.6884095912592,-4.66592779523305 40.6883380799756,-
4.66592940948321 40.6882947550781,-4.66361610806541 40.6882303967087,-
4.66362109568266 40.6879725725771,-4.66282250591812 40.6879661090874,-
4.66280686106009 40.6882309197424,-4.66025006109181 40.6881306919352,-
4.66024783056965 40.6881586821299,-4.65790093550221 40.6880451863505,-
4.65787467695265 40.6883601347101,-4.65376032328849 40.6881338221282,-
4.65394917023359 40.6856041874287,-4.65287725915778 40.6858764041005,-
4.65254775077409 40.6849502872262,-4.65249721380694 40.6849535130964,-
4.6524830010915 40.6847683016481,-4.65130044566973 40.6814446091862,-
4.6519237422019 40.6813258791203,-4.65127726204014 40.6768544818004,-

4.65196088187741 40.6768161443772,-4.65117269355569 40.67349326824,-
4.65153425445238 40.6734534611722,-4.65111406705301 40.6699817757736,-
4.65189711611194 40.6699435051966,-4.65189271449909 40.6699057185452,-
4.65240587740992 40.6698775135659,-4.65234387299232 40.667315250242,-
4.65228391264994 40.667319816139,-4.65197652377736 40.6626891156573,-
4.6526454743483 40.6626698197005,-4.65191134980688 40.6589580561357,-
4.65229376651584 40.6589190812632,-4.65211306572285 40.6551299811406,-
4.65236435670402 40.6551268643992,-4.65235971280116 40.6529509307443,-
4.65225736552819 40.6529540279436,-4.65223577301393 40.6479506176966,-
4.65229978494622 40.6479507982338,-4.65190768017241 40.6444122642866,-
4.65430998248149 40.6442706897376,-4.6543053974741 40.6442210742269,-
4.65468008575032 40.644203866557,-4.65465653796616 40.6438490611504,-
4.65486769928321 40.6438357501304,-4.65474243982831 40.641150860626,-
4.65444836531175 40.6411344009312,-4.6547143417298 40.6362292222887,-
4.65705708216194 40.6362548890392,-4.65705759283628 40.6361701313608,-
4.66152002522478 40.6360895902495,-4.66150995126631 40.6364523314349,-
4.66393510924866 40.6363892553331,-4.66393501647576 40.6364133232982,-
4.66456433041103 40.6363970859164,-4.66456861946974 40.6362783240739,-
4.66904701616411 40.6362481350124,-4.66903681162423 40.6364405968791,-
4.66904909578876 40.6364401380545,-4.66904706156797 40.6362524900024,-
4.6735481805083 40.6361129317357,-4.67354689798159 40.6363779862929,-
4.67367392298109 40.6363765010538,-4.67367938366011 40.6360340368521,-
4.67824899736167 40.6359836299546,-4.67823734772371 40.6362922624037,-
4.67861911389127 40.6362825220705,-4.67861670365517 40.6362237546398,-
4.68319203880503 40.6359889238565,-4.68319214189093 40.6359926568023,-
4.68565467360911 40.6358639838717,-4.68565732206141 40.6359709654379,-
4.68576324087842 40.6359659395909,-4.6857648037376 40.6359295382172,-
4.68648975177182 40.6359314666555,-4.68818857946963 40.6358508572963,-
4.68818984658884 40.6359359890872,-4.69041555350938 40.6359419097026,-
4.6904095850314 40.6360371032471,-4.69053633100592 40.6360323071113,-
4.69053782953744 40.636004472564,-4.69112567529777 40.6360100060068,-
4.69255888072891 40.6359557727436,-4.69256177146971 40.6359109462922,-
4.69503710527564 40.635977143764,-4.69504916182074 40.6358001505201,-
4.69970482529672 40.6359250406735,-4.69965350676987 40.6364354983153,-
4.6998551206154 40.63643350842,-4.69985170734742 40.6361747511012,-
4.70446221193806 40.6360425658726,-4.70446141070778 40.6362399015169,-
4.70461267526084 40.6362396362069,-4.70461992473359 40.6356486914439,-
4.70935596095297 40.6355982545999,-4.70933877094666 40.6359398627647,-
4.71172350435137 40.6359326014988,-4.71170883175232 40.636219148567,-
4.71407502474515 40.6361712275871,-4.7140731448975 40.6363044739066,-
4.71416363374795 40.6363037539226,-4.71416752081755 40.6359812635089,-
4.71880050220453 40.6359255989162,-4.7186576121443 40.6398402047572,-
4.71890799500346 40.6398452352082,-4.71859139792388 40.6434282374679,-
4.7189119239854 40.6434278070732,-4.71864161870137 40.6486170852599,-
4.71688718612438 40.6485337634228,-4.71689482932281 40.6486650671231,-
4.71635416156487 40.6486682012766,-4.71638371174291 40.6495042641304,-
4.71595114149848 40.6495173328863,-4.7160580175909 40.6528819369246,-
4.71602240041751 40.6528832457012,-4.71627643448374 40.6562422835294,-
4.71610260075864 40.656252818311,-4.71614255438029 40.6598321824467,-
4.71613747735056 40.6598322963464,-4.71627022520603 40.6639596922878,-
4.7160756354054 40.6639655250178,-4.71628884321361 40.6674123739548,-
4.71595216895749 40.6674289524511,-4.71625286536979 40.670591219827,-
4.71614473558538 40.6705991754702,-4.71633843286155 40.6744219148859,-
4.71519434569432 40.6744721369502,-4.71542343954666 40.6783504582376,-
4.71519364459346 40.6783656402185,-4.71526834854351 40.6796135303811,-
4.71641506201053 40.6795958403848,-4.71640777785174 40.679823739221,-
4.7185583493894 40.6797585655985,-4.71856604110093 40.6847535522524,-

```
4.7182580663101 40.6847492778864,-4.71778377091319 40.6882777743066,-
4.71543962739736 40.6880826076166,-4.71539839361768 40.6883781793031,-
4.71315179143872 40.6881778645467,-4.71312960676871 40.6883646189602,-
4.71081275643144 40.6881862509078,-4.71080088533241 40.6883746714574,-
4.70851623134179 40.6882735939383,-4.70852028358709 40.688567025972,-
4.70614141568664 40.6885606247355,-4.70614064673907 40.6887952103438,-
4.70160620320633 40.6887267644342)):2006#
generic:ortho:1000:POLYGON((-9.44824218750001 41.6729118196021,-
9.66796875 43.8345267822368,-1.71386718749999
43.5803908556078,3.69140625 42.4558876419717,4.57031250000001
39.4022443402928,-4.65820312500001 35.924644531441,-7.95410156249998
35.9957853864203,-7.20703125 41.5085772974393,-9.44824218750001
41.6729118196021)):2009# generic:rect_east:1000:POLYGON((-
9.44824218750001 41.6729118196021,-9.66796875 43.8345267822368,-
1.71386718749999 43.5803908556078,3.69140625
42.4558876419717,4.57031250000001 39.4022443402928,-4.65820312500001
35.924644531441,-7.95410156249998 35.9957853864203,-7.20703125
41.5085772974393,-9.44824218750001 41.6729118196021)):2009#
generic:rect_north:1000:POLYGON((-9.44824218750001 41.6729118196021,-
9.66796875 43.8345267822368,-1.71386718749999
43.5803908556078,3.69140625 42.4558876419717,4.57031250000001
39.4022443402928,-4.65820312500001 35.924644531441,-7.95410156249998
35.9957853864203,-7.20703125 41.5085772974393,-9.44824218750001
41.6729118196021)):2009# generic:rect_south:1000:POLYGON((-
9.44824218750001 41.6729118196021,-9.66796875 43.8345267822368,-
1.71386718749999 43.5803908556078,3.69140625
42.4558876419717,4.57031250000001 39.4022443402928,-4.65820312500001
35.924644531441,-7.95410156249998 35.9957853864203,-7.20703125
41.5085772974393,-9.44824218750001 41.6729118196021)):2009#
generic:rect_west:1000:POLYGON((-9.44824218750001 41.6729118196021,-
9.66796875 43.8345267822368,-1.71386718749999
43.5803908556078,3.69140625 42.4558876419717,4.57031250000001
39.4022443402928,-4.65820312500001 35.924644531441,-7.95410156249998
35.9957853864203,-7.20703125 41.5085772974393,-9.44824218750001
41.6729118196021)):2009# countryortho:ortho:1001:POLYGON((-
9.44824218750001 41.6729118196021,-9.66796875 43.8345267822368,-
1.71386718749999 43.5803908556078,3.69140625
42.4558876419717,4.57031250000001 39.4022443402928,-4.65820312500001
35.924644531441,-7.95410156249998 35.9957853864203,-7.20703125
41.5085772974393,-9.44824218750001 41.6729118196021)):2007
```

16.7 GetUserAvailableOverlays Service

16.7.1 Description

This service returns all available overlays for a certain user.

16.7.2 Parameters

The parameters for this service are:

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.
- **format [mandatory]:** Response format. Possible values are:
  - **txt:** The result is a text string describing each of the datalayers that comply with the requests. Records are separated by "#". Fields in each record are separated by ":".
  - **Xml:** The result is an XML file describing each of the datalayers that comply with the requests.
  - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.

16.7.3 Examples

This example returns javascript data: A function returning an array with the name of all available overlays.

```
http://www.blomurbex.com/v02/GetUserAvailableOverlays?usertoken=********&format=js&function=myfunction
```

The response returned will be:

```
function myfunction(){
 return ['generic', 'teleatlas', 'empty'];
}
```

Same example as above, just changing the output to XML to see how the result is given

```
http://www.blomurbex.com/v02/GetUserAvailableOverlays?usertoken=********&format=xml
```

The result is an XML file with the overlay names description

<result>

```
        <Overlay>
            generic
        </Overlay>
        <Overlay>
            teleatlas
        </Overlay>
        <Overlay>
            empty
        </Overlay>

</result>
```

16.8 GetUserAvailableBaselayers Service

16.8.1 Description

This service returns a list of available baselayers for one user. If, some baselayer (one, or several baselayer separated by commas) are specified in request, the service will interpret that the requester wants to know if he has permission to see that particular baselayers specified and it will send the names of the ones the user has permission to see.

16.8.2 Parameters

The parameters for this service are:

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

- **format [mandatory]:** Response format. Possible values are:
  - **txt:** The result is a text string describing each of the baselayers that comply with the requests. Records are separated by "#". Fields in each record are separated by ":".
  - **Xml:** The result is an XML file describing each of the baselayers that comply with the requests.
  - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return

16.8.3 Examples

```
http://www.blomurbex.com/v02/GetUserAvailableBaselayers?usertoken=********&format=js&function=myfunction
```

The response returned will be:

```
function myfunction(){
 return ['satellite', 'countryortho', 'pictometry', 'lidar', 'demos',
'oym', 'ubiestBS'];
}
```

```
Same Example as above but adding baselayers to check the permission on
ubiestBS and oym layer
```

```
http://www.blomurbex.com/v02/GetUserAvailableBaselayers?usertoken=********&baselayers=ubiestBS,oym&format=js&function=myfunction
```

The response returned will be:

```
function myfunction(){
```

```
 return ['ubiestBS', 'oym'];
```

```
}
```

16.9 GetUserAvailableObliquelayers Service

16.9.1 Description

This service returns a list of available Oblique layers for one user. If, some oblique layer (one, or several oblique layer separated by commas) are specified in request, the service will interpret that the requester wants to know if he has permission to see that particular oblique layers specified and it will send the names of the ones the user has permission to see.

16.9.2 Parameters

The parameters for this service are:

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

- **format [mandatory]:** Response format. Possible values are:
  - **txt:** The result is a text string describing each of the oblique layers that comply with the requests. Records are separated by "#". Fields in each record are separated by ":".
  - **Xml:** The result is an XML file describing each of the oblique layers that comply with the requests.
  - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return
  - **json:** The result is Json file.

16.9.3 Examples

```
http://www.blomurbex.com/v02/GetUserAvailableObliquelayers?usertoken=********&
format=js&function=myfunction
```

The response returned will be:

```
function myfunction(){
 return ['oblique', 'obliquehr'];
}
```

```
Same Example as above but adding oblique layers to check the
permission on obliquehr layer
```

```
http://www.blomurbex.com/v02/GetUserAvailableObliquelayers?usertoken=********&
baselayers=obliquehr&format=js&function=myfunction
```

The response returned will be:

```
function myfunction(){ return ['obliquehr'];}
```

16.10 GetUserAvailableStreetlayers Service

16.10.1 Description

This service returns a list of available Street layers for one user. If, some street layer (one, or several street layer separated by commas) are specified in request, the service will interpret that the requester wants to know if he has permission to see that particular street layers specified and it will send the names of the ones the user has permission to see.

16.10.2 Parameters

The parameters for this service are:

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

- **format [mandatory]:** Response format. Possible values are:
  - **txt:** The result is a text string describing each of the oblique layers that comply with the requests. Records are separated by "#". Fields in each record are separated by ":".
  - **Xml:** The result is an XML file describing each of the oblique layers that comply with the requests.
  - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return
  - **json:** The result is Json file.

16.10.3 Examples

```
http://www.blomurbex.com/v02/GetUserAvailableStreetLayers?usertoken=********&format=js&function=myfunction
```

The response returned will be:

```
function myfunction(){
 return ['streetst'];
}
```

Same Example as above but adding street layers to check the permission on **streetst** layer

```
http://www.blomurbex.com/v02/GetUserAvailableStreetlayers?usertoken=********&layers=streetst&format=js&function=myfunction
```

The response returned will be:

```
function myfunction(){ return ['streetst'];}
```

16.11 GetUserAvailableLidarlayers Service

16.11.1 Description

This service returns a list of available Lidar layers for one user. If, some Lidar layer (one, or several Lidar layer separated by commas) are specified in request, the service will interpret that the requester wants to know if he has permission to see that particular Lidar layers specified and it will send the names of the ones the user has permission to see.

16.11.2 Parameters

The parameters for this service are:

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

- **format [mandatory]:** Response format. Possible values are:
  - **txt:** The result is a text string describing each of the Lidar layers that comply with the requests. Records are separated by "#". Fields in each record are separated by ":".
  - **Xml:** The result is an XML file describing each of the Lidar layers that comply with the requests.
  - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return
  - **json:** The result is Json file.

16.11.3 Examples

```
http://www.blomurbex.com/v02/GetUserAvailableLidareLayers?usertoken=********&format=json
```

The response returned will be:

```
[
  "blom_lidar_10",
  "terratec_lidar_st",
  "blom_lidar_20",
  "blom_lidar_st"
]
```

16.12 GetUserAvailableDtm3DLayers Service

16.12.1 Description

This service returns a list of available Dtm3D layers for one user. If, some Dtm3D layer (one, or several Dtm3D layer separated by commas) are specified in request, the service will interpret that the requester wants to know if he has permission to see that particular Dtm3D layers specified and it will send the names of the ones the user has permission to see.

16.12.2 Parameters

The parameters for this service are:

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

- **format [mandatory]:** Response format. Possible values are:
  - **txt:** The result is a text string describing each of the Dtm3D layers that comply with the requests. Records are separated by "#". Fields in each record are separated by ":".
  - **Xml:** The result is an XML file describing each of the Dtm3D layers that comply with the requests.
  - **json:** The result is a Json function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return
  - **json:** The result is Json file.

16.12.3 Examples

```
http://www.blomurbex.com/v02/GetUserAvailableDtm3DLayers?usertoken=********&format=json
```

The response returned will be:

```
[
  "srtm15",
  "norway10m"
]
```

16.13 GetUserAvailable3DMeshLayers Service

16.13.1 Description

This service returns a list of available 3D Mesh layers for one user. If, some 3D Mesh layer (one, or several 3D Mesh layer separated by commas) are specified in request, the service will interpret that the requester wants to know if he has permission to see that particular 3D Mesh layers specified and it will send the names of the ones the user has permission to see.

16.13.2 Parameters

The parameters for this service are:

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

- **format [mandatory]:** Response format. Possible values are:
  - **txt:** The result is a text string describing each of the 3D Mesh layers that comply with the requests. Records are separated by "#". Fields in each record are separated by ":".
  - **Xml:** The result is an XML file describing each of the 3D Mesh layers that comply with the requests.
  - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return
  - **json:** The result is Json file.

16.13.3 Examples

```
http://www.blomurbex.com/v02/GetUserAvailable3DMeshLayers?usertoken=********&format=json
```

The response returned will be:

```
[
  "blom3dmesh",
  "demo3dmesh"
]
```

16.14 GetUserStreetProjects Service

16.14.1 Description

This service returns a list of available Street projects for one user.

16.14.2 Parameters

The parameters for this service are:

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

- **format [mandatory]:** Response format. Possible values are:
  - **txt:** The result is a text string describing each of the 3D Mesh layers that comply with the requests. Records are separated by "#". Fields in each record are separated by ":".
  - **Xml:** The result is an XML file describing each of the 3D Mesh layers that comply with the requests.
  - **js:** The result is a JavaScript function that will return the value. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return
  - **json:** The result is Json file.

16.14.3 Examples

```
http://www.blomurbex.com/v02/GetUserStreetProjects?usertoken=********&format=json
```

The response returned will be:

```
[
 {
  "code": "DKSODK17",
  "longitude": "9.71006864508701",
  "latitude": "55.517907045682875"
 },
 {
  "code": "DKSYDD18",
  "longitude": "9.972014171053756",
  "latitude": "55.440459734216"
 }
]
```

16.15 GetUserAvailableProviders Service

16.15.1 Description

This service returns a list of available external providers for a user. It will deliver information about external providers and services of those external providers allowed to the user.

16.15.2 Parameters

The parameters for this service are:

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

- **format [mandatory]:** Response format. Possible values are:
  - **Xml:** The result is an XML file describing each of the services allowed to the user and the external providers he is entitled to use.
  - **js:** The result is a JavaScript function that will returning a JSON object describing all allowed services and providers of that services for the user. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return

16.15.3 Examples

```
http://www.blomurbex.com/v02/GetUserAvailableProviders?usertoken=********&format=js&function=myfunction
```

The response returned will be:

```
function myfunction () {
return {"geocoding": ['oym','ubiest','osm'],"routing":
['ubiest','oym'],"map": ['oym','ubiest','osm'],"Hosted POIs":
['geomobile']};
}
```

And, in XML format:

```
- <result>
  - <service name="geocoding">
      <provider>oym</provider>
      <provider>ubiest</provider>
      <provider>osm</provider>
    </service>
  - <service name="routing">
      <provider>ubiest</provider>
      <provider>oym</provider>
    </service>
  - <service name="map">
      <provider>oym</provider>
      <provider>ubiest</provider>
      <provider>osm</provider>
    </service>
  - <service name="Hosted POIs">
      <provider>geomobile</provider>
    </service>
  </result>
```

16.16 GetUserBlomSTREETCredentials Service

16.16.1 Description

This service returns BlomSTREETCredentials for a given usertoken, in order to connect to BlomSTREET service. Those credentials would be the api key, the userlogin and the password.

16.16.2 Parameters

The parameters for this service are:

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

- **format [mandatory]:** Response format. Possible values are:
  - **txt:** The result would be the three credentials separated by commas (",").
  - **Xml:** The result is an XML file with the three credentials.
  - **js:** The result is a JavaScript function that will return an object with the three parameters. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return

16.16.3 Examples

```
http://www.blomurbex.com/v02/GetUserBlomSTREETCredentials?usertoken=********&format=xml
```

The response returned will be:

```
<result><apikey>
CLIVi7wxWbIV2fcpang1GHw1Sngl2jYBqvkhyDw5K3oJ6r0VejopGXQR-3ANWfwN
</apikey><user>
Blom_Spain
</user><password>
sPa2ad
</password></result>
```

16.17 GetUserLayers Service

16.17.1 Description

This service returns a list of available User layers for a user. The User Layer will contain all necessary information for the access to a determinate data set according the next properties:

- Layer: Base layer, Overlay layer, Oblique Layer or Street Layer name
- Date Interval: Data available between dates.
- Extent: Extension available for a data set.

16.17.2 Parameters

The parameters for this service are:

- **usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested to Blom.

- **format [mandatory]:** Response format. Possible values are:
  - **txt:** The result is a text string describing each of the oblique layers that comply with the requests. Records are separated by "#". Fields in each record are separated by ":".
  - **Xml:** The result is an XML file describing each of the oblique layers that comply with the requests.
  - **js: Json:** Response is Json format.

16.17.3 Examples

http://www.blomurbex.com/v02/GetUserLayers?**usertoken**=********&**format**=json

The response returned will be:

```
{
 {
  "name": "World satellite",
  "description": "Satellite coverage of the whole world",
  "daterange": "[-infinity,infinity)",
  "layer": "satellite",
  "extent": "POLYGON((179.99 85,179.99 -85,-179.99 -85,-179.99
85,179.99 85))",
 },
 {
  "name": "PNOA",
  "description": "PNOA aerial image Spain",
  "daterange": "[2000-01-01,2011-01-01)",
```

```
  "layer": "countryortho",
  "extent": "MULTIPOLYGON(((-1.97753906250001
43.9928145004899,4.17480468749999 42.4234565179383,4.48242187500001
37.3002752813443,-4.08691406249999 35.1019340572461,-14.2822265625
27.5667214304097,-19.423828125 27.2546295778001,-11.162109375
44.213709909702,-1.97753906250001 43.9928145004899)))",
 }
}
```

16.18 GetAutologin Service

16.18.1 Description

This service returns the token for a user that's been granted automatic login by IP permissions when he performs a request from that address.

16.18.2 Parameters

It has no parameters, as IP information is extracted from the request itself.

16.18.3 Examples

```
http://www.blomurbex.com/v02/GetAutologin
```

The response returned will be a user token in clear text if access is granted, or the string

```
UNAUTHORIZED
```

Otherwise.

IMPORTANT: This kind of automatic authentication is integrated in the rest of the services, so when a request with no credentials is received its source IP is checked and if automatic access is granted from that address the request is completed using the credentials of the user to which it is assigned.

16.19 WFSObliqueHR service

16.19.1 Description

The WFS Oblique HR service is a web service based in the WFS standard protocol for serving vector information.

BlomURBEX™ offers the WFS service to allow users to access to vector information related with the Street Image centroids and metadata information following OGC's regulation and directives.

The service uses the http protocol with the parameters normalized by OGC, which define version, type of request, coordinate system, bounding box, etc.

The service supports several input projection systems, but vector information will provided on EPSG:4326 projection.

The WFS supported versions are 1.0.0 & 1.1.0.

16.19.2 Connection

To connect to the BlomURBEX™ WFS ObliqueHR service from a GIS client, it is necessary to setup the following URL:

```
http://www.blomurbex.com/v02/WFSProxyObliqueHR?REQUEST=GetCapabilities&SERVICE
=WFS&ServiceName=WFS&version=1.1.0&userToken=****************
```

In this request, `usertokenstring` must be replaced by the unique platform access token provided by Blom.

GetCapabilities returns information about available features, capabilities and spatial operators.

16.19.3 Parameters

Parameters defined by OGC standard are:

- **Version [mandatory]**: This parameter is the WFS version that is used for this request. Only two values are accepted in the current API version:
    - **1.0.0**
    - **1.1.0**
- **Request [mandatory]:** Possible values are:
    - GetCapabilites: used to obtain the web services descriptor xml file

- DescribeFeatureType: returns the feature descriptor xml file.
- GetFeature: used to retrieving a feature query.

- **Service [mandatory]:** The service name: WFS
- **Typename:** mandatory for GetFeature and DescribeFeatureType requests. It is the name of the feature layer provided by the service. Street images will be provided by the name of 'panorama'.
- **Usertoken [mandatory]**: User's Token for authentication purposes. This token is unique for each client and must be requested from Blom.
- **Bbox [optional]:** Optional for GetFeature request. It defines the bounding box extent of the query.
- **SrsName [optional]:** it defines the projection system of the input information (i.e. bbox).

16.19.4 Result

The result of the GetCapabilities & DescribeFeatureType will be a Xml file according the OGC standard definition.

The output of GetFeature request is a Gml file with next parameters:

- **Geometry**: location of Oblique HR image centroid (x,y,z). Provided in EPSG:4326 projection
- **Id_obliq:** Oblique image identifier.
- **Url:** This is the link url for the load of the single Oblique HR image in a web browser.
- **Shot_date:** Date of the image capture. Format: YYYY-MM-DD hh:mm:ss
- **Shot_orient:** Orientation according the cardinal directions.

16.19.5 Example

```
http://www.blomurbex.com/v02/WFSProxyObliqueHR?SERVICE=WFS&REQUEST=GetFeature&
ServiceName=WFS&version=1.1.0&userToken=484441ce101a3bbc562b7f0ca5eb65b6&SRS=u
rn:ogc:def:crs:EPSG::4326&TYPENAME=obliquehr&BBOX=61.1530305,7.1282690,61.1608
730,7.1467680,urn:x-ogc:def:crs:EPSG:6.9:4326
```

The result of the previous url:


```
▼<wfs:FeatureCollection xmlns:ms="http://mapserver.gis.umn.edu/mapserver" xmlns:gml="http://www.opengis.net/gml"
 xmlns:wfs="http://www.opengis.net/wfs" xmlns:ogc="http://www.opengis.net/ogc" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
 instance" xsi:schemaLocation="http://mapserver.gis.umn.edu/mapserver http://www.blomurbex.com/v02/WFSProxyObliqueHR?
 SERVICE=WFS&VERSION=1.1.0&REQUEST=DescribeFeatureType&TYPENAME=obliquehr&OUTPUTFORMAT=text/xml;%20subtype=gml/3.1.1
 http://www.opengis.net/wfs http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
  ▼<gml:boundedBy>
    ▼<gml:Envelope srsName="EPSG:4326">
       <gml:lowerCorner>61.154320 7.130997</gml:lowerCorner>
       <gml:upperCorner>61.159492 7.144350</gml:upperCorner>
     </gml:Envelope>
   </gml:boundedBy>
  ▼<gml:featureMember>
    ▼<ms:obliquehr gml:id="obliquehr.E+007.132647_61.156463_170425">
      ▼<gml:boundedBy>
        ▼<gml:Envelope srsName="EPSG:4326">
           <gml:lowerCorner>61.156464 7.132648</gml:lowerCorner>
           <gml:upperCorner>61.156464 7.132648</gml:upperCorner>
         </gml:Envelope>
       </gml:boundedBy>
      ▼<ms:geometry>
        ▼<gml:Point srsName="EPSG:4326">
           <gml:pos>61.156464 7.132648</gml:pos>
         </gml:Point>
       </ms:geometry>
       <ms:id_obliq>E+007.132647_61.156463_170425</ms:id_obliq>
      ▼<ms:url>
         http://172.16.1.202?usertoken=484441ce101a3bbc562b7f0ca5eb65b6&lat=7.132647&lon=61.156463&orientation=E&view=single
       </ms:url>
       <ms:shotdate>2017-04-25 12:00:00</ms:shotdate>
       <ms:shotorient>E</ms:shotorient>
     </ms:obliquehr>
   </gml:featureMember>
  ▼<gml:featureMember>
    ▼<ms:obliquehr gml:id="obliquehr.N+007.137430_61.155784_170425">
      ▼<gml:boundedBy>
        ▼<gml:Envelope srsName="EPSG:4326">
           <gml:lowerCorner>61.155785 7.137430</gml:lowerCorner>
           <gml:upperCorner>61.155785 7.137430</gml:upperCorner>
         </gml:Envelope>
       </gml:boundedBy>
      ▼<ms:geometry>
        ▼<gml:Point srsName="EPSG:4326">
           <gml:pos>61.155785 7.137430</gml:pos>
         </gml:Point>
       </ms:geometry>
       <ms:id_obliq>N+007.137430_61.155784_170425</ms:id_obliq>
      ▼<ms:url>
         http://172.16.1.202?usertoken=484441ce101a3bbc562b7f0ca5eb65b6&lat=7.13743&lon=61.155784&orientation=N&view=single
       </ms:url>
       <ms:shotdate>2017-04-25 12:00:00</ms:shotdate>
       <ms:shotorient>N</ms:shotorient>
     </ms:obliquehr>
   </gml:featureMember>
  ▼<gml:featureMember>
    ▼<ms:obliquehr gml:id="obliquehr.S+007.134578_61.155361_170425">
      ▼<gml:boundedBy>
        ▼<gml:Envelope srsName="EPSG:4326">
           <gml:lowerCorner>61.155361 7.134578</gml:lowerCorner>
```

16.20 GetSRSInfo Service

16.20.1 Description

This service returns the projection system information for a given EPSG code or search the available projections info that matches with a given name.

16.20.2 Parameters

The parameters for this service are:

- **name [optional]**: This is the text string used for the projection searching by name.
- **epsg [optional]:** This is the EPSG code used for the projection search.

  One of those parameters is necessary.

- **format [mandatory]:** Format used for the response. Possible values are:
  - **txt:** The service response will be the list of transformed coordinates separated by commas (“,”).
  - **Xml:** The service response is an XML file, with the list of coordinates transformed to the new projection system.
  - **js:** The service response is a JavaScript function that will return the values. This is for internal use of the JavaScript API. The **function** parameter is required to specify the name of the function to return.
  - **json:** The service response is a json file.
- **usertoken [mandatory]**: User’s Token for authentication purposes. This token is unique for each client and must be requested to Blom.

16.20.3 Examples

16.20.3.1 By Name

In this example the service is called to search the projection systems that matches with the word ‘ntm’.

```
http://www.blomurbex.com/v02/GetSRSInfo?usertoken=********&name=ntm&format=txt
```

The result will be a list hash-separted of EPSG codes and short names:

5130, ETRS89 / NTM zone 30#5129, ETRS89 / NTM zone 29#5128, ETRS89 / NTM zone 28#5127, ETRS89 / NTM zone 27#5126, ETRS89 / NTM zone 26#5125, ETRS89 / NTM zone 25#5124, ETRS89 / NTM zone 24#5123, ETRS89 / NTM zone 23#5122, ETRS89 / NTM zone 22#5121, ETRS89 / NTM zone 21#5120, ETRS89 / NTM zone 20#5119, ETRS89 / NTM zone 19#5118, ETRS89 / NTM zone 18#5117, ETRS89 / NTM zone 17#5116, ETRS89 / NTM zone 16#5115, ETRS89 / NTM zone 15#5114, ETRS89 / NTM zone 14#5113, ETRS89 / NTM zone 13#5112, ETRS89 / NTM zone 12#5111, ETRS89 / NTM zone 11#5110, ETRS89 / NTM zone 10#5109, ETRS89 / NTM zone 9#5108, ETRS89 / NTM zone 8#5107, ETRS89 / NTM zone 7#5106, ETRS89 / NTM zone 6#5105, ETRS89 / NTM zone 5

16.20.3.2 By EPSG Code

In this example the service is called to search the projection systems that matches with the code '5110'.

http://www.blomurbex.com/v02/GetSRSInfo?**usertoken**=********&**name**=ntm&**format**=jon

The result is a json with the short name, epsg code, proj4 & projcs definitions:


```
{
  "NAME": "ETRS89 / NTM zone 10",
  "EPSG": 5110,
  "PROJ4": "+proj\u003dtmerc +lat_0\u003d58 +lon_0\u003d10.5 +k\u003d1
       +x_0\u003d100000 +y_0\u003d1000000 +ellps\u003dGRS80
       +towgs84\u003d0,0,0,0,0,0,0 +units\u003dm +no_defs ",
  "PROJCS": "PROJCS[\"ETRS89 / NTM zone
10\",GEOGCS[\"ETRS89\",DATUM[\"European_Terrestrial_Reference_System_1
989\",SPHEROID[\"GRS
1980\",6378137,298.257222101,AUTHORITY[\"EPSG\",\"7019\"]],TOWGS84[0,0
,0,0,0,0,0],AUTHORITY[\"EPSG\",\"6258\"]],PRIMEM[\"Greenwich\",0,AUTHO
RITY[\"EPSG\",\"8901\"]],UNIT[\"degree\",0.0174532925199433,AUTHORITY[
\"EPSG\",\"9122\"]],AUTHORITY[\"EPSG\",\"4258\"]],PROJECTION[\"Transve
rse_Mercator\"],PARAMETER[\"latitude_of_origin\",58],PARAMETER[\"centr
al_meridian\",10.5],PARAMETER[\"scale_factor\",1],PARAMETER[\"false_ea
sting\",100000],PARAMETER[\"false_northing\",1000000],UNIT[\"metre\",1
,AUTHORITY[\"EPSG\",\"9001\"]],AUTHORITY[\"EPSG\",\"5110\"]]"

}
```