

Get API

Introduction

The GET-API is meant to be a simplified way of communicating with the WCM backend for external partners. In the long run, functionality is planned for geocoding, reverse geocoding, coordinate transformation and routing, as well as a static map API. This document is a guide to these APIs, both in terms of what they allow you to do and how you go about using them. For all the APIs, using the functionality they expose is just a matter of executing a HTTP GET call with the appropriate parameters.

Note: parameters sent to the API should be UTF8-encoded; all responses are encoded with UTF8.

- Introduction
- Geocoding
 - Parameters
 - Request
 - Response
- Geo id search
 - Parameters
 - Example
- Reverse Geocoding
 - Parameters
 - Request
 - Response
- Geo suggest
 - Parameters
 - Request
 - Response
- Front door
 - Parameters
 - Request
 - Response
 - Examples
 - XML
 - JSON
 - JSON-P
 - Size
 - Profile
- Maptypes
 - Parameters
 - Request
- Routing
 - Parameters
 - Request
 - Response
 - Examples
 - Resolution (res)
 - Way points (waypoints)
 - Language (lang)
 - Summary (instr, geo)
 - Avoid (avoid)
 - Preference (pref)
 - Content type (contentType)
- Coordinate Transformation
 - Parameters
 - Request
 - Response
 - Examples
 - Bounding box (bbox) RT90 to WGS84
 - Coordinate (p) RT90 to WGS84
- StatMap
 - Parameters
 - Map size, format and type
 - Geographical content
 - Point objects
 - Pre-defined symbols
 - Geometry (line, polygon) objects
 - Routing requests
 - RT90 or WGS84

- Examples
 - Bounding box (bbox)
 - Center coordinate (cc) and buffer
 - Center coordinate (cc) and zoom
 - Image size (iwidth, iheight)
 - Image format (ifformat)
 - Image type (itype)
 - Point object (p) - default symbol
 - Point object (p) - pre-defined symbols
 - Point object (p) - user defined symbols
 - Point object (p) - default, pre-defined and user defined symbols
 - Prefix (prefix) - with user defined symbols
 - Polygon geometry (g) - default rendering
 - Line geometry (g) - default rendering
 - Routing
 - RT90 (grid)
- Tileversion
 - Parameters
 - Request examples
- Future development
 - Points/lines/polygons with cc and zoom - StatMap
 - Combinations of StatMap parameters

Geocoding

The geocoding service is to be used when you/a user wants to search for an address, a placename or a street in order to get more geographical information. The functionality is similar to that available in the normal map. The search can be limited to a particular country, or be pan-nordic; it can also be limited to a particular bounding box. The result of the search is ordered by relevance and is limited to a maximum of 25.

Please note that the 'any' option for the 'type'-parameter is currently implicit, i.e. leaving it out yields an 'any'-search whereas sending 'any' as a parameter-value doesn't. This is a known issue and will be addressed.

Parameters

Parameter	Description	Type	Required	Valid values	Default
name	The name of the location, e.g. "Blekingegatan, Stockholm"	String	true	Any string	None
type	Limit the search to a particular type	String	false	any, address, street, city	any
country	Limit the search to a particular country	String	false	se, dk, fi, no	all
bbox	Limit the search to a bounding box (WGS84 decimal coordinates)	String	false	A comma-separated string of minlon,minlat;maxlon,maxlat	Null
hits	The number of search results returned	Integer	false	1, 2, ..., 100, Cannot be used together with count	25
contentType	What kind of content to return (JSON, XML or Atom+GeoRSS)	String	false	json, xml, atom	json
callback	The name of a callback function to wrap the result around (valid only if the contentType is JSON)	String	false	Any string	Null

offset	Number of results to offset the whole search result with	Integer	false	Any number, cannot be used together with count	0
count	Returns only the header with count information	String	false	true, false, cannot be used together with hits or offset	null
grid	The reference grid of the position coordinates (returned hits)	String	false	mercator, utm32, rt90 or wgs84	wgs84
geometry	Returns a geometry for each result. For city and address this is a point, and for street it is a line string describing the street.	String	false	true, false	false

Request

E.g. to do a search in Sweden for 'Storgatan':

```
http://kartor.eniro.se/api/geocode?country=se&name=Storgatan+Stockholm&type=street&contentType=xml&hits=25
```

Example of using count parameter, get count for 'Storgatan' in Sweden:

```
http://kartor.eniro.se/api/geocode?country=se&name=Storgatan&type=street&contentType=xml&count=true
```

More Geocode examples can be found on page [WCM:Geocode_examples](#).

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<eniro:geocodeResponse xmlns:eniro="http://wcm.eniro.com/api" searchType="exact exact" searchMessage="all words included" totalHits="1">
  <locations>
    <addressId>maps_place_fi;119291;22</addressId>
    <locationType>ROAD</locationType>
    <roadname>Uusikatu</roadname>
    <houenumber/>
    <placename/>
    <zip>20540</zip>
    <postarea/>
    <city>TURKU</city>
    <municipality>Turku</municipality>
    <country>fi</country>
    <placementCoordinate>
      <EPSG>4326</EPSG>
      <x>22.29451300</x>
      <y>60.45855200</y>
    </placementCoordinate>
    <accessRoadCoordinate>
      <EPSG>4326</EPSG>
      <x>22.29451300</x>
      <y>60.45855200</y>
    </accessRoadCoordinate>
    <boundingRectangle>
      <EPSG>4326</EPSG>
      <minX>22.29398100</minX>
      <minY>60.45810600</minY>
      <maxX>22.29504900</maxX>
      <maxY>60.45901200</maxY>
    </boundingRectangle>
  </locations>
</eniro:geocodeResponse>
```

Geo id search

Use the geo id search to search for an address, a placename or a street with a known id, in order to get the details.

Please note that the 'any' option for the 'type'-parameter is currently implicit, i.e. leaving it out yields an 'any'-search whereas sending 'any' as a parameter-value doesn't. This is a known issue and will be addressed.

Parameters

Parameter	Description	Type	Required	Valid values	Default
id	The primary id of the address.	Numeric	false	Any numeric	None
cmp_id	An internal id. Only unique in combination with country and type.	Numeric	false	Any numeric	None
source_id	A secondary id of the address, as given by the data supplier, e.g. the KVHB code in Denmark. Note that addresses might be missing this id.	String	false	Any string	None
type	Limit the search to a particular type	String	false	any, address, street, city	any
country	Limit the search to a particular country	String	false	se, dk, fi, no	all
contentType	What kind of content to return (JSON, XML or Atom+GeoRSS)	String	false	json, xml, atom	json
callback	The name of a callback function to wrap the result around (valid only if the contentType is JSON)	String	false	Any string	Null
grid	The reference grid of the position coordinates (returned hits)	String	false	mercator, utm32, rt90 or wgs84	wgs84
geometry	Returns a geometry for each result. For city and address this is a point, and for street it is a line string describing the street.	String	false	true, false	false

Example

To do a source id search in Denmark for Avnhøjvej 1, Vejen:

http://map.krak.dk/api/getAddress?source_id=575_720_1

Reverse Geocoding

Reverse geocoding is the process of obtaining addresses near a specified point. It is especially useful in mobile applications where a user can get information of what addresses, roads and places are nearby his/her position.

Parameters

Parameter	Description	Type	Required	Valid values	Default
p	A point/coordinate. The position for which you want to expand your search from, in x,y format	Decimal	true	Any WGS84 coordinate	None
distance	Limits the search to a maximum search radius (in meters)	Decimal	false		1000
type	Limit the search to a particular type	String	false	any, address, street, city	any
country	Limit the search to a particular country	String	false	se, dk, no, pl	all countries
hits	The number of search results returned	Integer	false	1, 2, ..., 25, Cannot be used together with count	25
contentType	What kind of content to return (JSON, XML or Atom+GeoRSS)	String	false	json, xml, atom	json
callback	The name of a callback function to wrap the result around (valid only if the contentType is JSON)	String	false	Any string	Null
offset	Number of results to offset the whole search result with	Integer	false	Any number, cannot be used together with count	0
count	Returns only the header with count information (number of potential hits)	String	false	true, false, cannot be used together with hits or offset	null
grid	The reference grid of the position coordinates (returned hits)	String	false	mercator, utm32, rt90 or wgs84	wgs84
geometry	Returns a geometry for each result. For city and address this is a point, and for street it is a line string describing the street.	String	false	true, false	false

Request

Example of reverse geocode search in the middle of central Stockholm:

```
http://map.eniro.com/api/revgeocode.json?p=18.06964,59.32480&distance=1000&type=any&country=se&hits=10&contentType=json
```

Response

Same format as for normal geocoding queries.

Geo suggest

The geo suggest is used for returning candidates for geo locations based on the first letters.

Parameters

Parameter	Description	Type	Required	Valid values	Default
name	Name of location to search for	String	true	Any string	None
type	Limit the search to a particular type	String	false	any, address, street, city	address
country	Limit the search to a particular country	String	false	se, dk, no, pl	all countries
hits	The number of search results returned	Integer	false	1, 2, ..., 25	25
contentType	What kind of content to return (JSON, XML or Atom+GeoRSS)	String	false	json, xml, atom	json
callback	The name of a callback function to wrap the result around (valid only if the contentType is JSON)	String	false	Any string	Null
explain	Returns debug info	String	false	false	None

Request

Request a geo suggest for a address in Denmark starting with *Stor* and get the response in json:

<http://map.krak.dk/api/geoSuggest?country=dk&name=Storg&type=street&contentType=json&hits=25>

Response

```

{
  "search": {
    "geoSuggestResponse": {
      "items": [
        {
          "sug": "Storgårdsvej, Hirtshals, Hjørring"
        },
        {
          "sug": "Storgårdvej, Ejstrupholm, Ikast-Brande"
        },
        {
          "sug": "Storgårdvej, Herning"
        },
        {
          "sug": "Storgårdsvej, Hammel, Favrskov"
        },
        {
          "sug": "Storgårdsvej, Fuglebjerg, Næstved"
        },
        {
          "sug": "Storgårdsvej, Haslev, Faxe"
        },
        {
          "sug": "Storgårdsvej, Tappernøje, Næstved"
        },
        {
          "sug": "Storgade, Fårvang, Silkeborg"
        },
        {
          "sug": "Storgårdsvej, Videbæk, Ringkøbing-Skjern"
        },
        {
          "sug": "Storgårdsvej, Jægerspris, Frederikssund"
        },
        {
          "sug": "Storgade, Sorø"
        },
        {
          "sug": "Storgaards Vej, Jerup, Frederikshavn"
        },
        {
          "sug": "Storgaden, Viuf, Kolding"
        },
        {
          "sug": "Storgaardsvej, Saltum, Jammerbugt"
        }
      ]
    }
  }
}

```

Front door

The front door API allows for front door requests. The response is available in XML, JSON. Padded JSON can also be returned. The response contains a URL for a front door image and a URL to be used for linking in to street view mode.

Parameters

Parameter	Description	Type	Required	Valid values	Default
<tt>p	The coordinates to get front door for	Comma separated string	Yes	Coordinates on the form lat,lon	None

profile	The profile to get the front door for	String	Yes	Currently supported profiles are se, dk, dk_krak, fi, no, no_kvasir, no_tkb, pl	None
iwidth	Width of front door image	Integer	No	positive integers smaller than 800	230
iheight	Height of front door image	Integer	No	positive integers smaller than 600	130
callback	A callback for jsonp	String	No	Any string	None
contentType	The response content type	String	No	xml, json	json

Request

Request front door for a coordinate in sweden and get the response in xml:

```
http://kartor.eniro.se/api/frontdoor?p=59.290772,18.110529&profile=se&contentType=xml
```

Response

The response will look like:

```
<eniro:frontDoorResponse>

<frontDoor>http://kartor.eniro.se.test.eniro.net/streetview/image.jpg?path=stockholm_20090810/2173919_00201/20090810_2173919_201348&b=67&p=0&w=230&h=230&lod=2</frontDoor>

<streetView>http://kartor.eniro.se.test.eniro.net/query?what=streetView&lng=18.110281083&lat=59.290717861&heading=67</streetView>
</eniro:frontDoorResponse>
```

Examples

Some examples

XML

```
http://kartor.eniro.se/api/frontdoor?p=59.290772,18.110529&profile=se&contentType=xml
```

[XML example - Front door](#)

JSON

```
http://kartor.eniro.se/api/frontdoor?p=59.290772,18.110529&profile=se&contentType=json
```

[JSON example - Front door](#)

JSON-P

```
http://kartor.eniro.se/api/frontdoor?p=59.290772,18.110529&profile=se&contentType=json&callback=C
```

[JSON-P example - Front door](#)

Size

```
http://kartor.eniro.se/api/frontdoor?p=59.290772,18.110529&profile=se&contentType=xml&iwidth=250&iheight=180
```

[Size example - Front door](#)

Profile

```
http://kort.eniro.dk/api/frontdoor?p=55.675985015032,12.56926071153&profile=dk&contentType=xml
```


Maotypes

The maptype service is used for retrieving which map types are available for a specific bounding box (most likely the current map viewport). In addition to the available maotypes, the appropriate copyright text for each map type is also included in the response.

Parameters

Parameter	Description	Type	Required	Valid values	Default
bbox	The bounding box for which you want the available map types.	String	Yes	A comma-separated string of minlon,minlat;maxlon,maxlat	None
grid	The grid to be used when requesting coordinates	String	Optional	mercator, utm32, rt90 or wgs84	wgs84
contentType	What kind of content to return (json or xml)	String	No	json, xml	json
callback	The name of a callback function to wrap the result around (valid only if the contentType is JSON)	String	No	Any string	None

Request

<http://map.eniro.com/api/maotypes?bbox=18.06,59.31;18.09,59.32>

Example

<http://map.eniro.com/api/maotypes?bbox=18.06,59.31;18.09,59.32&grid=wgs84&callback=C>

Example with grid and callback

<http://map.eniro.com/api/maotypes?bbox=18.06,59.31;18.09,59.32&contentType=xml>

XML example

Routing

The routing API allows partners to request routing information. The result can be delivered in XML or JSON format. A number of parameters can be used to control how much information the response should contain. The parameters can in most cases be freely combined, with some exceptions which are listed below.

Parameters

Parameter	Description	Type	Required	Valid values	Default
waypoints	The start, via points, and endpoint of the route in WGS84 coordinates	Comma and semicolon separated string	Yes	Two or more set of coordinates on the form x1,y1;x2,y2;x3,y3	None
bbox	Limit the geometry returned to a bounding box in WGS84 coordinates	String	No	Comma and semicolon separated string on the form x1,y1;x2,y2	None

pref	Make route request for either fastest or shortest route	String	No	shortest, fastest	fastest
instr	Include or exclude textual instructions	String	No	true, false	true
geo	Include or exclude geometry for the route	String	No	true, false	true
res	Defines the resolution of the geometry returned	Integer	No if geo is provided and set to false	true, false	true
avoid	Avoid features along the route	String, multiple instances (e.g. avoid=highway, avoid=ferry)	No	ferry, highway, tollway	true
lang	Get textual instructions in preferred language	String	No	se, dk, fi, no	se
contentType	Get route result on preferred format	String	No	xml, json	xml

Request

To perform a route request between two coordinates in Stockholm city, exclude the geometry and get the textual information in Swedish.

<http://kartor.eniro.se/api/route?waypoints=18.024088,59.332775;18.028122,59.333759&geo=false&lang=se>

Response

The response will look like

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<multiRoute>
  <routeInstructions>
    <Feature>
      <Point>
        <coordinates>18.024086505478504</coordinates>
        <coordinates>59.33277651643381</coordinates>
      </Point>
      <properties>
        <property>
          <key eniro:type="ns3:string">instruction</key>
          <value eniro:type="ns3:string">Starta riktning nord på Mariebergsgatan, kör 0.3
km</value>
        </property>
      </properties>
      <bbox>18.024086</bbox>
      <bbox>59.332775</bbox>
      <bbox>18.025543</bbox>
      <bbox>59.335274</bbox>
    </Feature>
    <Feature>
      <Point>
        <coordinates>18.025544090089312</coordinates>
        <coordinates>59.33527227964519</coordinates>
      </Point>
      <properties>
        <property>
          <key eniro:type="ns3:string">instruction</key>
          <value eniro:type="ns3:string">Sväng höger in på Fleminggatan, kör 0.2 km</value>
        </property>
      </properties>
      <bbox>18.025543</bbox>
      <bbox>59.335007</bbox>
    </Feature>
  </routeInstructions>
</multiRoute>
```

```

        <bbox>18.02841</bbox>
        <bbox>59.335274</bbox>
    </Feature>
    <Feature>
        <Point>
            <coordinates>18.028409493195763</coordinates>
            <coordinates>59.33500799422891</coordinates>
        </Point>
        <properties>
            <property>
                <key eniro:type="ns3:string">instruction</key>
                <value eniro:type="ns3:string">Sväng höger in på Arbetargatan, kör 100 m</value>
            </property>
        </properties>
        <bbox>18.027714</bbox>
        <bbox>59.33382</bbox>
        <bbox>18.02841</bbox>
        <bbox>59.335007</bbox>
    </Feature>
    <Feature>
        <Point>
            <coordinates>18.027713389238645</coordinates>
            <coordinates>59.33382002586858</coordinates>
        </Point>
        <properties>
            <property>
                <key eniro:type="ns3:string">instruction</key>
                <value eniro:type="ns3:string">Sväng vänster in på Sankt Göransgatan, kör 24 m</value>
            </property>
        </properties>
        <bbox>18.027714</bbox>
        <bbox>59.333759</bbox>
        <bbox>18.028122</bbox>
        <bbox>59.33382</bbox>
    </Feature>
    <Feature>
        <Point>
            <coordinates>18.02812255665254</coordinates>
            <coordinates>59.3337610899088</coordinates>
        </Point>
        <properties/>
        <bbox>18.028123</bbox>
        <bbox>59.333761</bbox>
        <bbox>18.028123</bbox>
        <bbox>59.333761</bbox>
    </Feature>
    <id>ril</id>
</routeInstructions>
<routeGeometries>
    <Feature>
        <MultiLineString>
            <coordinateSet>
                <coordinates>
                    <coord eniro:type="ns3:doubleArray">
                        <item>18.024086</item>
                        <item>59.332775</item>
                    </coord>
                    <coord eniro:type="ns3:doubleArray">
                        <item>18.028122</item>
                        <item>59.333759</item>
                    </coord>
                </coordinates>
            </coordinateSet>
        </MultiLineString>
        <properties>
            <property>
                <key eniro:type="ns3:string">duration</key>
                <value eniro:type="ns3:long">46</value>
            </property>
            <property>
                <key instruction-id</key>
                <value eniro:type="ns3:string">ril</value>
            </property>
        </properties>
    </Feature>

```

```
        </property>
        <property>
          <key eniro:type="ns3:string">length</key>
          <value eniro:type="ns3:double">619.0</value>
        </property>
      </properties>
      <bbox>18.024086</bbox>
      <bbox>59.332775</bbox>
      <bbox>18.028122</bbox>
      <bbox>59.333759</bbox>
    </Feature>
  <bbox>18.024086</bbox>
  <bbox>59.332775</bbox>
  <bbox>18.028122</bbox>
  <bbox>59.333759</bbox>
</routeGeometries>
<totalDuration>46</totalDuration>
<totalLength>619.0</totalLength>
</multiRoute>
```

Examples

Resolution (res)

<http://kartor.eniro.se/api/route?waypoints=18.024088,59.332775;18.028122,59.333759&lang=se&res=200>

[Resolution \(res\) example - Route](#)

Way points (waypoints)

<http://kartor.eniro.se/api/route?waypoints=18.024086,59.332775;18.029453169272788,59.334845999871064;18.028122,59.333759&lang=se&res=200>

[Way points \(waypoints\) example - Route](#)

Language (lang)

<http://kartor.eniro.se/api/route?waypoints=18.024088,59.332775;18.028122,59.333759&lang=no&res=200>

[Language \(lang\) example - Route](#)

Summary (instr, geo)

<http://kartor.eniro.se/api/route?waypoints=18.024088,59.332775;18.028122,59.333759&instr=false&geo=false>

[Summary \(instr, geo\) example - Route](#)

Avoid (avoid)

<http://kartor.eniro.se/api/route?waypoints=18.024088,59.332775;18.028122,59.333759&lang=se&avoid=ferry&avoid=highway>

[Avoid \(avoid\) example - Route](#)

Preference (pref)

<http://kartor.eniro.se/api/route?waypoints=18.024088,59.332775;18.028122,59.333759&lang=se&pref=shortest>

[Preference \(pref\) example - Route](#)

Content type (contentType)

<http://kartor.eniro.se/api/route?waypoints=18.024088,59.332775;18.028122,59.333759&lang=se&contentType=json>

[Content type \(contentType\) example - Route](#)

Coordinate Transformation

The transform service is to be used when a transformation of a coordinate or a bounding box between two different coordinate systems is desired.

Parameters

Parameter	Description	Type	Required	Valid values	Default
p	A coordinate to transform.	String	No if bbox is provided, otherwise yes.	String on form lon,lat or x,y depending on what format to transform from.	None
bbox	A bounding box to transform.	String	No if p is provided, otherwise yes.	String on the form lon,lat;lon,lat or x,y;x,y depending on what format to transform from.	None
from	The coordinate system to transform from.	String	Yes	wgs84, rt90, mercator, utm32	wgs84
to	The coordinate system to transform to.	String	Yes	wgs84, rt90, mercator, utm32	wgs84

Request

To transform a coordinate from WGS84 to RT90:

```
http://kartor.eniro.se/api/transform?p=15.131858562535234,60.33456148739118&from=wgs84&to=rt90
```

Response

The response for a coordinate transformation will look like:

```
<eniro:transformResponse>
  <coordinate>
    <x>6690908.0</x>
    <y>1462812.0</y>
  </coordinate>
</eniro:transformResponse>
```

Examples

A few examples.

Bounding box (bbox) RT90 to WGS84

```
http://kartor.eniro.se/api/transform?bbox=6578743,1628416;6579915,1630086&from=rt90&to=wgs84
```

[Bounding box \(bbox\) example - Coordinate Transformation](#)

Coordinate (p) RT90 to WGS84

```
http://kartor.eniro.se/api/transform?p=6690908,1462812&from=rt90&to=wgs84
```

[Coordinate \(p\) example - Coordinate Transformation](#)

StatMap

NB: THE COORDINATE FORMAT HAS CHANGED; from 'lat lon,lat lon' to 'lon,lat;lon,lat'

NB: While the prod environment not is updated the dev server can be used instead; [rute1.gulesider.no:8080](#)

The StatMap API is a http get API that allows partners to request static map images in a stadard raster image format, ie. png. In addition to a map, the image can contain one or more points, lines and polygons displayed with pre-defined or user defined look.

The StatMap API is intended for use in applications missing support for Javascript - eg. mobile applications - and where a dynamic map is beyond the scope, ie. the result of a real estate for-sale search.

Parameters

Map size, format and type

The following parameters are used to specify the size, image format and type of map to use when rendering a map image.

Parameter	Description	Type	Required	Valid values	Default
iwidth	The width of the map image in pixels.	Integer	No	Integer value, min 50, max 1200	250
iheight	The height of the map image in pixels.	Integer	No	Integer value, min 50, max 1200	250
itype	The type of map used in the image.	String	No	map, ortho, hybrid, nautical	map
iformat	The format of the resulting image.	String	No	image/png, image/jpeg, image/gif	image/png
scale	Whether the map should be high res (scale=2)	Number	No	1, 2	1

Geographical content

The following parameters are used to specify the geographical content of the map.

Parameter	Description	Type	Required	Valid values	Default
bbox	The bounding box of the map area defined by lower left and upper right coordinate. Coordinates are defined in longitude/latitude - eg. '18.06,59.31;18.09,59.32'.	String	Optional when cc is used	'minlon,minlat;maxlon,maxlat' - eg. '18.06,59.31;18.09,59.32'	None
cc	The map center (latitude, longitude) coordinate - eg. '18.075,59.315'.	String	Optional when bbox is used	'lon,lat' - eg. '18.075,59.315'	None
buffer	A buffer around (added in all directions) the center coordinate. The buffer is defined in meters. The buffered area should always be present within the map regardless of the requested image dimension.	Float	No	Positive decimal value – eg. 1024 or 24.8	250
zoom	The zoom level of the map.	Integer	Optional when bbox or cc/buffer is used	TMS zoom value: 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17. Note: Level 5 is missing.	None

Point objects

The following parameters are used when point objects should be displayed in the map image. The map area is updated to display the given point objects.

Parameter	Description	Type	Required	Valid values	Default
p	The longitude/latitude coordinate for the point object together with information about which symbol to use when rendering the point object. The point object can be rendered with default, pre-defined or user defined symbols. Pre-defined symbols are listed below. The id of a user defined symbol must not be the same as a the name of a pre-defined symbol. Eg. '18.06,59.31;redcircle'.	String	Optional	'lon,lat[;pre-defined-symbol],user-defined-symbol]' - eg. '18.06,59.31' or '18.06,59.31;redcircle' or '18.06,59.31;mysymbol'	None
user-defined-symbol-id	Image URL and offset for a user defined point symbol. The offset is the distance in pixels from the top left corner of the image to the part of the image that should be centered on the coordinate of the point. Eg. 'http://myserver.com/image.png,9,32'.	String	Required for every unique user defined point symbol	'url[,offsetx[,offsety]]' - eg. 'http://myserver.com/image.png,9,32'	None
prefix	To shorten the URL:s a prefix can be specified and will work as a base URL and, if specified, will be prepended to all the image paths	String	No	Any valid path to an image	None

Pre-defined symbols

- redcircle
- bluecircle

Geometry (line, polygon) objects

The following parameters are used when line or polygon geometry objects should be displayed in the map image. The map area is updated to display the given geometry objects.

Parameter	Description	Type	Required	Valid values	Default
-----------	-------------	------	----------	--------------	---------

g	The longitude/latitude coordinates for the line or polygon geometry. The geometry is handled as a polygon if the first and last coordinate has the same longitude/latitude value.	String	Optional	'lon,lat;lon,lat...' - eg. '18.087,59.315;18.09,59.34;18.1134,59.3233;18.1,59.32'	None
---	---	--------	----------	---	------

Routing requests

The following parameters are used when requesting a static map with a routing geometry pre-rendered on it. Just as in the case of the geometry request above, the map area is updated to display the given geometry objects.

Parameter	Description	Type	Required	Valid values	Default
wp	The longitude/latitude coordinates of the routing points (waypoints).	String	True	'lon,lat;lon,lat...' - eg. '18.087,59.315;18.09,59.34;18.1134,59.3233;18.1,59.32'	None

RT90 or WGS84

It is possible to request the coordinates in RT90 as well as WGS84.

Parameter	Description	Type	Required	Valid values	Default
grid	The grid to be used when requesting coordinates	String	Optional	rt90 or wgs84	wgs84

Examples

Note: The resulting bounding box is sometimes rendered as debug.

Bounding box (bbox)

<http://kartor.eniro.se/api/statmap?bbox=18.06,59.31;18.09,59.32&iwidth=640&iheight=480>

[Bounding box \(bbox\) example - StatMap \(prod\)](#)

[Bounding box \(bbox\) example - StatMap \(test\)](#)

Center coordinate (cc) and buffer

<http://kartor.eniro.se/api/statmap?cc=18.075,59.315&buffer=1000&iwidth=640&iheight=480>

[Center coordinate \(cc\) and buffer example - StatMap \(prod\)](#)

[Center coordinate \(cc\) and buffer example - StatMap \(test\)](#)

Center coordinate (cc) and zoom

<http://kartor.eniro.se/api/statmap?cc=18.075,59.315&zoom=14&iwidth=640&iheight=480>

[Center coordinate \(cc\) and zoom \(14\) example - StatMap \(prod\)](#)

[Center coordinate \(cc\) and zoom \(14\) example - StatMap \(test\)](#)

<http://kartor.eniro.se/api/statmap?cc=18.075,59.315&zoom=15&iwidth=640&iheight=480>

[Center coordinate \(cc\) and zoom \(15\) example - StatMap \(prod\)](#)

[Center coordinate \(cc\) and zoom \(15\) example - StatMap \(test\)](#)

<http://kartor.eniro.se/api/statmap?cc=18.075,59.315&zoom=16&iwidth=640&iheight=480>

[Center coordinate \(cc\) and zoom \(16\) example - StatMap \(prod\)](#)
[Center coordinate \(cc\) and zoom \(16\) example - StatMap \(test\)](#)

Image size (iwidth, iheight)

<http://kartor.eniro.se/api/statmap?cc=18.075,59.315&buffer=1000&iwidth=500&iheight=500>

[Image size 500x500 example - StatMap \(prod\)](#)
[Image size 500x500 example - StatMap \(test\)](#)

Image format (ifformat)

<http://kartor.eniro.se/api/statmap?cc=18.075,59.315&zoom=14&iwidth=640&iheight=480>

[Png image format example - StatMap \(prod\)](#)
[Png image format example - StatMap \(test\)](#)

<http://kartor.eniro.se/api/statmap?cc=18.075,59.315&zoom=14&iwidth=640&iheight=480>

[Jpeg image format example - StatMap \(prod\)](#)
[Jpeg image format example - StatMap \(test\)](#)

<http://kartor.eniro.se/api/statmap?cc=18.075,59.315&zoom=14&iwidth=640&iheight=480&ifformat=image/gif>

[Gif image format example - StatMap \(prod\)](#)
[Gif image format example - StatMap \(test\)](#)

Image type (itype)

<http://kartor.eniro.se/api/statmap?cc=18.075,59.315&zoom=14&iwidth=640&iheight=480&itype=map>

[Map image type example - StatMap \(prod\)](#)
[Map image type example - StatMap \(test\)](#)

<http://kartor.eniro.se/api/statmap?cc=18.075,59.315&zoom=14&iwidth=640&iheight=480&itype=ortho>

[Ortho image type example - StatMap \(prod\)](#)
[Ortho image type example - StatMap \(test\)](#)

<http://kartor.eniro.se/api/statmap?cc=18.075,59.315&zoom=14&iwidth=640&iheight=480&itype=hybrid>

[Hybrid image type example - StatMap \(prod\)](#)
[Hybrid image type example - StatMap \(test\)](#)

Point object (p) - default symbol

<http://kartor.eniro.se/api/statmap?p=18.087,59.315&p=18.09,59.32&p=18.11341,59.3233&p=18.1,59.33&p=18.07,59.325&iwidth=640&iheight=480>

[Point object - default symbol - StatMap \(prod\)](#)
[Point object - default symbol - StatMap \(test\)](#)

Point object (p) - pre-defined symbols

<http://kartor.eniro.se/api/statmap?p=18.087,59.315;redcircle&p=18.1,59.33;bluecircle&p=18.07,59.325;bluecircle&iwidth=640&iheight=480>

[Pre-defined symbols - StatMap \(prod\)](#)
[Pre-defined symbols - StatMap \(test\)](#)

Point object (p) - user defined symbols

<http://kartor.eniro.se/api/statmap?p=18.07245,59.338583;mysymbol&mysymbol=http://www.spotify.com/wp-content/themes/spotify/images/feed-icon-14x14.png&iwidth=640&iheight=480>

[User defined symbol - centered - StatMap \(prod\)](#)
[User defined symbol - centered - StatMap \(test\)](#)

<http://kartor.eniro.se/api/statmap?p=18.07245,59.338583;mysymbol&mysymbol=http://maps.google.com/mapfiles/ms/micons/ylw-pushpin.png,12,30&iwidth=640&iheight=480>

[User defined symbol with offset - StatMap \(prod\)](#)
[User defined symbol with offset - StatMap \(test\)](#)

Point object (p) - default, pre-defined and user defined symbols

<http://kartor.eniro.se/api/statmap?p=18.087,59.315&p=18.09,59.34;redcircle&p=18.1134,59.3233;bluecircle&p=18.1,59.32;mysymbol2&p=18.07,59.325;mysymbol2&p=18.076,59.339;mysymbol1&p=18.07245,59.338583;mysymbol2&mysymbol1=http://www.spotify.com/wp-content/themes/spotify/images/feed-icon-14x14.png&mysymbol2=http://maps.google.com/mapfiles/ms/micons/ylw-pushpin.png,12,30&iwidth=640&iheight=480>

[Default, pre-defined and user defined symbols - StatMap \(prod\)](#)

[Default, pre-defined and user defined symbols - StatMap \(test\)](#)

Prefix (prefix) - with user defined symbols

<http://kartor.eniro.se/api/statmap?iwidth=400&iheight=400&p=15.966905529299545,59.518176369708364;yp-pin1&yp-pin1=yellow-1.png&p=13.509423324027402,59.376606705109914;yp-pin2&yp-pin2=yellow-2.png&prefix=http://kartor.eniro.se/media/markers/search/>

[User defined symbol - prefix - StatMap \(prod\)](#)

[User defined symbol - prefix - StatMap \(test\)](#)

Polygon geometry (g) - default rendering

<http://kartor.eniro.se/api/statmap?g=18.087,59.315;18.09,59.34;18.1134,59.3233;18.1,59.32;18.07,59.325;18.076,59.339;18.07245,59.338583;18.087,59.315&iwidth=640&iheight=480>

[Default, pre-defined and user defined symbols - StatMap \(prod\)](#)

[Default, pre-defined and user defined symbols - StatMap \(test\)](#)

Line geometry (g) - default rendering

<http://kartor.eniro.se/api/statmap?g=18.087,59.315;18.09,59.34;18.1134,59.3233;18.1,59.32;18.07,59.325;18.076,59.339;18.07245,59.338583&iwidth=640&iheight=480>

[Default, pre-defined and user defined symbols - StatMap \(prod\)](#)

[Default, pre-defined and user defined symbols - StatMap \(test\)](#)

Routing

<http://kartor.eniro.se/api/statmap?bbox=18.06,59.31;18.09,59.32&iwidth=640&iheight=480&wp=18.087,59.315;18.09,59.34;18.1134,59.3233;18.1,59.32>

[Routing request with bbox - StatMap \(prod\)](#)

[Routing request with bbox - StatMap \(test\)](#)

RT90 (grid)

<http://kartor.eniro.se/api/statmap?p=6579187,1629572&iwidth=640&iheight=480&grid=rt90>

[Point object in RT90 - StatMap \(prod\)](#)

[Point object in RT90 - StatMap \(test\)](#)

Tileversion

The tileversion service is used to find out what tile version to use when retrieving tiles for a specific map type. The result is returned as a single string (content type: text/plain).

Parameters

Parameter	Description	Type	Required	Valid values	Default
type	The map type	String	Yes	map, aerial, hybrid, nautical, streetview, oblique, obliquec3	None

format	Set to 'json' to get json response. Content type will be "application/json"	String	No	json	None
callback	jsonp callback. Content type will be "application/javascript"	String	No		None

Request examples

Example 1: <http://kartor.eniro.se/api/tileversion?type=map>

Example 2: <http://kartor.eniro.se/api/tileversion?type=aerial&format=json>

Example 3: <http://kartor.eniro.se/api/tileversion?type=map&format=json&callback=foo>

Future development

Points/lines/polygons with cc and zoom - StatMap

Should the API user be able to create a map with point objects, lines or polygons and then specify cc and zoom to be able to pan and zoom with the map? Cc and zoom will in this case override the calculation of bounding box for the given points/lines/polygons. The user will have to calculate cc and zoom by himself.

Combinations of StatMap parameters

What should the resulting map area be if any of the following combination of parameters are used?

- bbox + points/lines/polygons
- cc, buffer + points/lines/polygons
- cc, zoom + points/lines/polygons
- bbox + cc, buffer
- bbox + cc, zoom
- etc.

The current solution is to let the bounding box for the user objects (points/lines/polygons) override bbox. Bbox in turn overrides cc+buffer. Cc+buffer overrides cc+zoom. Is this ok? The user will not be able to pan or zoom with StatMap when points/lines/polygons has been added to the map.