

Documentation of ExtraAPI

Version 1.5.0 (e)

To use this functionality you must get a personal key and the IP address of server. Please, contact us by e-mail - info@gomap.az

Sending POST-request to the one of the functions which you get the response as JSON.

Example: *Example on JavaScript*

```
$.ajax({
  type: "POST",
  url: "http://xx.xx.xx.xx/Main.asmx/getRegions",
  data: {
    'region': 'Baku',
    'guid': your_guid_key
  },
  success: function (result) {
    console.log(result);
  },
  error: function(x, t, m) {
    alert("error");
  }
});
```

Web service **ExtraApi** consists of the following functions:

<http://xx.xx.xx.xx/Main.asmx/>

- **getRegions** – returns the list of the regions on the initial symbols
- **getRoads** – returns the list of the streets on the initial symbols
- **getBuildings** – returns the list of the buildings
- **getInfoBuilding** - returns the information about the building
- **searchObj** – Search objects
- **searchObjPage** - "The following 10 objects" (paging)
- **getRoute** – Route from the point A to the point B
- **getAddressByCoords** – the address returns on the coordinates

- **getFreeWaysList** - list of highways/freeways
 - **getCoordsFreeWays** - returns coordinates of kilometer
 - **searchNearBy** - returns 10 objects located in the vicinity of the coordinates
 - **searchNearBy50** - returns 50 objects located in the vicinity of the coordinates
 - **searchNearByPage** - returns the following 10 nearest objects
 - **getDescriptionByPoi_ID** - return description of object by ID
 - **getDescriptionByPoi_GUID** - return description of object by GUID
 - **getImageByPoi_GUID** - returns all images of the object by GUID
-

Detailed description of the functions

The functions **getRegions**, **getRoads**, **getBuildings** should be used in turn for getting of the coordinates up to the house / building. The obtained data from **getRegions** is used in **getRoads**, as the obtained data from **getRoads** is used in **getBuildings** function.

The function **getInfoBuilding** gives information related to the house/building more detail.

The function **searchObj** realizes the search of the objects by name and appears the first 10 objects.

The function **searchObjPage** returns the following 10 found objects.

The function **getRoute** returns the track on the points. The library <http://openlayers.org/> or other should be used for web or Android and IOS to describe this track on the client's map. The track is constructed on the point systematically.

The function **getAddressByCoords** address returns on the coordinates.

The function **getFreeWaysList** returns list of highways/freeways.

The function **getCoordsFreeWays** returns coordinates of kilometer.

The function **searchNearBy** - returns 10 objects located in the vicinity of the coordinates.

The function **searchNearBy50** - returns 50 objects located in the vicinity of the coordinates.

The function **searchNearByPage** - returns the following 10 nearest objects.

The function **getDescriptionByPoi_ID** - returns description of object by ID

The function **getDescriptionByPoi_GUID** - returns description of object by GUID

The function **getImageByPoi_GUID** - returns all images of the object by GUID

getRegions

Input parameters

region - The list of the regions will be formed from the symbols.

guid – Your personal guid key

Output

JSON

- **success** - The success of query fulfill returns to the value true/false
- **names** – Collection of the regions
 - **id** - Unique identifier
 - **name** - Name
 - **type** – The type related to the name of the region
 - **x** - Coordinate
 - **y** - Coordinate

Example:

region=Baku

```
{
  "success":true,
  "names":[
    {
      "ID":10,
      "Name":"Baku",
      "Type":2,
      "X":49.848170892888575,
      "Y":40.373912986305434
    },
    {
      "ID":117,
      "Name":"Baku / Binagadi",
      "Type":8,
      "X":49.848170892888575,
      "Y":40.373912986305434
    },
    ....
  ]
}
```

```
]
}
```

getRoads

Input parameters

tpIdRegion - The combination *type_id* is transferred which has been received from the request of getRegions.

Example: type=2 and id=10, then the combination will be 2_10

road – The list of the roads / streets will be formed from the symbols.

guid – Your personal guid key

Output

JSON

- **success** - The success of query fulfill returns to the value true/false
- **names** – collection of the roads /streets
 - **id** - Unique identifier
 - **name** - name
 - **x** - Coordinate
 - **y** - Coordinate

Example:

We want to activate the list of the streets in Baku region, in which there is "narimanov" name then using id and type we'll get:

tpIdRegion =2_10,

road=narimanov

```
{
  "success":true,
  "names":[
    {
      "ID":12390,
      "Name":"Muzaffar Narimanov",
      "X":49.897245856866853,
      "Y":40.41287638721213
    },
    {
      "ID":20509,
      "Name":"Muzaffar Narimanov, block 2054",
      "X":49.895431259992677,
      "Y":40.419107071250593
    },
    {
      "ID":20520,
```

```

    "Name": "Muzaffar Narimanov, block 2055",
    "X": 49.8949282034335,
    "Y": 40.420556939113951
  },
  {
    "ID": 20427,
    "Name": "Muzaffar Narimanov, block 2058",
    "X": 49.894658708848226,
    "Y": 40.421466510077792
  },
  {
    "ID": 20680,
    "Name": "Najaf Narimanov",
    "X": 49.836016687092744,
    "Y": 40.407206031726922
  },
  {
    "ID": 7166,
    "Name": "Nariman Narimanov",
    "X": 49.822919250248454,
    "Y": 40.383028131694523
  },
  .....
]
}

```

getBuildings

Input parameters

tpIdRegion – The combination *type_id* is transferred which has been received from the request of *getRegions*.

Example: *type=2* and *id=10*, then the combination will be *2_10*

st - ID roads/streets

guid – Your personal guid key

Output

JSON

- **success** - The success of query fulfill returns to the value true/false
- **blids** – Collection of buildings/houses
 - **id** - Unique identifiers
 - **name** - name
 - **x** - Coordinate
 - **y** - Coordinate

Example:

tpIdRegion= 2_10,

st=7166 (Nariman Narimanov)

```
{
  "success":true,
  "blds":[
    {
      "ID":434144,
      "Name":"1",
      "X":49.946743029028731,
      "Y":40.475123093916821
    },
    {
      "ID":386988,
      "Name":"2c",
      "X":49.728551229638562,
      "Y":40.320531803971576
    },
    {
      "ID":569448,
      "Name":"2",
      "X":49.996680375679887,
      "Y":40.537032237933268
    },
    {
      "ID":583652,
      "Name":"3B",
      "X":49.986340766758232,
      "Y":40.430021346369159
    },
    .....
  ]
}
```

getInfoBuilding

Input parameters

id - ID building

guid – Your personal guid key

Output

JSON

- **success** - The success of query fulfill returns to the value true/false
- **objType** - Type
- **id** - Unique identifier
- **guid** - Unique identifier of building
- **addr** - Address

- **x** - Coordinate
- **y** - Coordinate
- **postal** – Postal code

Example: *Building 3B, its id = 583652*
Id=583652

```
{
  "success": true,
  "objType": "building",
  "id": 583652,
  "guid": "55e3b8bfd83249cbb5eef7a1f7b511d0",
  "addr": "Nəriman Nərimanov, 3B, Əmircan, Suraxanı, Bakı",
  "x": 49.98634076675823,
  "y": 40.43002134636916,
  "postal": "AZ1042"
}
```

searchObj

Input parameters

name – Name of the object which should be found

lng - Language

guid – Your personal guid key

Output

JSON

- **success** - The success of query fulfill returns to the value true/false
- **msg** - System parameter
- **total** - Quantity of found objects. Example: 1^128^3593 Quantity:128
- **rows**- Collection of the objects
 - **id** - Unique identifiers
 - **name** - name
 - **x** - Coordinate
 - **y** - Coordinate
 - **guid** - Unique identifiers
 - **addr** - address
 - **nm** - name

Example:

name=28 may

lng=az

```
{
  "success": true,
  "msg": "3^0",
```

```

"total": "1^128^3593",
"rows": [
  {
    "id": 21947,
    "x": 49.8491334915161,
    "y": 40.3798526861589,
    "guid": "92d49fbcd56611e0ad4900226424597d",
    "addr": "Cəfər Cabbarlı Nəsimi, Bakı",
    "nm": "28 May"
  },
  {
    "id": 66348,
    "x": 49.8490530252457,
    "y": 40.3798240811278,
    "guid": "006b9f36d56611e0ad4900226424597d",
    "addr": "Cəfər Cabbarlı Nəsimi, Bakı",
    "nm": "Metro: 28 May"
  },
  ....
]
}

```

searchObjPage

Input parameters

numPage - number of page. 10 objects appear on the page

guid – Your personal guid key

Output

JSON

- **success** - The success of query fulfill returns to the value true/false
- **msg** – system parameter
- **Total** – Quantity of found objects. Example: 1^128^3593 Quantity:128
- **rows**- Collection of the objects
 - **id** - Unique identifiers
 - **name** - name
 - **x** - Coordinate
 - **y** - Coordinate
 - **guid** – Unique identifiers
 - **addr** - address
 - **nm** - name

Example:

numPage=2

```

{
  "success": true,

```



```

"msg": " ",
"total": "2^128",
"rows": [
  {
    "id": 101231,
    "x": 48.4936308860779,
    "y": 39.2098196959787,
    "guid": "c30f2e925f044e8e8b03947ec7d0ef1b",
    "addr": "28 May C?lilabad, C?lilabad",
    "nm": "28 May"
  },
  {
    "id": 102100,
    "x": 47.7471721172333,
    "y": 40.6561800022079,
    "guid": "080e9b890633482db78a1275af4a033b",
    "addr": "28 May Goycay, Goycay",
    "nm": "28 May"
  },
  .....
]
}

```

getRoute

Input parameters

- Ax** - Coordinate X of flag A
- Ay** - Coordinate Y of flag A
- Bx** - Coordinate X of flag B
- By** - Coordinate Y of flag B
- lng** – Language
- guid** – Your personal guid key

Output

JSON

- **report** - report
 - **nm** - name
- **ttlenght**- distance
- **ttlenght1**- distance (number value)
- **time** - time (hours_ minutes)
- **time1**- time (hours_ minutes)
- **coords** – coordinates for building of the routes
 - **x** - Coordinate
 - **y** - Coordinate

Example:

```

{
  "report": [
    {
      "nm": "Start: Keykab xanım Səfərəliyeva"
    },
    {
      "nm": "Davam et: (83 m.) Keykab xanım Səfərəliyeva"
    },
    ....
    {
      "nm": "Davam et: (25 m.) Müzəffər Həsənov"
    },
    {
      "nm": "Finiş: (Ümumi məsafə: 6,4 km.) Müzəffər Həsənov"
    }
  ],
  "ttllength": "Ümumi məsafə: 6,4 km.",
  "ttllength1": 6.4,
  "time": "Ümumi vaxt: 6 dəq. (ideal yol şəraiti üçün)",
  "time1": "6",
  "coords": [
    {
      "x": 49.8575663566589,
      "y": 40.3802245504575
    },
    {
      "x": 49.8578721284866,
      "y": 40.3795094249846
    },
    ....
    {
      "x": 49.8073124885559,
      "y": 40.4042606085295
    },
    {
      "x": 49.8070174455643,
      "y": 40.4042769483327
    }
  ]
}

```

getAddressByCoords

Input parameters

x - Coordinate X

y - Coordinate Y

lng - Language

guid – Your personal guid key

Output

JSON

- **success** – The success of query fulfill returns to the value true/false
- **info** – Address

Example:

x=49.831269761308

y=40.373406621172

lng=az

```
{
  "success": true,
  "info": "Molla Vəli Vidadi, 68"
}
```

getFreeWaysList

Input parameters

guid – Your personal guid key.

Output

JSON

- **success** - The success of query fulfill returns to the value true/false
- **data** – Collection of freeways
 - **road_id** - Unique identifier
 - **street_name** - name
 - **road_name** - name
 - **length_km** – length in km

Example:

```
{
  "success": true,
  "data": [
    {
      "road_id": 1,
      "street_name": "Bilasuvar-Iran",
      "road_name": "R-43",
      "length_km": 21
    },
    {
      "road_id": 2,
      "street_name": "Baku-Alat-Gazakh-Georgia",
      "road_name": "M-2",
    }
  ]
}
```

```

        "length_km": 674
    },
    {
        "road_id": 3,
        "street_name": "Hajji Zeynalabdin Taghiyev-Sahil",
        "road_name": "R-6",
        "length_km": 54
    },
    ....
]
}

```

getCoordsFreeWays

Input parameters

- road_id** - Unique identifier
- km** - Kilometer
- guid** – Your personal guid key

Output

JSON

- **success** – The success of query fulfill returns to the value true/false
- **street_name** – Name of freeway
- **x** - Coordinate X
- **y** - Coordinate Y

Example:

We want to know the coordinates of 10 kilometers of road Bilasuvar-Iran.

Input parameters are *road_id=1, km=10*

```

{
  "success": true,
  "street_name": "Bilasuvar-Iran",
  "x": 48.5512,
  "y": 39.4597
}

```

searchNearBy / searchNearBy50

Input parameters

- X** - Coordinate X
- Y** - Coordinate Y
- type** - Type of objects (*see the table below*)
- subtype** - Subtype of objects
- R** - Radius

lng - Language

guid – Your personal guid key

Output

JSON

- **success** - The success of query fulfill returns to the value true/false
- **msg** - System parameter
- **total** - Quantity of found objects. Example: 1^128^3593 Quantity:128
- **rows**- Collection of the objects
 - **id** - Unique identifiers
 - **nm** - name
 - **x** - Coordinate
 - **y** - Coordinate
 - **guid** - Unique identifiers
 - **addr** - address
 - **phone** - phones

Type	Subtype	Name
0	0	All type of objects
1	0	Food
2	0	Lodging
3	0	Attraction
3	6	Cinema
3	11	Park
3	14	Monument
4	0	Sport
5	0	Auto
5	1	Filling station
6	0	Communication
7	0	Service
7	10	ATM
7	11	Bank
8	0	Shopping
9	0	Transport
10	0	Organizations
10	4	High School / University
10	5	School

Example:

```
{  
  "success": true,  
  "msg": "1",  
  "total": "3^3",
```

```

"rows": [
  {
    "id": 28294,
    "x": 49.8677641153336,
    "y": 40.4349193520927,
    "guid": "92638822d56611e0ad4900226424597d",
    "nm": "Furniture factory Belezza ",
    "addr": "Darnegul 1, 1c Narimanov, Baku",
    "phone": "(+994 12) 412-54-12, (+994 12) 412-29-49"
  },
  {
    "id": 28268,
    "x": 49.8670399188995,
    "y": 40.4336699149094,
    "guid": "e3458917f85b4bd787fa1f1c7541c454",
    "nm": "Road building company Taleh-A.T. ",
    "addr": "Darnegul 1, 2095 Narimanov, Baku",
    "phone": "(+994 12) 447-08-47, (+994 12) 447-08-60, (+994 12) 565-06-71, (+994 12) 565-06-73, (+994 12) 565-06-72, (+99412) 565-06-74, ",
    "url": "http://www.taleh-at.az"
  },
  {
    "id": 28285,
    "x": 49.8693466186523,
    "y": 40.4345192080709,
    "guid": "92d134e4d56611e0ad4900226424597d",
    "nm": "Concrete-mixing plant",
    "addr": "Darnegul 1 Narimanov, Baku"
  }
]
}

```

searchNearByPage

Input parameters

numPage - number of page. 10 objects appear on the page

guid – Your personal guid key

Output

JSON

- **success** - The success of query fulfill returns to the value true/false
- **msg** - System parameter
- **total** - Quantity of found objects. Example: 1^128^3593 Quantity:128
- **rows**- Collection of the objects
 - **id** - Unique identifiers
 - **nm** - name
 - **x** - Coordinate

- **y** - Coordinate
- **guid** - Unique identifiers
- **addr** - address
- **phone** - phones

getDescriptionByPoi_ID

Input parameters

poi_id - POI ID (*see in search*)

lng - Language

guid – Your personal guid key

Output

JSON

- **success** - The success of query fulfill returns to the value true/false
- **name** - Name of object
- **text** - Description of object
- **type** - type of object
- **subtype** - subtype of object
- **street** - street
- **country** - country
- **countrydistrict** - country district
- **settlement** - settlement
- **settlementdistrict** - settlement district
- **postal** - postal
- **phone** - phone
- **fax** - fax
- **web_site** - web site
- **house** - house
- **email** - email

getDescriptionByPoi_GUID

Input parameters

poi_guid - POI GUID (*see in search*)

lng - Language

guid – Your personal guid key

Output

JSON

- **success** - The success of query fulfill returns to the value true/false
- **name** - Name of object

- **text** - Description of object
- **type** - type of object
- **subtype** - subtype of object
- **street** - street
- **country** - country
- **countrydistrict** - country district
- **settlement** - settlement
- **settlementdistrict** - settlement district
- **postal** - postal
- **phone** - phone
- **fax** - fax
- **web_site** - web site
- **house** - house
- **email** - email

getImageByPoi_GUID

Input parameters

poi_guid - POI GUID (*see in search*)

guid – Your personal guid key

Output

JSON

- **success** - The success of query fulfill returns to the value true/false
- **Image**- Collection of the objects
 - **nm** - name
 - **ImageURL** - URL of Image