



Geis Point SOAP Webservice Interface

Interface for insertion of Geis point into an order
at e-shop

Version 20180919

Valid since 05.12.2016

Content

General information	3
Geis Point Plugin WSDL	3
Description of functions	3
1) getRegions function	3
2) getCities function	5
3) getGPDetail function	6
4) searchGP function	8

General information

Geis Point Plugin WSDL

This is an webservices interface, which uses JSON format interface to return WSDL data based on input data. This data format can then be easily processed in PHP, ASP or Java languages and/or in iOS or Android applications.

WSDL URL: <http://plugin.geispoint.cz/>

At this address, one can find the definition of WSDL, list of functions and input parameters

Description of particular functions, input and output data and error conditions follows.

Description of functions

1) getRegions(countryCode string) function

This function returns lists of available regions of Geis points.

Input data

countryCode (string) – country code; **CZ** for Czech Republic

Output data

JSON object field:

a) if records are found:

idRegion – ID of region

name – name of region

b) in case of an error:

error – description of error, list of possible error conditions:

– Invalid country code. Use “CZ” or “SK” instead.

Example of input data

```
getRegions('CZ')
```

Example of output data (object field) with found results (after decoding of JSON data)

```
[0] => stdClass  
Object
```

```
(  
  [idRegion] => 19  
  [name] => Hl. m. Praha  
)
```

```
[1] => stdClass  
Object
```

```
(  
  [idRegion] => 27  
  [name] => Středočeský kraj  
)
```

```
...
```

Example of output data (object field) in case of an error (after decoding of JSON data)

```
[0] => stdClass  
Object
```

```
(  
  [error] => Invalid country code. Use "CZ" or "SK" instead.  
)
```

2) getCities(string countryCode, int idRegion) function

This function returns lists of available cities of Geis points.

Input data

countryCode (string) – country code; **CZ** for Czech Republic

idRegion (int) – ID of region returned by function getRegions()

Output data

JSON object field:

a) if records are found:

idRegion – ID of region

name – name of city

b) in case of an error:

error – description of error, list of possible error conditions:

- Invalid country code. Use “CZ” or “SK” instead.
- Region code must be an integer.
- Region code {entered region code} doesn't exist for the country {entered country code}.

Example of input data

```
getCities('CZ',19)
```

Example of output data (object field) with found results (after decoding of JSON data)

```
[0] => stdClass  
Object
```

```
(  
  [idRegion] => 19  
  [city] => Praha 1  
)
```

```
[1] => stdClass  
Object
```

```
(  
  [idRegion] => 19  
  [city] => Praha 10  
)
```

```
...
```

Example of output data (object field) in case of an error (after decoding of JSON data)

[0] => stdClass
Object

```
(  
  [error] => Invalid country code  
)
```

3) getGPDetail(string id_gp) function

This function returns detailed information about an Geis point (GP).

Input data

idGP (string) – ID of Geis point (GP), e.g. GP17015001

Output data

JSON object field:

a) if records are found:

idGP – ID of Geis point

idRegion – ID of region

name – name

city – city

street – street

postcode – ZIP

country – country

email – email

phone – phone

openiningHours – opening hours

holiday – holiday

mapUrl – map reference

gpsn – GPS latitude

gpse – GPS longitude

photoURL – URL with photograph of Geis point

note – remark

b) in case of an error:

error – description of error, list of possible error conditions:

- No Geis Point ID given.
- Given Geis Point ID doesn't exist.

Example of input data

```
getGPDetail('GP17015001')
```

Example of output data (object field) with found results (after decoding of JSON data)

```
[0] => stdClass  
Object
```

```
(  
  [idGP] => GP17015001  
  [idRegion] => 132  
  [name] => TRAFIKA,SKLO,PORCELÁN  
  [city] => Kopřivnice  
  [street] => Štefánikova 1239  
  [postcode] => 74221  
  [country] => ČR  
  [email] => maersro@post.cz  
  [phone] => 603 154 061  
  [openingHours] => Po-Pá 7:00-18:00, So 8:00-12:00  
  [holiday] =>  
  [mapUrl] => http://www.mapy.cz/s/6nOe  
  [gpsn] => 49.599178118658735  
  [gpse] => 18.146226100000035  
  [photoUrl] => http://data.e-shoppartner.cz/download/fotovydejny/58586169.jpg  
  [note] =>  
)
```

Example of output data (object field) in case of an error (after decoding of JSON data)

```
[0] => stdClass  
Object
```

```
(  
  [error] => No Geis Point ID given.  
)
```

4) searchGP(string countryCode, string postcode, string city, string idGP) function

This function searches for Geis points based on input parameters: ZIP, city or ID of Geis point (GP). Input parameters are linked using OR function; if you only enter State, all records for given state will be returned.

Input data

countryCode (string) – country code; **CZ** for Czech Republic

city (string) – name of city returned by getCities() function

postcode (string) – ZIP of requested location

idGP (string) – ID of Geis point (GP), e.g. GP17015001

Output data

JSON object field:

a) if records are found:

idGP – ID of Geis point

idRegion – ID of region

name – name

city – city

street – street

postcode – ZIP

country – country

email – email

phone – phone

openingHours – opening hours

holiday – holiday

mapUrl – map reference

gpsn – GPS latitude

gpse – GPS longitude

photoURL – URL with photograph of Geis point

note – remark

b) in case of an error:

error – description of error, list of possible error conditions:

– No corresponding results for given search parameters

– Use at least country code and one more parameter.

Example of input data

searchGP('CZ') .. returns all GPs for **Czech Republic**

searchGP('CZ', 'Kopřivnice') .. returns all GPs in the vicinity of **Kopřivnice**

searchGP('CZ', ' ', '74221') .. returns all GPs in the vicinity of city with ZIP **742 21**

searchGP('CZ', ' ', ' ', 'GP17015001 ') .. returns **GP with id GP17015001**

searchGP('CZ', 'Kopřivnice', '74221', 'GP17015001 ') .. returns **GP with id GP17015001**

Example of output data (object field) with found results (after decoding of JSON data)

[0] => stdClass

Object

```
(  
  [idGP] => GP17015001  
  [idRegion] => 132  
  [name] => TRAFIKA,SKLO,PORCELÁN  
  [city] => Kopřivnice  
  [street] => Štefánikova 1239  
  [postcode] => 74221  
  [country] => ČR  
  [email] => maersro@post.cz  
  [phone] => 603 154 061  
  [openingHours] => Po-Pá 7:00-18:00, So 8:00-12:00  
  [holiday] =>  
  [mapUrl] => http://www.mapy.cz/s/6nOe  
  [gpsn] => 49.599178118658735  
  [gpse] => 18.146226100000035  
  [photoUrl] => http://data.e-shoppartner.cz/download/fotovydejny/58586169.jpg  
  [note] =>  
)
```

Example of output data (object field) in case of no record (after decoding of JSON data)

[0] => stdClass

Object

```
(  
  [error] => No corresponding results for given search parameters.  
)
```