BC Physical Address Online Geocoder Javascript API v1.1 Specification

July 23, 2013


BRITISH
COLUMBIA


DataBC

Introduction

The BC Physical Address Online Geocoder Javascript API, available as http://apps.gov.bc.ca/pub/geocoder/js/geocode.js, helps to build geocode request URLs based on the request parameters. The user can decide what to do with the resulting URL, either show the contents in the browser or download it using the AJAX library of their choice and interpreting the results as appropriate.

For simplicity, the examples in this document make use of the following variable which represents the URL of the bgeo geocoder application:

var URL = "http://apps.gov.bc.ca/pub/geocoder"

Also, the complete list of supported output formats used as part of the .setOutputFormat(outputFormat) methods are:

csv, geojson, gml, kml, shpz, xhtml

GeocodeRequest

The GeocodeRequest javascript object is used to perform a geocode of an input address. Here is an example of how it could be used:

```
req = new GeocodeRequest(URL);
req.setOutputFormat("xhtml");
req.setMaxResults(10);
req.setAddress("1207 Douglas St, Victoria, BC");
window.location = req.getURL();
```

It is not required to set all parameters of the request; unset parameters will use the default values. However, an address to geocode must be provided, either in the free-form "address" parameter, or in one or more of the structured address fields such as streetName or locality.

Methods

Here is a list of the methods available from the GeocodeRequest object.

`GeocodeRequest(baseUrl)`
This method is the constructor for the GeocoderRequest object. The baseUrl parameter is the url of the bgeo geocoder application.

.setOutputFormat(outputFormat)
This method specifies the format of the return geocoding results. Valid outputFormat parameters are listed at the top of this document.

.setSetBack(setBack)
This method specifies the set-back distance from the road to be used when calculating locations. This is a number, in metres.

.setMinScore(minScore)
This method specifies the minimum acceptable score of returned results. Results with scores lower than this will not be returned. The score value ranges from 0-100; a minScore of 0 returns all results.

.setMaxResults(maxResults)
This method specifies the maximum number of results to return. This is applied after the minScore requirement. It is intended to prevent unnecessary overhead while waiting for a large, complete list of results to return when only a short list of the best matches is required. Setting maxResults to 0 will return the maximum number of results allowable by the system.

.setInterpolation(interpolation)
This method specifies the interpolation method for the geocoder to use. The valid interpolation values are “adaptive”, “linear” and “none”. Linear indicates standard block-face proportional interpolation. A setting of adaptive indicates that existing sites locations, if suitable, should be used to interpolate an intermediate address instead of interpolating off the block face. None indicates that no interpolation is to be performed – only an exact site match is an acceptable response.

.setLocationDescriptor(locationDescriptor)
This method specifies the preferred location descriptor for the output. Allowed values are “any”, “accessPoint”, “frontDoorPoint”, “parcelPoint”, “rooftopPoint”, “routingPoint”. Default value is “any”.

.setEcho(echo)
This method specifies whether to echo back unmatched address information in the results. For example, with echo enabled, an apartment number or site name specified in the address query, but which is not present in the database, is still included in the results. The valid echo values are “true” and “false”.

.setOutputSRS(outputSRS)
This method specifies the Spatial Referencing System (SRS) to use for the locations in the results. Many SRS's are supported but it only makes sense to use an SRS which is valid for area in which the results are. SRS's are specified using their EPSG code, an integer generally in the range of 1-1,000,000. Common examples are 3005 (BC Albers) and 4326 (Geographic) which is the default.

.setAddress(address)
.setSiteName(siteName)
.setUnitDesignator(unitDesignator)
.setUnitNumber(unitNumber)
.setUnitNumberSuffix(unitNumberSuffix)
.setCivicNumber(civicNumber)
.setCivicNumberSuffix(civicNumberSuffix)
.setStreetName(streetName)
.setStreetType(streetType)
.setStreetDirection(streetDirection)
.setStreetQualifier(streetQualifier)
.setLocality(locality)
.setProvince(province)

These methods specify the address to be geocoded. Either the entire address should be specified, as a single free-form address string in the address parameter, or the address components should be provided in the various other parameters. If any component of the address is unknown, or does not apply, it can be left out.

.getURL()

This method returns the URL from which to retrieve the results of the GeocodeRequest, as specified by previously called methods on the GeocodeRequest object. This URL can simply be used to redirect the browser, or can be used with an AJAX library such as jQuery to fetch and process in javascript.

.readForm(theForm)

This method reads all of the values for the geocodeRequest parameters from a form. The form fields must be named the same as the request parameters, and must be either simple text fields or select boxes with the correct values set.

siteIdRequest

The siteIdRequeset javascript object is used to query the details of a site based on an input site identifier. Here is an example of how it could be used:

```
req = new siteIdRequest(URL);
req.setOutputFormat("csv");
```

```
req.setSiteId("4d7e3b3e-bc01-4fcb-88db-0616f6765ba9");
window.location = req.getURL();
```

Methods

Here is a list of the methods available from the siteIdRequest object.

`siteIdRequest(baseUrl)`
This method is the constructor for the siteIdRequest object. The baseUrl parameter is the url of the bgeo geocoder application.

`.setSiteId(siteId)`
This method is used to specify the site identifier for site retrieval requests. The siteId parameter is a UUID string with hyphens separating the five hexadecimal digit groups.

`.setOutputSRS(outputSRS)`
This method specifies the Spatial Referencing System (SRS) to use for the locations in the results. Many SRS's are supported but it only makes sense to use an SRS which is valid for area in which the results are. SRS's are specified using their EPSG code, an integer generally in the range of 1-1,000,000. Common examples are 3005 (BC Albers) and 4326 (Geographic) which is the default.

`.setOutputFormat(outputFormat)`
This method specifies the format of the return geocoding results. Valid outputFormat parameters are listed at the top of this document.

`.getURL()`
This method returns the URL from which to retrieve the results of the siteIdRequest, as specified by previously called methods on the siteIdRequest object. This URL can simply be used to redirect the browser, or can be used with an AJAX library such as jQuery to fetch and process in javascript.

`.readForm(theForm)`
This method reads all of the values for the siteIdRequest parameters from a form. The form fields must be named the same as the request parameters, and must be either simple text fields or select boxes with the correct values set.

nearestSiteRequest

The nearestSiteRequest javascript object is used to locate the closest site to an input point coordinate, provided one exists within a maximum distance tolerance. Here is an example of it could be used:

```
req = new nearestSiteRequest(URL);
req.setOutputFormat("json");
req.setPoint("-123.3649883, 48.4254821");
window.location = req.getURL();
```

Methods

Here is a list of the methods available from the nearestSiteRequest object.

nearestSiteRequest(baseUrl)
This method is the constructor for the nearestSiteRequest object. The baseUrl parameter is the url of the bgeo geocoder application.

.setPoint(point)
This method sets the coordinate for use in specifying a target location to be used for locating the spatially closest site. Format for the point parameter is: longitude, latitude.

.setOutputSRS(outputSRS)
This method specifies the Spatial Referencing System (SRS) to use for the locations in the results. Many SRS's are supported but it only makes sense to use an SRS which is valid for area in which the results are. SRS's are specified using their EPSG code, an integer generally in the range of 1-1,000,000. Common examples are 3005 (BC Albers) and 4326 (Geographic) which is the default.

.setOutputFormat(outputFormat)
This method specifies the format of the return geocoding results. Valid outputFormat parameters are listed at the top of this document.

.getURL()
This method returns the URL from which to retrieve the results of the nearestSiteRequest, as specified by previously called methods on the nearestSiteRequest object. This URL can simply be used to redirect the browser, or can be used with an AJAX library such as jQuery to fetch and process in javascript.

.readForm(theForm)
This method reads all of the values for the nearestSiteRequest parameters from a form. The form fields must be named the same as the request parameters, and must be either simple text fields or select boxes with the correct values set.

sitesWithinRequest

The sitesWithinRequest javascript object is used to locate sites that spatially fall within an input bounding box. Result sets are limited by tolerance parameters (e.g. maximum perimeter, maximum results) to ensure the number of sites returned isn't too excessive. Here is an example of how the object could be used:

```
req = new sitesWithinRequest(URL);
req.setOutputFormat("shpz");
req.setBbox("-123.366, 48.424, -123.362, 48.427");
window.location = req.getURL();
```

Methods

Here is a list of the methods available from the sitesWithinRequest object.

sitesWithinRequest(baseUrl)
This method is the constructor for the sitesWithinRequest object. The baseUrl parameter is the url of the bgeo geocoder application.

.setBbox(bbox)
This method specifies the input bounding box (rectangular geographic area) within which site points are to be spatially queried. The bbox parameter is composed from a pair of coordinates in the form: longitude minimum, latitude minimum, longitude maximum, latitude maximum.

.setOutputSRS(outputSRS)
This method specifies the Spatial Referencing System (SRS) to use for the locations in the results. Many SRS's are supported but it only makes sense to use an SRS which is valid for area in which the results are. SRS's are specified using their EPSG code, an integer generally in the range of 1-1,000,000. Common examples are 3005 (BC Albers) and 4326 (Geographic) which is the default.

.setOutputFormat(outputFormat)
This method specifies the format of the return geocoding results. Valid outputFormat parameters are listed at the top of this document.

.getURL()
This method returns the URL from which to retrieve the results of the sitesWithinRequest, as specified by previously called methods on the sitesWithinRequest object. This URL can simply be used to redirect the browser, or can be used with an AJAX library such as jQuery to fetch and process in javascript.

.readForm(theForm)
This method reads all of the values for the sitesWithinRequest parameters from a form. The form fields must be named the same as the request parameters, and must be either simple text fields or select boxes with the correct values set.

intersectionIdRequest

The intersectionIdRequest javascript object is used to query the details of an intersection based on an input intersection identifier. Here is an example of how it could be used:

```
req = new intersectionIdRequest(URL);
req.setOutputFormat("gml");
req.setIntersectionId("260ece4f-c3d8-440f-aa8e-7c67112957f0");
window.location = req.getURL();
```

Methods

Here is a list of the methods available from the intersectionIdRequest object.

intersectionIdRequest(baseUrl)
This method is the constructor for the intersectionIdRequest object. The baseUrl parameter is the url of the bgeo geocoder application.

.setIntersectionId(intersectionId)
This method is used to specify the intersection identifier for intersection retrieval requests. The intersectionId parameter is a UUID string with hyphens separating the five hexadecimal digit groups.

.setOutputSRS(outputSRS)
This method specifies the Spatial Referencing System (SRS) to use for the locations in the results. Many SRS's are supported but it only makes sense to use an SRS which is valid for area in which the results are. SRS's are specified using their EPSG code, an integer generally in the range of 1-1,000,000. Common examples are 3005 (BC Albers) and 4326 (Geographic) which is the default.

.setOutputFormat(outputFormat)
This method specifies the format of the return geocoding results. Valid outputFormat parameters are listed at the top of this document.

.getURL()
This method returns the URL from which to retrieve the results of the intersectionIdRequest, as specified by previously called methods on the intersectionIdRequest object. This URL can simply be used to redirect the browser, or can be used with an AJAX library such as jQuery to fetch and process in javascript.

.readForm(theForm)
This method reads all of the values for the intersectionIdRequest parameters from a form. The form fields must be named the same as the request parameters, and must be either simple text fields or select boxes with the correct values set.

nearestIntersectionRequest

The nearestIntersectionRequest javascript object is used to locate the closest intersection to an input point coordinate, provided one exists within a maximum distance tolerance. Here is an example of it could be used:

```
req = new nearestIntersectionRequest(URL);
req.setOutputFormat("kml");
req.setPoint("-123.3649883, 48.4254821");
window.location = req.getURL();
```

Methods

Here is a list of the methods available from the nearestIntersectionRequest object.

nearestIntersectionRequest(baseUrl)
This method is the constructor for the nearestIntersectionRequest object. The baseUrl parameter is the url of the bgeo geocoder application.

.setPoint(point)
This method sets the coordinate for use in specifying a target location to be used for locating the spatially closest intersection. Format for the point parameter is: longitude, latitude.

.setOutputSRS(outputSRS)
This method specifies the Spatial Referencing System (SRS) to use for the locations in the results. Many SRS's are supported but it only makes sense to use an SRS which is valid for area in which the results are. SRS's are specified using their EPSG code, an integer generally in the range of 1-1,000,000. Common examples are 3005 (BC Albers) and 4326 (Geographic) which is the default.

.setOutputFormat(outputFormat)
This method specifies the format of the return geocoding results. Valid outputFormat parameters are listed at the top of this document.

.getURL()
This method returns the URL from which to retrieve the results of the nearestIntersectionRequest, as specified by previously called methods on the nearestIntersectionRequest object. This URL can simply be used to redirect the browser, or can be used with an AJAX library such as jQuery to fetch and process in javascript.

.readForm(theForm)
This method reads all of the values for the nearestIntersectionRequest parameters from a form. The form fields must be named the same as the request parameters, and must be either simple text fields or select boxes with the correct values set.

intersectionWithinRequest

intersectionsWithinRequest

The intersectionsWithinRequest javascript object is used to locate sites that spatially fall within an input bounding box. Result sets are limited by tolerance parameters (e.g. maximum perimeter, maximum results) to ensure the number of sites returned isn't too excessive. Here is an example of how the object could be used:

```
req = new intersectionsWithinRequest(URL);
req.setOutputFormat("geojson");
req.setBbox("-123.366, 48.424, -123.362, 48.427");
window.location = req.getURL();
```

Methods

Here is a list of the methods available from the intersectionsWithinRequest object.

`intersectionsWithinRequest(baseUrl)`
This method is the constructor for the intersectionsWithinRequest object. The baseUrl parameter is the url of the bgeo geocoder application.

`.setBbox(bbox)`
This method specifies the input bounding box (rectangular geographic area) within which intersection points are to be spatially queried. The bbox parameter is composed from a pair of coordinates in the form: longitude minimum, latitude minimum, longitude maximum, latitude maximum.

`.setOutputSRS(outputSRS)`
This method specifies the Spatial Referencing System (SRS) to use for the locations in the results. Many SRS's are supported but it only makes sense to use an SRS which is valid for area in which the results are. SRS's are specified using their EPSG code, an integer generally in the range of 1-1,000,000. Common examples are 3005 (BC Albers) and 4326 (Geographic) which is the default.

`.setOutputFormat(outputFormat)`
This method specifies the format of the return geocoding results. Valid outputFormat parameters are listed at the top of this document.

`.getURL()`
This method returns the URL from which to retrieve the results of the intersectionsWithinRequest, as specified by previously called methods on the intersectionsWithinRequest object. This URL can simply be used to redirect the browser, or can be used with an AJAX library such as jQuery to fetch and process in javascript.

`.readForm(theForm)`
This method reads all of the values for the intersectionsWithinRequest parameters from a form. The form fields must be named the same as the request parameters, and must be either simple text fields or

select boxes with the correct values set.