Open Geospatial Consortium Inc.

Date:   2012-06-22

Reference number of this OGC® project document:    OGC 12-061r1

Version: 0.0.4

Category: OGC® Implementation Specification

Editor:   Clemens Portele, Satish Sankaran

GeoServices REST API — Part 8: Geocoding Service

**Copyright notice**

Copyright © 2012 Open Geospatial Consortium, Inc. All Rights Reserved.  
To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

**Warning**

This document is not an OGC Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard.

Document type:    OGC® Publicly Available Standard

Document subtype:    n/a

Document stage:    Draft

Document language:  English

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Draft copyright:

Copyright: Esri Inc.

The companies listed above have granted the Open Geospatial Consortium, Inc. (OGC) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version.

Contents

1 Scope 1

2 Conformance 1

3 References 2

4 Terms and Definitions 2

5 Conventions 3

6 Geocoding Service overview 3

7 Geocoding Service Core 4

7.1 Overview 4

7.2 Geocoding Service Root 4

7.2.1 Geocoding Service Root URI 4

7.2.2 Geocoding Service Root resources 5

7.2.3 Example 6

7.3 Find Address Candidates 6

7.3.1 Overview 6

7.3.2 Find Address Candidates URI 7

7.3.3 Address Candidates resources 8

7.3.4 Example 9

7.4 Reverse Geocode 9

7.4.1 Overview 9

7.4.2 Reverse Geocode URI 10

7.4.3 Address resources 11

7.4.4 Example 12

A.1 Conformance class: gcservice 13

A.1.1 Test: gcservice/root 13

A.1.2 Test: gcservice/find 13

A.1.3 Test: gcservice/rev 14

Preface

The “Esri GeoServices REST Specification Version 1.0” was originally developed by Esri to provide interoperability between ArcGIS Server and the broader information technology community. The Esri specification had been widely implemented by Esri users and business partners over 4 years. In 2010 it was released as a non-proprietary open specification and has been implemented by developers outside of the Esri user community.

In 2011, Esri has offered the GeoServices REST API for consideration to become an OGC standard. An OGC Standards Working Group was formed to document the specification in conformance with the modular specification policy of the OGC and to address comments received from the OGC membership and during the public review.

This candidate standard is designed to be implemented without the use of Esri products.

Submitting organizations

The following organizations submitted this Implementation Specification to the Open Geospatial Consortium Inc.:

Esri Inc.

interactive instruments GmbH

Oracle USA

52°North

Submission contact points

All questions regarding this submission should be directed to the editor or the submitters:

|  |  |
| --- | --- |
| CONTACT | COMPANY |
| 1. Satish Sankaran | 1. Esri Inc. |
| 1. Keith Ryden | 1. Esri Inc. |
| 1. Clemens Portele | 1. interactive instruments GmbH |
| 1. John Herring | 1. Oracle USA |
| 1. Andreas Wytzisk | 1. 52° North |

Revision history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Release | Author | Paragraph modified | Description |
| 1. 23 Jan 2012 | 1. 0.0.1 | 1. CP, SS | 1. All | 1. First draft |
| 1. 22 May 2012 | 1. 0.0.2 | 1. CP | 1. All | 1. Include changes identified in review process |
| 1. 31 May 2012 | 1. 0.0.3 | 1. CP |  | 1. RFC package for OAB review |
| 1. 22 Jun 2012 | 1. 0.0.4 | 1. CP |  | 1. References updated 2. Editorial changes 3. Incomplete text revised |

Changes to the OGC® Abstract Specification

The OGC**®** Abstract Specification does not require changes to accommodate this OGC**®** standard.

Versioning Rules

See the “Versioning Rules” section in OGC document 12-054r1, GeoServices REST API – Part 1: Core.

Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. Open Geospatial Consortium Inc. shall not be held responsible for identifying any or all such patent rights. However, to date, no such rights have been claimed or identified.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.

This document is part 8 of the GeoServices REST API series:

Part 1: Core

Part 2: Catalog

Part 3: Map Service

Part 4: Feature Service

Part 5: Geometry Service

Part 6: Image Service

Part 7: Geoprocessing Service

Part 8: Geocoding Service

The relationship with other parts of the OGC standards baseline is described in document 12-062r1.

# Scope

The GeoServices REST API provides a standard way for web clients to communicate with geographic information system (GIS) servers based on Representational State Transfer (REST) principles. Clients issue requests to the resources on the server identified by structured URLs. The server responds with map images, text-based geographic information, or other representations of resources that satisfy the request.

This document specifies the geocoding resources in an implementation of the GeoServices REST API and extends the GeoServices REST API – Core standard.

# Conformance

Conformance with this standard shall be checked using all the relevant tests specified in Annex A (normative) of this document. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing web site[[1]](#footnote-1).

This Standard establishes one requirements class and a corresponding conformance class, extending the core conformance class of the GeoServices REST API series.

All requirements-classes and conformance-classes described in this document are owned by the standard identified as **http://www.opengis.net/spec/gsr-gcs/1.0**. Requirements and conformance test URIs defined in this document are relative to this URI unless they start with "http://" and are absolute URIs.

Any implementation claiming conformance with a conformance class shall pass all the tests in the associated abstract test suite. Table 1 summarizes the requirements and conformance tests associated per conformance class.

Table 1 – Conformance class summary

|  |  |  |
| --- | --- | --- |
| **gcservice** | **Title** | Geocoding Service Core |
| **Standardization target type** | Web service |
| **Dependencies** | **http://www.opengis.net/spec/gsr/1.0/conf/core**  **http://www.opengis.net/spec/gsr/1.0/conf/geometry** |
| **Requirements** | All requirements in Clause 7 |
| **Conformance tests** | Annex A.1 |

Figure 1 – Single conformance class

# References

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

GeoServices REST API – Core, Version 1.0 (2012), OGC document 12-054r1

# Terms and Definitions

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r9], which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

# Conventions

See Clause 5 in the GeoServices REST API – Core document.

# Geocoding Service overview

Geocoding is the process of assigning a location, usually in the form of coordinate values (points), to an address by comparing the descriptive location elements in the address to those present in the reference material. Addresses come in many forms, ranging from the common address format of a house number followed by the street name to other location descriptions such as postal zone or census tract. An address includes any type of information that distinguishes a place.

The GeoServices REST API Geocoding Service provides a web service that performs geocoding. The root resource provides basic information associated with the service such as the service description, the address fields, spatial reference, and locator properties.

The Geocoding Service supports two operations via controller resources:

* Find Address Candidates: Returns a list of candidates based on address and location
* Reverse Geocode: Returns information about all the address fields pertaining to the reverse geocoded address, as well as the address's exact location

The following figure provides an overview of the resources in a Geocoding Service. Resources in green color are controller resources (also called "operations") that query the underlying layers/tables and create resources that are not persistently stored on the server and made available with their own URI, but returned in the response from the controller resource. These resources are shown in white color.

Figure 2 – Resource overview

# Geocoding Service Core

## Overview

An implementation of the GeoServices REST API Geocoding Service Core provides capabilities that are needed by most applications using geocoding services. Currently no additional capabilities are specified in additional conformance classes that depend on this core.

Table 2 – Geocoding Service Core overview

|  |  |  |
| --- | --- | --- |
| **Resource** | **Parameters** | **Resource representation** |
| Geocoding Service Root | f=json | JSON representation valid  All JSON schema elements supported |
| Find Address Candidates | f=json  outfields  outSR  *address field names listed in the addressFields property of the Geocoding Service Root resource* | JSON representation valid  All JSON schema elements supported |
| Reverse Geocode | f=json  location  distance  outSR | JSON representation valid  All JSON schema elements supported |

## Geocoding Service Root

### Geocoding Service Root URI

In the following URI templates, these variables are used:

* gcServiceRootURI: the URL of the service

If the Geocoding Service is referenced from a Catalog Service, gcServiceRootURI is the same as

{catServiceRootURI}/{gcServiceName}/GeocodeServer

Table 3 – Geocoding Service Root reference

|  |  |
| --- | --- |
| **URI template** | {+gcServiceRootURI}{?f} |
| **HTTP methods** | GET |
| **Parent Resource Type** | - |
| **Child Resource Types** | Find Address Candidates  Reverse Geocode |

Table 4 – Geocoding Service Root parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Details** | |
| f | The response format. | |
| Required | Yes |
| Syntax | "json" |
| Example | f=json |

**Request Requirements**

|  |
| --- |
| * + 1. The Geocoding Service Root resource SHALL accept requests that conform to the URI template in Table 3 and use any HTTP method identified in the same table.   gcservice/request |

|  |
| --- |
| * + 1. The Geocoding Service Root resource SHALL support all parameters and values specified in Table 4.   gcservice/parameters |

### Geocoding Service Root resources

|  |
| --- |
| * + 1. The JSON representation of a Geocoding Service Root resource SHALL validate against the JSON Schema **http://schemas.opengis.net/gsr-gcs/1.0/root.json** or in case of an exception against JSON Schema http://schemas.opengis.net/gsr/1.0/exception.json.   gcservice/valid |

The Geocoding Service Root resource lists three types of fields:

* field names of address resources supported by the service (these may also be used to search addresses)
* field names reported for candidate addresses that are not intersections
* field names reported for candidate addresses that are intersections

### Example

URL for the geocoding web service GeocodeUSA:

http://example.com/rest/services/GeocodeUSA/GeocodeServer?f=json

**Request**

GET /rest/services/GeocodeUSA/GeocodeServer?f=json HTTP/1.1

Host: example.com

**Response**

HTTP/1.1 200 OK

Content-Type: application/json

Content-Length: nnn

{

"serviceDescription" : "Test Geocode Service Description",

"addressFields": [

{"name" : "Street", "alias" : "Street or Intersection", "required" : true, "type" : "FieldTypeString"},

{"name" : "Zone", "alias" : "Zip Code", "required" : false, "type" : "FieldTypeString"}

],

"singleLineAddressField" : {"name" : "Single Line Input","type" : "FieldTypeString","alias" : "Full Address","required" : false}

,

"candidateFields": [

{"name" : "Score", "alias" : "", "type" : "FieldTypeSmallInteger"},

{"name" : "StreetName", "alias" : "", "type" : "FieldTypeString"}

],

"intersectionCandidateFields": [

{"name" : "Score", "alias" : "", "type" : "FieldTypeSmallInteger"},

{"name" : "StreetName1", "alias" : "", "type" : "FieldTypeString"}

],

"spatialReference" : {"wkid" : 4326},

"locatorProperties": {

"MinimumCandidateScore" : "10",

"SideOffsetUnits" : "ReferenceDataUnits",

"SpellingSensitivity" : "80",

"MinimumMatchScore" : "60",

"IntersectionConnectors" : "& | @"

}

}

## Find Address Candidates

### Overview

The Find Address Candidates operation is performed on a controller resource of the geocoding service. The result of this operation provides a Address Candidates resource with information about candidates, including the address, location, and score. The resource is not stored on the server, it is returned directly in the response to the request.

Users provide arguments to the export operation as query parameters.

### Find Address Candidates URI

In the following URI templates, these variables are used:

* gcServiceURI: URL of a Geocoding Service Root resource without any parameter

Table 5 – Find Address Candidates reference

|  |  |
| --- | --- |
| **URI template** | {+gcServiceURI}/findAddressCandidates{?f,OUTFIELDS,OUTSR} |
| **HTTP methods** | GET  POST (application/x-www-form-urlencoded) |
| **Parent Resource** | Geocoding Service Root |

Table 6 – Find Address Candidates parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Details** | |
| f | The response format. | |
| Required | Yes |
| Syntax | "json" / "image" |
| Example | f=json |
| outfields | The list of fields to be included in the returned result set. This is a comma-delimited list of field names.  If the shape field is specified in the list of return fields, it is ignored.  For nonintersection addresses, specify the candidate fields from the Geocode Service resource. For intersection addresses, specify the intersection candidate fields from the Geocode Service resource.  outFields=\* returns all fields. | |
| Required | No. Default: all fields. |
| Syntax | NAME \*("," NAME) / "\*" |
| Example | outFields=StreetName,StreetType |
| outSR | The well-known ID of the spatial reference or a spatial reference JSON object for the returned address candidates. | |
| Required | No. Default: the spatial reference provided in the Geocoding Service Root resource is used. |
| Syntax | POSINT / JSON |
| Example | outSR=4326 |
| *address field names listed in the addressFields property of the Geocoding Service Root resource* | The various address fields accepted by the corresponding geocode service. | |
| Required | No. Default: No filter on the address field. |
| Syntax | STRING |
| Example | Suppose that addressFields of a Geocode Service Root resource includes fields with the following names: Street, City, State, and Zone. To perform the Find Address Candidates operation by providing values for Street and Zone, set the query parameters as such:  Street=380+New+York+St&Zone=92373 |

**Request Requirements**

|  |
| --- |
| * + 1. The Find Address Candidates resource SHALL accept requests that conform to the URI template in Table 5 and use any HTTP method identified in the same table.   gcservice/findRequest |

|  |
| --- |
| * + 1. The Find Address Candidates resource SHALL support all parameters and values specified in Table 6.   gcservice/findParameters |

### Address Candidates resources

|  |
| --- |
| * + 1. The JSON representation of a response to a request on an Find Address Candidates resource SHALL validate against the JSON Schema **http://schemas.opengis.net/gsr-gcs/1.0/addresses.json** or in case of an exception against JSON Schema http://schemas.opengis.net/gsr/1.0/exception.json.   gcservice/findValid |

|  |
| --- |
| * + 1. The spatial reference in the Address Candidates resource SHALL be the spatial reference requested.   gcservice/findOutSR |

|  |
| --- |
| * + 1. The attributes of each candidate in the Address Candidates resource SHALL be the address fields requested.   gcservice/findAttributes |

### Example

Geocode an address (380 New York Street, Redlands, CA 92373):

http://example.com/rest/services/GeocodeUSA/GeocodeServer/findAddressCandidates?f=json&Address=380+New+York+Street&City=Redlands&State=CA&Zip=92373

**Request**

GET /rest/services/GeocodeUSA/GeocodeServer/findAddressCandidates?f=json&Address=380+New+York+Street&City=Redlands&State=CA&Zip=92373 HTTP/1.1

Host: example.com

**Response**

HTTP/1.1 200 OK

Content-Type: application/json

Content-Length: nnn

{

"spatialReference": {"wkid" : 4326},

"candidates" : [

{

"address" : "1 MASON ST",

"location" : { "x" : -122.408951, "y" : 37.783206 },

"score" : 75,

"attributes" : {"StreetName" : "MASON", "StreetType" : "ST"}

},

{

"address" : "49 MASON ST",

"location" : { "x" : -122.408986, "y" : 37.783460 },

"score" : 27,

"attributes" : {"StreetName" : "MASON", "StreetType" : "ST"}

}

]

}

## Reverse Geocode

### Overview

The Reverse Geocode operation is performed on a controller resource of the geocoding service. The result of this operation provides a Address resource with information about about all the address fields pertaining to the reverse geocoded address, as well as its exact location. The resource is not stored on the server, it is returned directly in the response to the request.

Users provide arguments to the export operation as query parameters.

### Reverse Geocode URI

In the following URI templates, these variables are used:

* gcServiceURI: URL of a Geocoding Service Root resource without any parameter

Table 7 – Reverse Geocode reference

|  |  |
| --- | --- |
| **URI template** | {+gcServiceURI}/reverseGeocode{?f,location,distance,outSR} |
| **HTTP methods** | GET  POST (application/x-www-form-urlencoded) |
| **Parent Resource** | Geocoding Service Root |

Table 8 – Reverse Geocode parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Details** | |
| f | The response format. | |
| Required | Yes |
| Syntax | "json" / "image" |
| Example | f=json |
| location | The point at which to search for the closest address. The structure of the point is the same as the structure of the JSON point object returned by the GeoServices REST Specification. In addition to the JSON structure, the location can be specified with a simple comma-separated syntax.  If not specified in the JSON object or if using the simple comma-separated syntax, the location is assumed to be in the same spatial reference as that of the geocode service. | |
| Required | Yes |
| Syntax | JSON / X "," Y |
| Example | location={"x":-122.4,"y":37.7}  location=-122.4,37.7 |
| distance | The distance in meters from the given location within which a matching address should be searched. | |
| Required | No. Default: "0". |
| Syntax | NUMBER |
| Example | distance=100 |
| outSR | The well-known ID of the spatial reference or a spatial reference JSON object for the returned address. | |
| Required | No. Default: the spatial reference provided in the Geocoding Service Root resource is used. |
| Syntax | POSINT / JSON |
| Example | outSR=4269 |

**Request Requirements**

|  |
| --- |
| * + 1. The Reverse Geocode resource SHALL accept requests that conform to the URI template in Table 7 and use any HTTP method identified in the same table.   gcservice/revRequest |

|  |
| --- |
| * + 1. The Reverse Geocode resource SHALL support all parameters and values specified in Table 8.   gcservice/revParameters |

|  |
| --- |
| * + 1. If the location parameter uses a JSON syntax, it SHALL validate against the JSON Schema **http://schemas.opengis.net/gsr/1.0/point.json** .   gcservice/revLocation |

### Address resources

|  |
| --- |
| * + 1. The JSON representation of a response to a request on a Reverse Geocode resource SHALL validate against the JSON Schema **http://schemas.opengis.net/gsr-gcs/1.0/address.json** or in case of an exception against JSON Schema http://schemas.opengis.net/gsr/1.0/exception.json.   gcservice/revValid |

|  |
| --- |
| * + 1. The location of the Address resource SHALL be in the spatial reference requested and within the circle specified by the location and distance parameters.   gcservice/revLocationValid |

### Example

Reverse geocode based on a point:

http://example.com/rest/services/GeocodeUSA/GeocodeServer/reverseGeocode?location=-117.195681386,34.057517097&distance=0&f=json

**Request**

GET /rest/services/GeocodeUSA/GeocodeServer/reverseGeocode?location=-117.195681386,34.057517097&distance=0&f=json HTTP/1.1

Host: example.com

**Response**

HTTP/1.1 200 OK

Content-Type: application/json

Content-Length: nnn

{

"address" : {

"Street" : "771 TUNNEL AVE",

"Zone" : "94005"

},

"location" : { "x" : -122.400260954336, "y" : 37.7000445053795,"spatialReference": {"wkid": 4269} }

}

Annex A  
(normative)  
  
Abstract Test Suite

Conformance class: gcservice

* 1. Test: gcservice/root

|  |  |
| --- | --- |
| Requirements | **gcservice/request, gcservice/parameters, gcservice/valid** |
| Test purpose | Verify that the Geocoding Service Root resource supports the request and response requirements. |
| Test method | Set up a test service. Construct valid requests for the Geocoding Service Root resource.  Inspect the responses and validate them against the JSON Schema http://schemas.opengis.net/gsr-gcs/1.0/root.json or for exceptions against http://schemas.opengis.net/gsr/1.0/exception.json. |
| Test type | Capability |

* 1. Test: gcservice/find

|  |  |
| --- | --- |
| Requirements | **gcservice/findRequest, gcservice/findParameters, gcservice/findValid, gcservice/findOutSR, gcservice/findAttributes** |
| Test purpose | Verify that the Geocoding Service Root resource supports the request and response requirements. |
| Test method | Set up a test service. Construct valid requests for the Find Address resource.  Inspect the responses and validate them against the JSON Schema http://schemas.opengis.net/gsr-gcs/1.0/addresses.json or for exceptions against http://schemas.opengis.net/gsr/1.0/exception.json.  Inspect the Address Candidates resource and verify that   * the spatial reference is the spatial reference requested * the attributes of each candidate are the address fields requested |
| Test type | Capability |

* 1. Test: gcservice/rev

|  |  |
| --- | --- |
| Requirements | **gcservice/revRequest, gcservice/revParameters, gcservice/revLocation, gcservice/revValid, gcservice/revLocationValid** |
| Test purpose | Verify that the Geocoding Service Root resource supports the request and response requirements. |
| Test method | Set up a test service. Construct valid requests for the Reverse Geocode resource.  Inspect the responses and validate them against the JSON Schema http://schemas.opengis.net/gsr-gcs/1.0/address.json or for exceptions against http://schemas.opengis.net/gsr/1.0/exception.json.  Inspect the Address resource and verify that the location is in the spatial reference requested and within the circle specified by the location and distance parameters. |
| Test type | Capability |

1. [www.opengeospatial.org/cite](http://www.opengeospatial.org/cite) [↑](#footnote-ref-1)