



## *TM Forum Specification*

# Geographic Location Management API Conformance Profile

**TMF675B**  
**Release 17.5.0**  
**January 2018**

<b>Release: TM Forum Release 17.5.0</b>	<b>Document Status: Member Evaluation</b>
<b>Version: 1.0.2</b>	<b>IPR Mode: RAND</b>

## NOTICE

Copyright © TM Forum 2017. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to TM FORUM, except as needed for the purpose of developing any document or deliverable produced by a TM FORUM Collaboration Project Team (in which case the rules applicable to copyrights, as set forth in the [TM FORUM IPR Policy](#), must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by TM FORUM or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and TM FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Direct inquiries to the TM Forum office:

4 Century Drive, Suite 100

Parsippany, NJ 07054, USA

Tel No. +1 973 944 5100

Fax No. +1 973 944 5110

TM Forum Web Page: [www.tmforum.org](http://www.tmforum.org)

## TABLE OF CONTENTS

NOTICE.....	2
Table of Contents.....	3
List of Tables.....	4
Introduction .....	5
API DESCRIPTION.....	6
RESOURCE MODEL CONFORMANCE .....	7
Geographic Location API MANDATORY AND OPTIONAL RESOURCES.....	7
Geographic Location resource MANDATORY AND OPTIONAL ATTRIBUTES .....	7
API OPERATIONS CONFORMANCE .....	9
Geographic Location MANDATORY AND OPTIONAL OPERATIONS.....	9
API GET FILTERING OPERATION CONFORMANCE.....	10
Filtering in Geographic Location resource .....	10
GET /location/geographicLocation.....	10
GET /location /geographicLocation/{locationId}.....	11
API CONFORMANCE TEST SCENARIOS.....	12
Geographic Location resource TEST CASES .....	12
Acknowledgements.....	16
Release History.....	16
Contributors to Document.....	16

## LIST OF TABLES

N/A

## INTRODUCTION

The following document is the REST API Conformance for the Geographic Location API.

## API DESCRIPTION

This API covers the operations to manage (create, read, delete) geographic locations that can be associated to many types of entities.

This API defines a Geographic Location as a convenience class that allows to easily refer location(s) important to other entities.

This API allows the following operations

- Retrieve a list of geographic locations stored in a server filtered by a given criteria
- Retrieve a specific geographic location

## RESOURCE MODEL CONFORMANCE

### Geographic Location API MANDATORY AND OPTIONAL RESOURCES

Resource Name	Mandatory or Optional	Comments
GeographicLocation	M	Only GET operation
RetrieveGeographicLocation	O	
RetrieveLocationRelation	O	

### Geographic Location resource MANDATORY AND OPTIONAL ATTRIBUTES

Attribute Name	Mandatory or Optional	Comments
Id	M in response messages O otherwise	Only valid in request if the server allows the client to define the id for the resource to be created
href	M in response messages O otherwise	
name	M	
geometryType	M (if href not provided)	
accuracy	M	
spatialRef	M	
@type	O	

Attribute Name	Mandatory or Optional	Comments
@schemaLocation	O	
@baseType	O	
geometry	M (if href not provided)	Array of geographicPoints
x	M if geographicPoint included	
u	M if geographicPoint included	
z	O	



## API OPERATIONS CONFORMANCE

For every single resource use the following templates and define what operations are optional and what operations are mandatory.

### Geographic Location MANDATORY AND OPTIONAL OPERATIONS

Uniform API Operation	Mandatory/Optional	Comments
GET	M	GET must be used to retrieve a representation of a resource

## API GET FILTERING OPERATION CONFORMANCE

### Definitions

**Filtered Search:** A filtered search can be applied using query parameters in order to obtain only the resource entities that meet the criteria defined by the filtering parameters included in the query request. Several elements can be applied to the filtered search. In that case logic, a logical AND is applied to combine the criteria (e.g.:?name=<value> &owner.id=<value>)

**Filtered Response Data (Response Attribute selection):** In order to apply a filter and limit the number of attributes included in the response, the GET request can include the “?fields=” query parameter. Several elements can be applied to the filter. In that case, a logical AND is applied to combine the values (e.g.:?fields=name, description will provide in the response only the values assigned to attributes name and description). Attribute selection capabilities are the same for collections retrieval and individual resource queries

### Filtering in Geographic Location resource

Attribute name	Filtered search First Level	Filtered search N Level	Response Attribute Selection First Level	Response Attribute Selection N Level
Id	NA	NA	M	NA
href	NA	NA	M	NA
name	O	NA	M	NA
type	O	NA	M	NA

### GET /location/geographicLocation

**Filtered Search:** A filtered search can be applied using the following filtering criteria

- name: to obtain the locations with a given name
- type: to obtain the locations with a given type

**Filtered Response Data:** A filtered response can be requested for the following attributes using the “?fields=” query parameter

- Any of the attributes in the first level of GeographicLocation resource definition

GET /location /geographicLocation/{locationId}

**Filtered Response Data:** A filtered response can be requested for the following attributes using the “?fields=” query parameter

- Any of the attributes in the first level of Geographic Location resource definition

## API CONFORMANCE TEST SCENARIOS

This section describes the test scenarios required for the basic CONNECT certification of Geographic Location API.

Test Cases must be executed in the order defined for each resource because the result from one of the scenarios will be input for the next one.

Requests must be addressed to the endpoint provided for certification, specifically they must be addressed to the URI defined by the concatenation of the {apiRoot} and the specific resource, where the {apiRoot} is defined as {serverRoot}/location, being {serverRoot} the certification endpoint

### Geographic Location resource TEST CASES

#### Nominal Scenarios

##### TC\_GeoLocation\_N1 – Retrieve a geographicLocation from its id

This test case needs to have GeographicLocation pre-existing already on the server side

```
{
  "id": "7",
  "href": " https://host:port/location/geographicLocation/7",
  "name": "London",
  "geometryType": "point",
  "accuracy": "0.00001",
  "spatialRef": "WGS84",
  "geometry": [
    {
      "x": "1.430937",
      "y": "43.597208",
      "z": ""
    }
  ]
}
```

- Send a GET message to /{apiRoot}/geographicLocation/7
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one geographicLocation resource referring to {7}
  - The response message includes all mandatory parameters
  - The body of the response for the resource with identifier {7} matches the values described above

## TC\_GeoLocation\_N2 – Search for geographic location with specific characteristics

This test case needs to have GeographicLocation pre-existing already on the server side:

```
{
  "id": "9",
  "href": "https://host:port/location/geographicLocation/9",
  "name": "Road segment",
  "geometryType": "line",
  "accuracy": "0.0001",
  "spatialRef": "WGS84",
  "geometry": [
    {
      "x": "1.430937",
      "y": "43.597208",
      "z": ""
    },
    {
      "x": "1.430950",
      "y": "48.54688",
      "z": ""
    }
  ]
}
```

- Send a GET message to `/{apiRoot}/geographicLocation/?spatialRef=WGS84`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes two geographicLocation resources, referring to {7} and {9}
  - The body of the response for the resource with each identifier matches the values in the corresponding original request
- Send a GET message to `/{apiRoot}/geographicLocation/?name=London`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one Geographiclocation resource, referring to {7}
  - The response message includes all mandatory parameters
  - The body of the response for the resource with identifier {7} matches the values in the original request

- Send a GET message to `/{apiRoot}/geographicLocation/?type=line`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one `geographicLocation` resource, referring to `{9}`
  - The response message includes all mandatory parameters
  - The body of the response for the resource with identifier `{9}` matches the values in the original request

#### **TC\_GeoLocation\_N3 – Filtered retrieval of site data**

- Send a GET message to `/{apiRoot}/geographicLocation/{7}?fields=name,type`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one `GeographicLocation` resource, referring to `{7}` and including only attributes `name` and `type`, matching the values in the original request
- Send a GET message to `/{apiRoot}/geographicLocation/{9}?fields=id,geographicPoint`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one `GeographicLocation` resource, referring to `{9}` and including only attributes `id` and `geographicPoint`, matching the values in the original request

Notice that this test case is using parameters “name”, “geometryType”, “id” and “geometry” to filter the data included in the response but any other parameter could be used

#### **TC\_GeoLocation\_N4 – Filtered Search and Filtered data response**

- Send a GET message to `/{apiRoot}/geographicLocation/?name=Road segment&fields=geometryType`
- Wait for a response from the server with the following characteristics

- Response Code 200-OK
- The body of the response includes one geographicLocation resource, referring to {9}
- The body of the response for the resource with each identifier includes only attribute *geometryType*, matching the values in the corresponding original request

Notice that this test case is using parameters “geometryType” to filter the data included in the response but any other parameter could be used

## Error Scenarios

### TC\_GeoLocation\_E1 – Unknown Site

- Send a GET message to `{apiRoot}/geographicLocation/{8}`, where {8} does not match any of the identifiers previously created in the server
- Wait for a response from the server with the following characteristics
  - Response Code 404-Not Found

## ACKNOWLEDGEMENTS

### RELEASE HISTORY

Release Number	Date	Release led by:	Description
Release 17.5.0 Version 1.0.0	20-Oct-2017	Ludovic Robert	First Release of Draft Version of the Document
Release 17.5.0 Version 1.0.1	04-Dec-2017	Ludovic Robert	Aligned with Guidelines 3.0 Taking into account Patrick Huls comments
Release 17.5.0 Version 1.0.2	22-Jan-2018	Adrienne Walcott	Formatting/style edits prior to publishing

### CONTRIBUTORS TO DOCUMENT

Ludovic Robert	Orange
Luis Velarde	Telefonica
Dirk Rejahl	Bearing Point
Pierre Gauthier	TM Forum
Patrick Huls	ESRI