TM Forum Specification

API GEOGRAPHIC ADDRESS CONFORMANCE

TMF673B

Release 17.5.0

January 2018

Release: TM Forum Release 17.5.0	Status: Member Evaluation
Version: 1.0.2	IPR Mode: RAND

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

TABLE OF CONTENTS

INTRODUCTION - API DESCRIPTION

The following document is the Conformance Profile of the REST API for geographic address management. It includes the model definition as well as all available operations.

The Geographic Address Management API provides a standardized client interface to an Address management system.

It can also be used to validate geographic address data, to be sure that it corresponds to a real geographic address.

Finally, it can be used to look for a geographic address by: searching an area as a start (city, town …), then zooming on the streets of this area, and finally listing all the street segments (numbers) in a street. **This structured search is an optional feature**.

RESOURCE MODEL CONFORMANCE

API MANDATORY AND OPTIONAL RESOURCES

For the resources defined by the API fill the following table and indicate which ones are mandatory and which ones are optional.

Resource Name	Mandatory or Optional	Comments
GeographicAddress	M	
GeographicSubAddress	M	
GeographicAddressValidation	M	
Area	O	
Street	O	
StreetSegment	O	

GEOGRAPHIC ADDRESS MANDATORY AND OPTIONAL ATTRIBUTES

The table below summarizes mandatory and optional attributes for resource "GeographicAddress"

Attribute Name	Mandatory or Optional	Comments
city	O	At least one fields between locality or city or postcode must be fielded in the request
country	M	
href	M (in response messages) O (otherwise)	Url for the created resource
id	M (in response messages) O (otherwise)	Generated by the server

Attribute Name	Mandatory or Optional	Comments
locality	O	At least one fields between locality or city or postcode must be fielded in the request
postcode	O	At least one fields between locality or city or postcode must be fielded in the request
stateOrProvince	O	
streetName	O	
streetNr	O	
streetNrLast	O	
streetNrLastSuffix	O	
streetNrSuffix	O	
streetSuffix	O	
streetType	O	
geographicLocation	O	If a geographicLocation is provided, attributes spatialRef and accuracy are mandatories. If href is not provided, both geometryType and geometry (at least one geometryPoint) are mandatories.
geographicSubAddress	O	
@type	O	
@schemaLocation	O	

GEOGRAPHICADDRESS VALIDATION MANDATORY AND OPTIONAL ATTRIBUTES

The table below summarizes mandatory and optional attributes for resource "GeographicAddressValidation"

Attribute Name	Mandatory or Optional	Comments
href	M (in response messages) O (otherwise)	Url for the created resource
id	M (in response messages) O (otherwise)	Generated by the server
provideAlternative	O	By default 'no' should be considered
status	O	
validationDate	O	
validationResult	O	
alternateGeographicAddress	O	
validAddress	O	
@type	O	
@schemaLocation	O	

AREA MANDATORY AND OPTIONAL ATTRIBUTES

The table below summarizes mandatory and optional attributes for resource "Area"

Attribute Name	Mandatory or Optional	Comments
id	M (in response messages) O (otherwise)	Generated by the server
name	O	
type	O	
characteristic	O	

STREET MANDATORY AND OPTIONAL ATTRIBUTES

The table below summarizes mandatory and optional attributes for resource "Street"

Attribute Name	Mandatory or Optional	Comments
id	M (in response messages) O (otherwise)	Generated by the server
name	O	
type	O	
address	O	
characteristic	O	

STREET SEGMENT MANDATORY AND OPTIONAL ATTRIBUTES

The table below summarizes mandatory and optional attributes for resource "StreetSegment"

Attribute Name	Mandatory or Optional	Comments
id	M (in response messages) O (otherwise)	Generated by the server
number	O	
numberLast	O	
numberLastSuffix	O	
numberSuffix	O	
geographicAddress	O	

NOTIFICATION MODEL CONFORMANCE

The Pub/Sub models are common and described in the TMF REST Design Guidelines. Use the following templates to describe the Hub Mandatory and Optional attributes and filtering support.

API MANDATORY AND OPTIONAL NOTIFICATIONS

For the Notifications defined by the API the following table indicates which ones are mandatory and which ones are optional.

Notification Name	Mandatory or Optional	Comments
GeographicAddressValidationCreationNotification	M	
GeographicAddressValidationStateChangeNotification	M	
GeographicAddressValidationRemoveNotification	M	

All attributes of the resource associated with the notification are mandatory

API OPERATIONS CONFORMANCE

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

API MANDATORY AND OPTIONAL OPERATIONS

The following table indicates which ones are mandatory and which ones are optional for each one of the resources in the API (default is for all resources).

Uniform API Operation	Mandatory/Optional	Comments
GET	M for resources: GeographicAddress GeographicAddressValidation O for resources: Area Street StreetSegment	GET must be used to retrieve a representation of a resource
POST	M for resource: GeographicAddressValidation	POST must be used to create a new resource
PATCH	M for resource: AddressValidation	PATCH must be used to partially update a resource
DELETE	M for resource: AddressValidation	DELETE must be used to remove a resource

API GET FILTERING OPERATION CONFORMANCE

For every single resource use the following template to specify the mandatory and optional features supported by the GET operation.

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.:?severity=<value> &status=<value>)

Attribute selection (Filtered Response Data): 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=severity,status) will provide in the response only the values assigned to attributes category and channel. Attribute selection capabilities are the same for collections retrieval and individual resource queries

All the GET operations in this API share the same status code pattern.

GET	M/O	The GET operation is MANDATORY for the resources GeographicAddress & GeographicAddressValidation The GET Operation is optional for the resources Area, street & streetSegment
Response Status Code 200	M	
Other Status Codes	NA	

/GEOGRAPHICADDRESS?FIELDS=…&{FILTERING}

This operation list geographic addresses entities.
Attribute selection is mandatory for all first level attributes.
Filtering is mandatory for geogaphicAddress to avoid answer with too many geograpgicAddress occurrences.

Example of each filtering to be used:

- Geographic location reference

- streetNr+streetName+postcode+country

- streetNr+streetName+(city or locality)+country

- interval of streetNr+streetName+streetType +(city or locality)+country

/GEOGRAPHICADDRESS/{ID}?FIELDS=…&{FILTERING}

This operation retrieves a geographic address entity.
Attribute selection is mandatory for all first level attributes.
Filtering on sub-resources is optional for all compliance levels.

/GEOGRAPHICADDRESSVALIDATION?FIELDS=…&{FILTERING}

This operation list geographic address validation entities.
Attribute selection is mandatory for all first level attributes.
Filtering is mandatory for first compliance level (L1) and optional otherwise.

/GEOGRAPHICADDRESSVALIDATION/{ID}?FIELDS=…&{FILTERING}

This operation retrieves a geographic address validation entity.
Attribute selection is mandatory for all first level attributes.
Filtering on sub-resources is optional for all compliance levels.

/AREA?FIELDS=…&{FILTERING}

This operation list area entities.
Attribute selection is mandatory for all first level attributes.
Filtering is mandatory for first compliance level (L1) and optional otherwise.

/AREA/{ID}?FIELDS=…&{FILTERING}

This operation retrieves an area entity.
Attribute selection is mandatory for all first level attributes.
Filtering on sub-resources is optional for all compliance levels.

/STREET?FIELDS=…&{FILTERING}

This operation list street entities.
Attribute selection is mandatory for all first level attributes.
Filtering is mandatory for first compliance level (L1) and optional otherwise.

/STREET/{ID}?FIELDS=…&{FILTERING}

This operation retrieves a street entity.
Attribute selection is mandatory for all first level attributes.
Filtering on sub-resources is optional for all compliance levels.

/STREETSEGMENT?FIELDS=…&{FILTERING}

This operation list street segment entities.
Attribute selection is mandatory for all first level attributes.
Filtering is mandatory for first compliance level (L1) and optional otherwise.

/STREETSEGMENT/{ID}?FIELDS=…&{FILTERING}

This operation retrieves a street segment entity.
Attribute selection is mandatory for all first level attributes.
Filtering on sub-resources is optional for all compliance levels.

All the POST operations in this API share the same status code pattern.

POST	M	
Status Code 201	M	
Other Status Codes	NA	

/GEOGRAPHICADDRESSVALIDATION

This operation creates a geographic address validation entity.

Mandatory and Non Mandatory Attributes

All attributes are optional in a POST but at least

- streetName + streetType + (locality or city) + (stateorProvince or postcode) + country
- or GeographicLocation with a minimum of one geographicPoint

should be provided.

Hidden Attributes

The following attributes should not be used when creating a Geographic AddressValidation

status, validationDate, validationResult, alternateAddress

The response from the server must include a BODY with the contents of the new resource created, filled with at least the same information elements that were included in the request and are supported by the server. Notice that the value stored by the server may be different than the one set in the request (e.g.: status may be differently set by the server after processing than the one requested by the requestor)

If the POST request includes optional parameters (as per the model resource definition) that are not supported by the server, then the server must reject the request (replying with a 4xx error response) indicating the parameter not supported.

All the PATCH operations in this API share the same status code pattern.

PATCH	O	
Status Code 201	M	
Other Status Codes	NA	

/GEOGRAPHICADDRESSVALIDATION/{ID}

This operation allows partial updates of a geographic address validation entity. Support of json/merge (https://tools.ietf.org/html/rfc7386) is mandatory, support of json/patch (http://tools.ietf.org/html/rfc5789) is optional.

Note: If the update operation yields to the creation of sub-resources or relationships, the same rules concerning mandatory sub-resource attributes and default value settings in the POST operation applies to the PATCH operation. Hence these tables are not repeated here.

Patchable and Non Patchable Attributes

The tables below provide the list of patchable and non patchable attributes, including constraint rules on their usage.
Notice that patching is possible only for 'admin' API users.

Attribute Name	Patchable (YES/NO)	Rule
id	NO	
href	NO	
provideAlternative	YES	Only is status = inProgress
status	NO	
validationDate	NO	
validationResult	NO	
alternateAddress	NO	
validAddress	YES	Only is status = inProgress

Additional Rules

The following pre-conditions apply for this operation.

Pre-conditions

PATCH allowed if status is still inProgress

API DELETE OPERATION CONFORMANCE

All the DELETE operations in this API share the same status code pattern.

DELETE	O	
Status Code 200	M	
Other Status Codes	NA	

/GEOGRAPHICADDRESSVALIDATION/{ID}

This operation deletes a geographic address validation entity.

API CONFORMANCE TEST SCENARIOS

This section describes the test scenarios required for the basic CONNECT certification of the GeographicAddress 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}/geographicAddressManagement/v1**, where {serverRoot} defines the certification endpoint

GeographicAddress resource TEST CASES

Nominal Scenarios

TC_GeoAddress_N1 – Retrieve a GeographicAddress from its id

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

```
{
"id":"7",
"href":" https://host:port/location/geographicAddress/7",
"streetNr": "29",
"streetName": "Rambeau",
"streetType": "Rue",
"city": "Merville",
"postcode": "31330",
"country": "France",
"@type": "UrbanPropertyAddress "
}
```

- Send a GET message to /{apiRoot}/geographicAddress/7

- Wait for a response from the server with the following characteristics

    - Response Code 200-OK

    - The body of the response includes one geographicAdress 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_GeoAddress_N2 – Search for GeographicAddress with specific characteristics

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

```
{
"id":"7",
"href":" https://host:port/location/geographicAddress/7",
"streetNr": "29",
"streetName": "Rambeau",
"streetType": "Rue",
"city": "Merville",
"postcode": "31330",
"country": "France",
"@type": "UrbanPropertyAddress ",
}

{
"id":"85",
"href":" https://host:port/location/geographicAddress/85",
"streetNr": "27",
"streetName": "Rambeau",
"streetType": "Rue",
"city": "Merville",
"postcode": "31330",
"country": "France",
"@type": "UrbanPropertyAddress "
}
```

- Send a GET message to
  /{apiRoot}/geographicAddress/?streetName="Rambeau"&postcode=31330

- Wait for a response from the server with the following characteristics
    - Response Code 200-OK

    - The body of the response includes two geographic addresses referring to {7} and {85}

    - The body of the response for the resources with identifier {7} & {85} matches the values described above

- Send a GET message to /{apiRoot}/
  geographicAddress/?streetName="Rambeau"&postcode=31330&streetNr=27

- Wait for a response from the server with the following characteristics

    - Response Code 200-OK

- The body of the response includes one geographicAdress resource referring to {85} and there is no reference to geographicAdress resource {7}

- The response message includes all mandatory parameters

- The body of the response for the resource with identifier {85} matches the values described above

TC_GeoAddress_N3 – Filtered retrieval of GeographicAddress data

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

```
{
"id":"7",
"href":" https://host:port/location/geographicAddress/7",
"streetNr": "29",
"streetName": "Rambeau",
"streetType": "Rue",
"city": "Merville",
"postcode": "31330",
"country": "France",
"@type": "UrbanPropertyAddress "
}
```

- Send a GET message to /{apiRoot}/geographicAddress/7?fields=city,postcode,country

- Wait for a response from the server with the following characteristics
  - Response Code 200-OK

  - The body of the response includes one geographicAddress resource, referring to {7} and including only attributes *city, postcode* and *country*, matching the values decribed above

Notice that this test case is using parameters " city", "postcode", and "country" to filter the data included in the response but any other parameter could be used

TC_GeoAddress_N4 – Filtered retrieval of GeographicAddress data

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

```
{
"id":"7",
"href":" https://host:port/location/geographicAddress/7",
"streetNr": "29",
"streetName": "Rambeau",
"streetType": "Rue",
"city": "Merville",
```

"postcode": "31330",
"country": "France",
"@type": "UrbanPropertyAddress "
}

{
"id":"85",
"href":" https://host:port/location/geographicAddress/85",
"streetNr": "27",
"streetName": "Rambeau",
"streetType": "Rue",
"city": "Merville",
"postcode": "31330",
"country": "France",
"@type": "UrbanPropertyAddress "
}

- Send a GET message to /{apiRoot}/geographicAddress/?streetName="Rambeau"&postcode=31330&fields=city,postcode,country

- Wait for a response from the server with the following characteristics
  - Response Code 200-OK

  - The body of the response includes 2 geographicAddress resource, referring to {7} & {85} and including only attributes *city, postcode* and *country*, matching the values decribed above

  Notice that this test case is using parameters "city", "postcode", and "country" to filter the data included in the response but any other parameter could be used

Error Scenarios

TC_ GeoAddress _E1 – Unknown geographicAddress identifier

- Send a GET message to /{apiRoot}/}/geographicAddress/{IDGeoAddress}, where { IDGeoAddress} does not match any of the identifiers previously creted in the server

- Wait for a response from the server with the following characteristics

  o Response Code 404-Not Found

Nominal Scenarios

TC_GeoAddVal_N1 – Create a new GeographicAddressValidation

- Send a POST message to {apiRoot}/geographicAddressValidation/ with the following contents in the BODY

```
{
  "provideAlternative": true,
  "validAddress":
    {
    "streetName": " Rambeau ",
    "streetType": "Rue",
    "postcode": "31330",
    "city": "Merville",
    "country": "France",
    "@type": "UrbanPropertyAddress "
  }
}
```

- Wait for a response from the server with the following characteristics

  - Response Code 201-Created

  - Include a location header in the body set to {apiRoot}/ geographicAddressValidation /{GeoAddrVal01} where { GeoAddrVal01} indicates the identifier assigned by the server to the new geographicAddressValidation resource

  - The response message includes all mandatory parameters

  - The body of the response matches the values in the original request

- Send a GET message to /{apiRoot}/ geographicAddressValidation /

- Wait for a response from the server with the following characteristics

  - Response Code 200-OK

  - The body of the response includes one geographicAddressValidation resource with ID set to { GeoAddrVal01}, the same identifier as assigned by the server to the new resource created

  - The response message includes all mandatory parameters

  - The body of the response for the resource with identifier {GeoAddrVal01} matches the values in the original request

- Send a GET message to /{apiRoot}/ geographicAddressValidation /{ GeoAddrVal01}

- Wait for a response from the server with the following characteristics

  - Response Code 200-OK

  - The response message includes all mandatory parameters

  - The body of the response includes a geographicAddressValidation resource structure that matches the values in the original request**.**

TC_GeoAddVal_N2 – Search for GeographicAddressValidation with specific characteristics

- Send a POST message to {apiRoot}/geographicAddressValidation/ with the following contents in the BODY

```
{
    "provideAlternative": false,
    "validAddress":
      {
      "streetName": "Clemenceau ",
      "streetType": "Avenue",
      "postcode": "33000",
      "city": "Bordeaux",
      "country": "France",
      "@type": "UrbanPropertyAddress "
    }
}
```

- Wait for a response from the server with the following characteristics

  - Response Code 201-Created

  - Include a location header in the body set to {apiRoot}/ geographicAddressValidation /{GeoAddrVal02} where {GeoAddrVal02} indicates the identifier assigned by the server to the new geographicAddressValidation resource

  - The response message includes all mandatory parameters

- Send a GET message to /{apiRoot}/ geographicAddressValidation /

- Wait for a response from the server with the following characteristics

  - Response Code 200-OK

- The body of the response includes 2 geographicAddressValidation resources with ID set to {GeoAddrVal01} and {GeoAddrVal02}
- 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}/ geographicAddressValidation /?postcode=33000

- Wait for a response from the server with the following characteristics

  - Response Code 200-OK

  - The body of the response includes one geographicAddressValidation refering to {GeoAddrVal02}

  - The response message includes all mandatory parameters

  - The body of the response for the resource with identifier {GeoAddrVal02} matches the values in the original request

TC_GeoAddVal_N3 – Filtered retrieval of GeographicAddressValidation data

- Send a GET message to {apiRoot}/geographicAddressValidation/{GeoAddrVal01}?fields=postcode, city
- Wait for a response from the server with the following characteristics

  o Response Code 200-OK

  o The body of the response includes one GeographicAddressValidation resource, referring to {GeoAddrVal 01} and including only attributes *city* and *postcode*, matching the values in the original request

Notice that this test case is using parameters "postcode" and "city" to filter the data included in the response but any other parameter could be used

TC_GeoAddVal_N4 – Filtered retrieval and filtered data of GeographicAddressValidation

- Send a GET message to {apiRoot}/geographicAddressValidation/?country="France"&fields=postcode, city

- Wait for a response from the server with the following characteristics

  o Response Code 200-OK

o   The body of the response includes 2 GeographicAddressValidation resource, referring to {GeoAddrVal 01} and {GeoAddrVal 02} including only attributes *city* and *postcode*, matching the values in the original request

Notice that this test case is using parameters "postcode" and "city" to filter the data included in the response but any other parameter could be used

TC_GeoAddVal_N5 – Parameters not included in GeographicAddressValidation creation request but provided in the response

- Send a POST message to to {apiRoot}/geographicAddressValidation/ with the following contents in the BODY

```
{
  "provideAlternative": false,
  "validAddress":
    {
    "streetName": "Louis Pello",
    "streetType": "Chemin",
    "postcode": "31000",
    "city": "Toulouse",
    "country": "France",
    "@type": "UrbanPropertyAddress "
  }
}
```

- Wait for a response from the server with the following characteristics

    - Response Code 201-Created

    - Include a location header in the body including the identifier assigned by the server to the new geographicAddressValidation resource

    - The body of the response includes parameters "id", "href" and "status"

TC_GeoAddVal_E1 – Unknown GeographicAddressValidation

- Send a GET message to {apiRoot}/geographicAddressValidation/{GeoAddrVal04} where {GeoAddrVal04} does not match any of the identifiers previously created in the server

- Wait for a response from the server with the following characteristics

    o   Response Code 404-Not Found

TC_GeoAddVal_E3 – Invalid Request – Missing mandatory parameter

- Send a POST message to to {apiRoot}/geographicAddressValidation/ with the following contents in the BODY

```
{
  "provideAlternative": false,
  "validAddress":
    {
    "streetName": "Pelletier",
    "streetType": "Rue",
    "country": "France",
  }
}
```

Notice that this request is missing mandatory parameter: "At least one fields between locality or city or postcode must be fielded in the request".

- Wait for an error response from the server indicating the mandatory parameter is missing in the request

RELEASE HISTORY

Release Number	Date	Release led by:	Description
Release 17.5.0 Version 1.0.0	Nov 3rd, 2017	Ludovic Robert (Orange)	First Release of the Document.
Release 17.5.0 Version 1.0.1	04/12/2017	Ludovic Robert (Orange)	Aligned with Guidelines 3.0 taking into account Patrick Huls' comments
Release 17.5.0 Version 1.0.2	19-Jan-2018	Adrienne Walcott	Formatting/style edits prior to publishing