*Frameworx Specification*

# Address API REST Specification

**TMF647**
**Release 16.0**
**June 2016**

| | |
|---|---|
| Latest Update: Frameworx Release 1550 | Member Evaluation |
| Version 2.0.0 | IPR Mode: RAND |

## NOTICE



Direct inquiries to the TM Forum office:

240 Headquarters Plaza,
East Tower – 10th Floor,
Morristown, NJ 07960 USA
Tel No. +1 973 944 5100
Fax No. +1 973 944 5110
TM Forum Web Page: www.tmforum.org

## TABLE OF CONTENTS

# LIST OF TABLES

N/A

# INTRODUCTION

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

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

It allows to look for worldwide addresses.

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

Finally, it can be used to look for an 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.

## SAMPLE USE CASES

N/A

# RESOURCE MODEL

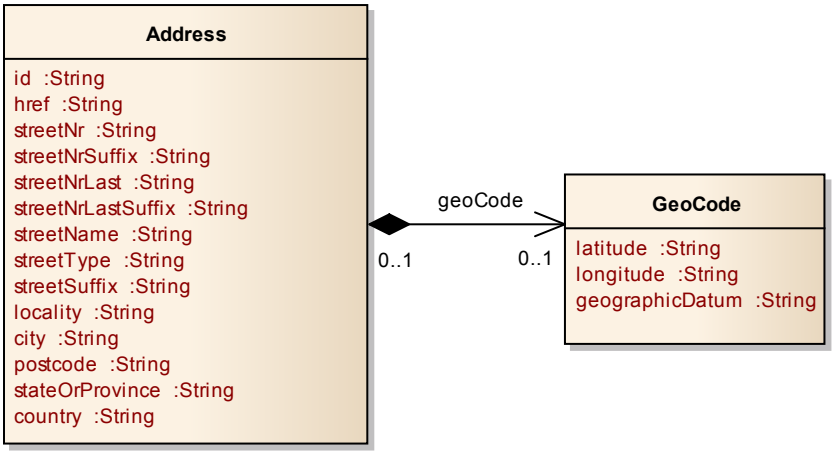
## Managed Entity and Task Resource Models

### ADDRESS RESOURCE

Structured textual way of describing how to find a Property in an urban area (country properties are often defined differently).

*Nota* : Address corresponds to SID UrbanPropertyAddress.

**Resource model**




**Lifecycle**

No state machine for the resources detailed in this API

**Json representation sample**

We provide below the json representation of an example of an Address resource object


```
{
 "id": "7660828",
 "href": "address/7660828",
 "streetNr": "225",
 "streetNrSuffix": "B",
 "streetNrLast": "",
 "streetNrLastSuffix": "",
 "streetName": "Strathmore",
 "streetType": "Terrace",
 "streetSuffix": "",
 "locality": "Brighton",
```

```
  "city": "Brighton",
  "postcode": "5004",
  "stateOrProvince": "SA",
  "country": "Australia",
  "geoCode": {
     "latitude": "1.430937",
     "longitude": "43.597208",
     "geographicDatum": "WGS84"
  }
}
```


### Field descriptions

*Address* fields

| Field | Description |
|---|---|
| **id** | Unique identifier of the address |
| **href** | An URI used to access to the address resource |
| **streetNr** | Number identifying a specific property on a public street. It may be combined with streetNrLast for ranged addresses |
| **streetNrSuffix** | the first street number suffix |
| **streetNrLast** | Last number in a range of street numbers allocated to a property |
| **streetNrLastSuffix** | Last street number suffix for a ranged address |
| **streetName** | Name of the street or other street type |
| **streetType** | alley, avenue, boulevard, brae, crescent, drive, highway, lane, terrace, parade, place, tarn, way, wharf ? |
| **streetSuffix** | A modifier denoting a relative direction |
| **locality** | "An area of defined or undefined boundaries within a local authority or other legislatively defined area, usually rural or semi rural in nature." [ANZLIC-STREET], or a suburb "a bounded locality within a city, town or shire principally of urban character " [ANZLIC-STREET] |
| **city** | City that the address is in |
| **postcode** | A descriptor for a postal delivery area, used to speed and simplify the delivery of mail |
| **stateOrProvince** | the State or Province that the address is in |
| **country** | Country that the address is in |
| **geoCode** | Geographic coordinates to point to the address |

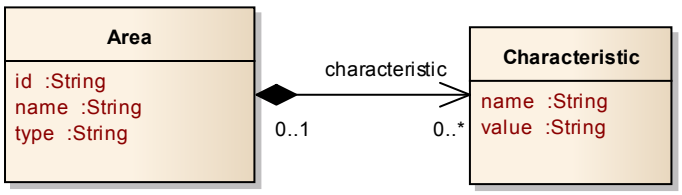
*geoCode* fields

| Field | Description |
|---|---|
| **latitude** | Latitude |
| **longitude** | Longitude |
| **geographicDatum** | Geocoding referential |

## AREA RESOURCE

It is used to represent areas defined on maps that relate to areas of settlement.
*Nota* : Area corresponds to SID GeographicArea.

### Resource model




### Lifecycle

No state machine for the resources detailed in this API

### Json representation sample

We provide below the json representation of an example of an Area resource object :

```
{
 "id": "7660828",
 "name": "Bagneux",
 "type": "city",
 "characteristic": [{
     "name": "aWayOfCodingCityInTheCountry",
     "value": "10304"
 }]
}
```

### Field descriptions

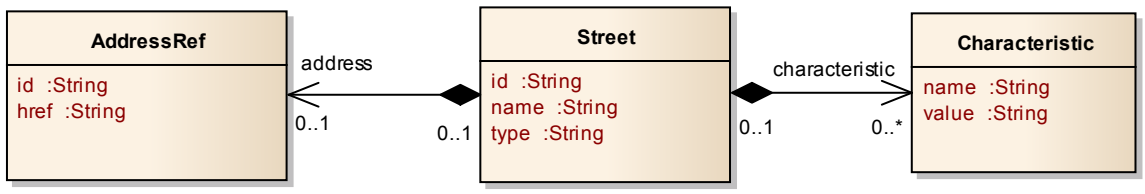
*area* fields

| Field | Description |
|---|---|
| **id** | Unique identifier of the Area |
| **name** | The defined name of the municipality |
| **type** | SUBURB, LOCALITY, CITY, TOWN, BOROUGH, … |
| **characteristic** | Name/value pairs, used to extra characterized the Area (e.g. if a standard code set has been defined for the GeographicArea type, etc.) |

## STREET RESOURCE

It is used to represent streets within an Area.
*Nota* : Street corresponds to SID Street.

**Resource model**




**Lifecycle**

No state machine for the resources detailed in this API

**Json representation sample**

We provide below the json representation of an example of a Street resource object :

```
{
  "id":"2173322",
  "type":"Rue",
  "name":"OLIVETTES",
  "characteristic": [{
      "name": "aWayOfCodingStreetsInTheCountry",
      "value": "34556"
  }]
}
```

In some cases, “streets” can represent small villages / hamlets which exist in a town and that are directly addressable (without any street specification).

In this last case, it can represent a complete address, which is accessible through hyperlinking.

```
{
  "id":"2173322",
  "type":"Hamlet",
  "name":"The Wires",
  "characteristic": [{
      "name": "aWayOfCodingHamletsInTheCountry",
      "value": "34556"
  }],
  "address":{
```

```
      "href":"address/21885630",
      "id":"21885630"
   }
}
```


**Field descriptions**

*street* fields

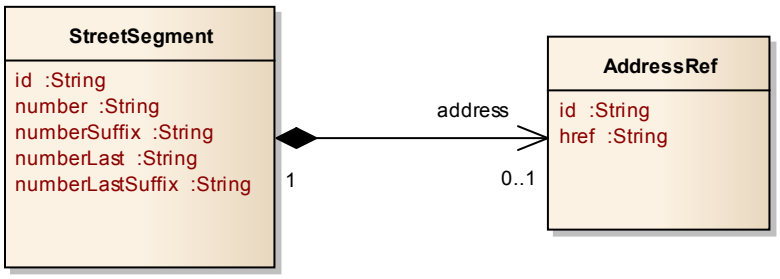
| Field | Description |
|---|---|
| **id** | Unique identifier of the Street |
| **name** | The defined name of the street |
| **characteristic** | Name/value pairs, used to extra characterized the Area (e.g. if a standard code set has been defined for the GeographicArea type, etc.) |
| **type** | alley, avenue, etc. |
| **address** | a linking to the corresponding geographic address |

## STREETSEGMENT RESOURCE

It is used to represent segments in a given street: this can be directly street numbers (22), or group of numbers materializing a geographic address, e.g. 22-24.

This level can represent an entire address, which is accessible through hyperlinking
*Nota* : StreetSegment corresponds to SID StreetSegment.

**Resource model**




**Lifecycle**

No state machine for the resources detailed in this API

**Json representation sample**

We provide below the json representation of an example of an StreetSegment resource object :


```
{
   "id":"21885630",
```

```
  "number":"1",
  "numberSuffix":"",
  "numberLast":"",
  "numberLastSuffix":"",
  "address":{
      "href":"address/21885630",
      "id":"21885630"
  }
}
```


**Field descriptions**

*streetSegment* fields

| Field | Description |
|---|---|
| **id** | Unique identifier of the Street Segment |
| **number** | number identifying a specific property on a public street. It may be combined with streetNrLast for ranged addresses |
| **numberSuffix** | the first street number suffix |
| **numberLast** | the last number in a range of street numbers allocated to a property |
| **numberLastSuffix** | the last street number suffix for a ranged address |
| **address** | a linking to the corresponding geographic address |

## Notification Resource Models

There is no notification in the Address API

## API OPERATIONS

Remember the following Uniform Contract:

| Operation on Entities | Uniform API Operation | Description |
|---|---|---|
| Query Entities | GET Resource | GET must be used to retrieve a representation of a resource. |
| Create Entity | POST Resource | POST must be used to create a new resource |
| Partial Update of an Entity | PATCH Resource | PATCH must be used to partially update a resource |
| Complete Update of an Entity | PUT Resource | PUT must be used to completely update a resource identified by its resource URI |
| Remove an Entity | DELETE Resource | DELETE must be used to remove a resource |
| Execute an Action on an Entity | POST on TASK Resource | POST must be used to execute Task Resources |
| Other Request Methods | POST on TASK Resource | GET and POST must not be used to tunnel other request methods. |

Filtering and attribute selection rules are described in the TMF REST Design Guidelines.

Notifications are also described in a subsequent section.

# OPERATIONS ON ADDRESS

## LIST ADDRESSES

GET /api/address?{fields=attributes}&{filtering expression}&.fullText={fullTextSearch}&.fuzzy={true/false}

**Description**

This operation is used to retrieve an address corresponding to search criteria.

Filtering is allowed on all attributes. See example below.
Attribute selection is possible for all attributes. See example below.

Two special search fields can also be used:

- ".fullText" : which can be used for full text searches, for example when you have a single textarea to capture address search infos
- ".fuzzy" : which can be used for approximative searches (sounds like, etc.)

Behavior :

- Returns HTTP/1.1 status code 200 if the request was successful

**Usage Samples**

Here's an example of a request for retrieving Address resources.

| **Request** |
| --- |
| GET /api/address?geoCode.latitude=1.296349&geoCode.longitude=43.717627<br>Accept: application/json |
| **Response** |
| 200<br>Content-Type: application/json<br>[<br>{<br>"id":"7513180",<br>"href":"address/7513180",<br>"streetNr": "29",<br>"streetName": "Rambeau", |

```
  "streetType": "Rue",
  "city": "Merville",
  "postcode": "31330",
  "country": "France",
  "geoCode": {
    "latitude": "1.296349",
    "longitude": "43.717627",
    "geographicDatum": "WGS84"
  }
}
]
```

## VALIDATE AN ADDRESS

POST /api/address/validate

**Description**

This operation can be used to validate an address if you give a sufficient part or the entire attribute set of the address to validate.

Behavior :

- Returns HTTP/1.1 status code 201 and the detailed validated address as payload if the request was successful
- Returns HTTP/1.1 status code 404 (Not found) if the address does not exist

**Usage Samples**

**Request**

```
POST /api/address/validate
Content-type: application/json

{
 "streetNr": "225",
 "streetNrSuffix": "B",
 "streetName": "Strathmore",
 "streetType": "Terrace",
 "city": "Brighton",
 "country": "Australia"
}
```

**Response**

```
201
Content-Type: application/json

[
{
  "id": "7660828",
  "href": "address/7660828",
  "streetNr": "225",
  "streetNrSuffix": "B",
  "streetName": "Strathmore",
  "streetType": "Terrace",
  "locality": "Brighton",
  "city": "Brighton",
  "postcode": "5004",
  "stateOrProvince": "SA",
  "country": "Australia",
  "geoCode": {
    "latitude": "1.430937",
    "longitude": "43.597208",
    "geographicDatum": "WGS84"
  }
 }
]
```

## GET AN ADDRESS

```
GET /api/address/{id}
```

### Description

Retrieves an address using its unique ID. This ID should be retrieve either using the address completion process (cf. completion), or in another API of the ecosystem (party, appointment, etc.)

Behavior :

- Returns HTTP/1.1 status code 200 if the request was successful
- Returns HTTP/1.1 status code 404 (Not found) if the address does not exist

### Usage Samples

| **Request** |
| --- |
| GET /api/address/7660828<br>Accept: application/json |
| **Response** |
| 200<br>Content-Type: application/json<br><br>{<br>"id": "7660828",<br>"href": "address/7660828",<br>"streetNr": "225",<br>"streetNrSuffix": "B",<br>"streetName": "Strathmore",<br>"streetType": "Terrace",<br>"locality": "Brighton",<br>"city": "Brighton",<br>"postcode": "5004",<br>"stateOrProvince": "SA",<br>"country": "Australia",<br>"geoCode": {<br>"latitude": "1.430937",<br>"longitude": "43.597208",<br>"geographicDatum": "WGS84"<br>}<br>} |

## ADDRESS COMPLETION PROCESS : STEP1 - LIST GEOGRAPHIC AREA

GET api/area?{fields=attributes}&{filtering expression}&.fuzzy={true/false}

### Description

This operation is the first step of an address completion process, allowing to retrieve geographic areas:

- Step 1: I look for a geographic area (city, locality, district, etc.) using its name
- Step 2: I look for the streets inside this geographic area
- Step 3: I get all the street segments (numbers) existing in the street

Filtering is allowed on all attributes. See example below.
Attribute selection is possible for all attributes. See example below.

A special search field can also be used:

- ".fuzzy" : which can be used for approximative searches (sounds like, etc.)

Behavior:

- Returns HTTP/1.1 status code 200 if the request was successful

**Usage Samples**

**Request**

```
GET /api/area?name=Kensington&fields=id,name,type
Accept: application-json
```

**Response**

```
200
Content-Type: application/json
[
{
 "id": "7660828",
 "name": "Royal Borough of Kensington and Chelsea",
 "type": "borough"
},
{
 "id": "7660855",
 "name": "North Kensington",
 "type": "district"
},
{
 "id": "7660821",
 "name": "South Kensington",
 "type": "district"
},
{
 "id": "7660834",
 "name": "Kensington",
 "type": "district"
}
]
```

## ADDRESS COMPLETION PROCESS : STEP2 - LIST STREETS

GET /api/street?{fields=attributes}&{filtering expression}&.fuzzy={true/false}

**Description**

This operation is the second step of an address completion process, allowing to retrieve streets:

- Step 1: I look for a geographic area (city, locality, district, etc.) using its name
- Step 2: I look for the streets inside this geographic area
- Step 3: I get all the street segments (numbers) existing in the street

Filtering is allowed on all attributes. See example below.
Attribute selection is possible for all attributes. See example below.

A special search field can also be used:

- ".fuzzy" : which can be used for approximative searches (sounds like, etc.)

Behavior:

Returns HTTP/1.1 status code 200 if the request was successful

**Usage Samples**

**Request**

```
GET /api/street?name=Cromwell&area.id=7660821&fields=id,name,type
Accept: application/json
```

**Response**

```
200
Content-Type: application/json
[
{
  "id":"2173322",
  "type":"Road",
  "name":"Cromwell"
},
{
  "id":"2173323",
  "type":"Place",
```

```
   "name":"Cromwell"
},
{
   "id":"2173324",
   "type":"Mews",
   "name":"Cromwell"
}
]
```

## ADDRESS COMPLETION PROCESS : STEP3 - LIST SEGMENTS

GET /api/street/{id}/segment

**Description**

This operation is the last step of an address completion process, allowing to retrieve numbers in a street:

- Step 1: I look for a geographic area (city, locality, district, etc.) using its name
- Step 2: I look for the streets inside this geographic area
- Step 3: I get all the street segments (numbers) existing in the street

Behavior:

Returns HTTP/1.1 status code 200 if the request was successful

**Usage Samples**

**Request**

```
GET /api/street/2173323/segment
Accept: application/json
```

**Response**

```
200
Content-Type: application/json
[
{
   "id":"21733233",
   "number":"3",
```

```
   "address":{
     "href":"address/217332332",
     "id":"217332332"
   }
},
{
   "id":"21733234",
   "number":"4",
   "address":{
     "href":"address/217332342",
     "id":"217332342"
   }
},
etc.
]
```

## API NOTIFICATIONS

N/A

# ACKNOWLEDGEMENTS

## RELEASE HISTORY

| Release Number | Date | Release led by: | Description |
|---|---|---|---|
| Release 1.0 | 09/06/2015 | Maxime Delon<br>Orange<br>maxime.delon@orange.com | First Release of Draft Version of the Document. |
| Release 1.1 | 25/03/2016 | Ludovic Robert<br>Orange<br>ludovic.robert@orange.com | Updated for delete of subAddress resource not necessary |
| | | | |

## CONTRIBUTORS TO DOCUMENT

| | |
|---|---|
| | |
| | |