TM Forum Specification

Federated ID API
REST Specification

TMF691
Release 18.0.0
June 2018

Latest Update: TM Forum Release 18.0.0	Member Evaluation
Version 1.0.2	IPR Mode: RAND

NOTICE

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

LIST OF TABLES

N/A

INTRODUCTION

The following document is the specification of the REST API for Federated ID Management. It includes the model definition as well as all available operations for SID userinfo entity.

This API covers the operations required to allow an application (for instance a selfcare mobile app) request identity information about the individual that is making use of the functionality provided by such application (the user), or in general to allow an application to request identity related information about an individual to the system holding such identity information.

This API, instead of defining new operations, relies on the use of industry standard for identity information such as OpenID Connect (http://openid.net/specs/openid-connect-core-1_0.html) and OAuth2.0 (RFC6749).

This API manages Federated Identity because it defines the operations that must be supported by any system in order to allow providing identity related information (i.e.: the set of attributes related to the individual, such as name, family name, primary phone number, gender, birthdate, …) to different requesting applications, provided they are authorized to perform such request. Authorization can be granted by providing a valid proof of authorization (e.g.: an OAuth2.0 token) granted by the individual whose identity is being requested.

It is up to each implementation to define how authorization can be provided (and confirmed), this API only defines the operations required to request identity related information by an authorized application. As indicated before industry standard mechanisms (OpenID Connect and OAuth2.0) are considered for the implementation of this API, therefore it is assumed in this specification that the request for user´s identity includes an Authorization header with a valid token.

This API relates with existing TM Forum PartyManagement API (TMF632) because a user is actually an individual (i.e.: a party) acting with the role of “user” when interacting with the application requesting the identity information, therefore the so called “userinfo” can be actually either an extension or a subset of the specific party data information, which typically encompasses identity related and any other additional information related to such individual (such as for instance the date when the individual registered in the system, the status or the list of other related parties which is not identity-specific information).

This API defines a set of extended claims over those standard claims defined in OpenID Connect, in order to provide not only the basic user identity information but also the so called “userAssets” or list of assets that can be managed by the individual whose identity information is being requested.

SAMPLE USE CASES

This section includes a set of main use cases that can be performed with this API. Additional use cases can be generated using the operations and resources defined in this specification.

- An application that is being used by an individual and provides functionality to that individual (i.e.: End user) based on his identity (for instance a selfCare application that greets the user displaying his name on the screen), needs to identify such individual.

  In order to obtain identity information of the individual, the app performs the following steps

  1. The application obtains a proof of authorization by the individual to request his identity information (e.g.: an OAUth2.0 token)

  2. The application requests the identity-related information of the user (i.e.: the userinfo) associated to the individual whose proof of authorization is provided

- An application that is being used by an individual and provides functionality to that individual (i.e.: End user) based on the information stored in a system (for instance a selfCare application that will request billing information about the billing accounts associated to the user), needs to identify such individual and know what assets that user can manage.

  In order to obtain identity information of the individual, the app performs the following steps

  1. The application obtains a proof of authorization by the individual to request his identity information (e.g.: an OAUth2.0 token)

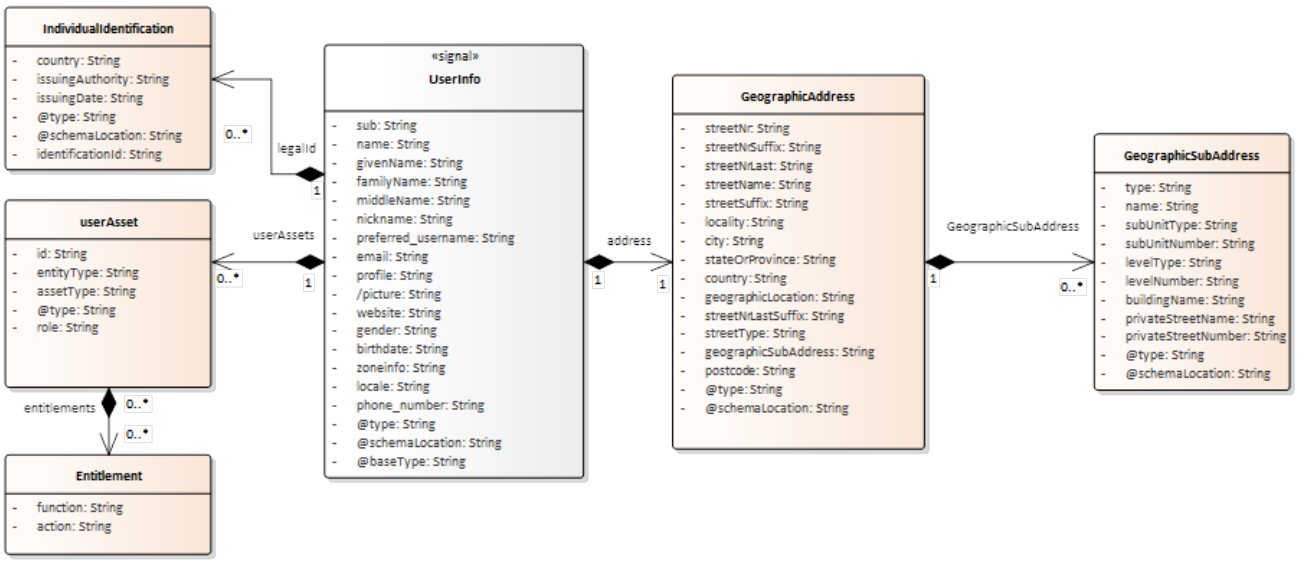
  2. The application requests the identity-related information of the user (i.e.: the userinfo and userAssets) associated to the individual whose proof of authorization is provided

RESOURCE MODEL

USERINFO RESOURCE

The Userinfo resource represents a class that allows to define identity related information of an individual (i.e.: a Party).

Resource model



Lifecycle

No state machine for the resources detailed in this API

Field descriptions

UserInfo fields

Field	*Mandatory in API messages*	*Description*
sub	Yes in response	A string. Subject - Unique Identifier for the user
name	Yes in response	A string. User´s full name in displayable form including all name parts, possibly including titles and suffixes, ordered according to the user's locale and preferences
given_name	No	A string. Given name(s) or first name(s) of the user
family_name	No	A string. Surname(s) or last name(s) of the user
middle_name	No	A string. Middle name(s) of the user
nickname	No	A string. Casual name of the user that may or may not be the same as the given name. For instance, a

		nickname value of Mike might be returned alongside a given name value of Michael
preferred_username	No	A string. Shorthand name by which the user wishes to be referred to at the RP, such as janedoe or j.doe
email	No	A string. User's preferred e-mail address. If the party entity includes multiple contact media of type email, this would be the preferred one
phone_number	No	A string. User's preferred telephone number. If the party entity includes multiple contact media of type tel-nr, this would be the preferred one
gender	No	A string. User's gender. Values defined by this specification are female and male
birthdate	No	A string. User's birthday, represented as an [ISO8601-2004] YYYY-MM-DD format
locale	No	A string. User's locale, represented as a [RFC5646] language tag. This is typically an [ISO639-1] language code in lowercase and an [ISO3166-1] country code in uppercase, separated by a dash. For example, en-US or fr-CA
zoneinfo	No	A string. String from zoneinfo time zone database representing the End-User's time zone. For example, Europe/Paris or America/Los_Angeles
profile	No	A string. URL of the user's profile page
picture	No	A string. URL of the user's profile picture. This URL MUST refer to an image file (for example, a PNG, JPEG, or GIF image file),
website	No	A string. URL of the user's Web page or blog
address	No	An address (GeographicAddress) passed by value, providing the preferred postal address. If the party entity includes multiple contact media of type postal-address, this would be the preferred one
legalId	No	A proof of legal identification (IndividualIdentification[*]) of an individual, such as a passport, a driver's license, social security number, …
userAssets	No	List of entities (userAsset[*]) that can be managed by an individual
@type	No	A string. Indicates the type of the resource in case polymorphism applies
@baseType	No	A string. Indicates the base type of the resource
@schemaLocation	No	A string. A Link to the schema describing this REST Resource

GeographicAddress sub-resource

Address reference. Defines an address and/or identifies an existing address entity

An address allows textual description of an existing place over the surface of the Earth

This resource must be invoked as value for this API

Field	*Mandatory in API messages*	*Description*
streetNr	Yes	A string. Number identifying a specific property on a public street. It may be combined with streetNrLast for ranged addresses.
streetNrSuffix	No	A string. the first street number suffix.
streetNrLast	No	A string. Last number in a range of street numbers allocated to a property.
streetNrLastSuffix	No	A string. Last street number suffix for a ranged address.
streetName	Yes	A string. Name of the street or other street type.
streetType	Yes	A string. Alley, avenue, boulevard, brae, crescent, drive, highway, lane, terrace, parade, place, tarn, way, wharf.
streetSuffix	No	A string. A modifier denoting a relative direction.
postcode	Yes	A string. Descriptor for a postal delivery area, used to speed and simplify the delivery of mail (also known as zipcode).
locality	Yes	A string. "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 " [ANZLICSTREET].
city	No	A string. City that the address is in.
stateOrProvince	Yes	A string. the State or Province that the address is in.
country	Yes	A string. Country that the address is in.
geographicSubAddress	No	A list of sub addresses (GeographicSubAddress [*]). Representation of a SubAddress It is used for addressing within a property in an urban area (country properties are often defined differently). It may refer to a building, a building cluster, or a floor of a multistory building.
@type	No	A string. Indicates the type of the resource referenced. Here can be 'UrbanPropertyAddress', ‘FormattedAddress’, ‘JapanesePropertyAddress’ , ‘AustralianPropertyAddress’, etc…

@schemaLocation	No	A string. A Link to the schema describing this REST Resource. The resource described 'UrbanPropertyAddress' but a schema could be used for other property address description.

GeographicSubAddress sub-resource

Representation of a SubAddress
It is used for addressing within a property in an urban area (country properties are often defined differently). It may refer to a building, a building cluster, or a floor of a multistory building.

Field	*Mandatory in API messages*	*Description*
type	No	A string. type of subAddress : it can be a subunit or a private street.
name	No	A string. Name of the subAddress to identify it with a meaningful identification.
subUnitType	No	A string. the type of subunit e.g.BERTH, FLAT, PIER, SUITE, SHOP, TOWER, UNIT, WHARF.
subUnitNumber	No	A string. the discriminator used for the subunit often just a simple number e.g. FLAT 5, may also be a range.
levelType	No	A string. describes level types within a building.
levelNumber	No	A string. used where a level type may be repeated e.g. BASEMENT 1, BASEMENT 2.
buildingName	No	A string. allows for buildings that have well-known names.
privateStreetName	No	A string. private streets internal to a property (e.g. a university) may have internal names that are not recorded by the land title office.
privateStreetNumber	No	A string. private streets numbers internal to a private street.
@type	No	A string. Type of the resource for thus subResource
@schemaLocation	No	A string. A Link to the schema describing the structure of this REST Resource to allow for extensions

IndividualIdentification sub-resource

Defines a proof of legal identification of an individual, such as a passport, a driver´s license, social security number, …

Field	*Mandatory in API messages*	*Description*
@type	Yes	A string. Identification type (passport, national identity number, driver´s license, …)
identificationId	Yes	A string. Unique identifier of the proof of individual identification
country	Yes	A string. Country where the identification was issued
issuingAuthority	No	A string. Authority that issued the legal identification (e.g.: social security, town hall, …)
issuingDate	No	A string (date-time). Date when identification was issued
@schemaLocation	No	A string. A Link to the schema describing the structure of this REST Resource to allow for extensions where specific proof of identification requires additional attributes

UserAsset sub-resource

Address reference. Defines an address and/or identifies an existing address entity

An address allows textual description of an existing place over the surface of the Earth

This resource could be invoked as reference or value

Field	*Mandatory in API messages*	*Description*
entityType	Yes	Type of managed asset (e.g.: customer, account, product, resource, service)
assetType	No	A string. Specific Type of the resource (e.g.: mobile line subscription, video platform license, mobile equipment, billingAccount) that can be managed by a user. Second level to define the type of managed element.
id	Yes	Identifier of the asset (within the entity/asset type pair, e.g.:customerId, accountId, mobile line number…)

role	No if entitlement is included	A string. Represents the part played by an individual in relation to being granted a set of entitlements for manageable assets (e.g.: owner, user, viewer, …). In order to use this attribute, the user roles must be defined in the system as specified in TMF632.
entitlement	No	List of entitlements (Entitlement[]) including information about individual entitlements to define access levels to operate over different functions that can be defined in an asset. If not included and no "role" attribute is provided, then the authorization will be understood for all functions and all actions (same as role of "owner").

Entitlement sub-resource

Defines information of individual access entitlement (access level authorization)

Field	*Mandatory in API messages*	*Description*
function	Yes	A string. Specific function that can be managed over a given asset (e.g.: all, account configuration, sport package, usage monitoring, ….)
action	Yes	A string. Level of access granted to the specific function (e.g.: all, read, write, read-and-write, view, record)

Json representation sample

The example below provides the json representation of a 'Userinfo' resource object

```
{
 "sub": "412d606f-4937-443b-b5e7-a8d0f63ef0bc",
 "name": "John Doe",
 "given_name": "John",
 "family_name": "Doe",
 "nickname": "Jonny",
```

```
"email": "johndoe@myserver.com",
"website": "http://johnsweb.com ",
"gender": "male",
"birthdate": "1970-02-20T00:00:00.000Z",
"legalId": [
 {
  "@type": "passport",
  "country": "England",
  "identificationId": "01234567BBC"
 }
],
"phone_number": "+447123456789",
"address": {
    "streetNr": "56",
    "streetName": "Arlington",
    "streetType": "Road",
    "postcode": "W45E02",
    "locality": "London",
    "city": "London",
    "stateOrProvince": "Great London",
    "country": "England",
},
"userAssets": [
 {
  "id": "cst123",
  "entityType": "customer",
 },
 {
  "id": "acc-ABC",
  "entityType": "account",
  "assetType": "billingAccount",
  "entitlements": [
   {
    "function": "billing",
    "action": "read"
   }
  ]
 },
 {
  "id": "7123456789",
  "entityType": "product",
  "assetType": "mobile",
  "role": "owner"
 },
 {
  "id": "7999333222",
  "entityType": "product",
  "assetType": "mobile",
```

```
    "entitlements": [
     {
      "function": "consumption",
      "action": "read"
     }
    ]
   }
 ],{
  "@type": "user",
  "@baseType": "user",
  "@schemaLocation": "https://www.somewhere.com/schemas/user"
 }
}
```


Notification Resource Models

No notifications are defined for this API

API OPERATION TEMPLATES

For every single of operation on the entities use the following templates and provide sample REST requests and responses.

Remember that the following Uniform Contract rules must be used:

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 Not required in this API
Partial Update of an Entity	PATCH Resource	PATCH must be used to partially update a resource Not required in this API
Complete Update of an Entity	PUT Resource	PUT must be used to completely update a resource identified by its resource URI Not required in this API
Remove an Entity	DELETE Resource	DELETE must be used to remove a resource Not required in this API

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

OPERATIONS ON USERINFO

RETRIEVE INDIVIDUAL´S IDENTITY INFORMATION

GET /openid/v1/userinfo

Description

This operation retrieves identity related information of an individual. The individual whose identity information is requested will be identified by the proof of authorization included (e.g.: OAuth2.0 token provided in Authorization header)

Attribute selection is enabled for all first level attributes.

Behavior:

Status Code	Description
200	the site information was returned successfully
401	Unauthorized. The requestor cannot request information of the user (e.g.: invalid token)
400	Request Error
500	The server encountered an unexpected condition which prevented it from fulfilling the request
Other	The server may use other HTTP error status codes to reflect the error, the client must be processed in accordance with the error messages in another HTTP specification.

Usage Samples

The example below includes the minimum attributes within the Userinfo resource model that must be included in the query response

REQUEST

```
GET https://{serverRoot}/openid/v1/userinfo
Accept: application/json
Authorization: Bearer <user-specific-token>
```

RESPONSE

```
200
Content-Type: application/json
{
 "sub": "412d606f-4937-443b-b5e7-a8d0f63ef0bc",
 "name": "John Doe"
 }
```

The example below includes the typical attributes within the Userinfo resource model that can be included in the query response when no user assets information is provided

REQUEST

```
GET https://{serverRoot}/openid/v1/userinfo
Accept: application/json
Authorization: Bearer <user-specific-token>
```

RESPONSE

```
200
Content-Type: application/json
{
 "sub": "412d606f-4937-443b-b5e7-a8d0f63ef0bc",
 "name": "John Doe",
 "given_name": "John",
 "family_name": "Doe",
 "nickname": "Jonny",
 "email": "johndoe@myserver.com",
 "website": "http://johnsweb.com ",
 "gender": "male",
 "birthdate": "1970-02-20T00:00:00.000Z",
 "legalId": [
```

```
  {
    "@type": "passport",
    "country": "England",
    "identificationId": "01234567BBC"
  }
 ],
 "phone_number": "+447123456789",
 "address": {
      "streetNr": "56",
      "streetName": "Arlington",
      "streetType": "Road",
      "postcode": "W45E02",
      "locality": "London",
      "city": "London",
      "stateOrProvince": "Great London",
      "country": "England",
 }
}
```


The example below includes the attributes within the Userinfo resource model that can be included in the query response when assets information is provided

REQUEST

```
GET https://{serverRoot}/openid/v1/userinfo
Accept: application/json
Authorization: Bearer <user-specific-token>
```

RESPONSE

```
200
Content-Type: application/json
{
 "sub": "412d606f-4937-443b-b5e7-a8d0f63ef0bc",
 "name": "John Doe",
 "phone_number": "+447123456789",
 "userAssets": [
```

```
  {
   "id": "7123456789",
   "entityType": "product",
   "assetType": "mobile",
   "entitlements": [
    {
     "function": "plan",
     "action": "read-and-modify"
    },
    {
     "function": "consumption",
     "action": "read"
    }
   ]
  }
 ]
}
```


The example below includes the attributes within the Userinfo resource model that can be included in the query response when assets information is provided and entitlements are provided via a user role. In order to use this approach, the user roles must be defined in teh system as specified in TMF632.

REQUEST

```
GET https://{serverRoot}/openid/v1/userinfo
Accept: application/json
Authorization: Bearer <user-specific-token>
```

RESPONSE

```
200
Content-Type: application/json
{
 "sub": "412d606f-4937-443b-b5e7-a8d0f63ef0bc",
 "name": "John Doe",
```

```
 "phone_number": "+447123456789",
  {
   "id": "7123456789",
   "entityType": "product",
   "assetType": "mobile",
   "role": "owner"
  },
  {
   "id": "7999333222",
   "entityType": "product",
   "assetType": "mobile",
   "role": "user"
  }
 ]
}
```

ACKNOWLEDGEMENTS

VERSION HISTORY

Version Number	Date	Release led by:	Description
1.01	07/03/2018	Luis Velarde (Telefónica)	First Release of Draft Version of the Document.
1.0.2	29-Jun-2018	Adrienne Walcott	Formatting/style edits prior to R18 publishing.

RELEASE HISTORY

Release Number	Date	Release led by:	Description
Release 18.0.0	07/03/2018	Luis Velarde (Telefónica)	Initial Release

CONTRIBUTORS TO DOCUMENT

Luis Velarde Guillermo Martínez	Telefonica
Steve Harrop	Vodafone
Pierre Gauthier	TM Forum