**User Roles & Permissions API
Conformance Profile**

**Document Number TMF672B**

**June 2017**

|  |  |
| --- | --- |
| **Release: Frameworx Release 17.0** | **Status: Member Evaluation** |
| **Version: 1.0.3** | **IPR Mode: RAND** |

# 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](http://www.tmforum.org/IPRPolicy/11525/home.html), 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](#_Toc497296531)

[Table of Contents 3](#_Toc497296532)

[List of Tables 4](#_Toc497296533)

[Introduction 5](#_Toc497296534)

[API DESCRIPTION 6](#_Toc497296535)

[RESOURCE MODEL CONFORMANCE 7](#_Toc497296536)

[UserRoles&Permissions API MANDATORY AND OPTIONAL RESOURCES 7](#_Toc497296537)

[Permission resource MANDATORY AND OPTIONAL ATTRIBUTES 7](#_Toc497296538)

[UserRole resource MANDATORY AND OPTIONAL ATTRIBUTES 9](#_Toc497296539)

[API OPERATIONS CONFORMANCE 11](#_Toc497296540)

[UserRoles&Permissions MANDATORY AND OPTIONAL OPERATIONS 11](#_Toc497296541)

[API GET FILTERING OPERATION CONFORMANCE 12](#_Toc497296542)

[Filtering in Permission resource 12](#_Toc497296543)

[GET /usersandroles/v1/permission 13](#_Toc497296544)

[GET /usersandroles/v1/permission/{permissionId} 13](#_Toc497296545)

[API POST OPERATION CONFORMANCE 14](#_Toc497296546)

[POST /usersandroles/v1/permission 14](#_Toc497296547)

[API PUT OPERATION CONFORMANCE 18](#_Toc497296548)

[API PATCH OPERATION CONFORMANCE 19](#_Toc497296549)

[API DELETE OPERATION CONFORMANCE 20](#_Toc497296550)

[API CONFORMANCE TEST SCENARIOS 21](#_Toc497296551)

[Permission resource TEST CASES 21](#_Toc497296552)

[UserRole resource TEST CASES 28](#_Toc497296553)

# List of Tables

**No table of figures entries found.**

# Introduction

The following document is the REST API Conformance for the UserRoles & Permissions API.

# API DESCRIPTION

The UserRoles & Permissions API covers operations to manage permissions granted by one individual (granter) to other individual (users) in order to allow accessing assets (products/services) owned by the granter, …

A typical use case that can be enabled with this API is the one where a user that owns a TV service can grant access to another user in order to make use of some of the functions within the service, for instance view only children movies, configure TV service or view documentaries over that TV service owned by the grantee

# RESOURCE MODEL CONFORMANCE

## UserRoles&Permissions API MANDATORY AND OPTIONAL RESOURCES

|  |  |  |
| --- | --- | --- |
| Resource Name | Mandatory or Optional | Comments |
| Permission | M |  |
| User Role | O | The description of resources accessible by an user may be managed independently per user instead of via a user role assigned to an user |

## Permission resource MANDATORY AND OPTIONAL ATTRIBUTES

|  |  |  |
| --- | --- | --- |
| Attribute Name | Mandatory or Optional | Comments |
| id | M (in response messages)O (otherwise) | Generated by the server and provided in the response upon resource creation.Accepted in entity-creation requests if the server supports the incoming identifier as the reference to create new resources |
| href | M (in response messages)O (otherwise) | Value in response must be the same as the one set in Location header provided upon entity creation |
| date | M (in response messages)O (otherwise) | Only required in response messages from the server |
| description | O |  |
| period | M |  |
|  | startDateTime | M | If null indicates from current time |
|  | endDateTime | O | If not included indicates for ever |
| user | M |  |
|  | id | M |  |
|  | href | M (in response messages)O (otherwise) |  |
|  | name | O |  |
| granter | M (in response messages)O (otherwise) | Only required in response messages from the server |
|  | id | M |  |
|  | href | M (in response messages)O (otherwise) |  |
|  | name | O |  |
| privilege | M (if not assetUserRoles included)O (otherwise) | Array of structures Mandatory if not assetUserRoles included, including at least one entity |
|  | manageableAsset | M |  |
|  |  | id | M |  |
|  |  | href | O |  |
|  |  | entityType | M |  |
|  | function | O |  |
|  | action | M |  |
| assetUserRole | M (if not privileges included)O (otherwise) | Array of structures Mandatory if not privileges included, including at least one entity |
|  | manageableAsset | M |  |
|  |  | id | M |  |
|  |  | href | O |  |
|  |  | entityType | M |  |
|  | userRole | M |  |
|  |  | id | M |  |
|  |  | href | M (in response messages)O (otherwise) |  |
|  |  | role | O |  |

## UserRole resource MANDATORY AND OPTIONAL ATTRIBUTES

Since support for UserRole resource is optional and not included in the basic CONNECT certification this is not applicable in this conformance document

# API OPERATIONS CONFORMANCE

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

## UserRoles&Permissions MANDATORY AND OPTIONAL OPERATIONS

|  |  |  |
| --- | --- | --- |
| Uniform API Operation | Mandatory/Optional | Comments |
| GET  | M | GET must be used to retrieve a representation of a resource |
| POST  | M | POST must be used to create a new resource |
| PUT  | O | PUT must be used to completely update a resource identified by its resource URI  |
| PATCH (JSON-PATH) | O | PATCH must be used to partially update a resource |
| DELETE  | O | DELETE must be used to remove 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.:?category=<value> &isBundle=<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=category,channel) 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

## Filtering in Permission 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 |
| date | O | NA | M | NA |
| description | O | NA | M | NA |
| period | NA | O | M | O |
| user | NA | M (.id)O (other) | M | O |
| granter | NA | M (.id)O (other) | M | O |
| privilege | NA | O | M | O |
| assetUserRole | NA | O | M | O |

## GET /usersandroles/v1/permission

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

* date: to obtain the list of permissions that have been created on a given date (it can be expanded to define date ranges startDate/endDate)
* user.id: To obtain the list of permissions that have been granted to a given user
* granter.id: To obtain the list of permissions that have been granted by a given user
* other optional attributes as defined in the table above

**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 Permission resource definition

## GET /usersandroles/v1/permission/{permissionId}

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

* Any of the attributes in thefirst level of Permission resource definition

# API POST OPERATION CONFORMANCE

## POST /usersandroles/v1/permission

This Uniform Contract operation is used to create a Permission resource in the server.

The response to this operation must include a Location header set to /usersandroles/v1/permissions/{permissionId} where {permissionId} indicates the identifier assigned by the server to the new Permission resource created

|  |  |  |
| --- | --- | --- |
| POST | M |  |
| Response Status Code 201 | M  |  |
| Other Status Codes | NA |  |

The following table indicates attributes that are required to be sent when creating a new Permission resource as well as attributes with special considerations. All other attributes defining the resource are not required to be sent as part of the BODY of the POST request message:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute name | Mandatory | Default | Rule |
| period | Y |  |  |
| period.startDateTime | Y |  | It can be null |
| user.id | Y |  |  |
| privileges | N |  | Mandatory if not assetUserRoles included At least one entity in the array |
| privilege.manageableAsset | N |  | Mandatory if privileges included |
| privilege.manageableAsset.id | N |  | Mandatory if privileges included |
| privilege.manageableAsset.entityType | N |  | Mandatory if privileges included |
| privilege.action | N |  | Mandatory if privileges included |
| assetUserRole | N |  | Mandatory if not privileges included At least one entity in the array |
| assetUserRole.manageableAsset | N |  | Mandatory if assetUserRoles included |
| assetUserRole.manageableAsset.id | N |  | Mandatory if assetUserRoles included |
| assetUserRole.manageableAsset.entityType | N |  | Mandatory if assetUserRoles included |
| assetUserRole.userRole | N |  | Mandatory if assetUserRoles included |
| assetUserRole.userRole.id | N |  | Mandatory if assetUserRoles included |

The response from the server must include a BODY with the contents of the new resource created

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.

The following parameters must be supported by the server when included in the request to create a new resource

* date
* description
* period
* user
* user.id
* user.name
* granter
* granter.id
* granter.name
* privilege
* privilege.manageableAsset
* privilege.manageableAsset.id
* privilege.manageableAsset.entityType
* privilege.function
* privilege.action

The BODY of the response from the server must include attribute “href” set to the same value as the one in the Location header.

The BODY of the response from the server must include the following attributes that are mandatory in the definition of a Permission as per the resource model defined even if they are not included in the request

* id
* href
* date
* granter
* granter.href

# API PUT OPERATION CONFORMANCE

This Uniform Contract operation is used to completely update a Permission resource existing in the server.

Since PUT operation is optional and not included in the basic CONNECT certification this is not applicable in this conformance document.

# API PATCH OPERATION CONFORMANCE

This section defines which attributes are patchable.

Since PATCH operation is optional and not included in the basic CONNECT certification this is not applicable in this conformance document.

# API DELETE OPERATION CONFORMANCE

This section defines what operations can be used to delete a Product Offering or Product Specification resource.

Since DELETE operation is optional and not included in the basic CONNECT certification this is not applicable in this conformance document

# API CONFORMANCE TEST SCENARIOS

This section describes the test scenarios required for the basic CONNECT certification of UserRoles&Permissions 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}/usersandroles/v1**, being {serverRoot} defines the certification endpoint

## Permission resource TEST CASES

**Nominal Scenarios**

**TC\_Prmsn\_N1 – Create new single Permission**

* Send a POST message to {apiRoot}/permission/ with the following contents in the BODY

{
 "period":

 {

 "startDateTime":"<any value with correct datetime format>",

 "endDateTime": "<any value with correct datetime format>"

 },

 "user":

 {

 "id"="u123"

 },

 "privilege":

 [

 {

 "manageableAsset":

 {

 "id"= "Asset987",

 "entityType"="IPTV license"

 },

 "action":"R&W"

 },

 {

 "manageableAsset":

 {

 "id"= "Asset987",

 "entityType"="IPTV license"

 },

 "function":"Sport basic package",

 "action":"watch"

 },

 {

 "manageableAsset":

 {

 "id"="Asset123",

 "entityType"="mobile line"

 },

 "action":"R/O"

 }

 ]

}

* 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}/permission/{Prmsn01} where {Prmsn01} indicates the identifier assigned by the server to the new Permission 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}/permission/
* Wait for a response from the server with the following characteristics
	+ Response Code 200-OK
	+ The body of the response includes one Permission resource with ID set to {Prmsn01}, 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 { Prmsn01} matches the values in the original request
* Send a GET message to /{apiRoot}/permission/{Prmsn01}
* 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 Permission resource structure that matches the values in the original request

**TC\_ Prmsn \_N2 – Search for Permission with specific characteristics**

* Send a POST message to {apiRoot}/permission/ with the following contents in the BODY

{
 "description"="this is the second permission",

 "period":

 {

 "startDateTime":"<any value with correct datetime format>",

 "endDateTime": "<any value with correct datetime format>"

 },

 "user":

 {

 "id"="u555"

 },

 "granter":

 {

 "id"="u444"

 },

 "privilege":

 [

 {

 "manageableAsset":

 {

 "id"="Asset555",

 "entityType"="mobile line"

 },

 "action":"R&W"

 }

 ]

}

* 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}/permission/{Prmsn02} where {Prmsn02} indicates the identifier assigned by the server to the new Permission 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}/permission
* Wait for a response from the server with the following characteristics
	+ Response Code 200-OK
	+ The body of the response includes two Permission resources, referring to {Prmsn01} and {Prmsn02}
	+ 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}/permission/?user.id=u123
* Wait for a response from the server with the following characteristics
	+ Response Code 200-OK
	+ The body of the response includes one Permission resource, referring to {Prmsn01}
	+ The response message includes all mandatory parameters
	+ The body of the response for the resource with identifier {Prmsn01} matches the values in the original request
* Send a GET message to /{apiRoot}/permission/?user.id=u555
* Wait for a response from the server with the following characteristics
	+ Response Code 200-OK
	+ The body of the response includes one Permission resource, referring to {Prmsn02}
	+ The response message includes all mandatory parameters
	+ The body of the response for the resource with identifier {Prmsn02} matches the values in the original request

**TC\_Prmsn\_N3 – Filtered retrieval of Permission data**

* Send a GET message to /{apiRoot}/permission/{Prmsn02}?fields=period,description
* Wait for a response from the server with the following characteristics
	+ Response Code 200-OK
	+ The body of the response includes one Permission resource, referring to { Prmsn02} and including only attributes date and description, matching the values in the original request
* Send a GET message to /{apiRoot}/permission/{Prmsn02}?fields=period,user
* Wait for a response from the server with the following characteristics
	+ Response Code 200-OK
	+ The body of the response includes one ProductOffering resource, referring to {Prmsn02} and including only attributes period and user, matching the values in the original request

Notice that this test case is using parameters ” period”, ”user” and ”description” to filter the data included in the response but any other parameter could be used

**TC\_Prmsn\_N4 – Filtered Search and Filtered data response**

* Send a GET message to /{apiRoot}/permission/?user.id=555&fields= period,user,granter
* Wait for a response from the server with the following characteristics
	+ Response Code 200-OK
	+ The body of the response includes one Permission resource, referring to {Prmsn02}
	+ The body of the response for the resource with each identifier includes only attributes period, user and granter, matching the values in the corresponding original request

Notice that this test case is using parameters ” period” , ”user” and ”granter” to filter the data included in the response but any other parameter could be used

**TC\_ProdOff\_N5 – Parameter not included in permission request but provided in response**

* Send a POST message to {apiRoot}/permission/ with the following contents in the BODY

{
 "description"="this is the third permission",

 "period":

 {

 "startDateTime":"<any value with correct datetime format>",

 "endDateTime": "<any value with correct datetime format>"

 },

 "user":

 {

 "id"="u888"

 },

 "privilege":

 [

 {

 "manageableAsset":

 {

 "id"="Asset888",

 "entityType"="mobile line"

 },

 "action":"R&W"

 }

 ]

}

* 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 Permission resource
	+ The body of the response includes parameters ”date” and ”href”

**Error Scenarios**

**TC\_Prmsn\_E1 – Unknown Permission**

* Send a GET message to /{apiRoot}/permission/{Prmsn04}, where {Prmsn04} does not match any of the identifiers previously creted in the server
* Wait for a response from the server with the following characteristics
	+ Response Code 404-Not Found

**TC\_Prmsn\_E2 – Invalid Request – Missing mandatory parameter**

* Send a POST message to {apiRoot}/permission/ with the following contents in the BODY.

{
 "user":

 {

 "id"="u555"

 },

 "granter":

 {

 "id"="u444"

 },

 "privilege":

 [

 {

 "manageableAsset":

 {

 "id"="Asset555",

 "entityType"="mobile line"

 },

 "action":"R&W"

 }

 ]

}

Notice that this request is missing mandatory parameter ”period” but any other mandatory parameter could be used

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

**TC\_Prmsn\_E3 – Invalid Request – Missing parameter mandatory in context**

* Send a POST message to {apiRoot}/permission/ with the following contents in the BODY.

{
 "period":

 {

 "startDateTime":"<any value with correct datetime format>",

 "endDateTime": "<any value with correct datetime format>"

 },

 "user":

 {

 "id"="u999"

 },

 "privilege":

 [

 {

 "manageableAsset":

 {

 "id"="Asset555",

 "entityType"="mobile line"

 }

 }

 ]

}

Notice that this request is missing mandatory parameter “action” when parameter “privilege” is included but any other parameter that becomes mandatory based on the context could be used

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

## UserRole resource TEST CASES

Since support for UserRole resource is optional and not included in the basic CONNECT