

# TM Forum Specification

## Process Flow Management API Conformance Profile

**TMF701B**

**Release 19.0.1**

**November 2019**

<b>Release: Release 19.0.1</b>	<b>TM Forum Approved</b>
<b>Version: 4.0.1</b>	<b>IPR Mode: RAND</b>

## NOTICE

Copyright © TM Forum 2019. 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.

TM FORUM invites any TM FORUM Member or any other party that believes it has patent claims that would necessarily be infringed by implementations of this TM Forum Standards Final Deliverable, to notify the TM FORUM Team Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the TM FORUM Collaboration Project Team that produced this deliverable.

The TM FORUM invites any party to contact the TM FORUM Team Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this TM FORUM Standards Final Deliverable by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the TM FORUM Collaboration Project Team that produced this TM FORUM Standards Final Deliverable. TM FORUM may include such claims on its website but disclaims any obligation to do so.

TM FORUM takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this TM FORUM Standards Final Deliverable or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on TM FORUM's procedures with respect to rights in any document or deliverable produced by a TM FORUM Collaboration Project Team can be found on the TM FORUM website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this TM FORUM Standards Final Deliverable, can be obtained from the TM FORUM Team Administrator. TM FORUM makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

Direct inquiries to the TM Forum office:

4 Century Drive, Suite 100

Parsippany, NJ 07054 USA

Tel No. +1 973 944 5100

Fax No. +1 973 998 7916

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

# Table of Contents

NOTICE .....	2
Table of Contents.....	4
INTRODUCTION - API DESCRIPTION.....	5
RESOURCE MODEL CONFORMANCE.....	6
API MANDATORY AND OPTIONAL RESOURCES .....	6
ProcessFlow MANDATORY AND OPTIONAL ATTRIBUTES .....	6
TaskFlow MANDATORY AND OPTIONAL ATTRIBUTES .....	8
NOTIFICATION MODEL CONFORMANCE .....	11
ProcessFlow MANDATORY AND OPTIONAL NOTIFICATIONS .....	11
TaskFlow MANDATORY AND OPTIONAL NOTIFICATIONS .....	11
API OPERATIONS CONFORMANCE.....	12
ProcessFlow MANDATORY AND OPTIONAL OPERATIONS.....	12
TaskFlow MANDATORY AND OPTIONAL OPERATIONS.....	13
API GET FILTERING OPERATION CONFORMANCE.....	14
All the GET operations in this API share the state status code pattern: .....	14
/processFlowManagement/v4/processFlow?fields=...&{filtering}.....	14
/processFlowManagement/v4/processFlow/{ID}?fields=.....	15
/processFlowManagement/v4/ processFlow/{ID}/taskFlow?fields=...&{filtering} .....	15
/processFlowManagement/v4/ processFlow/{ID}/taskFlow/{ID}?fields=.....	15
API POST OPERATION CONFORMANCE .....	16
/processFlowManagement/v4/processFlow/ .....	16
API PATCH OPERATION CONFORMANCE.....	18
/processFlowManagement/v4/ processFlow/{ID}/taskFlow/ .....	18
API DELETE OPERATION CONFORMANCE.....	19
API CONFORMANCE TEST SCENARIOS.....	20
ProcessFlow resource TEST CASES.....	20
TaskFlow resource TEST CASES.....	27
Document History.....	32
Version History .....	32
Release History .....	32
Contributors to Document.....	32

## INTRODUCTION - API DESCRIPTION

This document is the REST API conformance profile for the Process Flow Management REST API (TMF 701).

The ProcessFlow API allows management of business process. It provided all required information to achieve business task requiring manual action:

- A ProcessFlow will describe an orchestration of TaskFlow
- In event-based architecture the processFlow are triggered as consequence of event
- TaskFlow could be completed automatically (rules, event triggered, process delegation) or requiring manual action
- Operations on taskFlow allow updating taskFlow

ProcessFlow API manages ProcessFlow and TaskFlow resource:

- A ProcessFlow represents an instance of a pre-designed ProcessFlow type described in in BPMN engine or hard coded. It defines a start point, a sequence of TaskFlow (that could be dynamically instantiated) and end point(s).
- A TaskFlow is a “work” to be done within a ProcessFlow. It could an automated work (send an email, execute a business rule, wait for an event triggered by an external system) or a human work (a form to be completed). A taskFlow is an instantiation of a taskFlow type (defined design-time).

## 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
processFlow	M	
taskFlow	M	

### ProcessFlow MANDATORY AND OPTIONAL ATTRIBUTES

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

Attribute Name	Mandatory or Optional	Comments
id	M (in response messages) O (otherwise)	Initially generated by the server and provided in the response upon resource creation. Not available in POST request
href	M (in response messages) O (otherwise)	Initially generated by the server and provided in the response upon resource creation Value in response must be the same as the one set in Location header provided upon entity creation Not available in POST request
processFlowDate	M (in response messages)	Not available in POST request
processFlowType	O	Must be provided in POST request
state	M (in response messages) O (otherwise)	Only required in response messages from the server when providing the Process Flow entity contents (responses to create-POST and read-GET the entity)
@baseType	O	
@type	O	
@schemaLocation	O	

Attribute Name	Mandatory or Optional	Comments
channel	O	Array of channel where this ProcessFlow is executed
id	M	
href	O	
name	O	
@baseType	O	
@schemaLocation	O	
@type	O	
@referredType	O	
characteristic	O/M	Array of process flow characteristic(s)
name	M	
valueType	O	string is considered by default
value	M	This attribute type is any in the swagger
@baseType	O	
@schemaLocation	O	
@type	O	
relatedEntity	O	Array of entity in relation with this process flow
id	M	
href	O	
name	O	
role	M	
@baseType	O	
@schemaLocation	O	
@type	O	
@referredType	M	
relatedParty	O/M	Array of related Party related to this process flow
id	M	
href	O	
role	M/O	The role is mandatory if the id is referring a party resource. The role is optional is the id is referring a partyRole resource.
name	O	
@referredType	M	It allows to get id type : partyRole or Party or Party specialization: individual or organization
@baseType	O	

Attribute Name	Mandatory or Optional	Comments
@schemaLocation	O	
@type	O	
taskFlow	O Must not be provided in POST request	An array of task flow to be or executed for this process flow.
id	M	
href	O	
name	O	
@baseType	O	
@schemaLocation	O	
@type	O	
@referredType	O	

## TaskFlow MANDATORY AND OPTIONAL ATTRIBUTES

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

Note: TaskFlow is a ProcessFlow subResource (processFlow id is in the path)

Attribute Name	Mandatory or Optional	Comments
id	M (in response messages)	Initially generated by the server and provided in response messages.
href	M (in response messages)	Initially generated by the server and provided in response messages
completionMethod	O	
isMandatory	O	By default false
priority	O	
taskFlowDate	M	
taskFlowType	O	Must be provided in POST request
state	M (in response messages)	Only required in response messages from the server.
@baseType	O	
@type	O	
@schemaLocation	O	
channel	O	Array of channel where this task flow is executed
id	M	
href	O	



Attribute Name	Mandatory or Optional	Comments
name	O	
@baseType	O	
@schemaLocation	O	
@type	O	
@referredType	O	
characteristic	O/M	Array of task flow characteristic(s)
name	M	
valueType	O string is considered by default	
value	M	This attribute type is any in the swagger
@baseType	O	
@schemaLocation	O	
@type	O	
relatedEntity	O	Array of entity in relation with this task flow
id	M	
href	O	
name	O	
role	M	
@baseType	O	
@schemaLocation	O	
@type	O	
@referredType	M	
relatedParty	O/M	Array of related Party related to this task flow
id	M	
href	O	
role	M/O	The role is mandatory if the id is referring a party resource. The role is optional is the id is referring a partyRole resource.
name	O	
@referredType	M	It allows to get id type : partyRole or Party or Party specialization: individual or organization
@baseType	O	
@schemaLocation	O	
@type	O	
taskFlowRelationship	O	An array of taskFlowRelationship
relationshipType	M	

Attribute Name	Mandatory or Optional	Comments
@baseType	O	
@schemaLocation	O	
@type	O	
taskFlow (in taskFlowRelationship)	M	A task flow targeted by a relationship.
id	M	
href	O	
name	O	
@baseType	O	
@schemaLocation	O	
@type	O	
@referredType	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.

### ProcessFlow MANDATORY AND OPTIONAL NOTIFICATIONS

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

Here's an example of a request for registering a listener:

Notification Name	Mandatory or Optional	Comments
processFlowCreateEvent	M	
processFlowStateChangeEvent	M	
processFlowDeleteEvent	O	
processFlowAttributeValueChangeEvent	O	

All attributes of the resource associated with the notification are mandatory.

### TaskFlow MANDATORY AND OPTIONAL NOTIFICATIONS

Here's an example of a request for registering a listener:

Notification Name	Mandatory or Optional	Comments
taskFlowCreateEvent	M	
taskFlowStateChangeEvent	M	
taskFlowDeleteEvent	O	
taskFlowInformationRequired	M	
taskFlowAttributeValueChange	O	

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.

### ProcessFlow MANDATORY AND OPTIONAL OPERATIONS

The following table indicates which ones are mandatory and which ones are optional for ProductOrder resource:

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
DELETE	O	DELETE must be used to remove a resource

**TaskFlow MANDATORY AND OPTIONAL OPERATIONS**

The following table indicates which ones are mandatory and which ones are optional for CancelProductOrder resource:

Uniform API Operation	Mandatory/Optional	Comments
GET	M	GET must be used to retrieve a representation of a resource
PATCH	M	PATCH must be used to update 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 filtering 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.: `processFlowType=<value> &state=<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=processFlowType,state`) will provide in the response only the values assigned to attributes category and state. Attribute selection capabilities are the same for collections retrieval and individual resource queries

All the GET operations in this API share the state status code pattern:

GET	M	
Response Status Code 200	M	With JSON appointment objects as body response.
Response Status Code 404	M	When resource can't be found
Other Status Codes	N/A	

## `/processFlowManagement/v4/processFlow?fields=...&{filtering}`

### Filtered Search:

Providing filtering criteria is mandatory to avoid too many answers retrieved. See example below:

- processFlowType
- state + processFlowDate interval (`processFlowDate > 11/4/2018 & processFlowDate > 11/9/2017`)
- ...

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

- Any of the attributes in the ProcessFlow resource definition

**/processFlowManagement/v4/processFlow/{ID}?fields=...**

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

- Any of the attributes in ProcessFlow resource definition

**/processFlowManagement/v4/  
processFlow/{ID}/taskFlow?fields=...&{filtering}****Filtered Search:**

Providing filtering criteria is mandatory to avoid too many answers retrieved. See example below:

- state
- taskFlowType
- processFlowId
- ...

**Filtered 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 TaskFlow resource definition

**/processFlowManagement/v4/ processFlow/{ID}/taskFlow/{ID}?fields=...**

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

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

## API POST OPERATION CONFORMANCE

### /processFlowManagement/v4/processFlow/

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

The response to this operation must include a Location header set to /processFlowManagement/processFlow/{ID} where {ID} indicates the identifier assigned by the server to the new processFlow 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 processFlow as well as attributes with special considerations:

Attribute name	Mandatory	Default	Rule
id	NA		Must not be in POST request
href	NA		Must not be in POST request
processFlowDate	NA		Must not be in POST request
processFlowType	M		
state	NA		Must not be in POST request
channel	O		If present an id must be provided
relatedParty	O		If present an id must be provided. If id is not a partyRole, the role is mandatory. @referredType is mandatory
characteristic	O		name and value are mandatory. by default, valueType as 'string' value.
relatedEntity	O		If present then id, role and @referredType must be provided.
taskFlow	NA		Must not be in POST request

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.: relatedParty.Name)

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

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

- id
- href
- state
- processFlowDate

We must note that some attributes must not be sending in the POST request and could be not valued in the POST answer:

- taskFlow

The BODY of the response from the server could include attribute “href” (or “reference”) under each one of the entities within the ProductOrder model that can be addressed individually and were included in the response. This applies to any of the following entities

- relatedParty
- relatedEntity
- taskFlow

## API PATCH OPERATION CONFORMANCE

### /processFlowManagement/v4/ processFlow/{ID}/taskFlow/

This section defines which attributes are patchable.

The PATCH operation is not supported for processFlow but it is a mandatory operation for taskFlow.

This Uniform Contract operation is used to update a TaskFlow resource in the server.

PATCH	M	
Response Status Code 200	M	
Other Status Codes	NA	

The following table indicates attributes that are (or not) patch-able

Attribute name	PATCH- eable	Rule
id	No	
href	No	
completionMethod	Yes	
taskFlowSpecification	No	
isMandatory	Yes	For Admin only
priority	Yes	For Admin only
state	Yes	For Admin only
channel	Yes	
relatedParty	Yes	
characteristic	Yes	
relatedEntity	Yes	
taskFlowRelationship	Yes	For Admin Only

The response from the server must include a BODY with the contents of the resource modified, filled with at least the same information elements that were included in the request and are supported by the server.

## API DELETE OPERATION CONFORMANCE

This section defines what operations can be used to delete a Process Flow resource (for admin reason)

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 ProcessFlow management 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}/tmf-api/processFlowManagement/v4, being {serverRoot} defines the certification endpoint

## ProcessFlow resource TEST CASES

### Nominal Scenarios

#### TC\_ProcessFlow\_N1 – Create new Process Flow

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

```
{
  "processFlowType": "orderCapture",
  "channel": [
    {
      "id": "1",
      "name": "TMFWebStore",
      "@type": "Channel"
    }
  ],
  "relatedParty": [
    {
      "id": "456-f69",
      "href": ".../partyManagement/v4/individual/456-f69",
      "name": "Jean Pontus",
      "role": "Customer",
      "@type": "RelatedParty",
      "@referredType": "Individual"
    }
  ],
  "@type": "ProcessFlow",
  "@baseType": "ProcessFlow"
}
```

- 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/processFlow/{IDPF1}` where `{IDPF1}` indicates the identifier assigned by the server to the new ProcessFlow resource
  - The response message includes all mandatory parameters (**including id, state, and processFlowDate that were not sent in the original request**)
  - The body of the response matches the values set in the original request
- Send a GET message to `/apiRoot/processFlow/{IDPF1}`
  - Wait for a response from the server with the following characteristics
    - Response Code 200-OK
    - The body of the response includes one ProcessFlow resource with ID set to `{IDPF1}`, 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 `{IDPF1}` matches the values set in the original request

#### TC\_ProcessFlow\_N2 – Create new ProcessFlow with minimum set of parameters supported by server

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

```
{
  "processFlowType": "orderCapture",
  "channel": [
    {
      "id": "2",
      "name": "TMFMobileApp",
      "@type": "Channel"
    }
  ],
  "relatedParty": [
    {
      "id": "899-yyu",
      "href": "../partyManagement/v4/individual/899-yyu",
      "name": "Sophie Bouleau",
      "role": "Customer",
      "@type": "RelatedParty",
      "@referredType": "Individual"
    }
  ],
  "@type": "ProcessFlow",
  "@baseType": "ProcessFlow"
}
```

}

- 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}/processFlow/{IDPF2}` where `{IDPF2}` indicates the identifier assigned by the server to the new ProcessFlow resource
  - The response message includes all mandatory parameters (including `processFlowDate` and state that were not sent in the original request)
  - The body of the response matches the values set in the original request
- Send a GET message to `/{apiRoot}/processFlow/{IDPF2}`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one ProcessFlow resource with ID set to `{IDPF2}`, 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 `{IDPF2}` matches the values set in the original request

#### TC\_ProcessFlow\_N3 – Search for ProcessFlow with specific characteristics

- Send a GET message to `/{apiRoot}/processFlow?processFlowType=orderCapture`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes at least two ProcessFlow resources referring to `{IDPF1}` and `{IDPF2}`
  - 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}/processFlow?processFlowType=orderCapture&relatedParty.id=456-f69`
- Wait for a response from the server with the following characteristics

- Response Code 200-OK
  - The body of the response includes one ProcessFlow resource referring to {IDPF1} and there is no reference to ProcessFlow resource {IDPF2}
  - The response message includes all mandatory parameters
  - The body of the response for the resource with identifier {IDPF1} matches the values in the original request
- Send a GET message to `/{apiRoot}/processFlow?processFlowType=orderCapture&channel.id=2`
  - Wait for a response from the server with the following characteristics
    - Response Code 200-OK
    - The body of the response includes one ProcessFlow resource referring to {IDPF2} and there is no reference to ProcessFlow resource {IDPF1}
    - The response message includes all mandatory parameters
    - The body of the response for the resource with identifier {IDPF2} matches the values in the original request

#### TC\_ProcessFlow\_N4 – Filtered retrieval of ProcessFlow

- Send a GET message to `/{apiRoot}/processFlow/{IDPF2}?fields=id,href,processFlowType,state`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one ProcessFlow resource referring to {IDPF2} and including only attributes id,href,externalId, priority and state:

```
{
  "id": "IDPF2",
  "href": "https://host:port/processFlowManagement/v4/processFlow/IDPF2",
  "processFlowType": "orderCapture",
  "state": "active"
}
```

- Send a GET message to `/{apiRoot}/processFlow/{IDPF1}?fields=id,state,taskFlow.id,taskFlow.name`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one ProcessFlow resource referring to {IDPF1} and including only attributes id, state, and an array of task flow with attributes: id and name:

```
{
  "id": "IDPF1",
  "taskFlow": [
    {
      "id": "4fd-ty56-gg55-1",
      "name": "selectMainOffer "
    }
  ],
  "state": "active",
}
```

Notice that this test case is using parameters "state", "id", "taskFlow.id" and "taskFlow.name" to filter the data included in the response but any other ProcessFlow parameter could be used

#### TC\_ProcessFlow\_N5 – Filtered Search and Filtered data response

- Send a GET message to `/{apiRoot}/processFlow/?processFlowType=orderCapture=orderCapture&fields=id,state,relatedParty.name`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one ProcessFlow resource referring to {IDPF1} and there is no reference to ProcessFlow resource {IDPF2}
  - The body of the response for the resource includes only attribute: id, state and relatedParty.name matching the value in the corresponding original request:



```
{
  "id": "IDPF1",
  "state": "active",
  "relatedParty": [
    {
      "name": "Jean Pontus",
    }
  ],
}
```

Notice that this test case is using the parameters `id`, `state` and `relatedParty.name` to filter the data included in the response but any other parameter could be used

### Error Scenarios

#### TC\_ProcessFlow\_E1 – Unknown process flow identifier

- Send a GET message to `{apiRoot}/processFlow/{IDPF3}`, where `{IDPF3}` 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

#### TC\_ProcessFlow\_E2 – Invalid Request – unexpected parameter

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

```
{
  "id": "1",
  "processFlowType": "orderCapture",
  "state": "active",
  "channel": [
    {
      "id": "1",
      "name": "TMFWebStore",
      "@type": "Channel"
    }
  ],
  "relatedParty": [
    {
      "id": "456-f69",
      "href": ".../partyManagement/v4/individual/456-f69",
      "name": "Jean Pontus",
    }
  ]
}
```

```

    "role": "Customer",
    "@type": "RelatedParty",
    "@referredType": "Individual"
  }
],
"@type": "ProcessFlow",
"@baseType": "ProcessFlow"
}

```

Notice that this request features unaccepting parameters as id and state

- Wait for an error response from the server indicating the unexpected parameters in the request

#### TC\_ProcessFlow\_E3 – Invalid Request – Missing parameter mandatory in context

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

```

{
  "channel": [
    {
      "id": "1",
      "name": "TMFWebStore",
      "@type": "Channel"
    }
  ],
  "relatedParty": [
    {
      "href": ".../partyManagement/v4/individual/456-f69",
      "name": "Jean Pontus",
      "role": "Customer",
      "@type": "RelatedParty",
      "@referredType": "Individual"
    }
  ],
  "@type": "ProcessFlow",
  "@baseType": "ProcessFlow"
}

```

Notice that this request is missing a processFlowType and a relatedParty.id in the request.

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

## TaskFlow resource TEST CASES

### Nominal Scenarios

#### TC\_TaskFlow\_N1 – Retrieve a taskFlow from its id

This test case needs to have processFlow & (sub) taskFlow pre-existing already on the server side:

```
{
  "id" : "4fd-ty56-gg55-3",
  "href": ".../processFlowManagement/v4/processFlow/4fd-ty56-gg55/taskFlow/4fd-ty56-gg55-3",
  "completionMethod": "automatic",
  "isMandatory": true,
  "priority": 0,
  "taskFlowType": "SCinit",
  "relatedEntity": [
    {
      "id": "qq12-gg22",
      "href": ".../shoppingCartManagement/v4/shoppingCart/qq12-gg22",
      "name": "web ShoppingCart for Jean Pontus",
      "role": "shoppingCart",
      "@referredType": "ShoppingCart"
    }
  ],
  "state": "completed",
  "taskFlowRelationship": [
    {
      "relationshipType": "nextTask",
      "taskFlow": {
        "id": "4fd-ty56-gg55-4",
        "href": ".../processFlowManagement/v4/processFlow/4fd-ty56-gg55/taskFlow/4fd-ty56-gg55-4",
        "name": "configureShoppingCart"
      }
    }
  ]
}
```

- Send a GET message to `{apiRoot}/processFlow/4fd-ty56-gg55/TaskFlow/4fd-ty56-gg55-3`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK

- The body of the response includes one TaskFlow resource with ID set to {4fd-ty56-gg55-3}.
- The response message includes all mandatory parameters
- The body of the response for the resource with identifier {4fd-ty56-gg55-3} matches the values set in the original request

### TC\_TaskFlow\_N2 – Search for TaskFlow with specific characteristics

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

```
{
  "id" : "4fd-ty56-gg55-4",
  "href": ".../processFlowManagement/v4/ processFlow/4fd-ty56-gg55/taskFlow/4fd-ty56-
gg55-4",
  "completionMethod": "userSelection",
  "isMandatory": true,
  "priority": 0,
  "processFlowId": "4fd-ty56-gg55",
  "taskFlowType": "configureShoppingCart",
  "relatedEntity": [
    {
      "id": "qq12-gg22",
      "href": ".../shoppingCartManagement/v4/shoppingCart/qq12-gg22",
      "name": "web ShoppingCart for Jean Pontus",
      "role": "shoppingCart",
      "@referredType": "ShoppingCart",
      "@type": "RelatedEntity"
    }
  ],
  "state": "active",
  "@type": "TaskFlow",
  "@baseType": "TaskFlow"
}
```

- Send a GET message to `{/apiRoot}/processFlow/4fd-ty56-gg55/TaskFlow/4fd-ty56-gg55-4`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one TaskFlow resource with ID set to {4fd-ty56-gg55-4}.
  - The response message includes all mandatory parameters
  - The body of the response for the resource with identifier {4fd-ty56-gg55-4} matches the values set in the original request

- Send a GET message to `{/apiRoot}/processFlow/4fd-ty56-gg55/taskFlow?processFlowId= 4fd-ty56-gg55`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes at least two TaskFlow resources referring to `{4fd-ty56-gg55-3}` and `{4fd-ty56-gg55-4}`
  - The body of the response for the resource with each identifier matches the values in the corresponding original request

### TC\_ProcessFlow\_N3 – Update an existing Task Flow

- Send a PATCH message to `{apiRoot}/processFlow/4fd-ty56-gg55/taskFlow/4fd-ty56-gg55-4` with the following contents in the BODY

```
{
  "id": "4fd-ty56-gg55-4",
  "completionMethod": "userInput",
  "taskFlowType": "SCconfig",
  "characteristic": [
    {
      "name": "cartItem",
      "valueType": "object",
      "@schemaLocation": "../../../../schema/characteristic/cartItem.json",
      "@type": "CartItem",
      "cartItem": {
        "id": "1",
        "action": "add",
        "quantity": 1,
        "status": "active",
        "productOffering": {
          "id": "45d-666",
          "name": "TMF mobile Line"
        }
      }
    },
    {
      "name": "cartItem",
      "valueType": "object",
      "@schemaLocation": "../../../../schema/characteristic/cartItem.json",
      "@type": "CartItem",
      "cartItem": {
        "id": "2",
```

```

        "action": "add",
        "quantity": 1,
        "status": "active",
        "productOffering": {
            "id": "489-uuy",
            "name": "TMF mobile plan"
        },
        "cartItemRelationship": [
            {
                "id": "1",
                "type": "reliesOn"
            }
        ]
    },
    {
        "name": "cartItem",
        "valueType": "object",
        "@schemaLocation": ".../schema/characteristic/cartItem.json",
        "@type": "CartItem",
        "cartItem": {
            "id": "3",
            "action": "add",
            "quantity": 1,
            "status": "active",
            "productOffering": {
                "id": "ze2-gt1",
                "name": "TMF Sim Only Offer"
            },
            "cartItemRelationship": [
                {
                    "id": "1",
                    "type": "reliesOn"
                }
            ]
        }
    }
]
}

```

- Wait for a response from the server with the following characteristics
  - Response Code 200-Updated
  - The response message includes all parameters (**including all characteristic updated**)
  - The body of the response matches the values set in the update request
- Send a GET message to `{/apiRoot}/processFlow/4fd-ty56-gg55/taskFlow/{ 4fd-ty56-gg55-4}`

- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one ProcessFlow resource with ID set to { 4fd-ty56-gg55-4},
  - The response message includes all taskFlow parameters
  - The body of the response for the resource matches the values set in the update request

## Document History

### Version History

Version Number	Date	Release led by:	Description
4.0.0	11-Aug-2019	Ludovic Robert	First Release of the Document.
4.0.1	04-Nov-2019	Adrienne Walcott	Updated to reflect TM Forum Approved Status

### Release History

Release Number	Date	Release led by:	Description
19.0.0	11-Aug-2019	Ludovic Robert	First Release of the Document.
19.0.1	04-Nov-2019	Adrienne Walcott	Updated to reflect TM Forum Approved Status

### Contributors to Document

Ludovic Robert	Orange
Pierre Gauthier	TM Forum
Henrique Rodrigues	TMForum