*Frameworx Specification*

 Appointment API

 REST Specification

 TMF621

 Release 14.5.1

 September 2015

|  |  |
| --- | --- |
| **Latest Update: Frameworx Release 14.5** | **TM Forum Approved** |
| **Version 1.3.5** | **IPR Mode: RAND** |

# NOTICE

Copyright © TM Forum 2015. 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.

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:

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](http://www.tmforum.org/)

# Table of Contents

[NOTICE 2](#_Toc433204193)

[Table of Contents 4](#_Toc433204194)

[List of Tables 5](#_Toc433204195)

[Introduction 6](#_Toc433204196)

[SAMPLE USE CASES 7](#_Toc433204197)

[RESOURCE MODEL 8](#_Toc433204198)

[Managed Entity and Task Resource Models 8](#_Toc433204199)

[UML model 17](#_Toc433204200)

[Notification Resource Models 18](#_Toc433204201)

[Appointment Status Change Notification 18](#_Toc433204202)

[Appointment reschedule Notification 18](#_Toc433204203)

[Appointment creation Notification 19](#_Toc433204204)

[Appointment delete Notification 21](#_Toc433204205)

[Appointment value change Notification 21](#_Toc433204206)

[API OPERATION for Appointments 23](#_Toc433204207)

[**1 - A party wants to check free periods into a calendar** 23](#_Toc433204208)

[**2- A party wants to create an appointment** 23](#_Toc433204209)

 [**3- A party wants to delete an appointment** 23](#_Toc433204210)

[**4** **- A party wants to update an appointment** 23](#_Toc433204211)

[**5 - A party wants to find appointments with criterias** 23](#_Toc433204212)

[**6 – A party wants to reschedule an appointment** 23](#_Toc433204213)

[POST /api/appointment/schedule 24](#_Toc433204214)

[POST api/appointment 28](#_Toc433204215)

[PATCH api/appointment/{id} 33](#_Toc433204216)

[GET api/appointment/{Id} 37](#_Toc433204217)

[GET api/appointment?{field selector}&{filtering expression} 40](#_Toc433204218)

[POST api/appointment/{id}/schedule 43](#_Toc433204219)

[API NOTIFICATION 46](#_Toc433204220)

# List of Tables

N/A

# Introduction

The following document is the specification of the REST API for appointment management. It includes the model definition as well as all available operations. Possible actions are to check free slots and, then, creating, updating and retrieving appointment.

The appointment API provides a standardized mechanism to book an appointment with all the necessary appointment characteristics. First, the API consists in searching free slots based on parameters, as for example a party. Then, the appointment is created. The appointment has characteristics such as nature of appointment, place of appointment…

Appointment API performs the following operations:

* Retrieval of free slots depending on filter criteria
* Creation of appointment
* Deletion of appointment
* Update of appointment
* Rescheduling of an appointment

# SAMPLE USE CASES

The following table maps out the UC case.

|  |
| --- |
| **UC** |
| Free slots are checked according to criteria |
| A new appointment is created |
| An existing appointment should be updated because the status has changed or party availabilities has changed  |
| An appointment or a collection of appointment should be retrieved |
| An existing appointment is deleted |
| An existing appointment has to be rescheduled |

# RESOURCE MODEL

## Managed Entity and Task Resource Models

## APPOINTMENT RESSOURCE

Structured textual way of describing what is an appointment.

An appointment is a meeting with several persons, in one place, in order to do an action (an intervention, a sale, …). This action has a root, for example a trouble ticket.

Example of the JSON representation of an appointment:

|  |
| --- |
| {  "id":"21",  "href":"<http://xxxx/appointment/21>",  "externalId":" anExternalIDIfNeeded432113",  "category":"intervention",  "description":" A useful text to describe the appointment…", "status":"Missed",  "creationDate":"2015-09-01T14:40:43.071Z",  "lastUpdate":"2015-09-01T14:40:43.071Z",  "startDate":"2015-09-01T14:00:43.071Z",  "endDate":"2015-09-01T16:00:43.071Z",  "alarm":true,  "alarmAction":"smsToCustomer",  "attachment":[  {  "href":"http://server/path/titi.pdf"  }  ],  "relatedParty": [ {  "id":"32",  "href":"[http://xxxxx/individual/ 32](http://xxxxx/individual/%2032)",  "role":"customer",  "name":"John Doe"  } ],  "address":{  "id":" jkfdjgkldjf ", "href" : "<http://xxx/address/jkfdjgkldjf>" },  "relatedObject":[  {  "involvement":" problemToSolve ",  "reference":"<http://xxxx/troubleTicket/789745465>"  }  "note":[  {  "date":" 2015-09-01T14:40:43.071Z ",  "author":"Arthur Ewans"  "text":" Alreay called the expert ",  }  ] } |
|  |

**Fields descriptions :**

Appointment details :

|  |  |
| --- | --- |
| **Field** | **Description** |
| id | Appointment unique identifier |
| href | Appointment unique identifier |
| externalID | Reference of an external identifier |
| description | Description of appointment |
| category | A business category: for example “intervention”, or in a finer grain “afterSalesIntervention”, orderDeliveryIntervention”... |
| status | Status corresponding to appointment lifecycle |
| creationDate | Date of creation of appointment |
| lastUpdate | Date of last update of appointment |
| startDate | Date of the beginning of appointment |
| endDate | Date of the end of appointment |
| alarm | Indicates if there is a reminder |
| alarmAction | The actions to be invoked when an alarm is triggered for all participants (send email, send sms, etc.) |
| relatedParty | Party who participates to appointment.It can be a person (customer, …), a team (intervention team, …)There are at least 2 parties involved in the appointment |
| relatedObject | It allows other resources to be linked to the appointment.For example, it can be an orderToDeliver,for an order, a problemToSolve for trouble ticket |
| attachment | URI attached to appointment |
| address | Place of appointment |
| note | Extra-information about the appointment |

Note details :

|  |  |
| --- | --- |
| **Field** | **Description** |
| date | date of the note |
| author | author of the note |
| text | text of the note |

## SEARCH TASK RESSOURCE

Example of the JSON representation of a SearchTask **input**:

This task has to be used to look for free slots before booking an appointment (cf. operations)

|  |
| --- |
|  {  "marketSegment":"B2C",  "favoriteAmpm":"PM", "weekNumber":"38", "startDate":"2015-09-01T14:00:43.071Z ", "endDate":"2015-09-01T16:00:43.071Z ", "category":"intervention", "limit":"10", "productSpecification":{  "id":" productSpec42"  },  "address":{  "id":" jkfdjgkldjf "  "href" : "<http://xxx/address/jkfdjgkldjf>" },  "relatedParty":{  "id":"32", "href":"[http://xxxxx/individual/ 32](http://xxxxx/individual/%2032)"  }  }  |

Example of the JSON representation of a SearchTask **output**:

|  |
| --- |
| { "freeSlot":[ { "startDate":"2015-09-01T14:00:43.071Z", "endDate":"2015-09-01T16:00:43.071Z" "relatedParty":{  "id":"32" "href":"[http://xxxxx/individual/ 32](http://xxxxx/individual/%2032)"  }  } ]} |

SearchTask **input** :

|  |  |
| --- | --- |
| **Field** | **Description** |
| marketSegment | The market segment linked to appointment |
| productSpecification | Product linked to appointment |
| favoriteAmpm | Favorite moment of the day for the party .Two values : AM or PM |
| weekNumber | The week where free slots are researched |
| startDate | The beginning date of the research |
| endDate | The end date of the research |
| category | Category of appointment |
| limit | Number of free slots researched |
| address | Address of appointment |
| relatedParty | Party who is the owner of the calendar on which we want to plan an appointment (for example : an intervention team) |

Searchtask output: **free slots**

FreeSlot is derivated from slot and slot is derivated from SID calendarEntry.

|  |  |
| --- | --- |
| **Field** | **Description** |
| startDate | Beginning of free slot |
| endDate | End of free slot |
| relatedParty | Party (with the associated role) available during this free slot (e.g. an intervention team) |

**Appointment Status**



The following table explains the meaning of the states.

|  |  |
| --- | --- |
| **State** | **Description** |
| Initialised | When an appointment is created, the status is ‘Initialised’ |
| Validated | When an appointment is confirmed by all parties, the status is ‘validated’ |
| Cancelled | When an appointment is not confirmed by, at least, one party, it is ‘cancelled’ |
| Attended | When an appointment took place and it is OK , the status is ‘attended’ |
| Missed | When an appointment took place and it is KO , the status is ‘missed’ |

**Ampm Status**

|  |  |
| --- | --- |
| **State** | **Description** |
| AM | Slot in the morning |
| PM | Slot in the afternoon |

##

## UML model



## Notification Resource Models

### Appointment Status Change Notification

|  |
| --- |
| {      "eventId":"00001",      "eventTime":"2013-04-19T16:42:25-04:00",      "eventType":"appointmentStatusChangeNotification",      "appointment": { "id":"21",  "href":"<http://xxxx/appointment/21>",  "status":"Missed"       }} |

### Appointment reschedule Notification

|  |
| --- |
| {      "eventId":"00001",      "eventTime":"2013-04-19T16:42:25-04:00",      "eventType":"appointmentRescheduleNotification",      "appointment": { "id":"21",  "href":"<http://xxxx/appointment/21>",  "startDate":"2015-09-01T14:00:00.071Z",        "endDate":"2015-09-01T16:00:00.071Z"      }} |

### Appointment category change Notification

|  |
| --- |
| {      "eventId":"00001",      "eventTime":"2013-04-19T16:42:25-04:00",      "eventType":"appointmentCategoryChangeNotification",      "appointment": { "id":"21",  "href":"<http://xxxx/appointment/21>",  "category":"intervention"      }} |

### Appointment creation Notification

|  |
| --- |
| {      "eventId":"00001",      "eventTime":"2013-04-19T16:42:25-04:00",      "eventType":"appointmentCreationNotification",      "appointment": {    "id" : "21",      "href" : "<http://xxxx/appointment/21>",      "externalId" : "anExternalIDIfNeeded432113",      "category" : "intervention",      "description" : "A useful text to describe the appointment…",      "status" : "initialised",      "startDate" : "2015-09-01T14:00:43.071Z",      "endDate" : "2015-09-01T16:00:43.071Z",      "alarm" : false,      "creationDate": "2015-08-28T08:00:43.071Z",    "lastUpdate": "2015-08-28T08:00:43.071Z ",    "attachment":[         {            "href" : "<http://myDocumentServer/myRep/myDocument.pdf>"         }      ],      "relatedParty" : [       {            "id" : "32",            "href" : "[http://xxxxx/individual/ 32](http://xxxxx/individual/%2032)",            "role" : "customer",            "name" : "John Doe"         }    ],      "address" : {         "id" : "jkfdjgkldjf",       "href" : "<http://xxx/address/jkfdjgkldjf>"    },      "relatedObject":[         {            "involvement" : "problemToSolve",            "reference" : "<http://xxxx/troubleTicket/789745465>"         }      ],      "note":[        {           "date":" 2015-09-01T14:40:43.071Z ",           "author":"Arthur Ewans"           "text":" Alreay called the expert ",        }     ]   }      }} |

### Appointment delete Notification

|  |
| --- |
| {      "eventId":"00001",      "eventTime":"2013-04-19T16:42:25-04:00",      "eventType":"appointmentDeleteNotification",      "appointment": { "id":"21",  "href":"<http://xxxx/appointment/21>",       }} |

### Appointment value change Notification

|  |
| --- |
| {      "eventId":"00001",      "eventTime":"2013-04-19T16:42:25-04:00",      "eventType":"appointmentValueChangeNotification",      "appointment": { "id":"21",  "href":"<http://xxxx/appointment/21>",  "alarm":"false"       }} |

# API OPERATION for Appointments

Summary of operations :

**1 - A party (customer service representative, customer, etc.) wants to check free periods into a calendar**

A party wants to book an appointment for a customer: the party checks free periods in a calendar.

This calendar can be one of a single person, or an aggregation of persons (a team).

In case of a team calendar, the party identification (competent/relevant team to perform an intervention) is realized via a context: a product specification (FTTH, Copper, etc.), a marketSegment (Pro / Residential), a place/localization, etc.

1. **- A party wants to create an appointment**

A party books a slot, this slot will be used to realize a task (an intervention, etc.) or to meet a customer service representative.

This booking is done on an organization calendar (a team, a shop, etc.).

1. **- A party wants to delete an appointment**

If, for example, the customer is not available anymore, the party can cancel the appointment.

1. **- A party wants to update an appointment**

If, for example, the customer is not available anymore, the party can update the appointment by changing the slot.

**5 - A party wants to find appointments with criterias**

A party can search all the appointments booked by a customer in a determined period for example.

**6 – A party wants to reschedule an appointment**

Avaliabilies of parties have changed. The appointment must be rescheduled.

## POST /api/freeSlot/search

This operation is used to retrieve relevant free slots, available to book an appointment on, and matching a set of criterias.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute name** | **Mandatory** | **Default** | **Rule** |
| marketSegment | Y |  |  |
| productSpecification | Y |  |  |
| startDate | N |  | StartDate is mandatory if weekNumber is empty.If startDate is filled, endDate must be filled |
| endDate | N |  | EndDate is mandatory if weekNumber is empty.If endDate is filled, startDate must be filled |
| weekNumber | N |  | WeekNumber is mandatory if startDate and endDate are empty |
| favoriteAmpm | N |  |  |
| relatedParty | N |  | When party is specified, either id or href must be filled |
| address | Y |  | Either id or href must be filled |
| category | Y |  |  |
| limit  | N | Value to define by the project |  |

Behavior :

* Return status codes
	+ 201 Created - the request was successful
	+ 400 Bad Request – error, for example to cover these functional error cases:
		- startDate must not be in the past
		- endDate must not be in the past
		- endDate must be superior to startDate

|  |
| --- |
| **REQUEST** |
| POST /api/freeSlot/searchContent-type: application/json{         "marketSegment" : "B2C",         "favoriteAmpm": "PM",       "weekNumber": "36",       "category": "intervention",       "limit": "5",       "productSpecification": {            "id":"productSpec42"         },         "address":{            "id":"78798744541",        "href" : "<http://xxx/address/jkfdjgkldjf>"       },       "relatedParty" : {           "id" : "32",           "href" : "[http://xxxxx/organisation/ 32](http://xxxxx/organisation/%2032)"        }}   |
| **RESPONSE** |
| 201Content-Type: application/json{   "freeSlot":[      {       "relatedParty" : {           "id" : "32",           "href" : "[http://xxxxx/organisation/ 32](http://xxxxx/individual/%2032)",         },         "startDate":"2015-09-01T14:00:00.071Z",         "endDate":"2015-09-01T16:00:00.071Z"      },      {       "relatedParty" : {           "id" : "32",           "href" : "[http://xxxxx/organisation/ 32](http://xxxxx/individual/%2032)",         },         "startDate":"2015-09-01T16:00:00.071Z",         "endDate":"2015-09-01T18:00:00.071Z"      },      {       "relatedParty" : {           "id" : "32",           "href" : "[http://xxxxx/organisation/ 32](http://xxxxx/individual/%2032)",         },         "startDate":"2015-09-01T14:00:00.071Z",         "endDate":"2015-09-01T16:00:00.071Z"      },      {       "relatedParty" : {           "id" : "32",           "href" : "[http://xxxxx/organisation/ 32](http://xxxxx/individual/%2032)",         }         "startDate":"2015-09-03T14:00:00.071Z",         "endDate":"2015-09-03T16:00:00.071Z"      },      {       "relatedParty" : {           "id" : "32",           "href" : "[http://xxxxx/organisation/ 32](http://xxxxx/individual/%2032)",         }         "startDate":"2015-09-05T14:00:00.071Z",         "endDate":"2015-09-05T16:00:00.071Z"      }   ]} |

## POST api/appointment

After checking free slots, this operation is used to create an appointment with all its characteristics.

POST should be used without specifying the id and the href, the system is in charge of

generating the id + href for the appointment.

When performing a POST, these are the allowed attributes (all attributes that are

not in that list MUST NOT be used during POST, they are calculated/defined by the system) :

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute name** | **Mandatory** | **Default** | **Rule** |
| externalId | N | N/A |  |
| category | Y |  |  |
| description | N | N/A |  |
| status | N | initialised |  |
| startDate | Y |  |  |
| endDate | Y |  |  |
| alarm | N | N/A |  |
| alarmAction | N | N/A | If alarm is false, alarmAction doesn’t appear |
| attachment.href | N | N/A |  |
| address | Y |  | Either id or href must be filled at least |
| relatedParty | Y |  | At least one party must be linked to the appointment (customer, …) |
| relatedParty.id | Y |  | Either id or href must be filled at least |
| relatedParty.href | Y |  | Either id or href must be filled at least |
| relatedParty.role | N |  |  |
| relatedParty.name | N |  |  |
| relatedObject | N | N/A | For example, orderToDeliver, problemToSolve for trouble ticket |
| relatedObject.involvement | N | N/A | If a relatedObject is selected, involvement and reference must be filled.  |
| relatedObject.reference | N | N/A | If a relatedObject is selected, involvement and reference must be filled.  |
| note | N | N/A |  |
| note.date | N | N/A |  |
| note.author | N | N/A |  |
| note.text | N | N/A |  |

Behavior :

* Return status codes
	+ 201 Created - the request was successful
	+ 400 Bad Request – error, for example to cover these functional use cases :
		- startDate must not be in the past
		- endDate must not be in the past
		- endDate must be superior to startDate

- appointment on a slot already booked

- status lifecycle is not respected

|  |
| --- |
| **REQUEST** |
| POST api/appointmentContent-type: application/json{     "externalId" : "anExternalIDIfNeeded432113",     "category" : "intervention",     "description" : "A useful text to describe the appointment…",     "status" : "initialised",     "startDate" : "2015-09-01T14:00:43.071Z",     "endDate" : "2015-09-01T16:00:43.071Z",     "alarm" : false,     "attachment":[        {           "href" : "<http://myDocumentServer/myRep/myDocument.pdf>"        }     ],     "relatedParty" : [      {           "id" : "32",           "href" : "[http://xxxxx/individual/ 32](http://xxxxx/individual/%2032)",           "role" : "customer",           "name" : "John Doe"        }   ],     "address" : {        "id" : "jkfdjgkldjf",      "href" : "<http://xxx/address/jkfdjgkldjf>"   },     "relatedObject":[        {           "involvement" : "problemToSolve",           "reference" : "<http://xxxx/troubleTicket/789745465>"        }     ]   "note":[  {  "date":" 2015-09-01T14:40:43.071Z ",  "author":"Arthur Ewans"  "text":" Alreay called the expert ",  }    ]  } |
| **RESPONSE** |
| 201Content-Type: application/json{     "id" : "21",     "href" : "<http://xxxx/appointment/21>",     "externalId" : "anExternalIDIfNeeded432113",     "category" : "intervention",     "description" : "A useful text to describe the appointment…",     "status" : "initialised",     "startDate" : "2015-09-01T14:00:43.071Z",     "endDate" : "2015-09-01T16:00:43.071Z",     "alarm" : false,     "creationDate": "2015-08-28T08:00:43.071Z",   "lastUpdate": "2015-08-28T08:00:43.071Z ",   "attachment":[        {           "href" : "<http://myDocumentServer/myRep/myDocument.pdf>"        }     ],     "relatedParty" : [      {           "id" : "32",           "href" : "[http://xxxxx/individual/ 32](http://xxxxx/individual/%2032)",           "role" : "customer",           "name" : "John Doe"        }   ],     "address" : {        "id" : "jkfdjgkldjf",      "href" : "<http://xxx/address/jkfdjgkldjf>"   },     "relatedObject":[        {           "involvement" : "problemToSolve",           "reference" : "<http://xxxx/troubleTicket/789745465>"        }     ]  "note":[  {  "date":" 2015-09-01T14:40:43.071Z ",  "author":"Arthur Ewans"  "text":" Alreay called the expert ",  }    ]  } |

## PATCH api/appointment/{id}

This operation can be used to update an appointment if information has changed.

This is also used to cancel an appointment by modifying the status. The new status is ‘cancelled’.

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Patchable** | **Rule** |
| id | N |  |
| href | N |  |
| externalId | N |  |
| category | Y |  |
| description | Y |  |
| status | Y | To manage the appointment process (cf. appointment lifecycle) |
| creationDate | N |  |
| lastUpdate | N |  |
| startDate | Y |  |
| endDate | Y |  |
| alarm | Y |  |
| alarmAction | Y |  |
| attachment.href | Y |  |
| relatedParty | Y | Only when the party is not the customer |
| address | N |  |
| relatedObject | Y |  |
| note | Y |  |

Behavior :

* Return status codes
	+ 200 OK - the request was successful
	+ 400 Bad Request – error, for example to cover these errors :\*
		- startDate must not be in the past
		- endDate must not be in the past
		- endDate must be superior to startDate

- appointment on a slot already booked

- status lifecycle is not respected

|  |
| --- |
| **REQUEST** |
| PATCH api/appointment/21Content-type: application/json{     "status":"Cancelled"} |
| **RESPONSE** |
| 201Content-Type: application/json{     "id" : "21",     "href" : "<http://xxxx/appointment/21>",     "externalId" : "anExternalIDIfNeeded432113",     "category" : "intervention",   "description" : "A useful text to describe the appointment…",  ***"status":"Cancelled"***    "creationDate": "2015-08-28T08:00:43.071Z",   "lastUpdate": "2015-08-28T08:00:43.071Z ",   "startDate" : "2015-09-01T14:00:43.071Z",     "endDate" : "2015-09-01T16:00:43.071Z",     "alarm" : false,     "attachment":[        {           "href" : "<http://myDocumentServer/myRep/myDocument.pdf>"        }     ],     "relatedParty" : [      {           "id" : "32",           "href" : "[http://xxxxx/individual/ 32](http://xxxxx/individual/%2032)",           "role" : "customer",           "name" : "John Doe"        }   ],     "address" : {        "id" : "jkfdjgkldjf",      "href" : "<http://xxx/address/jkfdjgkldjf>"   },     "relatedObject":[        {           "involvement" : "problemToSolve",           "reference" : "<http://xxxx/troubleTicket/789745465>"        }     ]   "note":[  {  "date":" 2015-09-01T14:40:43.071Z ",  "author":"Arthur Ewans"  "text":" Alreay called the expert ",  }    ]  } |

## GET api/appointment/{Id}

This operation is used to research an appointment by its identifier.

Note that collections can be retrieved via GET /api/appointment with no {ID}

|  |
| --- |
| **REQUEST** |
| GET api/appointment/21Accept: application/json |
| **RESPONSE** |
| 200Content-Type: application/json{     "id" : "21",     "href" : "<http://xxxx/appointment/21>",     "externalId" : "anExternalIDIfNeeded432113",     "category" : "intervention",   "description" : "A useful text to describe the appointment…",     "status":"Validated",     "creationDate": "2015-08-28T08:00:43.071Z",   "lastUpdate": "2015-08-28T08:00:43.071Z ",   "startDate" : "2015-09-01T14:00:43.071Z",     "endDate" : "2015-09-01T16:00:43.071Z",     "alarm" : false,     "attachment":[        {           "href" : "<http://myDocumentServer/myRep/myDocument.pdf>"        }     ],     "relatedParty" : [      {           "id" : "32",           "href" : "[http://xxxxx/individual/ 32](http://xxxxx/individual/%2032)",           "role" : "customer",           "name" : "John Doe"        }   ],     "address" : {        "id" : "jkfdjgkldjf",      "href" : "<http://xxx/address/jkfdjgkldjf>"   },     "relatedObject":[        {           "involvement" : "problemToSolve",           "reference" : "<http://xxxx/troubleTicket/789745465>"        }     ]   "note":[  {  "date":" 2015-09-01T14:40:43.071Z ",  "author":"Arthur Ewans"  "text":" Alreay called the expert ",  }    ]  } |

Behavior :

* Return status codes
	+ 200 OK - the request was successful
	+ 404 Not found - the supplied ID does not match a known appointment

##

## GET api/appointment?{field selector}&{filtering expression}

This operation is used to retrieve appointment information using filter criteria. Especially criteria like period or party

Filtering selection is enabled on all attributes: first level and inner classes

Behavior :

* Return status codes
	+ 200 OK - the request was successful (includes situation in which no orders matched supplied criteria)
	+ 400 Bad Request – error

|  |
| --- |
| **REQUEST** |
| GET api/appointment?relatedParty.id=32&relatedParty.role=customer&startDate.gt=2015-08-31&startDate.lt=2015-09-04Accept: application/json |
| **RESPONSE** |
| 200Content-Type: application/json[{     "id" : "21",     "href" : "<http://xxxx/appointment/21>",     "externalId" : "anExternalIDIfNeeded432113",     "category" : "intervention",   "description" : "A useful text to describe the appointment…",     "status":"Validated"     "creationDate": "2015-08-28T08:00:43.071Z",   "lastUpdate": "2015-08-28T08:00:43.071Z ",   "startDate" : "2015-09-01T14:00:43.071Z",     "endDate" : "2015-09-01T16:00:43.071Z",     "alarm" : false,     "attachment":[        {           "href" : "<http://myDocumentServer/myRep/myDocument.pdf>"        }     ],     "relatedParty" : [      {           "id" : "32",           "href" : "[http://xxxxx/individual/ 32](http://xxxxx/individual/%2032)",           "role" : "customer",           "name" : "John Doe"        }   ],     "address" : {        "id" : "jkfdjgkldjf",      "href" : "<http://xxx/address/jkfdjgkldjf>"   },     "relatedObject":[        {           "involvement" : "problemToSolve",           "reference" : "<http://xxxx/troubleTicket/789745465>"        }     ]   "note":[  {  "date":" 2015-09-01T14:40:43.071Z ",  "author":"Arthur Ewans"  "text":" Alreay called the expert ",  }    ]  }] |

## POST api/freeSlot/search

This operation is used to retrieve relevant free slots, available for rescheduling an existing appointment on.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute name** | **Mandatory** | **Default** | **Rule** |
| appointmentId | Y |  |  |

Behavior :

* Return status codes
	+ 201 Created - the request was successful
	+ 400 Bad Request – error, for example for :
		- startDate must not be in the past
		- endDate must not be in the past
		- endDate must be superior to startDate

|  |
| --- |
| **REQUEST** |
| POST /api/freeSlot/searchContent-type: application/json{         "appointmentId": "36",}   |
| **RESPONSE** |
| 201Content-Type: application/json{   "freeSlot":[      {       "relatedParty" : {           "id" : "32",           "href" : "[http://xxxxx/organisation/ 32](http://xxxxx/organisation/%2032)",         }         "startDate":"2015-09-01T14:00:00.071Z",         "endDate":"2015-09-01T16:00:00.071Z"      },      {       "relatedParty" : {           "id" : "32",           "href" : "[http://xxxxx/organisation/ 32](http://xxxxx/organisation/%2032)",         }         "startDate":"2015-09-01T16:00:00.071Z",         "endDate":"2015-09-01T18:00:00.071Z"      },      {       "relatedParty" : {           "id" : "32",           "href" : "[http://xxxxx/individual/ 32](http://xxxxx/individual/%2032)",         }         "startDate":"2015-09-01T14:00:00.071Z",         "endDate":"2015-09-01T16:00:00.071Z"      },      {       "relatedParty" : {           "id" : "32",           "href" : "[http://xxxxx/individual/ 32](http://xxxxx/individual/%2032)",         }         "startDate":"2015-09-03T14:00:00.071Z",         "endDate":"2015-09-03T16:00:00.071Z"      },      {       "relatedParty" : {           "id" : "32",           "href" : "[http://xxxxx/individual/ 32](http://xxxxx/individual/%2032)",         }         "startDate":"2015-09-05T14:00:00.071Z",         "endDate":"2015-09-05T16:00:00.071Z"      }   ]} |

# API NOTIFICATIOn

It is assumed that the Pub/Sub uses the Register and UnRegister mechanisms described in the REST Guidelines reproduced below.

##  REGISTER LISTENER POST /hub

Description :

Sets the communication endpoint address the service instance must use to deliver information about its health state, execution state, failures and metrics. Subsequent POST calls will be rejected by the service if it does not support multiple listeners. In this case DELETE /api/hub/{id} must be called before an endpoint can be created again.

Behavior :

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 409 if request is not successful.

|  |
| --- |
| **REQUEST** |
| POST /api/hubAccept: application/json{"callback": "http://in.listener.com"} |
| **RESPONSE** |
| 201Content-Type: application/jsonLocation: /api/hub/42{"id":"42","callback":"http://in.listener.com","query":null} |

## UNREGISTER LISTENER DELETE hub/{id}

Description :

Clears the communication endpoint address that was set by creating the Hub.

Behavior :

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 404 if the resource is not found.

|  |
| --- |
| **REQUEST** |
| DELETE /api/hub/{id}Accept: application/json |
| **RESPONSE** |
| 204 |

## publish {EventTYPE} POST /listener

Description :

Provide the Event description

Behavior :

Returns HTTP/1.1 status code 201 if the service is able to set the configuration.

|  |
| --- |
| **REQUEST** |
| POST /client/listenerAccept: application/json{ "eventType": "EventType", "eventTime": "2014-09-27T05:46:25.0Z", "eventId": "1562231", "event":  { EVENT BODY  }} |
| **RESPONSE** |
| 201Content-Type: application/json |

Example see TMF REST Design Guidelines.