|  |
| --- |
| Integration Specification  REST Add Team Member  DARTMOUTH |

|  |
| --- |
| Customer: DARTMOUTH  Version: 1.0  Date: 04-28-2021  Author: Ahmed Ahizoune |

# Document Data

Table Document History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Date | Version | Status | Description | Deliverables version |
| Ahmed Ahizoune | 04/28/2021 | 1.0 |  | draft |  |
|  |  |  |  |  |  |

Table Distribution List

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Version | Department |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table Sources

|  |  |  |
| --- | --- | --- |
| Reference | Name | Date |
| Keywords |  |  |
| MS Teams or Trello |  |  |
| Others |  |  |

Table 4 Deliverables

|  |  |  |
| --- | --- | --- |
| Reference | Name | Date |
| .Jar File |  |  |
| Others |  |  |

# Content

[Integration Specification 1](#_Toc70503944)

[REST Add Team Member 1](#_Toc70503945)

[DARTMOUTH 1](#_Toc70503946)

[Document Data 2](#_Toc70503947)

[Content 3](#_Toc70503948)

[1 Introduction 4](#_Toc70503949)

[1.1 Process 4](#_Toc70503950)

[1.2 Execution 4](#_Toc70503951)

[1.3 Monitoring 4](#_Toc70503952)

[1.4 Environment 4](#_Toc70503953)

[2 Mapping 5](#_Toc70503954)

[2.1 Mapping Role Player Person (Team Member) 5](#_Toc70503955)

[3 Webservices 6](#_Toc70503956)

[3.1 Direction: Dartmouth à Planon 6](#_Toc70503957)

[3.1.1 Role Player Person (Team Member) 6](#_Toc70503958)

[3.1.1.1 Create Role Player Person (Team Member) 6](#_Toc70503959)

[4 Installation 7](#_Toc70503960)

[4.1 Configuration 7](#_Toc70503961)

[4.2 Access SDK URL 8](#_Toc70503962)

[5 Cloud specific Installation 8](#_Toc70503963)

1. Introduction

This document outlines the business, technical requirements and scope for one REST API import that will add team member to team.

* 1. Process

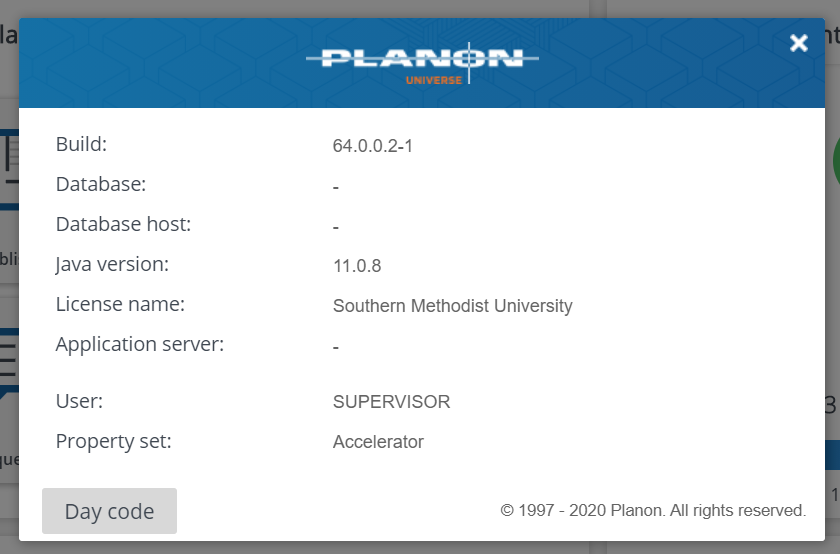
The intention of this integration is to add team member to Team in Planon.

In order to build the interface the Planon will create JAX-RS webservices that will be consumed by DARTMOUTH.

* 1. Execution
* Planon will expose JAX-RS webservice.
* Dartmouth will call the end-points to create team to team member.
* Planon will post a response back to Dartmouth which contains “Success” or “Failure” and the Team member created code.
* Planon will provide an API access-key for authentication
* Planon will provide an example of the JAX-RS payload to DARTMOUTH
  1. Monitoring

Standard log details can be found in the Log section of the Cloud Environment Gadget.

* 1. Environment



1. Mapping

BO overview:

|  |  |
| --- | --- |
| BO System name | BO description |
| RolePlayerPerson | Role players - personnel |
| UsrMaintenanceTeam | Maintenance team |
|  |  |
|  |  |

* 1. Mapping Role Player Person (Team Member)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Field type | System name | Mandatory | REST field |
| Code | Text | Code | No |  |
| Description | Text | Name | No |  |
| Person | Person | PersonRef | Yes |  |
| Start date | DateNeutral | BeginDate | Yes |  |
| End date | DateNeutral | EndDate | No |  |
| Role | Role | Role | Yes |  |
| Comment | Comment | CommentString | No |  |
| team |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. Webservices
   1. Direction: Dartmouth à Planon
      1. Role Player Person (Team Member)
         1. Create Role Player Person (Team Member)

**Triggering**

* the Planon webservice is triggered by Dartmouth.
* The Planon webservice returns the result of the action. (success or error)

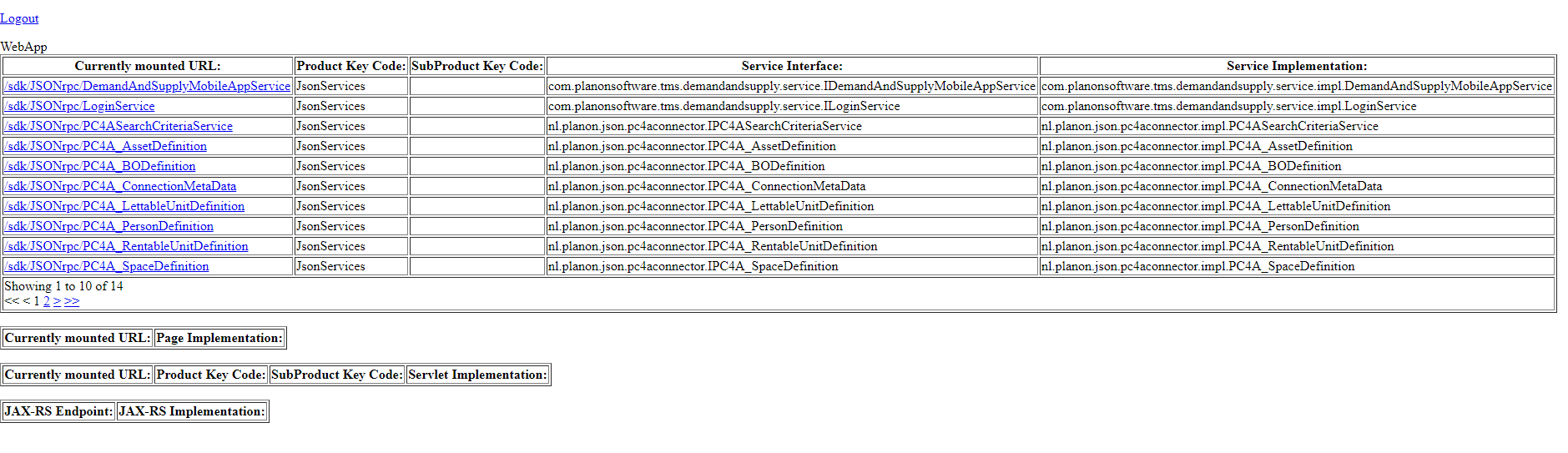
**Webservice Definition**

Description: a REST webservice is used by Dartmouth to create team Member.

|  |  |
| --- | --- |
| Webservice url | https://dartmouthsandbox-acc.planoncloud.com/sdk/jaxrs/planonwebservices/{teamCode}/addteammemeber |
| HTTP method | POST |
| Authentication method | Header: Authorization = PLANONKEY accesskey={PlanonAccessKey} |
| JSON data to be POSTED to webservice | A JSON array of “Team Member” objects (json)   * A “Team Member” object (json) can have the following properties:  |  |  |  |  | | --- | --- | --- | --- | | JSON property name | Planon RolePlayerPerson | Mandatory / max length | Type | | description | Name | 100 | String value | | startDate | BeginDate |  | Date format should be the format emitted by Date's toJSON method:  e.g. 2021-12-24 | | endDate | EndDate |  | Date format should be the format emitted by Date's toJSON method:  e.g. 2021-12-24 | | personCode | PersonRef/Code | 10 | String value | | roleCode | Role/Code | 10 | String value | | comment | CommentString | 2000 | String value |   Example:  {     "description":"Description",     "personCode":"John\_S",     "roleCode":"Role",     "startDate":"2021-12-24",     "endDate":"",     "comment":"comment"  } |
| JSON response from webservice | A json “result” object (json) will be returned   * *A “result” object (json) always*contains the following properties:  |  |  | | --- | --- | | JSON property of result | Comment | | isSuccess | true, if no error occurred,otherwise false | | errorMessage | Error message, only filled in when isSuccess=false |   Example:  {      "errorMessage": "",      "data": {          "code": "0056",          "description": "Description",          "teamCode": "0001",          "roleCode": "Role",          "personCode": "John\_S",          "startDate": "2021-12-24",          "endDate": "",          "comment": "comment"      },      "success": **true**  } |

1. Installation
   1. Configuration
2. Deploy the jars ‘**WebServicesJAXRS-1.0.0.jar**’ in folder <Planon Installation Directory Path> \Planonxxx\Server\tomcatxxx\bundles\sdk
3. User should have access to **SDK, JsonWebservices** and should able to see [https://DartMouth-dev.planoncloud.com/sdk/](https://smu-dev.planoncloud.com/sdk/)<nameofservice> in URLS.

Access : [https://DartMouth-dev.planoncloud.com/sdk/](https://smu-dev.planoncloud.com/sdk/)



* 1. Access SDK URL

URL for the sdk page would be

[https://DartMouth-dev.planoncloud.com/sdk/](https://smu-dev.planoncloud.com/sdk/)<nameofservice>

1. Cloud specific Installation
2. Login to https://DartMouth-dev.planoncloud.com/TMS By using any FileTransfer appication(WinSCP).
3. Then you will see two or more files like

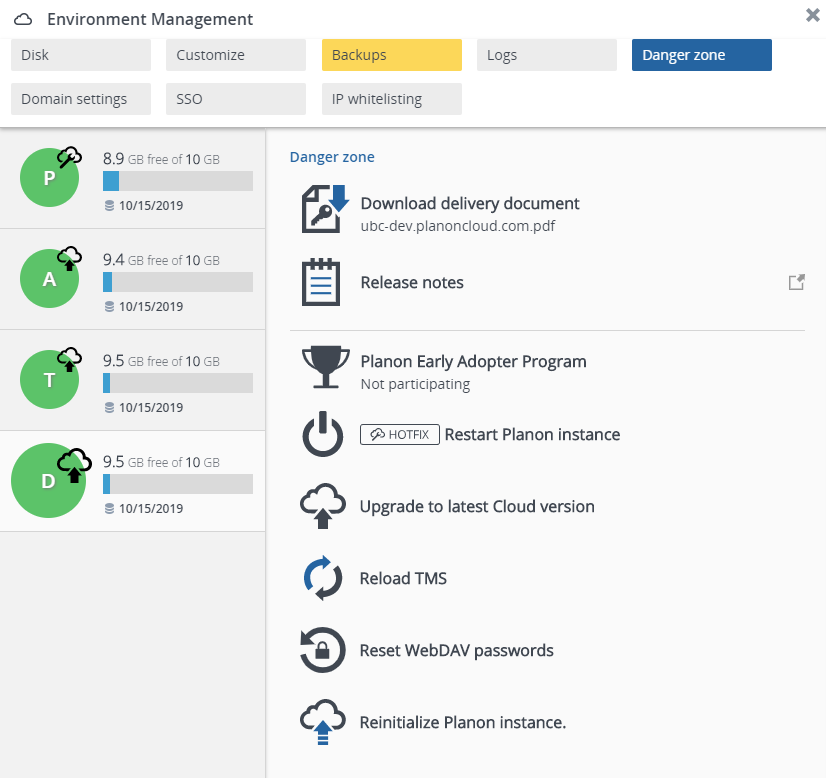
1)upload

2)status

1. Deploy the jars ‘**WebServicesJAXRS-1.0.0.jar**’ in folder upload/sdk

To load the bundle into Planon you have to log on to the Planon Cloud with an account with sufficient rights to ‘Environment Management’. Username and password are specified in the delivery document of the Cloud (chapter 9 Planon accounts).

* Go to tab ‘Danger zone’ in ‘Environment Management’



* Select ‘Reload TMS’ and choose ‘Reload TMS’ in the confirmation. The files in **/tms/upload/jboss** will be copied to the JBOSS/WILDFLY directory on the application server.

After this the change will automatically be processed by the JBOSS/WILDFLY application.

It can take a couple of minutes before the bundle is picked up by JBOSS.