

Administrator's Guide

CUSTOMER

SAP Predictive Maintenance and Service, cloud edition
Document Version: 1.6.1 – 2017-02-07

**SAP Predictive Maintenance and Service, cloud
edition**

June 2016



Typographic Conventions

Type Style	Description
<i>Example</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Textual cross-references to other documents.
Example	Emphasized words or expressions.
EXAMPLE	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
Example	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
Example	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE	Keys on the keyboard, for example, F2 or ENTER.

1 Document History

Version	Date	Change
1.6.1	2017-02-07	<ul style="list-style-type: none"> • Error in documentation, chapter deleted: "Set Machine to Active/Passive Using Service /sap/rsm/xs/services/dbAPI/Object.xsodata/ObjectActivityState" • Changes to chapter "oData Service sap.rsm.xs.services.Object.xsodata"
1.6	2016-06-28	<ul style="list-style-type: none"> • New section "Getting Users" • New chapter "Deleting a Machine Using Service /sap/rsm/xs/services/dbAPI/Object.xsjs" • New chapter "Set Machine to Active/Passive Using Service /sap/rsm/xs/services/dbAPI/Object.xsjs" • New chapter "Web Service sap.rsm.va.xs.services.masterdata.ThresholdOnboarding.xsjs" • Changes to chapter "Services for Vibration Analysis (Onboarding Objects and Configuration)"
1.5	2016-01-29	<ul style="list-style-type: none"> • Update of chapter "Business Process Integration" with information about HANA Cloud Integration • Rename of chapter "Machine Onboarding" to "Machine Onboarding Service" • Update of chapter 6.5 including addition of Web Service sap.rsm.va.xs.services.masterdata.MachineDelete.xsjs for vibration analysis. • Addition to chapter 6.2 of new service for deleting measuring data /sap/rsm/xs/services/measurement/deleteMeasurements.xsjs • New section 8.2 on oData services for vibration data
1.4	2015-10-16	<ul style="list-style-type: none"> • New chapter "Business Process Integration" • New section "Services for Vibration Analysis (Onboarding Objects and Configuration)" • The chapter "Machine Onboarding" has been updated and reordered
1.3	2015-07-28	<ul style="list-style-type: none"> • Update of chapter "Mapping Sensor Data to SAP Predictive Maintenance and Service, cloud edition - Reading Meta Data Using Service /sap/rsm/xs/services/classConfigurationData.xsodata" <ul style="list-style-type: none"> ◦ New chapter "General Information about Handling oData Services" ◦ New chapter "How to Get the Data for Feature Mapping" • Update of chapter "Optional Functions" with e-mail notifications • New chapter "oData Service for Chunked Measurements" • New chapter "Terminology"

Version	Date	Change
1.2	2015-03-31	<ul style="list-style-type: none">• Update of section "Useful Links"• Update of chapter "Machine Onboarding"• New chapter "Mapping Sensor Data to SAP Predictive Maintenance and Service, cloud edition - Reading Meta Data Using Service /sap/rsm/xs/services/classConfigurationData.xsodata"
1.1	2014-12-15	Update of SAP Notes in chapter " Support "
1.0	2014-11-03	Initial version

Table of Contents

1	Document History	3
2	Introduction	7
3	Support	9
4	General Information	10
4.1	Available Languages	10
4.2	Software Requirements	10
4.3	Getting Users.....	10
5	Getting Started.....	12
6	Machine Onboarding Services	13
6.1	Prerequisites for Machine Onboarding	13
6.1.1	Getting a User for Machine Onboarding.....	13
6.1.2	Obtaining an X-CSRF Token.....	13
6.1.3	General Information about Handling oData Services	14
6.2	Creating, Updating, and Deleting Machines in SAP Predictive Maintenance and Service, cloud edition	14
6.2.1	Creating a Machine Using Service /sap/rsm/xs/services/dbAPI/Object.xsjs.....	15
6.2.2	Updating a Machine Using Service /sap/rsm/xs/services/dbAPI/Object.xsjs.....	17
6.2.3	Refreshing Machine Data Using Web Service /sap/rsm/xs/services/dbAPI/refreshObjectProperties.xsjs	18
6.2.4	Deleting a Machine Using Service /sap/rsm/xs/services/dbAPI/Object.xsjs	19
6.2.5	Delete Measuring Data using Service /sap/rsm/xs/services/measurement/deleteMeasurements.xsjs	20
6.3	Example: Calling ODATA/REST Services from an ERP ABAP System.....	21
6.3.1	Getting the Certificate.....	21
6.4	Mapping Sensor Data in SAP HANA Internet of Things Edition to SAP Predictive Maintenance and Service, cloud edition	25
6.4.1	How to Get the Data for Feature Mapping	25
6.5	Services for Vibration Analysis (Onboarding Objects and Configuration)	26
6.5.1	Web Service sap.rsm.va.xs.services.masterdata.SensorOnboarding.xsjs	27
6.5.2	Web Service sap.rsm.va.xs.services.masterdata.ComponentOnboarding.xsjs	28
6.5.3	Web Service sap.rsm.va.xs.services.masterdata.PropertyOnboarding.xsjs	29
6.5.4	Web Service sap.rsm.va.xs.services.masterdata.MachineDelete.xsjs	30
6.5.5	Web Service sap.rsm.va.xs.services.masterdata.MachineOnboarding.xsjs	31
6.5.6	Web Service sap.rsm.va.xs.services.masterdata.ThresholdOnboarding.xsjs	32
7	Business Process Integration	36
7.1	Service for Reading and Updating Alert Data /sap/rsm/xs/services/alertAPI/alertIntegration.xsodata.....	36

8	Reporting Machine Data	37
8.1	oData Services for Business Data of a Machine	37
8.1.1	oData Service sap.rsm.xs.services.Object.xsodata	37
8.1.2	Examples For Calling The Service	39
8.1.3	oData Service for Measurement Data of a Machine	39
8.1.4	oData Service for Alerts of a Machine.....	40
8.1.5	oData Service for Chunked Measurements.....	40
8.2	oData Services for Vibration Data.....	40
8.2.1	oData Service MachineInformation.xsodata	40
8.2.2	Examples for Calling the Services	41
8.2.3	oData Service SpectrumData.xsodata	41
8.2.4	oData Service DiagnosisResults.xsodata	42
8.2.5	oData Service ComponentSpeedForSpectrum.xsodata	42
9	Internet of Things (IoT)	43
10	Security	44
10.1	Access Control.....	44
11	Optional Functions	45
11.1	E-Mail Notifications	45
11.2	Telephone Functions - Microsoft Lync	45
12	Terminology	46

2 Introduction

Welcome to *SAP Predictive Maintenance and Service, cloud edition*, your business solution for analyzing and monitoring your machines!

SAP Predictive Maintenance and Service, cloud edition is an Internet of Things (IoT) application in the SAP Cloud seamlessly integrated with *SAP HANA Internet of Things (IoT) Edition*.

SAP has prepared and released the system, and you have received system access information as well as your initial user and password. You can now start setting up your system.

About this Application

The aim of the application is to improve the visibility of machines at customer sites and to prevent machine failure and its consequences. Machine data is collected using an Internet of Things (IoT) platform and enriched with business information coming from, for example, SAP Business Suite. This allows real time monitoring of the machine so that customers can be contacted at the first sign of any unusual trends or behavior machine and can take appropriate action, potentially avoiding unnecessary service costs.

Typical users of *SAP Predictive Maintenance and Service, cloud edition* are remote service engineers and vibration experts. Additionally, the application has several roles for configuration and administration.

For more information about the application and its configuration, see SAP Help Portal at <http://help.sap.com/pdm-od>.

About this Document

This document provides information about the following topics:

- Service and support
- Software requirements
- Machine onboarding (creating machines)
- Reporting of machine data
- Internet of Things (IoT) integration

You can find the following information on SAP Help Portal at <http://help.sap.com/pdm-od>:

- How to invite users to the application
- Configuration activities required to create machines
- User guide (how to use the *SAP Predictive Maintenance and Service, cloud edition* application)

Target Audience

This guide is intended for users who handle the administration and configuration of the *SAP Predictive Maintenance and Service, cloud edition* application.

Useful Links

The following table lists all useful sources of information mentioned in this guide and their location:

Information	Location
User Guide <i>SAP Predictive Maintenance and Service, cloud edition</i> (including configuration)	SAP Help Portal at http://help.sap.com/pdm-od
User Guide <i>Internet of Things (IoT) Edition in the SAP Cloud</i>	SAP Support Portal http://support.sap.com/swdc → <i>Installation and Upgrades</i> → <i>A-Z Alphabetical List of my Products</i> → <i>IOT ENABLEMENT FOR HANA</i> → <i>IOT ENABLEMENT FOR HANA 1.0</i> → <i>Installation</i>
SAP HANA Cloud Documentation	https://help.hana.ondemand.com
SAP HANA Cloud Platform	http://hcp.sap.com

3 Support

If you need to report issues for the *SAP Predictive Maintenance and Service, cloud edition* application, create service tickets under the following components at <http://support.sap.com>:

- Condition monitoring issues **LOD-PDM-CM**
- Vibration analysis issues **LOD-PDM-VA**

SAP Notes

Make sure that you always have the up-to-date version of each SAP Note, which you can find on the SAP Marketplace at <http://service.sap.com/notes>.

The following SAP Notes are relevant for your implementation:

SAP Note	Short Text
2049732	With this note you can check that you always have the most up-to-date version of the User Guide for <i>Internet of Things (IoT) Edition in the SAP Cloud</i>

4 General Information

4.1 Available Languages

The *SAP Predictive Maintenance and Service, cloud edition* application is available in the following languages:

- Chinese
- English
- French
- German
- Portuguese
- Russian
- Spanish

4.2 Software Requirements

You need one of the following Internet browsers to use *SAP Predictive Maintenance and Service, cloud edition*:

- Internet Explorer 10.0 and higher.
- Firefox 24 and higher
- Chrome 30.0 and higher
- Safari 6.1 and higher

Note

We strongly recommend always using the latest available versions, support packages, and patches of operating systems and browsers.

4.3 Getting Users

To get the following users you need to create a service ticket for component LOD-PDM:

- ALERTAPIUSER - For reading and updating alert data. See [Service for Reading and Updating Alert Data /sap/rsm/xs/services/alertAPI/alertIntegration.xsodata](#)
- DELETEDMEASUREMENTSUSER - For deleting measuring data. See [Delete Measuring Data using Service /sap/rsm/xs/services/measurement/deleteMeasurements.xsjs](#)
- MACHINECARDUPLOADUSER - For user onboarding. See [Machine Onboarding Services](#)

You do this via the service system at support.sap.com with the following text:

Example:

Short text:

`<Customer> <IoT/LOD-PDM>: User for <Machine Onboarding>`

Description:

`Version:`

`<first name, last name> or <existing user>`

`<e-mail address>`

Within the ticket, user credentials are provided. Call the URL [Error! Hyperlink reference not valid.](#) to set the final password.

Confirm the ticket after you have set the final password.

5 Getting Started

This section gives an overview of the steps required to get users, configure and connect machines to the *SAP Predictive Maintenance and Service, cloud edition* application.

1. The master user* receives an e-mail with an invitation to join the *SAP Predictive Maintenance and Service, cloud edition* application as an administrator. He can then invite users to join the application and restrict the areas of the application they can access by assigning specific roles. For more information, see <http://help.sap.com/pdm-od> → *User Management and Roles*.
2. The master user needs to **invite a user** with the role **Configuration Manager** to make the configuration settings in the *SAP Predictive Maintenance and Service, cloud edition* application.
Configuration settings for control units and machine card tabs must be made in the *SAP Predictive Maintenance and Service, cloud edition* application. For more information, see SAP Help Portal at <http://help.sap.com/pdm-od> → *Configuration*.
3. If vibration analysis needs to be configured, the vibration analysis role Configuration Manager needs to be assigned to the user. Configuration settings like diagnosis codes, vibration source, components, and machine template can be configured with this role. For more information, see <http://help.sap.com/pdm-od> → *Vibration Analysis* → *Configuration Settings*.
4. The master user needs to **invite the user** that will make all the settings in the IoT system (SAP HANA Platform, Internet of Things (IoT) Edition) via the IT Workbench and assign the roles **Workbench User** and **Workbench Administrator**. These roles enable you to see the *IoT Workbench* tile in the application.
For more information about how to use the IoT Workbench for configuring the IoT system, refer to section 3 of the IoT Workbench User Guide. Download the latest version of the User Guide from SAP Support Portal <http://support.sap.com/swdc> → *Installation and Upgrades* → *A-Z Alphabetical List of my Products* → *IOT ENABLEMENT FOR HANA* → *IOT ENABLEMENT FOR HANA 1.0* → *Installation*.
5. **Connect the two data models.** Device Adapters (IoT Component of type "Device Adapter") are configured to read the required machine data. The measuring point IDs/machine IDs on the IoT side have to match the feature IDs/object IDs used in the *SAP Predictive Maintenance and Service, cloud edition* application.
6. Once the configuration settings have been made **machine onboarding** can begin. For more information, see "[Machine Onboarding](#)".
7. If the vibration analysis configurations have been made and the machine onboarding is done, vibration analysis specific objects and configurations can be onboarded.

* master user - the first user on the customer side.

6 Machine Onboarding Services

This section provides the following information:

- How to set up machines in the system by uploading machine data using services, and the prerequisites.
- Reference to the service to support mapping of sensor data (for example, in *SAP HANA Internet of Things Edition*) to *SAP Predictive Maintenance and Service, cloud edition*
- Example of how to call the oData services from an ERP ABAP system

6.1 Prerequisites for Machine Onboarding

- You must obtain a valid X-CSRF-token (Cross Site Request Forgery) for security purposes to prevent unauthorized persons/processes from creating or updating machines. For more information about X-CSRF-tokens, see "[Obtaining an X-CSRF Token](#)".
- In the *SAP Predictive Maintenance and Service, cloud edition* application you have configured the control units, machine card tabs, and if required, machine templates, that should be included in the machine. For more information, see <http://help.sap.com/pdm-od> → *Configuration*.
- For onboarding the first test machines you can use a REST client browser plug-in (for example, Postman) to call the service manually.
- For the productive onboarding process you must integrate the service into your existing IT environment, for example, an ERP ABAP system.
- Find out the host port of the *SAP Predictive Maintenance and Service, cloud edition* application.

6.1.1 Getting a User for Machine Onboarding

For more information, see [Getting Users](#).

6.1.2 Obtaining an X-CSRF Token

You must obtain a valid X-CSRF-token (Cross Site Request Forgery) for security purposes to prevent unauthorized persons/processes from creating or updating machines.

1. To obtain the X-CSRF Token the service has to be called with the HTTP GET method before it is called with the POST or PUT method.
2. The body of the HTTP request can be empty.
3. When calling the service with the GET method the following http header fields have to be provided:
 - *Content-Type: (Value) application/json; charset=utf-8*

- o *X-CSRF-Token: Fetch*

Result: The service returns a response containing a header field called *X-CSRF-Token*. Copy the value of this field for calling services using POST/PUT methods.

6.1.3 General Information about Handling oData Services

All data is read by oData services. oData services are called via a URL which consists of the server address and path to the service. This document only describes the path of the services. You need to add your unique server address of the HANA XS engine.

- The server address looks like this: `http(s)://<host of XS engine>`
- The service path looks like this:
`/sap/rsm/xs/services/classConfigurationData.xsodata/ControlUnit`
- To call the service you have to concatenate the server address and service path like this:
`http://<host of XS engine>/sap/rsm/xs/services/classConfigurationData.xsodata/ControlUnit`

Some URLs contain values in angle brackets like: <Control unit name>

These values (including the brackets) have to be replaced by the actual value you want to query.

- To call an oData URL a browser can be used. Some browsers recognize the XML of the oData result and provide a readable formatting. There are also tools available which can be used to read and format an oData call directly.
- For some browsers a JSON formatter addon can be installed to format a JSON output. The services can be switched to JSON output by adding the following to the URL: `&$format=json`

6.2 Creating, Updating, and Deleting Machines in SAP Predictive Maintenance and Service, cloud edition

We provide the following services for creating machines in the *SAP Predictive Maintenance and Service, cloud edition* system. You can call these services from your business system, for example, ERP:

- `/sap/rsm/xs/services/dbAPI/Object.xsjs` - to **create and update and delete** machine data
- `/sap/rsm/xs/services/dbAPI/refreshObjectProperties.xsjs` - to **refresh** machine data
- `/sap/rsm/xs/services/dbAPI/Object.xsjs` to **delete** a machine

Note the meaning of the following:

- Create - a new machine is created.
- Update - the values of an existing machine are updated and classes (with their features) may be added. For example, you can add additional control units or machine card tabs (classes). You can edit existing data such as machine name or customer.
- Refresh - features are added to the machine or classes are added to an assigned machine template.

i Background

Machine data (which we refer to in the following as 'features') is fed to the service in the following 'classes':

- Class: Machine card tabs
Subsections of the machine card that contains the machine's master data. The features are fields in the machine card tab.
- Class: Control units
Specify the sensor measurements and machine status information to be sent from the machine to the application. The control unit has subsections to group the different feature types, for example, alerts, measuring points, statuses, attributes.

The services create, update or refresh the machine's configuration according to the features in these classes.

You define the configuration of the control units and machine card tabs in the *SAP Predictive Maintenance and Service, cloud edition* application. For more information, see <http://help.sap.com/pdm-od> → *Configuration*.

Prerequisites

- You must obtain a valid X-CSRF-token (Cross Site Request Forgery) for security purposes to prevent unauthorized persons/processes from creating or updating machines. For more information about X-CSRF-tokens, see "[Obtaining an X-CSRF Token](#)".
- In the *SAP Predictive Maintenance and Service, cloud edition application* you have configured the control units, machine card tabs, and if required, machine templates, that should be included in the machine. For more information, see <http://help.sap.com/pdm-od> → *Configuration*.

6.2.1 Creating a Machine Using Service `/sap/rsm/xs/services/dbAPI/Object.xsjs`

The service creates a machine using specified classes (control units and machine card tabs) that you have already configured.

There are three ways to tell the service which classes it should use to create the machine:

- Specifying the required classes one by one for each machine.
- Copying an existing machine using a template ID.
- Using a machine template which specifies all classes to be assigned to the machine. This simplifies the onboarding process because if you have multiple machines of a specific type (for example, a decanter), you can simply specify the template rather than several classes one by one.

A machine template has no link to an existing machine. For more information about creating machine templates, see SAP Help Portal at <http://help.sap.com/pdm-od> → *Configuration* → *Configure Machine Template*.

To create a new machine the service must be called with the HTTP POST method.

Prerequisites

The following header fields must be set in the HTTP request:

- *Content-Type: application/json*
- *X-CSRF-Token: <a valid x-csrf-token>* for example, *43C3600D913FDA469D9DE9CDB2FD3AC4*

How to Create Machines

1. Call the service with the GET method to obtain an X-CSRF-Token.
2. Send a HTTP POST request to create the machine with its feature assignments.
 - *Content-Type: application/json: charset=utf-8*
 - *X-CSRF-Token: <a valid x-csrf-token>*, for example, *43C3600D913FDA469D9DE9CDB2FD3AC4*
3. In the body of the HTTP request you must provide the machine data in JSON format. (make sure 'raw' is selected)

In the following are examples for the body of the HTTP POST request.

Example

The machine will be created using the specified Siemens decanter control unit and a machine card tab containing fields for the decanter master data. A machine template is also specified to provide additional general decanter data.

```
{
  "ID": "MachineCreatedByService",
  "Name": "MachineName",
  "Text" : [{"Language": "D", "Text": "A description in german"},
            {"Language": "E", "Text": "A description in english"}],
  "Label1Value": "Machine Type",
  "Label2Value": "Machine Subtype",
  "Label3Value": "Fabrication Number",
  "ParentRef.ID": "ID of the parent machine, if there is a hierarchy of machines",
  "Classes": [{"Name": "<name of first control unit you want to add>"}, {"Name": "<name of second control unit you want to add>"}],
  "Templates": [{"Name": "<name of machine template you want to add>"}]
}
```

Note

In general only **one** machine template is required.

Note

To use the name of the class in the service the name must be unique. If the name of the class is not unique, you can assign the class by using its ID (for example "**Classes**": [{"**Class**": "14056"}, {"**Class**": "15832"}, ...]). Use the meta data service to find out the ID. For more information, see "[Mapping Sensor Data to SAP Predictive Maintenance and Service, cloud edition - Reading Meta Data Using Service /sap/rsm/xs/services/classConfigurationData.xsodata](#)".

Example

The machine will be created by **copying** the required classes from an existing machine using its template ID.


```

{
  "ID": "MachineCreatedByService",
  "Name": "Name of MachineCreatedByService",
  "Text" : [{"Language": "F", "Text": "A description in french"},
            {"Language": "I", "Text": "A description in italian"},
            ],
  "Label1Value": "Machine Type",
  "Label2Value": "Machine Subtype",
  "Label3Value": "Fabrication Number",
  "ParentRef.ID": "ID of the parent machine, if there is a hierarchy of machines",
  "TemplateID": "IDOfMachineThatShouldBeCopied"
}

```

4. Machine is created and is visible in the machine list. You can also see the assigned measuring points in the machine condition monitor.

6.2.2 Updating a Machine Using Service `/sap/rsm/xs/services/dbAPI/Object.xsjs`

The service allows you to:

- Update existing values of the following machine data:
 - *Name*
 - *Text*
 - *Label1Value (Machine Type)*
 - *Label2Value (Machine Subtype)*
 - *Label3Value (Fabrication Number)*
 - *ParentRef.ID*
- Add classes to the machine
- Update existing values of the machine features. This includes the standard delivered features and the customer-specific features that have been defined in the configuration. The features must be specified in the service. You can get the names of all available machine (object) features using the service call `/sap/rsm/xs/services/Object.xsodata/Object('<ID of the Object>')?$format=json&$expand=StandardFields,CustomFields`.
- Specify a **geolocation** when updating machine data. The reference coordinate system used is World Geodetic System (WGS) 84; SRID 4326. It is read with the service object.xsodata. See the coding example below.

To update data for an existing machine the service must be called with the HTTP PUT method

How to Update Machines

1. Call the service with the GET method to obtain an X-CSRF-Token
2. Send a HTTP PUT request to update the machine's data.

- o *Content-Type: application/json; charset=utf-8*
- o *X-CSRF-Token: <a valid x-csrf-token>* , for example, *43C3600D913FDA469D9DE9CDB2FD3AC4*

Example

Much of the existing machine data, for example, machine name, description and field names in the machine card will be updated.

```
{
  "ID": "MachineToBeUpdated",
  "Name": "Changed Name of Machine",
  "Text" : [{"Language": "E", "Text": "A changed description in english"}],
  "Label1Value": "Changed Machine Type",
  "Label2Value": "Changed Machine Subtype",
  "Label3Value": "Changed Fabrication Number",
  "ParentRef.ID": "Changed parent machine ID",
  "StandardFields": {
    "StandardMachineCardAttributeName1" : "Value for attribute1",
    "StandardMachineCardAttributeName2" : "Value for attribute2",
    "Geolocation":{
      "Longitude":97.1500,
      "Latitude":-44.100
    }
  },
  "CustomFields": {
    "CustomMachineCardAttributeName1" : "Value for custom attribute1",
    "CustomMachineCardAttributeName2" : "Value for custom attribute2",
  },
  "Classes": [
    {"Name": "NewControlUnitClass"},
    {"Name": "NewCustomerFields"}
  ]
}
```

6.2.3 Refreshing Machine Data Using Web Service `/sap/rsm/xs/services/dbAPI/refreshObjectProperties.xsjs`

Changes made in the UI configuration are not automatically refreshed in the machine's (object's) configuration. You have to manually trigger this service to see reflected changes. For example:

- When you add new features to a control unit or machine card tab.
- When you add a new class to an existing template

You can manually trigger the refresh in the following ways:

- On the UI directly if you are assigned the role *Configuration Manager*. You can refresh the object properties by choosing the *Update Machine Data Configuration* tile and starting the service from there.

Note that *Refresh* is called *Update* on the user interface.

- If you do not want to trigger assignment of the new features to an existing machine on the UI, proceed as follows:
 1. Get the machine ID. This is not visible on the UI. If you do not know the machine ID, do the following:
 1. Find the machine fabrication number from the machine list on the UI.
 2. Enter the service `/sap/rsm/xs/services/Object.xsodata/Object?$filter=(Label3Value eq '<FABRICATION NUMBER>')&$select=ID&$format=json` with the fabrication number added.
 3. Before you call the service additionally FETCH the X-CSRF-Token.
 4. Call the service with the method GET and note the machine ID.
 2. Enter the service `/sap/rsm/xs/services/dbAPI/refreshObjectProperties.xsjs`.
 3. Add the machine ID with the format `["<MACHINE ID>"]` and the new assignments are refreshed in the machine.

Note

If a control unit or machine card tab is assigned to several machines, the service must be called for every machine in which the control unit or machine card tab is assigned. For example `["<MACHINE ID>,<MACHINE ID2>"]`.

If you have added fields to several machine card tabs for one machine, then you only need to call the service once and all tabs will be refreshed.

6.2.4 Deleting a Machine Using Service `/sap/rsm/xs/services/dbAPI/Object.xsjs`

The service allows you to delete a (single) machine if its status is 'not yet connected'. All data related to this machine is irrevocably deleted from the system.

The service is called using the 'DELETE' method.

Example

```
{"ID": "MachineToBeDeleted"}
```

Prerequisites

The following header fields must be set in the HTTP request:

- *Content-Type*: `application/json`
- *X-CSRF-Token*: `<a valid x-csrf-token>` for example, `43C3600D913FDA469D9DE9CDB2FD3AC4`

How to delete machine data

1. Call the service with the GET method to obtain an X-CSRF-Token.
2. Send a HTTP POST request to update valid X-CSRF-Token
 - o *X-CSRF-Token: <a valid x-csrf-token>* for example, *43C3600D913FDA469D9DE9CDB2FD3AC4*
3. Call the service with the DELETE method providing the following information in the body.
 - o Example: {"ID":"MachineToBeDeleted"}

If the status is not 'not yet connected', or if the machine does not exist then an error message occurs.

6.2.5 Delete Measuring Data using Service `/sap/rsm/xs/services/measurement/deleteMeasurements.xsjs`

The service deletes measuring data of a single machine for a specific time frame. It deletes the following measuring data:

- Measurements of all measuring points of the machine (from HANA and Sybase IQ)
- Property Valuation (Alerts, Status)
- Alerts
- Thresholds will not be deleted

To delete measuring data the service must be called with the HTTP POST method.

Prerequisites

You need a special user. See [Getting Users](#).

The following header fields must be set in the HTTP request:

- *Content-Type: application/json*
- *X-CSRF-Token: <a valid x-csrf-token>* for example, *43C3600D913FDA469D9DE9CDB2FD3AC4*

How to delete measuring data for a single machine for a specific time frame

4. Call the service with the GET method to obtain an X-CSRF-Token.
5. Send a HTTP POST request to delete the measuring data for a single machine and a specific time frame.
 - o *Content-Type: application/json*
 - o *X-CSRF-Token: <a valid x-csrf-token>* for example, *43C3600D913FDA469D9DE9CDB2FD3AC4*
6. In the body of the HTTP request you must provide the parameters in JSON format to determine which measuring data shall be deleted.

In the following are examples for the body of the HTTP POST request.

Example

The measuring data for the machine with ID "MachineID" will be deleted for the time frame between 8.00 on 1st November 2015 and 15.00 on 2nd November 2015.

```
{
  "MachineID": " MachineID ",
  "FromTime": "2015-11-01T08:00:00.000Z",
  "ToTime": "2015-11-02T15:00:00.000Z"
}
```

Note

The timestamps for *FromTime* and *ToTime* have to be strings in ISO 8601 format.

7. The service response can have the following statuses:
 - *HTTP Status 200*: Everything is fine. The measuring data have been deleted. The service response contains the number of deleted rows in its response body.
 - *HTTP Status 400*: Bad Request. The sent request to the service was wrong. The body contains a failure description, which can be one of the following:
 - No user provided -> the session provided no user
 - No object ID provided -> the request was sent without or with an empty MachineID
 - Object does not exist -> the provided machine does not exist in the system
 - Invalid timestamp -> either the *From Date* or the *To Date* was in a wrong timestamp format. Please provide a timestamp in ISO 8601 format
 - *HTTP Status 500*: Internal Server Error. Something went wrong during the deletion of measuring data. The response body contains information where to find a detailed failure description. Usually the response will provide you an ID for the trace log where you will find the error.

6.3 Example: Calling ODATA/REST Services from an ERP ABAP System

The following section provides an example of how to connect a REST Service to an ERP ABAP System using ODATA Service </sap/rsm/xs/services/dbAPI/RSMMachine.xsodata>. This service can be used to create, for example, the customer-specific machine card data.

In general this procedure can also be used to call the services described in "[Reporting Machine Data](#)".

6.3.1 Getting the Certificate

1. Ensure sure you have access to the SAP HANA system and transaction **STRUST** in the ERP system.

2. Ensure that HTTPS is active in the ERP System in transaction **SMICM** → *Go To* → *Services*
3. Load the HTTP certificate from the Internet Browser. The (service) certificate must be downloaded in Internet Explorer so that it can be entered in ABAP transaction **STRUST**.
4. Enter the HTTPS URL of the service (for example,


```
https://<host>:<port>/sap/rsm/xs/services/dbapi/rsmmachine.xsodata/machinecard
```
5. In Internet Explorer, go to the *Details* tabpage in the *Certificate* dialog under *File* → *Properties* → *Certificate* or via the 'Lock' icon next to the URL field and copy the certificate to a local file.
4. Upload the certificate in transaction **STRUST** in the ERP System
5. In transaction **STRUST** this certificate must be imported under *SSL-Client SSL-Client (Standard)*
6. Select entry in the tree and choose *Import Certificate* (icon in bottom left of right screen area) and upload the saved .cer file
7. Choose *Add to Certificate List*.

6.3.1.1 Calling the Service from ABAP

In transaction **SM59** create an RFC connection of type *HTTP Connection to External Server* to call the service. It must be an HTTPS connection.

Example

The following coding example shows how the service can be called to create a machine. The service must firstly be called using the HEAD method and the header parameter *X-CSRF-Token* with value *FETCH* in order to receive a CSRF token. The token is needed for the POST call of the service, which creates the machine. The coding below shows a simple case in which the machine has been created with its ID, type and fabrication number.

```
*&-----*
*& Report  ZODATA_CALL
*&-----*
*& {
*& "ID":"MachineCreatedByService",
*& "Name":"Name of MachineCreatedByService",
*& "Classes": [{"Class":"DecanterSiemens"}, {"Class":"DecanterSeparatorMaster"}]
*& }
*&
*& {
*& "ID":"MachineCreatedByService",
*& "Type":"ServiceMachine",
*& "Subtype":"SubtypeServiceMachine",
*& "CountryCode": "DE",
*& "EndCustomerName": "Service Machine End Customer",
*& "EndCustomerAddress":"Walldorf, Dietmar-Hopp-Allee 16",
*& "FabricationNo":"FabricationNoServiceMachine"
*& }
*&
*&-----*
```

```

REPORT zodata_call.

DATA: lo_rsmmachine TYPE REF TO if_http_client,
      http_fields   TYPE      tihttpnvp,
      http_field    LIKE LINE OF http_fields,
      s_csrf_token  TYPE      string,
      s_body        TYPE      string,
      lo_error      TYPE REF TO cx_root.

TRY.
"-----
" instantiate an http client for the SM59 RFC destination
" that points to the machine card service
"-----
cl_http_client=>create_by_destination(
  EXPORTING
    destination = 'ZMP_MACHINE_CARD'
  IMPORTING
    client = lo_rsmmachine
).

"-----
" send HEAD Request in order to obtain an X-CSRF-Token
"-----
lo_rsmmachine->request->set_method( if_rest_request=>gc_method_head ).

lo_rsmmachine->request->set_header_field(
  name = if_rest_request=>gc_header_csrf_token
  value = 'Fetch'
).

lo_rsmmachine->send( ).
lo_rsmmachine->receive( ).

lo_rsmmachine->response->get_header_fields( CHANGING fields = http_fields ).

LOOP AT http_fields INTO http_field.
  TRANSLATE http_field-name TO UPPER CASE.
  IF http_field-name = 'X-CSRF-TOKEN'.
    s_csrf_token = http_field-value.
  ENDIF.
ENDLOOP.

lo_rsmmachine->refresh_request( ).
lo_rsmmachine->refresh_response( ).

"-----
" send POST Request to create the RSMMachineCard.Data entry
"-----

```

```

lo_rsmmachine->request->set_method( if_rest_request=>gc_method_post ).

lo_rsmmachine->request->set_header_field(
  name = 'Content-Type'
  value = 'application/json;charset=utf-8'
).

lo_rsmmachine->request->set_header_field(
  name = if_rest_request=>gc_header_csrf_token
  value = s_csrf_token
).

s_body = '{"ID":"MichaelsABAPMachine","Type":"Separator","FabricationNo":"45
25-889"}'.
lo_rsmmachine->request->set_cdata( data = s_body ).

lo_rsmmachine->send( ).
lo_rsmmachine->receive( ).

lo_rsmmachine->response->get_header_fields( CHANGING fields = http_fields ).

s_body = lo_rsmmachine->response->get_cdata( ).

WRITE s_body.

CATCH cx_root INTO lo_error.
  WRITE lo_error->textid.

ENDTRY.

```

6.3.1.2 Calling the Service From the Chrome Addon 'Postman'

1. Enter the URL of the service
`http://<host>:<port>/sap/rsm/xs/services/dbAPI/RSMMachine.xsodata/MachineCard`
2. Under *Basic Auth* enter user and password and choose *Refresh Headers*.
3. In the header enter the X-CSRF token with the value *Fetch* and send a GET request by pressing *Send*.
4. In the request answer under *Headers*, highlight the X-CSRF token copy it into the header.
5. Change the request method to *POST* and switch to *Raw*. In the header set the content type to 'application/json;charset=utf-8'. In the dropdown next to the *Raw* Button select JSON and enter the required machine data in JSON Format.
6. Choose *Send* to create the machine.

6.4 Mapping Sensor Data in SAP HANA Internet of Things Edition to SAP Predictive Maintenance and Service, cloud edition

Reading Meta Data Using Service

`/sap/rsm/xs/services/classConfigurationData.xsodata`

Service *ClassConfigurationData* was developed to support the **mapping of connectors with machine configurations**. It allows you to read parts of the configuration including **IDs** that are not displayed on the UI. In some cases it is necessary to use an ID, for example **class IDs** for the service *sap.rsm.xs.services.dbAPI.Object.xsjs* (for more information, see "[Creating a Machine Using Service /sap/rsm/xs/services/dbAPI/Object.xsjs](#)") or **feature IDs** for configuration of the IoT Workbench.

This chapter describes how to map the data from a connector (carries identifiers from physical sensors) with the machine configuration (control unit in the SAP Predictive Maintenance and Service (PdMS) machine data model).

6.4.1 How to Get the Data for Feature Mapping

1. To get the ID of the control unit call the following service:

```
/sap/rsm/xs/services/classConfigurationData.xsodata/ControlUnit?$filter(Name eq '<Control unit name>')
```

The name of your control unit can be seen in the PdMS configuration control units.

1. To get a list of all control units call:

```
/sap/rsm/xs/services/classConfigurationData.xsodata/ControlUnit
```

The **ID** of the **control unit** can be seen in tag *ID*.

2. To get all the **features** of the control unit call the following service:

```
/sap/rsm/xs/services/classConfigurationData.xsodata/ControlUnit('<control unit id>')/ControlUnitTag?$select=FeatureName,FeatureDescription,FeatureTypeDescription,ID&$orderby(FeatureName)
```

<control unit id> is the ID of your control unit returned by the first service.

Identify the Features

All features are listed for the given control unit. Use this information to map the features to IDs in the IoT application.

- The **feature ID** can be seen in tag *ID*.
- The **feature description** is shown in tag *FeatureDescription*.
- The **feature name** is shown in tag *FeatureName*.

The result is sorted by the FeatureName to make it easier to find the related features. The **FeatureTypeDescription** describes the role of the feature. Together with the following naming convention of the feature name it is possible to identify the features.

Feature Name	Semantics
<FeatureName>	Measuring point, status, alert, attributes
<FeatureName> Upper	Measuring point warning threshold
<FeatureName> Uppermost	Measuring point error threshold
<FeatureName> Error	Measuring point error
<FeatureName> Warning	Measuring point warning

Optionally a service to get the thresholds, warning and error values can be called for a specific measurement value. This service does not return status feature IDs.

```
/sap/rsm/xs/services/controlUnit.xsodata/Feature(ID='<Feature id>',ControlUnitRef='<Control unit id>')/FeatureThresholds?$select=(Feature,FeatureDescription,Threshold,ThresholdTypeDescription,Alert,AlertTypeDescription)
```

For some control units it may be more comfortable to call this service with the specific **measuring point ID**. The <Feature id> is the ID of the measuring point.

For more information, see "[Internet of Things \(IoT\)](#)".

6.5 Services for Vibration Analysis (Onboarding Objects and Configuration)

The onboarding process can be classified as below:

1. Sensor onboarding
2. Component onboarding
3. Property onboarding
4. Machine onboarding
5. Machine deletion

Prerequisites

- You have a user for machine onboarding.
- You have made the required configuration settings in Vibration Analysis. For more information, see <http://help.sap.com/pdm-od> → *Vibration Analysis* → *Configuration Settings*.
- The role `sap.rsm.va.xs.security::VACRTUPDMachineUser` has been assigned to your user. It authorizes you to call services for onboarding sensors, components, properties, and machines.
- You must obtain a valid X-CSRF-token (Cross Site Request Forgery) for security purposes to prevent unauthorized persons/processes from creating or changing machines. For more information about X-CSRF-tokens, see "[Obtaining an XCSRF Token](#)".

6.5.1 Web Service

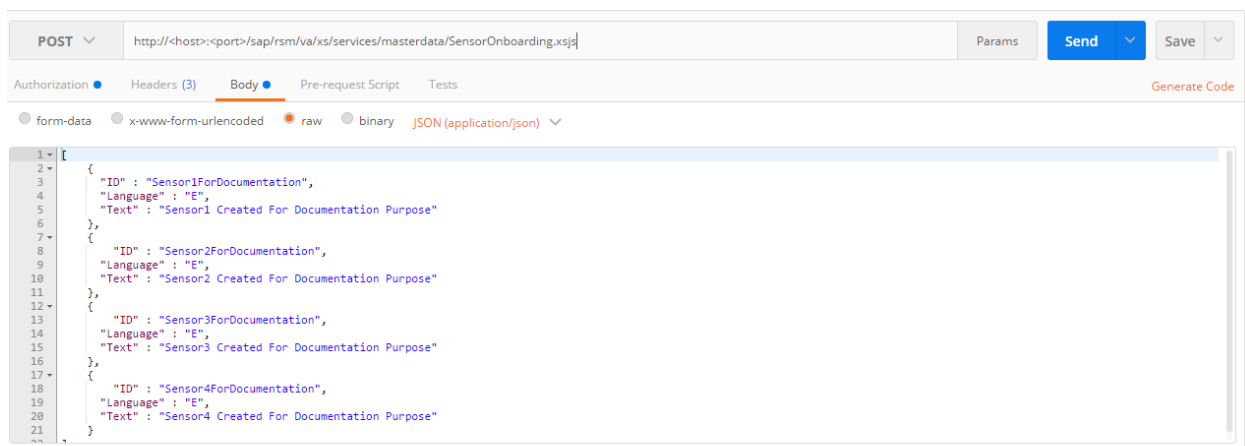
sap.rsm.va.xs.services.masterdata.SensorOnboarding.xsjs

This service is used to onboard the sensors in the application. The web service standard defines that to create a new entity, for example, a sensor, the service must be called with the HTTP POST method. Ensure that the sensor IDs used are Sensor1, Sensor2, Sensor3, and Sensor4.

Procedure

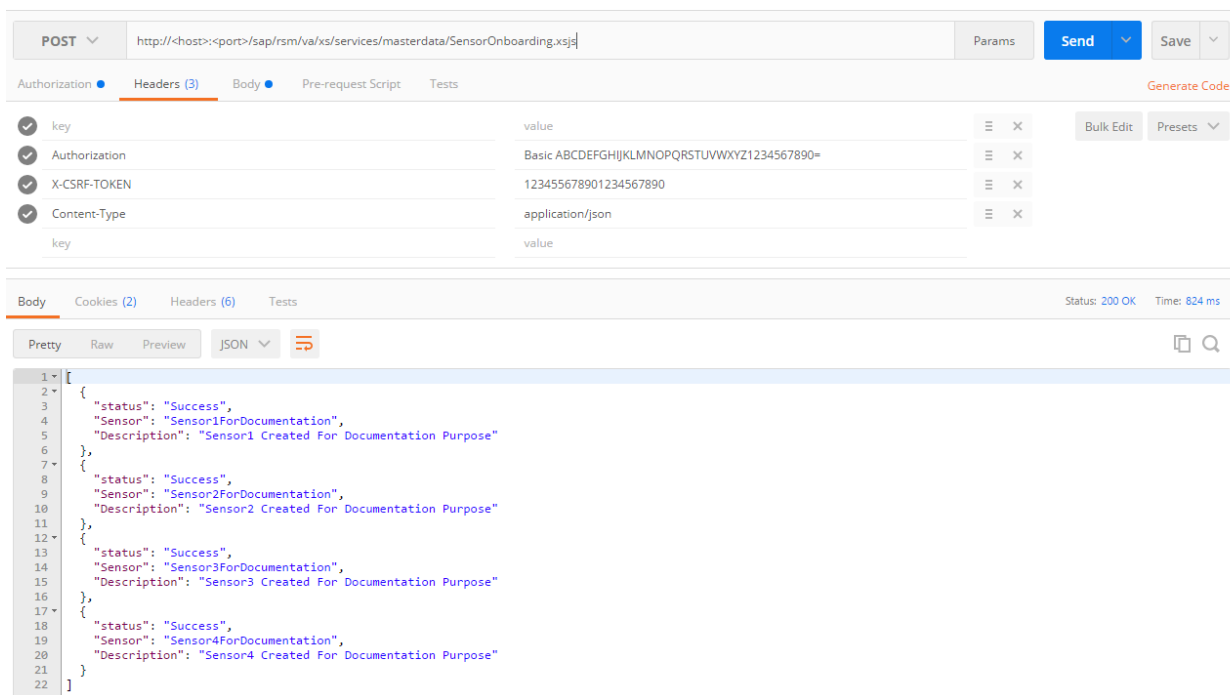
1. Call the service with the GET method to obtain an X-CSRF-Token.
2. Send a HTTP POST request to create the sensor with its orientation and description.
 - o Content-Type: application/json: charset=utf-8
 - o X-CSRF-Token: <a valid x-csrf-token>, for example, 43C3600D913FDA469D9DE9CDB2FD3AC4

The following example shows how the body of the HTTP POST request will look:



```
POST http://<host>:<port>/sap/rsm/va/xs/services/masterdata/SensorOnboarding.xsjs
Authorization
Headers (3)
Body
Pre-request Script
Tests
Generate Code
form-data x-www-form-urlencoded raw binary JSON (application/json)
1- [
2- {
3-   "ID" : "Sensor1ForDocumentation",
4-   "Language" : "E",
5-   "Text" : "Sensor1 Created For Documentation Purpose"
6- },
7- {
8-   "ID" : "Sensor2ForDocumentation",
9-   "Language" : "E",
10-  "Text" : "Sensor2 Created For Documentation Purpose"
11- },
12- {
13-   "ID" : "Sensor3ForDocumentation",
14-   "Language" : "E",
15-   "Text" : "Sensor3 Created For Documentation Purpose"
16- },
17- {
18-   "ID" : "Sensor4ForDocumentation",
19-   "Language" : "E",
20-   "Text" : "Sensor4 Created For Documentation Purpose"
21- }
22- ]
```

After successful execution, the following results display:



6.5.2 Web Service `sap.rsm.va.xs.services.masterdata.ComponentOnboarding.xsjs`

This service is used to onboard the components for which the base order is already known, such as bearing. The web service standard defines that to create a new entity, for example, a component, the service must be called with the HTTP POST method.

Procedure

1. Call the service with the GET method to obtain an X-CSRF-Token.
2. Send a HTTP POST request to create an entry for base order of the component per vibration source.
 - o **Content-Type:** `application/json; charset=utf-8`
 - o **X-CSRF-Token:** `<a valid x-csrf-token>`, for example, `43C3600D913FDA469D9DE9CDB2FD3AC4`

The following example shows how the body of the HTTP POST request can look:

```

[
  {
    "ComponentID" : "1234-000-001",
  }
]
  
```

```

        "VibrationSrcRefID" :           "VibrationSource1" ,
        "CompBaseOrder"      :         "0.1"
    },
    {
        "ComponentID"        :           "1234-000-002" ,
        "VibrationSrcRefID" :           "VibrationSource2" ,
        "CompBaseOrder"      :         "0.2"
    }
]

```

6.5.3 Web Service sap.rsm.va.xs.services.masterdata.PropertyOnboarding .xsjs

This service is used to onboard the properties' value of components, configured in the application for different component ID. The web service standard defines that to create a new entity, for example, a property, the service must be called with the HTTP POST method.

Procedure

1. Call the service with the GET method to obtain an X-CSRF-Token.

The output for the GET method is as below.

```

[{"ComponentID":null,"Category":"Motor","Property":"Speed","Value":null,"UoM":"X06"},
 {"ComponentID":null,"Category":"Pulley","Property":"Diameter","Value":null,"UoM":"MM"},
 {"ComponentID":null,"Category":"Belt","Property":"Length","Value":null,"UoM":"MM"},
 {"ComponentID":null,"Category":"DEMO_fan","Property":"DEMO_Speed","Value":null,"UoM":"X06"}]

```

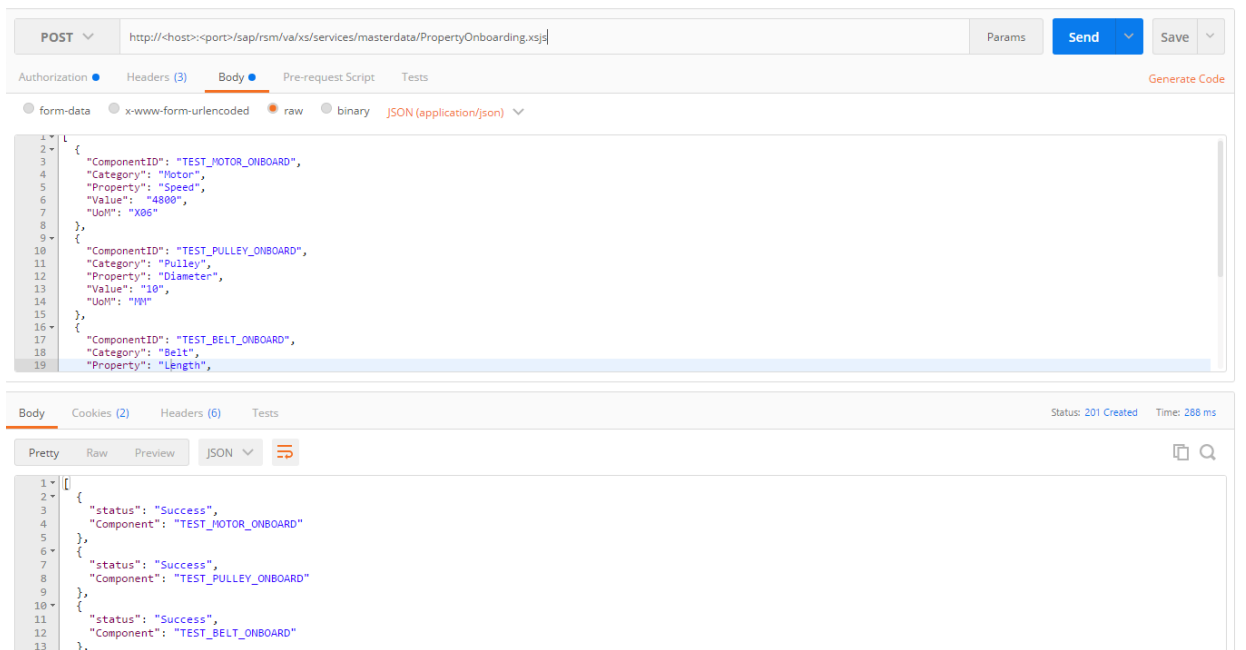
This output serves as the template to fill in while executing a POST method. For example, the category belts 1, 1, 2 and 3 are configured with property length, 1, 2 and 3 respectively. The component ID and value will be replaced and posted as mentioned in step 2.

2. Send a HTTP POST request to create the sensor with its orientation and description.

Content-Type: application/json: charset=utf-8

X-CSRF-Token: <a valid x-csrf-token>, for example, 43C3600D913FDA469D9DE9CDB2FD3AC4

The following example shows how the body of the HTTP POST request can look:



6.5.4 Web Service `sap.rsm.va.xs.services.masterdata.MachineDelete.xsjs`

This service is used to delete a machine in the *Vibration Analysis* application only. The web service standard defines that to delete an entity, for example, a machine, the service must be called with the **HTTP DELETE** method.

Procedure

1. Call the service with the GET method to obtain an X-CSRF-Token.
2. Send a HTTP DELETE request to delete a machine.
 - o *X-CSRF-Token*: `<a valid x-csrf-token>`, for example, `43C3600D913FDA469D9DE9CDB2FD3AC4`
3. The URL parameter key is machine and value is the machine ID you want to delete, for example, Error! Hyperlink reference not valid..

If the service call is successful, the following result is displayed:

```

1  {
2  "HTTP_STATUS_CODE": 200,
3  "ERROR_MESSAGE": "Machine deleted"
4  }

```

i Note

Only one machine can be deleted at a time. When deleting a machine, the associated spectrums, machine diagnosis and measuring point assigned are also deleted. Use this service with caution as this is not reversible.

6.5.5 Web Service **sap.rsm.va.xs.services.masterdata.MachineOnboarding .xsjs**

This service is used to onboard a machine in the *Vibration Analysis* application. The web service standard defines that to create a new entity, for example, a machine, the service must be called with the HTTP POST method.

Prerequisite

It is mandatory to onboard the machine in condition monitoring before onboarding the machine in Vibration Analysis.

Procedure

1. Call the service with the GET method to obtain an X-CSRF-Token. Ensure that you pass category parameter along with the request to get the defined template for the requested machine variant, for example, you could use

```
https://<host>/sap/rsm/va/xs/services/masterdata/MachineOnboarding.xsjs?Category=<category>
```

An example of a GET call is as below.

```
[{"MachineCategoryRef":"XD","InternalSequenceId":"1","ComponentCategoryRefID":"Bearing","ParentComponentID":"1","DescriptionRef":"Bearing","MachineID":null,"PartNumber":null}]
```

2. Send a HTTP POST request to onboard a machine in the *Vibration Analysis* application. Multiple machines can be onboarded at a time for a given machine variant. Each machine is sent as an array of JSON object. For multiple machines, an array of machines is sent as shown below:

- o *Content-Type*: **application/json; charset=utf-8**
- o *X-CSRF-Token*: *<a valid x-csrf-token>*, for example, **43C3600D913FDA469D9DE9CDB2FD3AC4**

An example, for POST call is as below.

[

[{

```
"MachineCategoryRef":"XD",  
  "InternalSequenceId":"1",  
  "ComponentCategoryRefID":"Bearing",  
  "ParentComponentID":"1",
```

```
        "DescriptionRef": "Bearing",
        "MachineID": "XYZ",
        "PartNumber": "0011-1100-001"
    }
}
]
```

6.5.6 Web Service

`sap.rsm.va.xs.services.masterdata.ThresholdOnboarding.xsjs`

This service must be called per machine to onboard threshold values.

Prerequisite

The machine must be onboarded in condition monitoring before the thresholds are onboarded.

Procedure

1. Call the service with the GET method to obtain an X-CSRF-Token. Ensure that you pass the MachineID parameter with the request to get the MachineID, component, and its harmonics for which the thresholds needs to be onboarded. For example, use

```
http://<host>/sap/rsm/va/xs/services/masterdata/ThresholdOnboarding.xsjs
```

An example of a GET call is:

```
[
  {
    "MachineID": "machine1",
    "InternalSequence": "1",
    "VibrationSourceID": "1X",
    "DiagnosisID": "UNBALANCE",
    "Dimension": "SPEED",
    "AnalysisMethod": "Line",
    "MachineCategory": "MachineCategory",
    "ComponentID": "Motor",
    "ComponentCategory": "Motor",
    "ThresholdLower": "4.5",
    "ThresholdUpper": "7",
    "ThresholdUOM": "MMS",
    "AmplitudeType": "RMS",
```



```

        "MinimumHarmonic": "1",
        "MaximumHarmonic": "2",
        "BaseOrderMultiplier": "1"
    },
    {
        "MachineID": "machine1",
        "InternalSequence": "2",
        "VibrationSourceID": "GMF",
        "DiagnosisID": "GMF",
        "Dimension": "ACCEL",
        "AnalysisMethod": "Line",
        "MachineCategory": "MachineCategory",
        "ComponentID": "Gear1",
        "ComponentCategory": "Gear",
        "ThresholdLower": null,
        "ThresholdUpper": null,
        "ThresholdUOM": null,
        "AmplitudeType": null,
        "MinimumHarmonic": "1",
        "MaximumHarmonic": "1",
        "BaseOrderMultiplier": "1"
    }
]

```

2. Change the following fields

- *ThresholdLower*
- *ThresholdUpper*
- *ThresholdUOM*
- *AmplitudeType*

Permitted Amplitude Type

RMS, PEAK, P2P, AVG

Permitted Threshold UOM

For dimension ACCEL

- MS2: m/s²
- X10: g
- X11: mg
- X12: mm/s²

For dimension SPEED

- 2M: cm/s

- 2X: m/min
 - CMH: cm/h
 - KMH: km/h
 - M/S: m/s
 - MMA: mm/a (millimeter per annum)
 - MMH: mm/h
 - MMS: mm/s
 - MTS: m/h
3. Send a HTTP POST request to onboard the thresholds for a given machine. In the POST call, change the threshold values and post it
- *Content-Type*: **application/json; charset=utf-8**
 - *X-CSRF-Token*: *<a valid x-csrf-token>*, for example, *43C3600D913FDA469D9DE9CDB2FD3AC4*

An example, for POST call is as below.

```
{
  "MachineID": "machine1",
  "InternalSequence": "1",
  "VibrationSourceID": "1X",
  "DiagnosisID": "UNBALANCE",
  "Dimension": "SPEED",
  "AnalysisMethod": "Line",
  "MachineCategory": "MachineCategory",
  "ComponentID": "Motor",
  "ComponentCategory": "Motor",
  "ThresholdLower": "5",
  "ThresholdUpper": "7",
  "ThresholdUOM": "MMS",
  "AmplitudeType": "RMS",
  "MinimumHarmonic": "1",
  "MaximumHarmonic": "2",
  "BaseOrderMultiplier": "1"
},
{
  "MachineID": "machine1",
  "InternalSequence": "2",
  "VibrationSourceID": "GMF",
  "DiagnosisID": "GMF",
  "Dimension": "ACCEL",
  "AnalysisMethod": "Line",
  "MachineCategory": "MachineCategory",
  "ComponentID": "Gear1",
  "ComponentCategory": "Gear",
```

```
"ThresholdLower": "0.5",  
"ThresholdUpper": "0.7",  
"ThresholdUOM": "X10",  
"AmplitudeType": "RMS",  
"MinimumHarmonic": "1",  
"MaximumHarmonic": "10",  
"BaseOrderMultiplier": "1"  
}  
]
```

7 Business Process Integration

It is possible to integrate *SAP Predictive Maintenance and Service, cloud edition* with other systems, for example SAP CRM, SAP ERP, or SAP C4C or third party systems.

Currently one integration scenario is available for SAP CRM. This can be used as a template to create and process SAP CRM service tickets for PDMS alerts.

Recommendation

We recommend using *SAP HANA Cloud Integration* to seamlessly integrate *SAP Predictive Maintenance and Service, cloud edition* with your business systems.

You can access the integration scenario here:

<https://cloudintegration.hana.ondemand.com/#/shell/discover/hcipackage/74e3f7c4664349b599915927c2e2493b> and find the documentation.

For general information about SAP HANA Cloud Integration see

<https://proddps.hana.ondemand.com/dps/d/preview/88a767b35adb4dc887ee1d545d301140/1/en-US/f830932fddf6453ebe1fd0c666592017.html>.

7.1 Service for Reading and Updating Alert Data `/sap/rsm/xs/services/ alertAPI/alertIntegration.xsodata`

You can use this service when integrating a service ticket system with *SAP Predictive Maintenance and Service, cloud edition* to optimize your alert handling.

The service `/sap/rsm/xs/services/alertAPI/alertIntegration.xsodata` enables

- Reading alert data based on different selection criteria
- Updating the ticket-related information of an alert

Prerequisites

- You need a special user. See [Getting Users](#).

The service allows you to update the existing values of the following alert data:

- TicketID
- TicketURL
- TicketStatus

8 Reporting Machine Data

You can download the measurement data saved in *SAP Predictive Maintenance and Service, cloud edition* to your business system.

8.1 oData Services for Business Data of a Machine

i Note

As oData is an open standard, all general information about the oData standard also holds true for the MachineReporting service. There is plenty of information available in the internet on this topic. Following the examples given below and the general oData documentation you can come up with your own reporting queries.

For more information, see "[General Information about Handling oData Services](#)".

8.1.1 oData Service `sap.rsm.xs.services.Object.xsodata`

This service can be used to retrieve all the different kinds of data related to a machine stored in *SAP Predictive Maintenance and Service, cloud edition*, for example, data about the machine itself, its alerts, its measuring points, and its measurement data.

For this purpose the service has four entity sets:

Object

This entity set contains data like the machine's ID, type, sub type and parent object.

- StandardFields - All machine features that are delivered as standard with the application. Every machine in the system will have these features when it is created. In detail these fields are:
 - *Connection Data*
 - *MAC Address*
 - *LastDataReceipt*
 - *Location Data*
 - *Street*
 - *House number*
 - *Postal Code*
 - *City*
 - *District*
 - *Country Code*
 - *Longitude*

- *Latitude*
- *Machine Data*
 - *Start up date*
- *Customer Data*
 - *Name*
 - *Name of Contact at Customer*
 - *Telephonenumber. Of Contact at Customer*
- *Service Data*
 - *Service Contract*
 - *Service Level*
 - *Next Service*
 - *Last Service*
 - *Service Org*
- *Service Organization*
 - *Service Organization Name*
 - *Street*
 - *House Number*
 - *Postal Code*
 - *City*
 - *District*
 - *Country Code*
 - *Name of Contact at Service Organization*
 - *Telephonenumber. Of Contact at Service Organization*
- *Source System*
 - *External Machine ID*
 - *Business Object*
 - *System*
 - *Client*
- **Custom Fields** - All machine card features that the customer has defined in the application's configuration.

MeasuringPointList

A list of the machines measuring points including the most recent measurement value for each point, the current thresholds and unit of measurement.

AlertList

All alerts that have been generated by the machine. As well already processed alerts as those which are still open or in work.

Measurement

All the machines measurement data

- **ValuationLatest** - latest values for alerts, statuses or thresholds of a machine

8.1.2 Examples For Calling The Service

Service	Description
<code>/sap/rsm/xs/services/Object.xsodata/\$metadata</code>	Returns information about the available features of all entity sets and about the associations between the entities. For example, it gives you the information that a machine has features like <i>ID</i> , <i>Name</i> etc. You can use the information provided by this call to build 'real' calls of the service to obtain the data you require.
<code>/sap/rsm/xs/services/Object.xsodata/Object('4711')</code>	Returns all data of the entity set <i>Machine</i> available for the machine with the ID 4711.
<code>/sap/rsm/xs/services/Object.xsodata/Object?\$select=ID,Name</code>	Returns the feature values <i>ID</i> and <i>Name</i> for all machines.
<code>/sap/rsm/xs/services/Object.xsodata/Object('4711')/MeasuringPointList</code>	Gives you back a list of all the measuring points of machine '4711'
<code>/sap/rsm/xs/services/Object.xsodata/Object('4711')?\$expand=MeasuringPointList</code>	Selects all data of machine 4711 including the complete list of measuring points. The difference to the example before is, that you get the machine data as well as the measuring point data in one call.
<code>/sap/rsm/xs/services/Object.xsodata/Object('4711')/Measurement?\$filter=(Timestamp ge datetime'2014-01-03T00:00:00.000Z' and Timestamp le datetime'2014-01-03T01:10:00.000Z')</code>	Returns the measurement data of machine 4711 that was taken between midnight and 1:10AM (in UTC, Coordinated Universal Time) on the third of January 2014.

8.1.3 oData Service for Measurement Data of a Machine

For your internal and external reporting *SAP Predictive Maintenance and Service, cloud edition* offers oData Services.

oData Services for measurement data of a single machine allow you to access measurement data in the cloud. You can then use the data for analysis, for example in MS Excel.

- The oData service provides **input parameters** for the selected machines, for example:
 - *Machine*
 - *Customer*
 - *Location*
 - *Fabrication number*

- *From date to date*
- The oData service provides **output parameters**: You get a list of measurement data for the selected machines.

8.1.4 oData Service for Alerts of a Machine

The oData service [alert.xsodata](#) is used to read the status of machine alerts.

8.1.5 oData Service for Chunked Measurements

The oData services [reportingMeasuringPoint.xsodata](#) and [ChunkedMeasurements.xsodata](#) can be used to read out measurements of a measuring point with a resolution of one thousand measurements.

Both services are located in the sub package [reporting](#) which do not authenticate via SAP Identity Provider. It makes it therefore easier for Business Intelligence tools to directly connect to the required data.

8.2 oData Services for Vibration Data

All services for Vibration Analysis data are located in the sub package `sap.rsm.va.xs.services.reporting`. These services do not authenticate via SAP Identity Provider. This will make it easier for Business Intelligence tools to directly connect to the required data.

Note

As oData is an open standard, all general information about the oData standard apply to the Vibration Analys reporting service, too. There is plenty of information available in the Internet on this topic. You can create your own reporting queries by following the examples given below and by general oData documentation.

For more information, see "[General Information about Handling oData Services](#)".

8.2.1 oData Service `MachineInformation.xsodata`

The information available for one specific machine can be retrieved using the service. For this, the service consists the following entity sets:

Components

This entity set contains basic information such as the ID, category, and the parent component of a component.

ComponentVibrationSources

This part will provide more information about the components including its part numbers, base order, threshold, vibration source, diagnosis, diagnosis harmonics and thresholds, amplitude type, and threshold units.

ComponentProperties

This service will return the properties of a machine including the values and UoM.

ComponentSensors

The EntitySet ComponentSensors contains all the sensors attached to the machines components.

MachineLevelDiagnosis

Machine Level Diagnosis - basically to get resonance and cavitation information.

8.2.2 Examples for Calling the Services

Service	Description
/sap/rsm/va/xs/services/reporting/MachineInformation.xsodata/Machine('xxxx-xx')/Components	Returns components of the machine with the ID XXXX-XX
/sap/rsm/va/xs/services/reporting/MachineInformation.xsodata/Machine('xxxx-xxx')/ComponentVibrationSources	Returns the vibration sources of the machine with the ID XXXX-XX
/sap/rsm/va/xs/services/reporting/MachineInformation.xsodata/Machine('xxxx-xxx')/ComponentProperties	Returns component properties of the machine with the ID XXXX-XX
/sap/rsm/va/xs/services/reporting/MachineInformation.xsodata/Machine('xxxx-xxx')/ComponentSensors	Returns all component sensors of the machine with the ID XXXX-XX
/sap/rsm/va/xs/services/reporting/MachineInformation.xsodata/Machine('xxxx-xxx')/MachineLevelDiagnosis	Returns the resonance and cavitation information for the machine with the ID XXXX-XX

8.2.3 oData Service SpectrumData.xsodata

Returns the spectrum data. For this, the service consists the following entity set:

Header

The header entity set contains basic information such as the MachineID, SpectrumId, SensorId, SpectrumType, AnalysisMethod, the UoM, and the timestamp.

To retrieve the details of one spectrum header the entity set has a sub set called *Data*.

Data

This entity set contains the amplitude, frequency index, and reference frequency.

8.2.4 oData Service DiagnosisResults.xodata

Returns the results from the vibration diagnosis. For this, the service consists the following entity set:

Results

The Results entity set will help to identify developing faults within the machine components after you triggered an automatic vibration diagnosis for a machine at certain time stamp.

8.2.5 oData Service ComponentSpeedForSpectrum.xodata

URL

`http://<host>:<port>/sap/rsm/va/xs/services/reporting/MachineInformation.xodata/ComponentSpeed`

Returns the machine component frequencies and corresponding amplitudes over a period of time. The service consists of the following entity set:

ComponentSpeed

The **ComponentSpeed** entity set consists of calculated speed (from formula, PLC, design) and runtime speed (actual speed found in spectrum), along with corresponding amplitude. User can filter component speed based on timestamp or machine.

9 Internet of Things (IoT)

SAP Predictive Maintenance and Service, cloud edition is an Internet of Things (IoT) application in the SAP Cloud seamlessly integrated with SAP HANA Internet of Things (IoT) Edition.

For more information, see SAP Support Portal <http://support.sap.com/swdc> → *Support Packages and Patches* → *Alphabetical List of my Products* → *IOT ENABLEMENT FOR HANA* → *IOT ENABLEMENT FOR HANA 1.0* → *Installation*.

10 Security

Compliance to Security Standards

SAP deploys modern elaborate processes to achieve high security level of its software. This product complies with the SAP Product Security Standards.

Additional Information

For more information about specific topics, consider the following resources:

Document	Location
Security	http://service.sap.com/security
SAP HANA Security Guide, Administration Guide etc.	http://help.sap.com/hana_appliance
SAP HANA Cloud Security	http://help.sap.com/hana_cloud
SAP HANA Internet of Things Edition	For more information about security, see Software Download Center (SWDC) http://support.sap.com/swdc → <i>Support Packages and Patches</i> → <i>A-Z Index</i> → <i>IOT ENABLEMENT FOR HANA</i> .

10.1 Access Control

SAP Predictive Maintenance and Service, cloud edition uses *SAP Cloud Identity Service* (SAP ID Service) as identity provider.

SAML framework (Security Assertion Markup Language) is used for exchanging authentication and authorization information.

For more information about access control, see "[Obtaining an X-CSRF Token](#)".

11 Optional Functions

11.1 E-Mail Notifications

During alert rule creation (under *Configure Alert Rules* on the UI) the configuration manager on the customer side can specify that an e-mail is sent to a generic customer e-mail address when specific alert information is received from a machine.

If you wish to enable this, please inform your account executive of the send-to and send-from e-mail addresses. The account executive will pass on the information to the installation team.

Customers can then define rules on their e-mail servers to ensure the mails are forwarded on to the appropriate recipients.

11.2 Telephone Functions - Microsoft Lync

In *SAP Predictive Maintenance and Service, cloud edition* Microsoft Lync can be integrated within the list of users.

Prerequisites

- Activate the add-on for Microsoft Lync.
- Use Internet Explorer as a browser.

12 Terminology

You can find a complete list of terminology and glossary entries for *SAP Predictive Maintenance and Service, cloud edition* at <http://sapterm.com> under the component PM-PDM.



www.sap.com/contactsap

Material Number

© 2016 SAP SE or an SAP affiliate company. All rights reserved.
No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors.

National product specifications may vary.

These materials are provided by SAP SE and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE in Germany and other countries. Please see www.sap.com/corporate-en/legal/copyright/index.epx#trademark for additional trademark information and notices.