

Valuekeep

API REST

Contents

Contents.....	2
Introduction	4
API REST.....	5
Login	6
Services	7
Assets and Locations Management.....	7
Assets	7
Asset Routes	9
Asset Families	11
Locations	13
Location Routes	15
Work Centers	17
Access Levels	19
Monitorings.....	23
Monitoring Groups.....	25
Maintenance Management	27
Failures	27
Failure Causes	29
Failure Actions.....	31
Failure Symptoms	33
Issues	35
Issue Reasons	37
Asset Shutdowns	39
Location Shutdowns	41
MaintenancePlans	43
TimeUnit.....	43
Work Orders	43
Assignment of Work Order Tasks.....	50
Execution of Work Order Tasks	53
Reporting Labor in the Work Order	57
Reporting Expenses in the Work Order	62

Reporting Consumptions in the Work Order.....	66
Reporting Contracts in the Work Order.....	75
Reporting Other Costs in the Work Order.....	81
Inventory Management.....	85
Items.....	85
Parties	87
Taxes.....	89
Extended Items	90
Extended Items Entry	94
Extended Items Issue.....	97
Purchases Management	100
Supplier.....	100
Sales Management.....	103
Customer.....	103
Administration.....	106
Exchange Rates	106
Employees	107
Internal Request Management.....	111
Extended Internal Requests.....	111
Configuration Tables.....	114
Maintenance setup	114
Base Currency and Reporting Currency set in Valuekeep.....	115
Integration	115
Integration Log.....	115
Entity Attachment Service.....	117
Upload	117
Download	117
Code examples	118
Login	118
Access Levels.....	119
Asset Routes.....	124
Location Routes	128
Work Centers.....	132
Work Orders	136
Tasks Assignment.....	154
Tasks Execution.....	155
Labor Report.....	160
Entity Attachment.....	164

Introduction

The last few years have seen the rise of REST APIs - services exposed through simple HTTP, instead of a more formal service contract (such as SOAP or WS *). Exposing the services this way can make the integration of features with a wide range of devices and client platforms easier. It also enables the creation of richer HTML experiences by using JavaScript in the browser. Most large websites on the web now expose REST APIs (some examples: Facebook, Twitter, LinkedIn, Netflix, etc.), and their use will increase even further in the next few years due to the booming amount of connected devices, as well as the demand for richer user experiences.

This manual documents the integration of external applications – developed by Valuekeep Partners – into the Valuekeep (VK) solution through services exposed via HTTP.

API REST

Valuekeep's REST API exposes a set of services and resources which allow for interaction with Valuekeep.

These HTTP services make it possible to perform read operations (read, search, listing and paged listing) and persistence (save, update and delete) on the different resources found in the system. The documentation about each of the services and about the resources included in each of them can be found online.

Items : Handles Item operations.

Show/Hide | List Operations | Expand Operations | Raw

GET /api/materialscore/items

Gets all Items.

Implementation Notes

Sample: GET api/item.

Response Class

Model | Model Schema

ItemResource {
 ItemKey (string): Item natural key,
 Description (string, optional): Item description,
 ComplementaryDescription (string, optional): Item long description,
 Keyword (string, optional): Item keyword,
 Barcode (string, optional): Item barcode,
 Weight (decimal?, optional): Item weight,
 Volume (decimal?, optional): Item volume,
 InventoryPostings (bool): Determines if the item does inventory postings or not,
 ModelName (string, optional): Item model name,
 StockBalance (decimal?, optional): Current stock balance,
 MaxStock (decimal?, optional): Maximum stockable quantity,
 MinStock (decimal?, optional): Minimum stock quantity,
 InventoryBalance (decimal?): Current inventory value,
 CalculatedUnitCost (decimal?): Calculated unit cost based on a weighted average formula,
 LastUnitCost (decimal?): Last unit cos for this item,
 Remarks (string, optional): Item remarks,
 IsSales (bool): Sales Item,
 IsPurchases (bool): Purchases Item,
 ItemType (ItemTypeResourceEnumeration) = ['Item' or 'Service']: Item type: 1 - Product; 2 - Service,
 Assortment (string, optional): Item assortment,
 Brand (string, optional): Item brand,
 BrandModel (string, optional): Brand dependent model of this item,
 Rank (RankResourceEnumeration) = ['RankA' or 'RankB' or 'RankC']: Item rank: 1 - Rank A; 2 - Rank B; 3 - Rank C,
 BaseUnit (string): Item base unit,
 WeightUnit (string, optional): Item weight for each unit,
 VolumeUnit (string, optional): Item volume for each unit,
 OriginCountry (string, optional): Country of origin of this item,
 ItemTaxSchema (string): Tax schema for this item,
 ItemWithholdingTaxSchema (string, optional): Withholding tax schema fro this item
 }

Response Content Type

application/json ▼

Response Status Codes

HTTP Status Code	Reason
200	Success.

Try it out!

Figure 1 - Example of a service's documentation

Login

The authentication process is carried out by the following process:

1. The client makes an authentication request by indicating its username and password:

```
POST /api/login HTTP/1.1
Host: localhost:60326
Content-Type: application/json
Body:
{ "Username": "<username>", "Password": "<password>" }
```

2. The server returns an object with the corresponding modules to which the client has access, as well as the respective Authorization Token for each of them:

```
{
  "Username": "<username>",
  "Name": "<username>",
  "Email": null,
  "Photo": null,
  "AuthorizedModules": [
    {
      "Key": "EAM.TOTAL",
      "AuthorizationToken": "5XPUw200UEGBFr8LtzPfXQ=="
    }
  ]
}
```

3. The client starts using the AuthorizationToken returned in all of the remaining requests by adding a new header to the subsequent requests:

```
GET /api/sales/invoices HTTP/1.1
Host: localhost:60326
Content-Type: application/json
Authorization: authorizationToken=j/18INarFE03JM1fBuIyKA==
```

The AuthorizationTokens issued are valid for 15 minute periods which are renewed by every request made afterwards.

Services

Each of the services exposes a common interface allowing for interaction with the different system resources. Below you can find an analysis of the methods concerning Assets/Locations Management, Maintenance Management and Inventory Management.

Assets and Locations Management

Assets

GET	/api/{tenant}/{organization}/maintenancemanagementcore/assets	Gets all Assets.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assets?assetKey={assetKey}	Gets a Asset by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assets/{id}	Gets a Asset by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assets/search/version/{version}	Gets all Assets with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assets/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assets?page={page}&pageSize={pageSize}	Gets all the Assets using a paged return.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/assets	Creates a Asset.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/assets?assetKey={assetKey}	Updates a Asset.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/assets/{id}	Updates a Asset by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/assets?assetKey={assetKey}	Deletes a specific Asset.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/assets/{id}	Deletes a specific Asset by it's unique identifier.

Figure 2 - Example of the methods exposed by the Assets service

The assets service makes it possible to obtain and manipulate Valuekeep's assets. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assets**

This method allows you to get the full list of existing Assets.

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/assets
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assets/{assetKey}**

This method allows you to obtain the details of a specific Asset through its natural key.

Example:

Obtaining the details of the Asset with the natural key "Sample001"

```
GET /api/mytenant/myorganization/maintenancemanagementcore/assets/Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assets? page={page}&pagesize={pagesize}**

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Assets:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/assets?
page=1&pagesize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assets? field={field}&value={value}**

This method allows you to search for a specific field of the Asset.

Example:

Obtaining all Assets of the tool type.

```
GET /api/mytenant/myorganization/maintenancemanagementcore/assets?
field=IsTool&value=1
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assets/GetAttachmentsByVersionHigherThan?version={value}**

This method allows you to search for a specific version of the Asset attachments.

Returns all the Asset attachments from a specific version onwards. The version parameter is a number with 12 digits which identifies the records to get. To obtain a full list use the "AAAAAAAAAAAA=" version.

Example:

Obtaining all Asset Attachments.

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/assets/GetAttachmentsByVersionHigherThan?version=AAAAAAAAAAAA=
```

- **POST /api/mytenant/myorganization/maintenancemanagementcore/assets**

This method allows you to create a new Asset. To that end, the http message should have the "body" filled in with the structure of the resource *AssetResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/assets
Body:
{
  "AssetKey": "Sample001",
  "Description": "Sample 001",
}
```

- **PUT /api/mytenant/myorganization/maintenancemanagementcore/assets/{assetKey}**

This method allows you to update an Asset. To that end, the http message should have the "body" filled in with the full structure of the resource *AssetResource* (including the unchanged ones).

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/assets/{assetKey}**

This method allows you to delete a specific Asset through its natural key.

Example:

Deleting the Asset with the natural key *Sample001*:

```
DELETE /api/mytenant/myorganization/maintenancemanagementcore/assets/Sample001
```

Asset Routes

GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetroutes	Gets all AssetRoutes.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetroutes?assetRouteKey={assetRouteKey}	Gets a Asset Route by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetroutes/{id}	Gets a Asset Route by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetroutes/search/version/{version}	Gets all AssetRoutes with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetroutes/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetroutes?page={page}&pageSize={pageSize}	Gets all the AssetRoutes using a paged return.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/assetroutes	Creates a Asset Route.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/assetroutes?assetRouteKey={assetRouteKey}	Updates a Asset Route.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/assetroutes/{id}	Updates a Asset Route by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/assetroutes?assetRouteKey={assetRouteKey}	Deletes a specific Asset Route.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/assetroutes/{id}	Deletes a specific Asset Route by it's unique identifier.

Figure 3 - Example of the methods exposed by the Asset Routes service

The Asset Routes service makes it possible to obtain and manipulate Valuekeep's Asset Routes. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assetroutes**

This method allows you to obtain the full list of existing Asset Routes.

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/assetroutes
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assetroutes/{assetRouteKey}**

This method allows you to obtain the details of a specific Asset Route through its natural key.

Example:

Obtaining the details of the Route with the natural key "Sample001"

```
GET /api/mytenant/myorganization/maintenancemanagementcore/assetroutes/Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assetroutes? page={page}&pagesize={pagesize}**

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Asset Routes:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/assetroutes?
page=1&pagesize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assetroutes? field={field}&value={value}**

This method allows you to search for a specific field of the Asset Route.

Example:

Obtaining all Asset Routes with the note: "Review".

```
GET /api/mytenant/myorganization/maintenancemanagementcore/assetroutes?
field=Note&value=Review
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assetroutes/GetAttachmentsByVersionHigherThan?version={value}**

This method allows you to search for a specific version of the Asset Route attachments.

Returns all the Asset Route attachments from a specific version onwards. The version parameter is a number with 12 digits which identifies the records to get. To obtain a full list use the "AAAAAAAAAAAA=" version.

Example:

Obtaining all Asset Route Attachments.

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/assetroutes/GetAttachment
sByVersionHigherThan?version=AAAAAAAAAAAA=
```

- **POST /api/mytenant/myorganization/maintenancemanagementcore/assetroutes**

This method allows you to create a new Asset Route. To that end, the http message should have the "body" filled in with the structure of the resource *AssetRouteResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/assetroutes
Body:
{
  "AssetRouteKey": "Sample001",
  "Description": "Sample 001",
}
```

- **PUT**

/api/mytenant/myorganization/maintenancemanagementcore/assetroutes/{assetRouteKey}

This method allows you to update an Asset Route. To that end, the http message should have the “body” filled in with the full structure of the resource *AssetRouteResource* (including the unchanged ones).

- **DELETE**

/api/mytenant/myorganization/maintenancemanagementcore/assetroutes/{assetRouteKey}

This method allows you to delete a specific Asset Route through its natural key.

Example:

Deleting the Route with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/assetroutes/Sample001
```

Asset Families

GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetfamilies	Gets all AssetFamilies.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetfamilies?familyKey={familyKey}	Gets a Family by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetfamilies/{id}	Gets a Family by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetfamilies/search/version/{version}	Gets all AssetFamilies with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetfamilies/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetfamilies?page={page}&pageSize={pageSize}	Gets all the AssetFamilies using a paged return.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/assetfamilies	Creates a Family.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/assetfamilies?familyKey={familyKey}	Updates a Family.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/assetfamilies/{id}	Updates a Family by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/assetfamilies?familyKey={familyKey}	Deletes a specific Family.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/assetfamilies/{id}	Deletes a specific Family by it's unique identifier.

Figure 4 - Example of the methods exposed by the Assets service

The Asset Routes service makes it possible to obtain and manipulate Valuekeep's Asset Families. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assetfamilies**

This method allows you to obtain the full list of existing Asset Families.

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/assetfamilies
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assetfamilies/{familyKey}**

This method allows you to obtain the details of a specific Asset Family through its natural key.

Example:

Obtaining the details of the Family with the natural key “Sample001”

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/assetfamilies/Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assetfamilies?page={page}&pagesize={pagesize}**

This method allows you to obtain a paged list.

The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Family Routes:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/assetfamilies?
page=1&pagesize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assetfamilies?field={field}&value={value}**

This method allows you to search for a specific field of the Asset Family.

Example:

Obtaining all Families with the note: "Review".

```
GET /api/mytenant/myorganization/maintenancemanagementcore/assetfamilies?
field=Notes&value=Review
```

- **POST /api/mytenant/myorganization/maintenancemanagementcore/assetfamilies**

This method allows you to create a new Asset Family. To that end, the http message should have the "body" filled in with the structure of the resource *AssetFamilyResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/assetfamilies
Body:
{
  "FamilyKey": "Sample001",
  "Description": "Sample 001",
}
```

- **PUT /api/mytenant/myorganization/maintenancemanagementcore/assetfamilies/{familyKey}**

This method allows you to update an Asset Family. To that end, the http message should have the "body" filled in with the full structure of the resource *AssetFamilyResource* (including the unchanged ones).

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/assetfamilies/{familyKey}**

This method allows you to delete a specific Asset Family through its natural key.

Example:

Deleting the Family with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/assetfamilies/Sample001
```

Locations

GET	/api/{tenant}/{organization}/maintenancemanagementcore/locations	Gets all Locations.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/locations?locationKey={locationKey}	Gets a Location by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/locations/{id}	Gets a Location by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/locations/search/version/{version}	Gets all Locations with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/locations/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/locations?page={page}&pageSize={pageSize}	Gets all the Locations using a paged return.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/locations	Creates a Location.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/locations?locationKey={locationKey}	Updates a Location.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/locations/{id}	Updates a Location by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/locations?locationKey={locationKey}	Deletes a specific Location.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/locations/{id}	Deletes a specific Location by it's unique identifier.

Figure 5 - Example of the methods exposed by the Locations service

The Locations service makes it possible to obtain and manipulate Valuekeep's Locations. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/locations**

This method allows you to get the full list of existing Locations.

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/locations
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/locations/{locationKey}**

This method allows you to obtain the details of a specific Location through its natural key.

Example:

Obtaining the details of the Location with the natural key "Sample001"

```
GET /api/mytenant/myorganization/maintenancemanagementcore/locations/Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/locations?page={page}&pagesize={pagesize}**

This method allows you to obtain a paged list.

The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Locations:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/locations?
page=1&pagesize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/locations?field={field}&value={value}**

This method allows you to search for a specific field of the Location.

Example:

Obtaining all Locations of the building type.

```
GET /api/mytenant/myorganization/maintenancemanagementcore/locations?
field=IsBuilding&value=1
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/locations/GetAttachmentsByVersionHigherThan?version={value}**

This method allows you to search for a specific version of the Location attachments.

Returns all the Location Attachments from a specific version onwards. The version parameter is a number with 12 digits which identifies the records to get. To obtain a full list use the "AAAAAAAAAAAA=" version.

Example:

Obtaining all Location Attachments.

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/locations/GetAttachmentsB
yVersionHigherThan?version=AAAAAAAAAAAA=
```

- **POST /api/mytenant/myorganization/maintenancemanagementcore/locations**

This method allows you to create a new Location. To that end, the http message should have the "body" filled in with the structure of the resource *LocationResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/locations
Body:
{
  "LocationKey": "Sample001",
  "Description": "Sample 001",
}
```

- **PUT /api/mytenant/myorganization/maintenancemanagementcore/locations/{locationKey}**

This method allows you to update a Location. To that end, the http message should have the "body" filled in with the full structure of the resource *LocationResource* (including the unchanged ones).

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/locations/{locationKey}**

This method allows you to delete a specific Location through its natural key.

Example:

Deleting the Location with the natural key *Sample001*:

```
DELETE /api/mytenant/myorganization/maintenancemanagementcore/locations
/Sample001
```

Location Routes

GET	/api/{tenant}/{organization}/maintenancemanagementcore/locationroutes	Gets all LocationRoutes.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/locationroutes?locationRouteKey={locationRouteKey}	Gets a Location Route by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/locationroutes/{id}	Gets a Location Route by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/locationroutes/search/version/{version}	Gets all LocationRoutes with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/locationroutes/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/locationroutes?page={page}&pageSize={pageSize}	Gets all the LocationRoutes using a paged return.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/locationroutes	Creates a Location Route.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/locationroutes?locationRouteKey={locationRouteKey}	Updates a Location Route.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/locationroutes/{id}	Updates a Location Route by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/locationroutes?locationRouteKey={locationRouteKey}	Deletes a specific Location Route.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/locationroutes/{id}	Deletes a specific Location Route by it's unique identifier.

Figure 6 - Example of the methods exposed by the Location Routes service

The Location Routes service makes it possible to obtain and manipulate Valuekeep's Location Routes. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/locationroutes**

This method allows you to obtain the full list of existing Location Routes.

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/locationroutes
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/locationroutes/{locationRouteKey}**

This method allows you to obtain the details of a specific Location Route through its natural key.

Example:

Obtaining the details of the Route with the natural key "Sample001"

```
GET /api/mytenant/myorganization/maintenancemanagementcore/locationroutes/Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/locationroutes? page={page}&pagesize={pagesize}**

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Location Routes:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/locationroutes?
page=1&pagesize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/locationroutes? field={field}&value={value}**

This method allows you to search for a specific field of the Location Route.

Example:

Obtaining all Routes with the note: "Review".

```
GET /api/mytenant/myorganization/maintenancemanagementcore/locationroutes?
field=Notes&value=Review
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/locationroutes/GetAttachmentsByVersionHigherThan?version={value}**

This method allows you to search for a specific version of the Location Route attachments.

Returns all the Location Route Attachments from a specific version onwards. The version parameter is a number with 12 digits which identifies the records to get. To obtain a full list use the "AAAAAAAAAAAA=" version.

Example:

Obtaining all Location Route Attachments.

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/locationroutes/GetAttachmentsByVersionHigherThan?version=AAAAAAAAAAAA=
```

- **POST /api/mytenant/myorganization/maintenancemanagementcore/locationroutes**

This method allows you to create a new Location Route. To that end, the http message should have the "body" filled in with the structure of the resource *LocationRouteResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/locationroutes
Body:
{
  "LocationRouteKey": "Sample001",
  "Description": "Sample 001",
}
```

- **PUT /api/mytenant/myorganization/maintenancemanagementcore/locationroutes /{locationRouteKey}**

This method allows you to update a Location Route. To that end, the http message should have the “body” filled in with the full structure of the resource *LocationRouteResource* (including the unchanged ones).

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/locationroutes/{locationRouteKey}**

This method allows you to delete a specific Location Route through its natural key.

Example:

Deleting the Route with the natural key *Sample001*:

```
DELETE /api/mytenant/myorganization/maintenancemanagementcore/locationroutes/Sample001
```

Work Centers

GET	/api/{tenant}/{organization}/maintenancemanagementcore/workcenters	Gets all WorkCenters.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workcenters?workCenterKey={workCenterKey}	Gets a Work Center by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workcenters/{id}	Gets a Work Center by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workcenters/search/version/{version}	Gets all WorkCenters with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workcenters/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workcenters?page={page}&pageSize={pageSize}	Gets all the WorkCenters using a paged return.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workcenters/GetWorkCenterAssetEmployeeInfos?version={version}	Gets the work center asset employee infos.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workcenters/GetWorkCenterLocationEmployeeInfos?version={version}	Gets the work center location employee infos.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/workcenters	Creates a Work Center.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workcenters?workCenterKey={workCenterKey}	Updates a Work Center.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workcenters/{id}	Updates a Work Center by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/workcenters?workCenterKey={workCenterKey}	Deletes a specific Work Center.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/workcenters/{id}	Deletes a specific Work Center by it's unique identifier.

Figure 7 - Example of the methods exposed by the Work Centers service

This service makes it possible to obtain and manipulate Valuekeep's Work Centers. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workcenters**

This method allows you to obtain the full list of existing Work Centers.

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/workcenters
```

- **GET**
/api/mytenant/myorganization/maintenancemanagementcore/workcenters/{workCenterKey}

This method allows you to obtain the details of a specific Work Center through its natural key.

Example:

Obtaining the details of the Center with the natural key "Sample001"

```
GET /api/mytenant/myorganization/maintenancemanagementcore/workcenters/Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workcenters?**
page={page}&pagesize={pagesize}

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Work Centers:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/workcenters?  
page=1&pagesize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workcenters?**
field={field}&value={value}

This method allows you to search for a specific field of the Center.

Example:

Obtaining all Centers of the internal type.

```
GET /api/mytenant/myorganization/maintenancemanagementcore/workcenters?  
field=IsInternal&value=1
```

- **POST /api/mytenant/myorganization/maintenancemanagementcore/workcenters**

This method allows you to create a new Work Center. To that end, the http message should have the "body" filled in with the structure of the resource *WorkCenterResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/workcenters  
Body:  
{  
  "WorkCenterKey": "Sample001",  
  "Description": "Sample 001",  
}
```

- **PUT**
/api/mytenant/myorganization/maintenancemanagementcore/workcenters/{workCenterKey}

This method allows you to update a Work Center. To that end, the http message should have the "body" filled in with the full structure of the resource *WorkCenterResource* (including the unchanged ones).

- **DELETE**

/api/mytenant/myorganization/maintenancemanagementcore/workcenters/{workCenterKey}

This method allows you to delete a specific Work Center through its natural key.

Example:

Deleting the Center with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/workcenters/Sample001
```

Access Levels

GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetaccesslevels	Gets all AssetAccessLevels.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetaccesslevels?assetLevelKey={assetLevelKey}	Gets a Access Level by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetaccesslevels/{id}	Gets a Access Level by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetaccesslevels/search/version/{version}	Gets all AssetAccessLevels with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetaccesslevels/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetaccesslevels?page={page}&pageSize={pageSize}	Gets all the AssetAccessLevels using a paged return.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetaccesslevels/GetAccessLevelAssetEmployeeInfos?version={version}	Gets the work center location employee infos.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetaccesslevels/GetAccessLevelLocationEmployeeInfos?version={version}	Gets the access level location employee infos.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/assetaccesslevels	Creates a Access Level.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/assetaccesslevels?assetLevelKey={assetLevelKey}	Updates a Access Level.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/assetaccesslevels/{id}	Updates a Access Level by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/assetaccesslevels?assetLevelKey={assetLevelKey}	Deletes a specific Access Level.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/assetaccesslevels/{id}	Deletes a specific Access Level by it's unique identifier.

AssetLevelAssets : Handles Asset Level Asset operations.

Show/Hide | List Operations | Expand Operations | Raw

GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevelassets	Gets all AssetLevelAssets.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevelassets?asset={asset}&assetAccessLevel={assetAccessLevel}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevelassets/{id}	Gets a Asset Level Asset by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevelassets/search/version/{version}	Gets all AssetLevelAssets with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevelassets/search/version?version={version}	Gets all AssetLevelAssets with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevelassets/search?field={field}&value={value}	Gets all AssetLevelAssets containing the search criteria on the specified field.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevelassets?page={page}&pageSize={pageSize}	Gets all the AssetLevelAssets using a paged return.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevelassets	Creates a Asset Level Asset.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevelassets?asset={asset}&assetAccessLevel={assetAccessLevel}	No Documentation Found.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevelassets/{id}	Updates a Asset Level Asset by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevelassets?asset={asset}&assetAccessLevel={assetAccessLevel}	No Documentation Found.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevelassets/{id}	Deletes a specific Asset Level Asset by it's unique identifier.

AssetLevelLocations : Handles Asset Level Location operations.			Show/Hide	List Operations	Expand Operations	Raw
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevellocations	Gets all AssetLevelLocations.				
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevellocations?location={location}&assetAccessLevel={assetAccessLevel}	No Documentation Found.				
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevellocations/{id}	Gets a Asset Level Location by the resource unique identifier.				
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevellocations/search/version/{version}	Gets all AssetLevelLocations with a version higher than the specified.				
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevellocations/search/version?version={version}	Gets all AssetLevelLocations with a version higher than the specified.				
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevellocations/search?field={field}&value={value}	Gets all AssetLevelLocations containing the search criteria on the specified field.				
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevellocations?page={page}&pageSize={pageSize}	Gets all the AssetLevelLocations using a paged return.				
POST	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevellocations	Creates a Asset Level Location.				
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevellocations?location={location}&assetAccessLevel={assetAccessLevel}	No Documentation Found.				
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevellocations/{id}	Updates a Asset Level Location by it's unique identifier.				
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevellocations?location={location}&assetAccessLevel={assetAccessLevel}	No Documentation Found.				
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/assetlevellocations/{id}	Deletes a specific Asset Level Location by it's unique identifier.				

Figure 8 - Example of the methods exposed by the Access Levels and associated services

This service makes it possible to obtain and manipulate Valuekeep's Access Levels. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assetaccesslevels**

This method allows you to obtain the full list of existing Access Levels.

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/assetaccesslevels
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assetaccesslevels/{assetLevelKey}**

This method allows you to obtain the details of a specific Access Level through its natural key.

Example:

Obtaining the details of the Access Level with the natural key "Sample001"

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/assetaccesslevels/Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assetaccesslevels? page={page}&pagesize={pagesize}**

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Access Levels:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/assetaccesslevels?
page=1&pagesize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assetaccesslevels? field={field}&value={value}**

This method allows you to search for a specific field of the Level.

Example:

Obtaining all Levels with the "Review" note.

```
GET /api/mytenant/myorganization/maintenancemanagementcore/assetaccesslevels?
field=Notes&value=Review
```

- **POST /api/mytenant/myorganization/maintenancemanagementcore/assetaccesslevels**

This method allows you to create a new Access Level. To that end, the http message should have the "body" filled in with the structure of the resource *AssetAccessLevelResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/assetaccesslevels
Body:
{
  "WorkCenterKey": "Sample001",
  "Description": "Sample 001",
}
```

- **PUT /api/mytenant/myorganization/maintenancemanagementcore/assetaccesslevels /{assetLevelKey}**

This method allows you to update an Access Level. To that end, the http message should have the "body" filled in with the full structure of the resource *AssetAccessLevelResource* (including the unchanged ones).

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/assetaccesslevels/{assetLevelKey}**

This method allows you to delete a specific Access Level through its natural key.

Example:

Deleting the Level with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/assetaccesslevels/Sample0
01
```

- **api/mytenant/myorganization/maintenancemanagementcore/assetlevelassets**
- **api/mytenant/myorganization/maintenancemanagementcore/assetlevellocations**

These methods are analogous to the Access Level service, but they are used to manage the association between existing Access Levels and Assets or Locations. To create, in this case you must only define the resource with the natural key of the Access Level and the natural key of the Asset/Location.

Monitorings

GET	/api/{tenant}/{organization}/maintenancemanagementcore/monitorings	Gets all Monitorings.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/monitorings?monitoringKey={monitoringKey}	Gets a Monitoring by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/monitorings/{id}	Gets a Monitoring by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/monitorings/search/version/{version}	Gets all Monitorings with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/monitorings/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/monitorings?page={page}&pageSize={pageSize}	Gets all the Monitorings using a paged return.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/monitorings	Creates a Monitoring.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/monitorings?monitoringKey={monitoringKey}	Updates a Monitoring.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/monitorings/{id}	Updates a Monitoring by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/monitorings?monitoringKey={monitoringKey}	Deletes a specific Monitoring.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/monitorings/{id}	Deletes a specific Monitoring by it's unique identifier.

Figure 9 - Example of the methods exposed by the Monitorings service

This service makes it possible to obtain and manipulate Valuekeep's Monitorings. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/monitorings**

This method allows you to obtain the full list of existing Monitorings.

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/monitorings
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/monitorings/{monitoringKey}**

This method allows you to obtain the details of a specific Monitoring through its natural key.

Example:

Obtaining the details of the Monitoring with the natural key "Sample001"

```
GET /api/mytenant/myorganization/maintenancemanagementcore/monitorings/Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/monitorings? page={page}&pagesize={pagesize}**

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Monitorings:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/monitorings?
page=1&pagesize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/monitorings? field={field}&value={value}**

This method allows you to search for a specific field of the Monitoring.

Example:

Obtaining all Monitorings with the "Review" note.

```
GET /api/mytenant/myorganization/maintenancemanagementcore/monitorings?
field=Notes&value=Review
```

- **POST /api/mytenant/myorganization/maintenancemanagementcore/monitorings**

This method allows you to create a new Monitoring. To that end, the http message should have the "body" filled in with the structure of the resource *MonitoringResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/monitorings
Body:
{
  "MonitoringKey": "Sample001",
  "Description": "Sample 001",
}
```

- **PUT /api/mytenant/myorganization/maintenancemanagementcore/monitorings /{monitoringKey}**

This method allows you to update a Monitoring. To that end, the http message should have the "body" filled in with the full structure of the resource *MonitoringResource* (including the unchanged ones).

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/monitorings/{monitoringKey}**

This method allows you to delete a specific Monitoring through its natural key.

Example:

Deleting the Monitoring with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/monitorings/Sample001
```


Monitoring Groups

GET	/api/{tenant}/{organization}/maintenancemanagementcore/monitoringgroups	Gets all MonitoringGroups.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/monitoringgroups?monitoringGroupKey={monitoringGroupKey}	Gets a Monitoring Group by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/monitoringgroups/{id}	Gets a Monitoring Group by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/monitoringgroups/search/version/{version}	Gets all MonitoringGroups with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/monitoringgroups/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/monitoringgroups?page={page}&pageSize={pageSize}	Gets all the MonitoringGroups using a paged return.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/monitoringgroups	Creates a Monitoring Group.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/monitoringgroups?monitoringGroupKey={monitoringGroupKey}	Updates a Monitoring Group.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/monitoringgroups/{id}	Updates a Monitoring Group by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/monitoringgroups?monitoringGroupKey={monitoringGroupKey}	Deletes a specific Monitoring Group.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/monitoringgroups/{id}	Deletes a specific Monitoring Group by it's unique identifier.

Figure 10 - Example of the methods exposed by the Monitoring Groups service

This service makes it possible to obtain and manipulate Valuekeep's Valuekeep's Monitoring Groups. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/monitoringgroups**

This method allows you to obtain the full list of existing Monitoring Groups.

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/monitoringgroups
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/monitoringgroups/{monitoringGroupKey}**

This method allows you to obtain the details of a specific Monitoring Group through its natural key.

Example:

Obtaining the details of the Group with the natural key "Sample001"

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/monitoringgroups/Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/monitoringgroups? page={page}&pagesize={pagesize}**

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Monitoring Groups:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/monitoringgroups?
page=1&pagesize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/monitoringgroups? field={field}&value={value}**

This method allows you to search for a specific field of the Group.

Example:

Obtaining all Groups with the "Review" note.

```
GET /api/mytenant/myorganization/maintenancemanagementcore/monitoringgroups?
field=Notes&value=Review
```

- **POST /api/mytenant/myorganization/maintenancemanagementcore/monitoringgroups**

This method allows you to create a new Monitoring Group. To that end, the http message should have the "body" filled in with the structure of the resource *MonitoringGroupResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/monitoringgroups
Body:
{
  "MonitoringGroupKey": "Sample001",
  "Description": "Sample 001",
}
```

- **PUT /api/mytenant/myorganization/maintenancemanagementcore/monitoringgroups /{monitoringGroupKey}**

This method allows you to update a Monitoring Group. To that end, the http message should have the "body" filled in with the full structure of the resource *MonitoringGroupResource* (including the unchanged ones).

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/monitoringgroups/{monitoringGroupKey}**

This method allows you to delete a specific Monitoring Group through its natural key.

Example:

Deleting the Group with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/monitoringgroups/Sample001
```

Maintenance Management

Failures

GET	/api/{tenant}/{organization}/maintenancemanagementcore/failures	Gets all Failures.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/failures?failureKey={failureKey}	Gets a Failure by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/failures/{id}	Gets a Failure by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/failures/search/version/{version}	Gets all Failures with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/failures/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/failures?page={page}&pageSize={pageSize}	Gets all the Failures using a paged return.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/failures	Creates a Failure.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/failures?failureKey={failureKey}	Updates a Failure.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/failures/{id}	Updates a Failure by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/failures?failureKey={failureKey}	Deletes a specific Failure.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/failures/{id}	Deletes a specific Failure by it's unique identifier.

Figure 11 - Example of the methods exposed by the Failures service

This service makes it possible to obtain and manipulate Valuekeep's Failures. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/failures**

This method allows you to get the full list of existing Failures.

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/failures
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/failures/{failureKey}**

This method allows you to obtain the details of a specific Failure through its natural key.

Example:

Obtaining the details of the Failure with the natural key "Sample001"

```
GET /api/mytenant/myorganization/maintenancemanagementcore/failures/Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/failures? page={page}&pagesize={pagesize}**

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Failures:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/failures?
page=1&pagesize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/failures? field={field}&value={value}**

This method allows you to search for a specific field of the Failure.

Example:

Obtaining all Failures with the "Review" note.

```
GET /api/mytenant/myorganization/maintenancemanagementcore/failures?
field=Remarks&value=Rever
```

- **POST /api/mytenant/myorganization/maintenancemanagementcore/failures**

This method allows you to create a new Failure. To that end, the http message should have the "body" filled in with the structure of the resource *FailureResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/failures
Body:
{
  "FailureKey": "Sample001",
  "Description": "Sample 001",
}
```

- **PUT /api/mytenant/myorganization/maintenancemanagementcore/failures /{failureKey}**

This method allows you to update a Failure. To that end, the http message should have the "body" filled in with the full structure of the resource *FailureResource* (including the unchanged ones).

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/failures/{failureKey}**

This method allows you to delete a specific Failure through its natural key.

Example:

Deleting the Failure with the natural key *Sample001*:

```
DELETE /api/mytenant/myorganization/maintenancemanagementcore/failures/Sample001
```

Failure Causes

GET	/api/{tenant}/{organization}/maintenancemanagementcore/failurecauses	Gets all FailureCauses.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/failurecauses?failureCauseKey={failureCauseKey}	Gets a Failure Cause by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/failurecauses/{id}	Gets a Failure Cause by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/failurecauses/search/version/{version}	Gets all FailureCauses with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/failurecauses/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/failurecauses?page={page}&pageSize={pageSize}	Gets all the FailureCauses using a paged return.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/failurecauses	Creates a Failure Cause.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/failurecauses?failureCauseKey={failureCauseKey}	Updates a Failure Cause.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/failurecauses/{id}	Updates a Failure Cause by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/failurecauses?failureCauseKey={failureCauseKey}	Deletes a specific Failure Cause.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/failurecauses/{id}	Deletes a specific Failure Cause by it's unique identifier.

Figure 12 - Example of the methods exposed by the Failure Causes service

This service makes it possible to obtain and manipulate Valuekeep's Failure Causes. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/failurecauses**

This method allows you to get the full list of existing Causes.

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/failurecauses
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/failurecauses/{failureCauseKey}**

This method allows you to obtain the details of a specific Cause through its natural key.

Example:

Obtaining the details of the Cause with the natural key "Sample001"

```
GET /api/mytenant/myorganization/maintenancemanagementcore/failurecauses/Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/failurecauses? page={page}&pagesize={pagesize}**

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Causes:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/failurecauses?
page=1&pagesize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/failurecauses? field={field}&value={value}**

This method allows you to search for a specific field of the Cause.

Example:

Obtaining all Causes with the "Review" note.

```
GET /api/mytenant/myorganization/maintenancemanagementcore/failurecauses?
field=Remarks&value=Rever
```

- **POST /api/mytenant/myorganization/maintenancemanagementcore/failurecauses**

This method allows you to create a new Failure Cause. To that end, the http message should have the "body" filled in with the structure of the resource *FailureCauseResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/failurecauses
Body:
{
  "FailureCauseKey": "Sample001",
  "Description": "Sample 001",
}
```

- **PUT /api/mytenant/myorganization/maintenancemanagementcore/failurecauses/{failureCauseKey}**

This method allows you to update a Cause. To that end, the http message should have the "body" filled in with the full structure of the resource *FailureCauseResource* (including the unchanged ones).

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/failurecauses/{failureCauseKey}**

This method allows you to delete a specific Cause through its natural key.

Example:

Deleting the Failure Cause with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/failurecauses/Sample001
```

Failure Actions

GET	/api/{tenant}/{organization}/maintenancemanagementcore/failureactions	Gets all FailureActions.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/failureactions?failureActionKey={failureActionKey}	Gets a Failure Action by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/failureactions/{id}	Gets a Failure Action by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/failureactions/search/version/{version}	Gets all FailureActions with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/failureactions/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/failureactions?page={page}&pageSize={pageSize}	Gets all the FailureActions using a paged return.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/failureactions	Creates a Failure Action.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/failureactions?failureActionKey={failureActionKey}	Updates a Failure Action.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/failureactions/{id}	Updates a Failure Action by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/failureactions?failureActionKey={failureActionKey}	Deletes a specific Failure Action.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/failureactions/{id}	Deletes a specific Failure Action by it's unique identifier.

Figure 13 - Example of the methods exposed by the Failure Actions service

This service makes it possible to obtain and manipulate Valuekeep's Failure Actions. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/failureactions**

This method allows you to get the full list of existing Actions.

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/failureactions
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/failureactions/{failureActionKey}**

This method allows you to obtain the details of a specific Action through its natural key.

Example:

Obtaining the details of the Action with the natural key "Sample001"

```
GET /api/mytenant/myorganization/maintenancemanagementcore/failureactions/Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/failureactions? page={page}&pagesize={pagesize}**

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Actions:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/failureactions?
page=1&pagesize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/failureactions? field={field}&value={value}**

This method allows you to search for a specific field of the Action.

Example:

Obtaining all Actions with the "Review" note.

```
GET /api/mytenant/myorganization/maintenancemanagementcore/failureactions?
field=Remarks&value=Rever
```

- **POST /api/mytenant/myorganization/maintenancemanagementcore/failureactions**

This method allows you to create a new Failure Action. To that end, the http message should have the "body" filled in with the structure of the resource *FailureActionResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/failureactions
Body:
{
  "FailureActionKey": "Sample001",
  "Description": "Sample 001",
}
```

- **PUT /api/mytenant/myorganization/maintenancemanagementcore/failureactions/{failureActionKey}**

This method allows you to update an Action. To that end, the http message should have the "body" filled in with the full structure of the resource *FailureActionResource* (including the unchanged ones).

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/failureactions/{failureActionKey}**

This method allows you to delete a specific Action through its natural key.

Example:

Deleting the Failure Action with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/failureactions/Sample001
```


Failure Symptoms

GET	/api/{tenant}/{organization}/maintenancemanagementcore/failuresymptoms	Gets all FailureSymptoms.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/failuresymptoms?failureSymptomKey={failureSymptomKey}	Gets a Failure Symptom by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/failuresymptoms/{id}	Gets a Failure Symptom by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/failuresymptoms/search/version/{version}	Gets all FailureSymptoms with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/failuresymptoms/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/failuresymptoms?page={page}&pageSize={pageSize}	Gets all the FailureSymptoms using a paged return.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/failuresymptoms	Creates a Failure Symptom.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/failuresymptoms?failureSymptomKey={failureSymptomKey}	Updates a Failure Symptom.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/failuresymptoms/{id}	Updates a Failure Symptom by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/failuresymptoms?failureSymptomKey={failureSymptomKey}	Deletes a specific Failure Symptom.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/failuresymptoms/{id}	Deletes a specific Failure Symptom by it's unique identifier.

Figure 14 - Example of the methods exposed by the Failure Symptoms service

This service makes it possible to obtain and manipulate Valuekeep's Failure Symptoms. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/failuresymptoms**

This method allows you to get the full list of existing Symptoms.

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/failuresymptoms
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/failuresymptoms/{failureSymptomKey}**

This method allows you to obtain the details of a specific Symptom through its natural key.

Example:

Obtaining the details of the Symptom with the natural key "Sample001"

```
GET /api/mytenant/myorganization/maintenancemanagementcore/failuresymptoms/Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/failuresymptoms? page={page}&pagesize={pagesize}**

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Symptoms:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/failuresymptoms?
page=1&pagesize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/failuresymptoms? field={field}&value={value}**

This method allows you to search for a specific field of the Symptom.

Example:

Obtaining all Symptoms with the "Review" note.

```
GET /api/mytenant/myorganization/maintenancemanagementcore/failuresymptoms?
field=Remarks&value=Rever
```

- **POST /api/mytenant/myorganization/maintenancemanagementcore/failuresymptoms**

This method allows you to create a new Failure Symptom. To that end, the http message should have the "body" filled in with the structure of the resource *FailureSymptomResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/failuresymptoms
Body:
{
  "FailureSymptomKey": "Sample001",
  "Description": "Sample 001",
}
```

- **PUT /api/mytenant/myorganization/maintenancemanagementcore/failuresymptoms/{failureSymptomKey}**

This method allows you to update a Symptom. To that end, the http message should have the "body" filled in with the full structure of the resource *FailureSymptomResource* (including the unchanged ones).

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/failuresymptoms/{failureSymptomKey}**

This method allows you to delete a specific Symptom through its natural key.

Example:

Deleting the Failure Symptom with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/failuresymptoms/Sample001
```

Issues

GET	/api/{tenant}/{organization}/maintenancemanagementcore/issues	Gets all Issues.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/issues?issueKey={issueKey}	Gets a Issue by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/issues/{id}	Gets a Issue by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/issues/search/version/{version}	Gets all Issues with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/issues/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/issues?page={page}&pageSize={pageSize}	Gets all the Issues using a paged return.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/issues	Creates a Issue.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/issues?issueKey={issueKey}	Updates a Issue.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/issues/{id}	Updates a Issue by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/issues?issueKey={issueKey}	Deletes a specific Issue.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/issues/{id}	Deletes a specific Issue by it's unique identifier.

Figure 15 - Example of the methods exposed by the Issues service

This service makes it possible to obtain and manipulate Valuekeep's Issues. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/issues**

This method allows you to get the full list of existing Issues.

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/issues
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/issues/{issueKey}**

This method allows you to obtain the details of a specific Issue through its natural key.

Example:

Obtaining the details of the Issue with the natural key "Sample001"

```
GET /api/mytenant/myorganization/maintenancemanagementcore/issues/Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/issues?page={page}&pageSize={pageSize}**

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Issues:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/issues?
page=1&pageSize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/issues?field={field}&value={value}**

This method allows you to search for a specific field of the Issue.

Example:

Obtaining all Issues with the "Review" note.

```
GET /api/mytenant/myorganization/maintenancemanagementcore/issues?
field=Remarks&value=Rever
```

- **POST /api/mytenant/myorganization/maintenancemanagementcore/issues**

This method allows you to create a new Issue. To that end, the http message should have the "body" filled in with the structure of the resource *IssueResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/issues
Body:
{
  "IssueKey": "Sample001",
  "Description": "Sample 001",
}
```

- **POST /api/mytenant/myorganization/maintenancemanagementcore/issues/CreateIssue**

This method offers a simple way of creating an Issue for a specific Asset, Location, Asset Route or Location Route. The "body" of the message should be filled in with the structure of the resource *IssueInfoResource*.

Example:

```
POST
/api/mytenant/myorganization/maintenancemanagementcore/issues/CreateIssue

Body:
{
  "IssueKey": "ISSUE-0025",
  "Description": "Issue",
  "IssueTypeKey": "EQUIP",
  "AssetKey": "AC-0022",
  "LocationKey": null,
  "AssetRouteKey": null,
  "LocationRouteKey": null,
  "IssuerKey": "0002",
  "MaintenancePlanKeys": ["PM-000012.1", "PM-000013.1"],
  "WorkCenterKey": "CTEGH",
  "PriorityLevelKey": "HIGH",
  "ApprovedPriorityLevelKey": "HIGH",
  "Remarks": "General Remarks",
  "IssueDate": "2015-11-25T13:37:00.000+00:00"
}
```

Note: If the Priority Level and the Approved Priority Level are not specified, the creation of the issue uses the default Priority Level and Approved Priority Level. The “MaintenancePlanKeys” field can use one or more maintenance plans and it must be filled in with the natural key for the maintenance plan. To specify to which asset, location or route you want to create the issue for, it is only necessary to fill in the respective key. If an asset, location or route is not specified, the issue is created in the “Pending” state, as a draft.

- **PUT /api/mytenant/myorganization/maintenancemanagementcore/issues/{issueKey}**

This method allows you to update an Issue. To that end, the http message should have the “body” filled in with the full structure of the resource *IssueResource* (including the unchanged ones).

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/issues/{issueKey}**

This method allows you to delete a specific Issue through its natural key.

Example:

Deleting the Issue with the natural key *Sample001*:

```
DELETE /api/mytenant/myorganization/maintenancemanagementcore/issues/Sample001
```

Issue Reasons

GET	/api/{tenant}/{organization}/maintenancemanagementcore/issuereasons	Gets all IssueReasons.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/issuereasons?reasonKey={reasonKey}	Gets a Issue Reason by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/issuereasons/{id}	Gets a Issue Reason by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/issuereasons/search/version/{version}	Gets all IssueReasons with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/issuereasons/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/issuereasons?page={page}&pageSize={pageSize}	Gets all the IssueReasons using a paged return.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/issuereasons	Creates a Issue Reason.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/issuereasons?reasonKey={reasonKey}	Updates a Issue Reason.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/issuereasons/{id}	Updates a Issue Reason by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/issuereasons?reasonKey={reasonKey}	Deletes a specific Issue Reason.

Figure 16 - Example of the methods exposed by the Issue Reasons service

This service makes it possible to obtain and manipulate Valuekeep's Issue Reasons. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/issuereasons**

This method allows you to get the full list of existing Issue Reasons.

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/issuereasons
```

- **GET**
/api/mytenant/myorganization/maintenancemanagementcore/issuereasons/{issueReasonKey}

This method allows you to obtain the details of a specific Issue Reason through its natural key.

Example:

Obtaining the details of the Reason with the natural key "Sample001"

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/issuereasons/Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/issuereasons?**
page={page}&pagesize={pagesize}

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Reasons:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/issuereasons?
page=1&pagesize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/issuereasons?**
field={field}&value={value}

This method allows you to search for a specific field of the Reason.

Example:

Obtaining all Reasons with the "Review" note.

```
GET /api/mytenant/myorganization/maintenancemanagementcore/issuereasons?
field=Remarks&value=Rever
```

- **POST /api/mytenant/myorganization/maintenancemanagementcore/issuereasons**

This method allows you to create a new Reason. To that end, the http message should have the "body" filled in with the structure of the resource *IssueReasonResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/issuereasons
Body:
{
  "IssueReasonKey": "Sample001",
  "Description": "Sample 001",
}
```

- **PUT**
/api/mytenant/myorganization/maintenancemanagementcore/issuereasons/{issueReasonKey}

This method allows you to update a Reason. To that end, the http message should have the "body" filled in with the full structure of the resource *IssueReasonResource* (including the unchanged ones).

- **DELETE**
/api/mytenant/myorganization/maintenancemanagementcore/issuereasons/{issueReasonKey}

This method allows you to delete a specific Reason through its natural key.

Example:

Deleting the Reason with the natural key *Sample001*:

DELETE
/api/mytenant/myorganization/maintenancemanagementcore/issuereasons/Sample001

Asset Shutdowns

GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetshutdowns	Gets all AssetShutdowns.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetshutdowns?shutdownKey={shutdownKey}	Gets a Asset Shutdown by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetshutdowns/{id}	Gets a Asset Shutdown by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetshutdowns/search/version/{version}	Gets all AssetShutdowns with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetshutdowns/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/assetshutdowns?page={page}&pageSize={pageSize}	Gets all the AssetShutdowns using a paged return.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/assetshutdowns	Creates a Asset Shutdown.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/assetshutdowns?shutdownKey={shutdownKey}	Updates a Asset Shutdown.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/assetshutdowns/{id}	Updates a Asset Shutdown by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/assetshutdowns?shutdownKey={shutdownKey}	Deletes a specific Asset Shutdown.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/assetshutdowns/{id}	Deletes a specific Asset Shutdown by it's unique identifier.

Figure 17 - Example of the methods exposed by the Asset Shutdowns service

This service makes it possible to obtain and manipulate Valuekeep's Asset Shutdowns. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assetshutdowns**

This method allows you to get the full list of existing Shutdowns.

Example:

GET /api/mytenant/myorganization/maintenancemanagementcore/assetshutdowns

- **GET**
/api/mytenant/myorganization/maintenancemanagementcore/assetshutdowns/{assetShutdownKey}

This method allows you to obtain the details of a specific Shutdown through its natural key.

Example:

Obtaining the details of the Shutdown with the natural key "Sample001"

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/assetshutdowns/Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assetshutdowns?page={page}&pagesize={pagesize}**

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Shutdowns:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/assetshutdowns?
page=1&pagesize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/assetshutdowns?field={field}&value={value}**

This method allows you to search for a specific field of the Shutdown.

Example:

Obtaining all Shutdowns with the "Review" note.

```
GET /api/mytenant/myorganization/maintenancemanagementcore/assetshutdowns?
field=Remarks&value=Rever
```

- **POST /api/mytenant/myorganization/maintenancemanagementcore/assetshutdowns**

This method allows you to create a new Shutdown. To that end, the http message should have the "body" filled in with the structure of the resource *AssetShutdownResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/assetshutdowns
Body:
{
  "AssetShutdownKey": "Sample001",
  "Description": "Sample 001",
}
```

- **PUT /api/mytenant/myorganization/maintenancemanagementcore/assetshutdowns/{assetShutdownKey}**

This method allows you to update a Shutdown. To that end, the http message should have the "body" filled in with the full structure of the resource *AssetShutdownResource* (including the unchanged ones).

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/assetshutdowns/{assetShutdownKey}**

This method allows you to delete a specific Shutdown through its natural key.

Example: Deleting the Shutdown with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/assetshutdowns/Sample001
```


Location Shutdowns

GET	/api/{tenant}/{organization}/maintenancemanagementcore/locationshutdowns	Gets all LocationShutdowns.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/locationshutdowns?shutdownKey={shutdownKey}	Gets a Location Shutdown by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/locationshutdowns/{id}	Gets a Location Shutdown by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/locationshutdowns/search/version/{version}	Gets all LocationShutdowns with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/locationshutdowns/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/locationshutdowns?page={page}&pageSize={pageSize}	Gets all the LocationShutdowns using a paged return.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/locationshutdowns	Creates a Location Shutdown.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/locationshutdowns?shutdownKey={shutdownKey}	Updates a Location Shutdown.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/locationshutdowns/{id}	Updates a Location Shutdown by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/locationshutdowns?shutdownKey={shutdownKey}	Deletes a specific Location Shutdown.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/locationshutdowns/{id}	Deletes a specific Location Shutdown by it's unique identifier.

Figure 18 - Example of the methods exposed by the Location Shutdowns service

This service makes it possible to obtain and manipulate Valuekeep's Location Shutdowns. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/locationshutdowns**

This method allows you to get the full list of existing Shutdowns.

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/locationshutdowns
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/locationshutdowns/{locationShutdownKey}**

This method allows you to obtain the details of a specific Shutdown through its natural key.

Example:

Obtaining the details of the Shutdown with the natural key "Sample001"

```
GET /api/mytenant/myorganization/maintenancemanagementcore/locationshutdowns/Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/locationshutdowns? page={page}&pagesize={pagesize}**

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Shutdowns:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/locationshutdowns?
page=1&pagesize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/locationshutdowns? field={field}&value={value}**

This method allows you to search for a specific field of the Shutdown.

Example:

Obtaining all Shutdowns with the "Review" note.

```
GET /api/mytenant/myorganization/maintenancemanagementcore/locationshutdowns?
field=Remarks&value=Rever
```

- **POST /api/mytenant/myorganization/maintenancemanagementcore/locationshutdowns**

This method allows you to create a new Shutdown. To that end, the http message should have the "body" filled in with the structure of the resource *LocationShutdownResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/locationshutdowns
Body:
{
  "LocationShutdownKey": "Sample001",
  "Description": "Sample 001",
}
```

- **PUT /api/mytenant/myorganization/maintenancemanagementcore/locationshutdowns /{locationShutdownKey}**

This method allows you to update a Shutdown. To that end, the http message should have the "body" filled in with the full structure of the resource *LocationShutdownResource* (including the unchanged ones).

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/locationshutdowns/{locationShutdownKey}**

This method allows you to delete a specific Shutdown through its natural key.

Example:

Deleting the Shutdown with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/locationshutdowns/Sample0
01
```

MaintenancePlans

TimeUnit

0	Value when the value list has no value.
1	Day Time Unit.
2	Hour Time Unit.
3	Minutes Time Unit.

Work Orders

GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorders	Gets all WorkOrders.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorders?workOrderKey={workOrderKey}	Gets a Work Order by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorders/{id}	Gets a Work Order by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorders/search/version/{version}	Gets all WorkOrders with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorders/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorders?page={page}&pageSize={pageSize}	Gets all the WorkOrders using a paged return.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorders/GetWorkOrderPlanTasksInfos?version={version}	Gets the WorkOrderPlanTasksInfos.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorders/GetWorkOrderStateTransitions?version={version}	Gets the work order state transitions.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorders/GetWorkOrderStates?version={version}	Gets the work order states.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorders/GetWorkOrdersByState?stateKey={stateKey}&version={version}	Gets the list of workorders by state.

Figure 19 - Example of the methods exposed by the Work Orders service (Part I)

POST	/api/{tenant}/{organization}/maintenancemanagementcore/workorders	Creates a Work Order.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/workorders/CreateWorkOrder	Creates the work order.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorders?workOrderKey={workOrderKey}	Updates a Work Order.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorders/{id}	Updates a Work Order by it's unique identifier.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorders/ChangeWorkOrderState	Changes the state of the work order.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/workorders?workOrderKey={workOrderKey}	Deletes a specific Work Order.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/workorders/{id}	Deletes a specific Work Order by it's unique identifier.

Figure 20 - Example of the methods exposed by the Work Orders service (Part II)

This service makes it possible to obtain and manipulate Valuekeep's Work Orders. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorders**

This method allows you to obtain the full list of existing Work Orders.

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/workorders
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorders/{workOrderKey}**

This method allows you to obtain the details of a specific Order through its natural key.

Example:

Obtaining the details of the Order with the natural key "Sample001"

```
GET /api/mytenant/myorganization/maintenancemanagementcore/workorders/Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorders?page={page}&pagesize={pagesize}**

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Work Orders:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/workorders?
page=1&pagesize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorders/search/version?version={value}**

This method allows you to search for a specific version of the Work Order.

Returns all the Work Order from a specific version onwards. The version parameter is a number with 12 digits which identifies the record update. To obtain a full list use the version "AAAAAAAAAAAA=".

Note: This method returns the WorkOrderAttachments and AssociatedAttachments.

Example:

Obtaining all Orders with the "Review" note.

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorders/search/version
?version=AAAAAAAAAAAA=
```

- **GET**
/api/mytenant/myorganization/maintenancemanagementcore/workorders/search/version/{version}

This method allows you to search for a specific version of the Work Order.

Returns all the Work Order from a specific version onwards. The version parameter is a number with 12 digits which identifies the record update. To obtain a full list use the version "AAAAAAAAAAAA=".

Note: This method returns the WorkOrderAttachments and AssociatedAttachments.

Example:

Obtaining all Orders with the "Review" note.

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorders/search/version
/AAAAAAAAAAAA=
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorders?field={field}&value={value}**

This method allows you to search for a specific field of the Work Order.

Example:

Obtaining all Orders with the "Review" note.

```
GET /api/mytenant/myorganization/maintenancemanagementcore/workorders?
field=Remarks&value=Rever
```

- **GET**
/api/mytenant/myorganization/maintenancemanagementcore/workorders/GetWorkOrderPlanTaskInfos? version={version}

Returns a summary (*WorkOrderPlanTasksInfo*) of all the Work Order tasks from a specific version onwards. The version parameter is a number with 12 digits which identifies the record update. To obtain a full list use the version "AAAAAAAAAAAA=".

Example:

Obtaining all task summaries of all Work Orders.

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorders/GetWorkOrderPlanTaskInfos?version=AAAAAAAAAAAA=
```

- **GET**

/api/mytenant/myorganization/maintenancemanagementcore/workorders/GetWorkOrderStateTransitions? version={version}

Returns a summary (*SchemaEntityStateTransitionInfo*) of all state transitions of the Work Orders from a specific version onwards. The version parameter is a number with 12 digits which identifies the record update. To obtain a full list use the version "AAAAAAAAAAAA=".

Example:

Obtaining all state transitions of the Work Orders.

```
GET /api/mytenant/myorganization/maintenancemanagementcore/workorders/
GetWorkOrderStateTransitions?version=AAAAAAAAAAAA=
```

- **GET**

/api/mytenant/myorganization/maintenancemanagementcore/workorders/GetWorkOrderStates ? version={version}

Returns a summary (*SchemaEntityStateInfo*) of all possible states for the Work Orders from a specific version onwards. The result includes the localized descriptions for each supported culture. The version parameter is a number with 12 digits which identifies the record update. To obtain a full list use the version "AAAAAAAAAAAA=".

Example:

Obtaining all Work Orders states.

```
GET /api/mytenant/myorganization/maintenancemanagementcore/workorders/
GetWorkOrderStates?version=AAAAAAAAAAAA=
```

- **GET**

/api/mytenant/myorganization/maintenancemanagementcore/workorders/GetWorkOrdersByState?stateKey={stateKey}&version={version}

Returns a list of all Work Orders which are in a specific state and from a specific version onwards. The version parameter is a number with 12 digits which identifies the record update. To obtain a full list use the version "AAAAAAAAAAAA=".

Example:

Obtaining all Work Orders in progress.

```
GET /api/mytenant/myorganization/maintenancemanagementcore/workorders/
GetWorkOrdersByState?stateKey=004&version=AAAAAAAAAAAA=
```

- **GET**

/api/mytenant/myorganization/maintenancemanagementcore/workorders/GetDeletedWorkOrders?version={version}

Returns a list of all Work Orders which are marked as deleted from a specific version onwards. The version parameter is a number with 12 digits which identifies the record update. To obtain a full list use the version "AAAAAAAAAAAA=".

Example:

Obtaining all Work Orders in progress.

```
GET /api/mytenant/myorganization/maintenancemanagementcore/workorders/
GetDeletedWorkOrders?version=AAAAAAAAAAAA=
```

- **GET**

/api/mytenant/myorganization/maintenancemanagementcore/workorders/GetWorkOrdersByStates?stateKey={stateKey}&daysLimit={daysLimit}&version={version}

This method allows you to obtain the list of Work Orders from multiple states and from a version onwards. It also allows you to filter the last work orders by {daysLimit} days. For example, if the value of {daysLimit} is 1, it will return all the task executions created between yesterday and today. The minimum value of the version is: "AAAAAAAAAAAA=";

Examples:

GET

```
/api/mytenant/myorganization/maintenancemanagementcore/workorders/GetWorkOrdersByStates?version=AAAAAAAAAAAA=
```

GET

```
/api/mytenant/myorganization/maintenancemanagementcore/workorders/GetWorkOrdersByStates?stateKey=002.WORKORDER&daysLimit=1000&version=AAAAAAAAAAAA=
```

GET

```
/api/mytenant/myorganization/maintenancemanagementcore/workorders/GetWorkOrdersByStates?stateKey=001.WORKORDER&stateKey=002.WORKORDER&daysLimit=1000&version=AAAAAAAAAAAA=
```

GET

```
/api/mytenant/myorganization/maintenancemanagementcore/workorders/GetWorkOrdersByStates?stateKey=001.WORKORDER&version=AAAAAAAAAAAA=
```

GET

```
/api/mytenant/myorganization/maintenancemanagementcore/workorders/GetWorkOrdersByStates?stateKey=002.WORKORDER&daysLimit=1000&version=AAAAAAAAAAAA=
```

Note: Only the stateKey field isn't optional. All the others can be left blank.

- **GET**

/api/mytenant/myorganization/maintenancemanagementcore/workorders/GetAttachmentsByVersionHigherThan?version={value}

This method allows you to search for a specific version of the Work Order attachments.

Returns all the Work Order attachments from a specific version onwards. The version parameter is a number with 12 digits which identifies the records to get. To obtain a full list use the "AAAAAAAAAAAA=" version.

Example:

Obtaining all Work Order Attachments.

GET

```
/api/mytenant/myorganization/maintenancemanagementcore/workorders/GetAttachmentsByVersionHigherThan?version=AAAAAAAAAAAA=
```

- **POST /api/mytenant/myorganization/maintenancemanagementcore/workorders**

This method allows you to create a new Work Order. To that end, the http message should have the "body" filled in with the structure of the resource *WorkOrderResource*.

Example:

```

POST /api/mytenant/myorganization/maintenancemanagementcore/workorders
Body:
{
  "WorkOrderKey": "Sample001",
  "Description": "Sample 001",
}

```

- **POST**
/api/mytenant/myorganization/maintenancemanagementcore/workorders/CreateWorkOrder

This method offers a simple way of creating a Work Order for a specific Asset, Location, Asset Route or Location Route. The “body” of the message should be filled in with the structure of the resource *WorkOrderInfoResource*.

Example:

```

POST
/api/mytenant/myorganization/maintenancemanagementcore/workorders/CreateWorkOrder

Body:
{
  "WorkOrderKey": "W0135",
  "Description": "Work Order",
  "WorkOrderTypeKey": "PR",
  "AssetKey": "AC-0022",
  "LocationKey": null,
  "AssetRouteKey": null,
  "LocationRouteKey": null,
  "MaintenancePlanKeys": ["PM-000012.1", "PM-000013.1"],
  "WorkCenterKey": "CTEGH",
  "PriorityLevelKey": "HIGH",
  "OperatingConditionKey": "ESTADO PARADO",
  "Remarks": "General Remarks",
  "Date": "2015-08-31T11:15:17.477+00:00"
}

```

Note: If the Work Center and the Priority Level are not specified, the creation of the work order uses the default Work Center and Priority Level. The “MaintenancePlanKeys” field can use one or more maintenance plans and it must be filled in with the natural key for the maintenance plan. This method replaces the following previous methods *CreateWorkOrderForAsset*, *CreateWorkOrderForLocation*, *CreateWorkOrderForAssetRoute*, *CreateWorkOrderForLocationRoute*. To specify to which asset, location or route you want to create the work order for, it is only necessary to fill in the respective key. If an asset, location or route is not specified, the work order is created in the “Pending” state, as a draft.

- **PUT** **/api/mytenant/myorganization/maintenancemanagementcore/workorders/{workOrderKey}**

This method allows you to update a Work Order. To that end, the http message should have the “body” filled in with the full structure of the resource *WorkOrderResource* (including the unchanged ones).

- **PUT**

/api/mytenant/myorganization/maintenancemanagementcore/workorders/ChangeWorkOrderState

This method allows you to change the state of several Work Orders. To do so, it is necessary to fill in the *StateResource*. The resource is filled in with the natural key of the WorkOrder (EntityKey), the natural key of the state (NewState), reason for the change (only used when it is rejected (reason)), and the date of the state change (StateChangeDate).

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/workorders/ChangeWorkOrderState
Body:
{
  "StateLines" : [
    {
      "EntityKey" : "OT0004",
      "NewState" : "003.WORKORDER",
      "Reason" : "TEST",
      "StateChangeDate" : "2015-07-27T11:15:17.477+00:00",
    },
    {
      "EntityKey" : "OT0005",
      "NewState" : "003.WORKORDER",
      "Reason" : "TEST3",
      "StateChangeDate" : "2015-07-27T11:15:17.477+00:00",
    }
  ]
}
```

- **DELETE**

/api/mytenant/myorganization/maintenancemanagementcore/workorders/{workOrderKey}

This method allows you to delete a specific Work Order through its natural key.

Example:

Deleting the Order with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/workorders/Sample001
```

Assignment of Work Order Tasks

GET	/api/{tenant}/{organization}/maintenancemanagementcore/workordertaskassignments	Gets all WorkOrderTaskAssignments.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workordertaskassignments?workOrderTaskAssignmentKey={workOrderTaskAssignmentKey}	Gets a WO Task Assignment Selection by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workordertaskassignments/{id}	Gets a WO Task Assignment Selection by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workordertaskassignments/search/version/{version}	Gets all WorkOrderTaskAssignments with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workordertaskassignments/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workordertaskassignments?page={page}&pageSize={pageSize}	Gets all the WorkOrderTaskAssignments using a paged return.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/workordertaskassignments	Creates a WO Task Assignment Selection.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workordertaskassignments?workOrderTaskAssignmentKey={workOrderTaskAssignmentKey}	Updates a WO Task Assignment Selection.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workordertaskassignments/{id}	Updates a WO Task Assignment Selection by it's unique identifier.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workordertaskassignments/AssignWorkOrderTask?workOrderKey={workOrderKey}&maintenancePlanKey={maintenancePlanKey}&taskKey={taskKey}&expertiseKey={expertiseKey}&employeeKey={employeeKey}	Assigns the work order task to an Expertise and/or employee.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/workordertaskassignments?workOrderTaskAssignmentKey={workOrderTaskAssignmentKey}	Deletes a specific WO Task Assignment Selection.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/workordertaskassignments/{id}	Deletes a specific WO Task Assignment Selection by it's unique identifier.

Figure 21 - Example of the methods exposed by the Assignment of Work Order Tasks service

This service makes it possible to obtain and manipulate Valuekeep's Assignments of Work Order Tasks. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workordertaskassignments**

This method allows you to get the full list of existing Assignments.

Example:

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workordertaskassignments
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workordertaskassignments/{workOrderTaskAssignmentKey}**

This method allows you to obtain the details of a specific Assignment through its natural key.

Example:

Obtaining the details of the Failure with the natural key "Sample001"

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workordertaskassignments/
Sample001
```

- **GET**
/api/mytenant/myorganization/maintenancemanagementcore/workordertaskassignments?page={page}&pagesize={pagesize}

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Assignments:

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workordertaskassignments?
page=1&pagesize=20
```

- **GET**
/api/mytenant/myorganization/maintenancemanagementcore/workordertaskassignments?field={field}&value={value}

This method allows you to search for a specific field of the Assignment.

Example:

Obtaining all Assignments for the Employee "Maria".

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workordertaskassignments?
field=Employee&value=Maria
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workordertaskassignments/GetDeletedWorkorderTaskAssignments?version={version}**

Returns a list of all Work Order Task Assignments which are marked as deleted from a specific version onwards. The version parameter is a number with 12 digits which identifies the record update. To obtain a full list use the version "AAAAAAAAAAAA=".

Example:

Obtaining all Work Orders in progress.

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workordertaskassignments/
GetDeletedWorkorderTaskAssignments?version=AAAAAAAAAAAA=
```

- **POST**
/api/mytenant/myorganization/maintenancemanagementcore/workordertaskassignments

This method allows you to create a new Assignment. To that end, the http message should have the "body" filled in with the structure of the resource *WorkOrderTaskAssignmentResource*.

Example:

```
POST
/api/mytenant/myorganization/maintenancemanagementcore/workordertaskassignments
Body:
{
```

```
"FailureKey": "Sample001",  
"Description": "Sample 001",  
}
```

- **PUT** `/api/mytenant/myorganization/maintenancemanagementcore/workordertaskassignments/{workOrderTaskAssignmentKey}`

This method allows you to update an Assignment. To that end, the http message should have the “body” filled in with the full structure of the resource *WorkOrderTaskAssignmentResource* (including the unchanged ones).

- **PUT**
`/api/mytenant/myorganization/maintenancemanagementcore/workordertaskassignments/AssignWorkOrderTask?workOrderKey={workOrderKey}&maintenancePlanKey={maintenancePlanKey}&taskKey={taskKey}&expertiseKey={expertiseKey}&employeeKey={employeeKey}`

This method allows you to make the Assignment of a specific Task to an Employee in a simple way. The parameters which identify the Work Order, Maintenance Plan, Task and finally the Employee or Specialty must be passed in the *QueryString*.

- **DELETE**
`/api/mytenant/myorganization/maintenancemanagementcore/workordertaskassignments/{workOrderTaskAssignmentKey}`

This method allows you to delete a specific Assignment through its natural key.

Example:

Deleting the Assignment with the natural key *Sample001*:

```
DELETE  
/api/mytenant/myorganization/maintenancemanagementcore/workordertaskassignments/  
Sample001
```

Execution of Work Order Tasks

GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderexecutions	Gets all WorkOrderExecutions.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderexecutions?workOrder={workOrder}	Gets a Work Order Execution by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderexecutions/{id}	Gets a Work Order Execution by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderexecutions/search/version/{version}	Gets all WorkOrderExecutions with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderexecutions/search/version?version={version}	Gets all WorkOrderExecutions with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderexecutions/search?field={field}&value={value}	Gets all WorkOrderExecutions containing the search criteria on the specified field.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderexecutions?page={page}&pageSize={pageSize}	Gets all the WorkOrderExecutions using a paged return.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderexecutions/GetWorkOrderTaskExecutionsByVersion?workOrderKey={workOrderKey}&version={version}&lastExecutionDays={lastExecutionDays}	Gets the work order task executions by version.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/workorderexecutions	Creates a Work Order Execution.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorderexecutions?workOrder={workOrder}	Updates a Work Order Execution.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorderexecutions/{id}	Updates a Work Order Execution by it's unique identifier.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorderexecutions/StartWorkOrderTask	Starts the work order task.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorderexecutions/CompleteWorkOrderTask	Completes the work order task.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/workorderexecutions?workOrder={workOrder}	Deletes a specific Work Order Execution.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/workorderexecutions/{id}	Deletes a specific Work Order Execution by it's unique identifier.

Figure 22 - Example of the methods exposed by the Executions of Work Order Tasks service

This service makes it possible to obtain and manipulate Valuekeep's Executions of Work Order Tasks. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions**

This method allows you to get the full list of existing Executions.

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions
```

- **GET**
/api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions/{workOrder}

This method allows you to obtain the details of a specific Execution through its natural key.

Example:

Obtaining the details of the Execution with the natural key "Sample001"

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions/Sample001
```

- **GET**
/api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions?page={page}&pagesize={pagesize}

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Executions:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions?page=1&pagesize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions?field={field}&value={value}**

This method allows you to search for a specific field of the Execution.

Example:

Obtaining all Executions created by the User "Carlos".

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions?field=CreatedBy&value=Carlos
```

- **GET**
/api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions/GetWorkOrderExecutionsTasksByVersion?WorkOrderKey={workOrderKey}&Version={version}&DaysLimit={daysLimit}

This method allows you to obtain the list of Work Order Task Executions from a specific work order and from a version onwards. It also allows you to filter the last task executions by *{daysLimit}* days. For example, if the value of *{daysLimit}* is 1, it will return all the task executions created between yesterday and today. The minimum value of the version is: "AAAAAAAAAAAA=";

Examples:

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions/GetWorkOrderTaskExecutionsByVersion?WorkOrderKey=&Version=AAAAAAAAAAAA=

/api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions/GetWorkOrderTaskExecutionsByVersion?WorkOrderKey=OT0010&Version=AAAAAAAAAAAA=
```

```
/api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions/  
GetWorkOrderTaskExecutionsByVersion?WorkOrderKey=OT0010&Version=AAAAAAAAAA=&DaysLimit=30
```

Note: All fields are optional. They can be left blank.

- **POST /api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions**

This method allows you to create a new Execution. To that end, the http message should have the “body” filled in with the structure of the resource *WorkOrderExecutionResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions  
Body:  
{  
  "WorkOrder": "Sample001",  
}
```

- **PUT**

/api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions/{workOrder}

This method allows you to update an Execution. To that end, the http message should have the “body” filled in with the full structure of the resource *WorkOrderExecutionResource* (including the unchanged ones).

- **PUT**

/api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions/StartWorkOrderTask

Example:

```
PUT  
/api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions/StartWorkOrderTask  
Body:  
{  
  "WorkOrderKey": "W00007",  
  "AssetKey": "AC-0021",  
  "TaskKey": "T-000001",  
  "EmployeeKey": "0002",  
  "MaintenancePlanKey": "PM-000005.2",  
  "Date": "2015-11-25T17:00:00.00+00:00",  
  "IsRestart" : "false"  
}
```

This method allows you to start a specific Task of a Work Order in a simple way. The parameters which identify the Work Order, Maintenance Plan and Task must be passed in the *TaskExecutionResource*; it is also necessary to identify the Employee who will start the Task as well as the context of that execution, that is, the Asset or Location. Optionally, it is also possible to indicate the start datetime for the execution.

This method can also be used to reopen tasks that have been completed. In this case, you must specify the attribute *IsRestart* with the value *true*, and in the *Date* attribute you must specify the date when the task is being reopened.

- **PUT**
/api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions/CompleteWorkOrderTask

Example:

```
PUT
/api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions/CompleteWorkOrderTask
Body:
{
  "WorkOrderKey": "W00007",
  "AssetKey": "AC-0021",
  "TaskKey": "T-000001",
  "EmployeeKey": "0002",
  "MaintenancePlanKey": "PM-000005.2",
  "Date": "2015-11-25T18:00:00.00+00:00"
}
```

This method allows you to finish a specific Task of a Work Order in a simple way. The parameters which identify the Work Order, Maintenance Plan and Task must be passed in the *TaskExecutionResource*; it is also necessary to identify the Employee who will start the Task as well as the context of that execution, that is, the Asset or Location. Optionally, it is also possible to indicate the end datetime for the execution.

- **DELETE**
/api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions/{workOrder}

This method allows you to delete a specific Execution through its natural key.

Example:

Deleting the Execution with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/workordertaskassignments/Sample001
```


Reporting Labor in the Work Order

GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportlabors	Gets all WorkOrderReportLabors.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportlabors?sequenceKey={sequenceKey}	Gets a Report Labor by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportlabors/{id}	Gets a Report Labor by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportlabors/search/version/{version}	Gets all WorkOrderReportLabors with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportlabors/search/version?version={version}	Gets all WorkOrderReportLabors with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportlabors/search?field={field}&value={value}	Gets all WorkOrderReportLabors containing the search criteria on the specified field.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportlabors?page={page}&pageSize={pageSize}	Gets all the WorkOrderReportLabors using a paged return.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportlabors/GetDeletedWorkOrderReportLabors?version={version}	Gets the deleted work order report labors with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportlabors/GetWorkOrderReportLaborByFilter?workOrderKey={workOrderKey}&employeeKey={employeeKey}&version={version}	Gets the work order report labor by filter.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportlabors	Creates a Report Labor.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportlabors/ReportLaborForWorkOrderTask	Reports the labor for work order task.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportlabors?sequenceKey={sequenceKey}	Updates a Report Labor.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportlabors/{id}	Updates a Report Labor by it's unique identifier.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportlabors/ChangeWorkOrderReportLaborExternalInfo	Changes the work order report labor external information.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportlabors?sequenceKey={sequenceKey}	Deletes a specific Report Labor.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportlabors/{id}	Deletes a specific Report Labor by it's unique identifier.

Figure 23 - Example of the methods exposed by the Work Order Report Labor service

This service makes it possible to obtain and manipulate Work Order Labor Reports in Valuekeep. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors**

This method allows you to get the full list of existing Reports.

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors/{sequenceKey}**

This method allows you to obtain the details of a specific Report through its natural key.

Example:

Obtaining the details of the Report with the natural key "Sample001"

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors/Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors/search/version?version=AAAAAAAAAAAA=**

This method allows you to obtain the list of Reports from a version onwards. The minimum value of the version is: "AAAAAAAAAAAA=".

Example:

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors/search/version?version=AAAAAAAAAAAA=
```

Notes: This method also returns the cost center key of the work order and the report notes in the following format:

```
[Work Order] - [Asset/Location/Asset Route/Location Route]
[Work Order Description]
[Remarks on each line of the report]
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors?page={page}&pagesize={pagesize}**

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Reports:

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors?page=1&pagesize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors?field={field}&value={value}**

This method allows you to search for a specific field of the Report.

Example:

Obtaining all Reports created by the User "Carlos".

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors?field=CreatedBy&value=Carlos
```

- **GET /api/{tenant}/{organization}/maintenancemanagementcore/workorderreportlabors/GetDeletedWorkOrderReportLabors?version=AAAAAAAAAAAA=**

This method allows you to obtain the list of Reports deleted from a version onwards, and the minimum value of the version is: "AAAAAAAAAAAA=".

Example:

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors/Get
DeletedWorkOrderReportLabors?version=AAAAAAAAAAAA=
```

Note: This method also returns the cost center key of the work order and the report notes in the following format:

```
[Work Order] - [Asset/Location/Asset Route/Location Route]
[Work Order Description]
[Remarks on each line of the report]
```

○ **GET**

/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportlabors/GetWorkOrderReportLaborByFilter?workOrderKey={workOrderKey}&employeeKey={employeeKey}&version={version}

This method allows you to obtain the list of Reports from a specific work order or employee and from a version onwards. The minimum value of the version is: "AAAAAAAAAAAA=";

Examples:

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors/Get
WorkOrderReportLaborByFilter?WorkOrderKey=&EmployeeKey=&Version=AAAAAAAAAAAA=

/api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors/Get
WorkOrderReportLaborByFilter?WorkOrderKey=&EmployeeKey=0002&Version=AAAAAAAAAAAA=

/api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors/Get
WorkOrderReportLaborByFilter?WorkOrderKey=&EmployeeKey=0003&Version=

/api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors/Get
WorkOrderReportLaborByFilter?WorkOrderKey=OT0010&EmployeeKey=&Version=
```

Note: All fields are optional. They can be left blank.

○ **POST /api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors**

This method allows you to create a new Labor Report. To that end, the http message should have the "body" filled in with the structure of the resource *WorkOrderReportLaborResource*.

Example:

```
POST
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors
Body:
{
  "SequenceKey": "Sample001",
  "Description": "Sample001",
}
```

Note: The Exchange Rate, End Date, Currency, Total Base and Total reporting fields are automatically updated, and the filled in values are ignored.

○ **POST**

/api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors/ReportLaborForWorkOrderTask

This method allows you to report Labor costs for a specific Task of a Work Order in a simple way. To that end, the http message should have the “body” filled in with the structure of the resource *ReportLaborForWorkOrderTaskResource*.

Example:

```
POST
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors/
ReportLaborForWorkOrderTask

Body:
{
  "WorkOrderKey": "OT0012",
  "MaintenancePlanKey": "PM-000005",
  "TaskKey": "T-000003",
  "EmployeeKey": "0002",
  "AssetKey": "AC-0021",
  "LocationKey": null,
  "StartDate": "2015-06-02T10:00:00+01:00",
  "EndDate": "2015-06-02T11:00:00+01:00",
  "Hours": "1"
}
```

The natural keys which identify the Work Order, Maintenance Plan, Task and the asset or location must be passed in the *resource*. It is also necessary to identify the Employee who spent hours to execute the Task. Optionally, it is also possible to indicate the start and end dates/times of the activity, in case the use of the task' execution start date and the current date/time is not specified.

The “Hours” field:

- specified, the “EndDate” will be automatically calculated by the “StartDate” plus the “Hours”
- not specified or value equals to zero, the “Hours” will be automatically calculated by the time elapsed time from “StartDate” to “EndDate”

Note: The fields *Exchange Rate*, *End Date*, *Currency*, *Total Base* and *Total reporting* are automatically updated, and, as such, the values filled in are ignored.

○ **PUT /api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors/{sequenceKey}**

This method allows you to update a Report. To that end, the http message should have the “body” filled in with the full structure of the resource *WorkOrderReportLaborResource* (including the unchanged ones).

Note: The fields *Exchange Rate*, *End Date*, *Currency*, *Total Base* and *Total reporting* are automatically updated, and, as such, the values filled in are ignored.

- **PUT**

/api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors/ChangeWorkOrderReportLaborExternalInfo

This method allows you to update a Report with the data ExternalId, ExternalVersion and IsExternallyManaged from several units of labor. To that end, the http message should have the “body” filled in with the full structure of the resource *ExternalInfoResource*.

Note: The fields *Exchange Rate*, *End Date*, *Currency*, *Total Base* and *Total reporting* are automatically updated, and, as such, the values filled in are ignored.

Example:

Changing the data of the Work Order Labor Reports with the SequenceKey = “WOL1” and “WOL 2”.

```
PUT
/api/mytenant/myorganization/maintenancemanagementcore/
workorderreportlabors/ChangeWorkOrderReportLaborExternalInfo

{
  "ExternalInfoLines" : [
    {
      "EntityKey" : "WOL1",
      "ExternalKey" : "externalKey1",
      "ExternalVersion" : "v1",
      "IsExternallyManaged" : "true"
    },
    {
      "EntityKey" : "WOL2",
      "ExternalKey" : "externalKey2",
      "ExternalVersion" : "v1",
      "IsExternallyManaged" : "true"
    }
  ]
}
```

- **DELETE**

/api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors/{sequenceKey}

This method allows you to delete a specific Report through its natural key.

Example:

Deleting the Report with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportlabors/Sample001
```

Reporting Expenses in the Work Order

GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportexpenses	Gets all WorkOrderReportExpenses.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportexpenses?sequenceKey={sequenceKey}	Gets a Report Expense by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportexpenses/{id}	Gets a Report Expense by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportexpenses/search/version/{version}	Gets all WorkOrderReportExpenses with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportexpenses/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportexpenses?page={page}&pageSize={pageSize}	Gets all the WorkOrderReportExpenses using a paged return.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportexpenses/GetDeletedWorkOrderReportExpenses?version={version}	Gets the deleted work order report expenses with a version higher than the specified.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportexpenses	Creates a Report Expense.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportexpenses?sequenceKey={sequenceKey}	Updates a Report Expense.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportexpenses/{id}	Updates a Report Expense by it's unique identifier.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportexpenses/ChangeWorkOrderReportExpenseExternalInfo	Changes the work order report expense external information.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportexpenses?sequenceKey={sequenceKey}	Deletes a specific Report Expense.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportexpenses/{id}	Deletes a specific Report Expense by it's unique identifier.

Figure 24 - Example of the methods exposed by the Work Order Expense Report service

This service makes it possible to obtain and manipulate Work Order Expense Reports in Valuekeep. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorderreportexpenses**

This method allows you to get the full list of existing Reports.

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/
workorderreportexpenses
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorderreportexpenses/sequenceKey={sequenceKey}**

This method allows you to obtain the details of a specific Report through its natural key.

Example:

Obtaining the details of the Report with the natural key "Sample001"

```
GET /api/mytenant/myorganization/maintenancemanagementcore/
workorderreportexpenses/sequenceKey=Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorderreportexpenses/search/version?version=AAAAAAAAAAAA=**

This method allows you to obtain the list of Reports from a version onwards. The minimum value of the version is: "AAAAAAAAAAAA=".

Example:

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportexpenses/search/version?version=AAAAAAAAAAAA=
```

Notes: This method also returns the cost center key of the work order and the report notes in the following format:

```
[Work Order] - [Asset/Location/Asset Route/Location Route]
[Work Order Description]
[Remarks on each line of the report]
```

- **GET**
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportexpenses?page={page}&pageSize={pagesize}

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Reports:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/
workorderreportexpenses?page=1&pageSize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorderreportexpenses?field={field}&value={value}**

This method allows you to search for a specific field of the Report.

Example:

Obtaining all Reports created by the User "Carlos".

```
GET /api/mytenant/myorganization/maintenancemanagementcore/
workorderreportexpenses?field=CreatedBy&value=Carlos
```

- **GET**
/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportexpenses/GetDeletedWorkOrderReportExpenses?version=AAAAAAAAAAAA=

This method allows you to obtain the list of deleted Reports from a version onwards. The minimum value of the version is: "AAAAAAAAAAAA=".

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/
Workorderreportexpenses/GetDeletedWorkOrderReportExpenses?version=AAAAAAAAAAAA=
```

Notes: This method also returns the cost center key of the work order and the report notes in the following format:

```
[Work Order] - [Asset/Location/Asset Route/Location Route]
[Work Order Description]
[Remarks on each line of the report]
```

- **POST /api/mytenant/myorganization/maintenancemanagementcore/workorderreportexpenses**

This method allows you to create a new Expenses Report. To that end, the http message should have the “body” filled in with the structure of the resource *WorkOrderReportExpensesResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/
workorderreportexpenses
Body:
{
  "SequenceKey": "Sample001",
  "Description": "Sample001",
}
```

- **PUT /api/mytenant/myorganization/maintenancemanagementcore/workorderreportexpenses/sequenceKey={sequenceKey}**

This method allows you to update a Report. To that end, the http message should have the “body” filled in with the full structure of the resource *WorkOrderReportExpensesResource* (including the unchanged ones).

- **PUT /api/mytenant/myorganization/maintenancemanagementcore/workorderreportexpenses/ChangeWorkOrderReportExpenseExternalInfo**

This method allows you to update a Report with the data ExternalId, ExternalVersion and IsExternallyManaged from several Expenses. To that end, the http message should have the “body” filled in with the full structure of the resource *ExternalInfoResource*.

Example:

Changing the data of the Work Order Expenses with the SequenceKey = “WOE1” and “WOE2”

```
PUT
/api/mytenant/myorganization/maintenancemanagementcore/
workorderreportexpenses/ChangeWorkOrderReportExpenseExternalInfo

{
  "ExternalInfoLines" : [
    {
      "EntityKey" : "WOE1",
      "ExternalKey" : "externalKey1",
      "ExternalVersion" : "v1",
      "IsExternallyManaged" : "true"
    },
    {
      "EntityKey" : "WOE2",
      "ExternalKey" : "externalKey2",
    }
  ]
}
```



```
"ExternalVersion" : "v1",
  "IsExternallyManaged" : "true"
}
]
}
```

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/workorderreportexpenses/sequenceKey={sequenceKey}**

This method allows you to delete a specific Report through its natural key.

Example:

Deleting the Report with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportexpenses/sequenceKey=Sample001
```

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/workorderreportexpenses/{id}**

This method allows you to delete a specific Report through its unique label.

Example:

Deleting the Report with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportexpenses/Sample001
```

Note: The following fields are recalculated in the POST and PUT methods, the values assigned in the body of the message are ignored:

- **WorkOrderReportExpense:** fields ExchangeRate and ExchangeRateDate;
- **WorkOrderTaskReportExpense:** field MaxValue.

Reporting Consumptions in the Work Order

GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions	Gets all WorkOrderReportConsumptions.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions?sequenceKey={sequenceKey}	Gets a Report Consumption by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions/{id}	Gets a Report Consumption by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions/search/version/{version}	Gets all WorkOrderReportConsumptions with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions/search/version?version={version}	Gets all WorkOrderReportConsumptions with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions/search?field={field}&value={value}	Gets all WorkOrderReportConsumptions containing the search criteria on the specified field.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions?page={page}&pageSize={pageSize}	Gets all the WorkOrderReportConsumptions using a paged return.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions/GetDeletedWorkOrderReportConsumptionVersion={version}	Gets the deleted work order report consumptions with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions/GetPendingWorkOrderReportConsumptionVersion={version}	Gets the pending work order report consumptions.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions/GetWorkOrderReportConsumptionByFilterworkOrderKey={workOrderKey}&createdBy={createdBy}&version={version}	Gets the work order report consumption by filter.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions	Creates a Report Consumption.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions?sequenceKey={sequenceKey}	Updates a Report Consumption.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions/{id}	Updates a Report Consumption by it's unique identifier.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions/ChangeWorkOrderReportConsumptionExternalInformation	Changes the work order report consumption external information.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions/ChangeWorkOrderReportConsumptionsPriority	No Documentation Found.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions?sequenceKey={sequenceKey}	Deletes a specific Report Consumption.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions/{id}	Deletes a specific Report Consumption by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions/DeleteWorkOrderReportConsumptionById{id}&referenceDate={referenceDate}	No Documentation Found.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions/DeleteWorkOrderReportConsumptionWithSequenceKey={sequenceKey}&referenceDate={referenceDate}	No Documentation Found.

Figure 25 - Example of the methods exposed by the Work Order Consumptions Report service

This service makes it possible to obtain and manipulate Work Order Consumptions Reports in Valuekeep. You can find an analysis of the methods made available by the service below:

- **GET**

/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions

This method allows you to get the full list of existing Reports.

Example:

```
GET/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions
```

- **GET**

/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/sequenceKey={sequenceKey}

This method allows you to obtain the details of a specific Report through its natural key.

Example:

Obtaining the details of the Report with the natural key "Sample001"

```
GET/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/sequenceKey=Sample001
```

- **GET**

/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/search/version?version=AAAAAAAAAAAA=

This method allows you to obtain the list of Reports from a version onwards. The minimum value of the version is: "AAAAAAAAAAAA=".

Example:

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/search/version?version=AAAAAAAAAAAA=
```

Notes: This method also returns the cost center key of the work order and the report notes in the following format:

```
[Work Order] - [Asset/Location/Asset Route/Location Route]
[Work Order Description]
[Remarks on each line of the report]
```

This method returns the field HasExternalItemIssue, which indicates whether the report has external stock issue or not.

- **GET**

/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions?page={page}&pageSize={pageSize}

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Reports:

```
GET/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions?page=1&pageSize=20
```

- **GET**

/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions?field={field}&value={value}

This method allows you to search for a specific field of the Report.

Example:

Obtaining all Reports created by the User “Carlos”.

```
GET/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions?field=CreatedBy&value=Carlos
```

- **GET**

/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions/GetDeletedWorkOrderReportConsumptions?version=AAAAAAAAAAAA=

This method allows you to obtain the list of Reports deleted from a version onwards, the minimum value of the version is: “AAAAAAAAAAAA=”.

Example:

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/GetDeletedWorkOrderReportConsumptions?version=AAAAAAAAAAAA=
```

Notes: This method also returns the cost center key of the work order and the report notes in the following format:

```
[Work Order] - [Asset/Location/Asset Route/Location Route]
[Work Order Description]
[Remarks on each line of the report]
```

This method returns the field HasExternalItemIssue, which indicates whether the report has external stock issue or not.

- **GET**

/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions/GetPendingWorkOrderReportConsumptions?version=AAAAAAAAAAAA=

This method allows you to get the list of integrated Reports, WOs which are not in “Executed”, “Closed” or “Canceled” state, from a version onwards. The minimum value of the version is: “AAAAAAAAAAAA=”.

Response is given with the structure of the resource *PendingConsumptionReportResource*.

Example:

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/GetPendingWorkOrderReportConsumptions?version=AAAAAAAAAAAA=

[
  {
    "ConsumptionId": "0acea48c-1513-485f-b836-1afb13184dfd",
    "ConsumptionNaturalKey": "I00001",
```

```

"ExternalId": "CONS.2016.1",
"HasExternalItemIssue": "true",
"PendingConsumptionLines": [
  {
    "ItemNaturalKey": "AIR-FILTER",
    "WarehouseNaturalKey": "A1",
    "Price": 15,
    "PriceBase": 15,
    "PriceReporting": 15,
    "UpdatedPrice": 0,
    "UpdatedPriceBase": 0,
    "UpdatedPriceReporting": 0,
    "Date": "2015-07-28T10:20:17.7622601+01:00",
    "Index": 0,
    "Id": "c634471f-d44b-e511-829b-f816541429a3",
    "IsActive": true,
    "IsDeleted": false,
    "Version": null
  },
  {
    "ItemNaturalKey": "AIR-FILTER",
    "WarehouseNaturalKey": "A1",
    "Price": 16,
    "PriceBase": 16,
    "PriceReporting": 16,
    "UpdatedPrice": 0,
    "UpdatedPriceBase": 0,
    "UpdatedPriceReporting": 0,
    "Date": "2015-08-25T15:47:32.8137027+01:00",
    "Index": 1,
    "Id": "c934471f-d44b-e511-829b-f816541429a3",
    "IsActive": true,
    "IsDeleted": false,
    "Version": null
  }
],
"Id": "ca34471f-d44b-e511-829b-f816541429a3",
"IsActive": true,
"IsDeleted": false,
"Version": null
}
]

```

Notes: This method returns the field HasExternalItemIssue, which indicates whether the report has external stock issue or not.

○ GET

/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportconsumptions/GetWorkOrderReportConsumptionByFilter?workOrderKey={workOrderKey}&createdBy={createdBy}&version={version}

This method allows you to obtain the list of Reports from a specific work order or employee and from a version onwards. The minimum value of the version is: "AAAAAAAAAAAA=";

Examples:

GET

```
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/GetWorkOrderReportConsumptionByFilter?WorkOrderKey=&CreatedBy=&Version=AAAAAA  
AAAAA=
```

```
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/GetWorkOrderReportConsumptionByFilter?WorkOrderKey=&CreatedBy=user1&Version=A  
AAAAA=
```

```
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/GetWorkOrderReportConsumptionByFilter?WorkOrderKey=&CreatedBy=user1&Version=
```

```
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/GetWorkOrderReportConsumptionByFilter?WorkOrderKey=OT0010&CreatedBy=&Version=
```

Note: All fields are optional. They can be left blank.

○ POST

/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions

This method allows you to create a new Consumption Report. To that end, the http message should have the “body” filled in with the structure of the resource *WorkOrderReportConsumptionResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/  
workorderreportconsumptions
```

Body:

```
{  
  "Description": "Consumo",  
  "ReportDate": "2015-11-17T11:40:49.1534494+00:00",  
  "IsPrinted": null,  
  "StockIssueEmitted": true,  
  "ExchangeRate": 1,  
  "ExchangeRateDate": "2015-11-17T00:00:00",  
  "ReturnsProcessed": false,  
  "WorkOrder": "OT0006",  
  "Location": null,  
  "Asset": "PORTATIL-0028",  
  "WorkOrderTaskReportConsumptions": [  
    {  
      "Date": "2015-11-17T11:41:02.6735047+00:00",  
      "Quantity": 1,  
      "Remarks": "",  
      "MaintenancePlan": "PMGEN-000002.1",  
      "Warehouse": "A1",  
      "Unit": "UN",  
      "Task": "T-000003",  
      "Item": "FILTO-AR",  
    }  
  ],  
  "Company" : "Default"  
}
```

- **PUT**

/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumption/sequenceKey={sequenceKey}

This method allows you to update a Report. To that end, the http message should have the “body” filled in with the full structure of the resource *WorkOrderReportConsumptionResource* (including the unchanged ones).

- **PUT**

/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/ChangeWorkOrderReportConsumptionExternalInfo

This method allows you to update a Report with the data ExternalId, ExternalVersion and IsExternallyManaged from several Consumptions. To that end, the http message should have the “body” filled in with the full structure of the resource *ExternalInfoResource*.

Example:

Changing the data of the Work Order Consumptions with the SequenceKey = “WOC1” and “WOC2”

```
PUT
/api/mytenant/myorganization/maintenancemanagementcore/
workorderreportconsumptions/ChangeWorkOrderReportConsumptionExternalInfo
{
  "ExternalInfoLines" : [
    {
      "EntityKey" : "WOC1",
      "ExternalKey" : "externalKey1",
      "ExternalVersion" : "v1",
      "IsExternallyManaged" : "true"
    },
    {
      "EntityKey" : "WOC2",
      "ExternalKey" : "externalKey2",
      "ExternalVersion" : "v1",
      "IsExternallyManaged" : "true"
    }
  ]
}
```

- **PUT /api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/ChangeWorkOrderReportConsumptionsPrice**

This method allows you to update ACP (average cost price) of a Consumption Report. To that end, the http message should have the “body” filled in with a list with the full structure of the resource *PendingConsumptionReportResource*.

Example:

Changing the Consumption data with the natural key = “100005” and “100006”:

```
PUT
/api/mytenant/myorganization/maintenancemanagementcore/
workorderreportconsumptions/ChangeWorkOrderReportConsumptionsPrice
[
  {
```

```

    "ConsumptionNaturalKey": "I00005",
    "PendingConsumptionLines": [
      {
        "Id": "f148a632-c94c-e511-93fe-000c29de8b5e",
        "Date": "2015-08-26T15:41:19.1280658+01:00",
        "ItemNaturalKey": "AIR-CONDITIONING",
        "WarehouseNaturalKey": "A1",
        "UpdatedPrice": 9,
      }
    ],
  },
  {
    "ConsumptionNaturalKey": "I00006",
    "PendingConsumptionLines": [
      {
        "Id": "0049a632-c94c-e511-93fe-000c29de8b5e",
        "Date": "2015-08-26T17:17:24.0444469+01:00",
        "ItemNaturalKey": "AIR-FILTER",
        "WarehouseNaturalKey": "A1",
        "UpdatedPrice": 11.1,
      }
    ]
  }
]

```

- **PUT /api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/ChangeWorkOrderReportConsumptionsPriceWithoutStockIssue**

This method allows you to update the price of a Consumption Report. To that end, the http message should have the “body” filled in with a list with the full structure of the resource *PendingConsumptionReportResource*. This method does not perform any stock issue operations.

Example:

Changing the Consumption data with the natural key = “I00005” and “I00006”:

```

PUT
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/ChangeWorkOrderReportConsumptionsPriceWithoutStockIssue

[
  {
    "ConsumptionNaturalKey": "I00005",
    "PendingConsumptionLines": [
      {
        "Id": "f148a632-c94c-e511-93fe-000c29de8b5e",
        "ItemNaturalKey": "AIR-CONDITIONING",
        "WarehouseNaturalKey": "A1",
        "UpdatedPrice": 5.5,
      }
    ],
  },
  {
    "ConsumptionNaturalKey": "I00006",

```



```

    "PendingConsumptionLines": [
      {
        "Id": "0acea48c-1513-485f-b836-1afb13184dfd ",
        "ItemNaturalKey": "AIR-FILTER",
        "WarehouseNaturalKey": "A1",
        "UpdatedPrice": 19.9,
      },
      {
        "Id": "0049a632-c94c-e511-93fe-000c29de8b5e",
        "ItemNaturalKey": "AIR-FILTER",
        "WarehouseNaturalKey": "A2",
        "UpdatedPrice": 25.5,
      }
    ]
  }
]

```

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/sequenceKey={sequenceKey}**

This method allows you to delete a specific Report through its natural key.

Example:

Deleting the Report with the natural key *Sample001*:

```

DELETE
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/sequenceKey=Sample001

```

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/{id}**

This method allows you to delete a specific Report through its unique label.

Example:

Deleting the Report with the natural key *Sample001*:

```

DELETE
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/Sample001

```

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/DeleteWorkOrderReportConsumptionWithReferenceDate?sequenceKey={sequenceKey}&referenceDate={referenceDate}**

This method allows you to delete a specific Report through its natural key and validates the removal on the reference date.

Example:

Deleting the Report with the natural key *Sample001*:

```

DELETE
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/deleteworkorderreportconsumptionwithreferencedate?sequenceKey=I00001&referenceDate=2015-01-01T01:00:00.00

```

- **DELETE** /api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/DeleteWorkOrderReportConsumptionByIdWithReferenceDate?id={id}&referenceDate={referenceDate}

This method allows you to delete a specific Report through its unique label and validates the removal on the reference date.

Example:

Deleting the Report with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportconsumptions/deleteworkorderreportconsumptionbyidwithreferencedate?id=BF7DEA92-71D9-478A-BCE6-71DA8B4D6AD6&referenceDate=2015-01-01T01:00:00.00
```

Note: The following fields are recalculated in the POST and PUT methods, the values assigned in the body of the message are ignored:

- **WorkOrderReportConsumption:** fields Currency, ExchangeRate and ExchangeRateDate (the report is always made in the base currency). Field StockIssueEmitted;
- **WorkOrderTaskReportConsumption:** fields Price and TotalCost.

Reporting Contracts in the Work Order

GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportcontracts	Gets all WorkOrderReportContracts.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportcontracts?sequenceKey={sequenceKey}	Gets a Work Order Report Contract by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportcontracts/{id}	Gets a Work Order Report Contract by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportcontracts/search/version/{version}	Gets all WorkOrderReportContracts with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportcontracts/search/version?version={version}	Gets all WorkOrderReportContracts with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportcontracts/search?field={field}&value={value}	Gets all WorkOrderReportContracts containing the search criteria on the specified field.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportcontracts?page={page}&pageSize={pageSize}	Gets all the WorkOrderReportContracts using a paged return.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportcontracts/GetDeletedWorkOrderReportContracts?version={version}	Gets the deleted work order report contract consumption with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportcontracts/GetPendingWorkOrderReportContracts?version={version}	Gets the pending work order report contracts.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportcontracts	Creates a Work Order Report Contract.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportcontracts?sequenceKey={sequenceKey}	Updates a Work Order Report Contract.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportcontracts/{id}	Updates a Work Order Report Contract by it's unique identifier.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportcontracts/ChangeWorkOrderReportContractExternalInfo	Changes the work order report contract consumption external information.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportcontracts?sequenceKey={sequenceKey}	Deletes a specific Work Order Report Contract.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportcontracts/{id}	Deletes a specific Work Order Report Contract by it's unique identifier.

Figure 26 - Example of the methods exposed by the Work Order Contracts Report service

This service makes it possible to obtain and manipulate Work Order Contracts Reports in Valuekeep. You can find an analysis of the methods made available by the service below:

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorderreportcontracts**

This method allows you to get the full list of existing Reports;

Example:

```
GET/api/mytenant/myorganization/maintenancemanagementcore/workorderreportcontracts
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorderreportcontracts/sequenceKey={sequenceKey}**

This method allows you to obtain the details of a specific Report through the natural key;

Example:

Obtaining the details of the Report with the natural key "Sample001"

```
GET/api/mytenant/myorganization/maintenancemanagementcore/workorderreportcontracts/sequenceKey=Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorderreportcontracts/search/version?version=AAAAAAAAAAAA=**

This method allows you to obtain the list of Reports from a version onwards. The minimum value of the version is: "AAAAAAAAAAAA=";

Example:

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportcontracts/search/version?version=AAAAAAAAAAAA=
```

Notes: This method also returns the cost center key of the work order and the report notes in the following format:

```
[Work Order] - [Asset/Location/Asset Route/Location Route]
[Work Order Description]
[Remarks on each line of the report]
```

This method returns the field HasExternalItemIssue, which indicates whether the report has external stock issue or not.

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorderreportcontracts?page={page}&pageSize={pageSize}**

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Reports:

```
GET/api/mytenant/myorganization/maintenancemanagementcore/workorderreportcontracts?page=1&pageSize=20
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorderreportcontracts?field={field}&value={value}**

This method allows you to search for a specific field of the Report.

Example:

Obtaining all Reports created by the User "Carlos".

```
GET/api/mytenant/myorganization/maintenancemanagementcore/workorderreportcontracts?field=CreatedBy&value=Carlos
```

- **GET /api/{tenant}/{organization}/maintenancemanagementcore/workorderreportcontracts/GetDeletedWorkOrderReportContracts?version=AAAAAAAAAAAA=**

This method allows you to obtain the list of Reports deleted from a version onwards, the minimum value of the version is: "AAAAAAAAAAAA=";

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/
workorderreportconsumptions/GetDeletedWorkOrderReportContracts?version=AAAAAA
AAAAA=
```

Notes: This method also returns the cost center key of the work order and the report notes in the following format:

```
[Work Order] - [Asset/Location/Asset Route/Location Route]
[Work Order Description]
[Remarks on each line of the report]
```

This method returns the field `HasExternalItemIssue`, which indicates whether the report has external stock issue or not.

○ **GET**

/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportcontracts/GetPendingWorkOrderReportContracts?version=AAAAAAAAAAAA=

This method allows you to get the list of integrated Reports, WOs which are not in “Executed”, “Closed” or “Canceled” state, from a version onwards. The minimum value of the version is: “AAAAAAAAAAAA”. Response is given with the structure of the resource *PendingConsumptionReportResource*.

Example:

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportcontrac
ts/GetPendingWorkOrderReportContracts?version=AAAAAAAAAAAA=

[
  {
    "ConsumptionId": "0acea48c-1513-485f-b836-1afb13184dfd",
    "ConsumptionNaturalKey": "I00001",
    "ExternalId": "CONT.2016.1",
    "HasExternalItemIssue": "false",
    "PendingConsumptionLines": [
      {
        "ItemNaturalKey": "AIR-FILTER",
        "WarehouseNaturalKey": "A1",
        "Price": 15,
        "PriceBase": 15,
        "PriceReporting": 15,
        "UpdatedPrice": 0,
        "UpdatedPriceBase": 0,
        "UpdatedPriceReporting": 0,
        "Date": "2015-07-28T10:20:17.7622601+01:00",
        "Index": 0,
        "Id": "c634471f-d44b-e511-829b-f816541429a3",
        "IsActive": true,
        "IsDeleted": false,
        "Version": null
      },
      {

```

```

        "ItemNaturalKey": "AIR-FILTER",
        "WarehouseNaturalKey": "A1",
        "Price": 16,
        "PriceBase": 16,
        "PriceReporting": 16,
        "UpdatedPrice": 0,
        "UpdatedPriceBase": 0,
        "UpdatedPriceReporting": 0,
        "Date": "2015-08-25T15:47:32.8137027+01:00",
        "Index": 1,
        "Id": "c934471f-d44b-e511-829b-f816541429a3",
        "IsActive": true,
        "IsDeleted": false,
        "Version": null
    }
],
    "Id": "ca34471f-d44b-e511-829b-f816541429a3",
    "IsActive": true,
    "IsDeleted": false,
    "Version": null
}
]

```

Notes: This method returns the field `HasExternalItemIssue`, which indicates whether the report has external stock issue or not.

- **POST /api/mytenant/myorganization/maintenancemanagementcore/workorderreportcontracts**

This method allows you to create a new Contract Consumption Report. To that end, the http message should have the “body” filled in with the structure of the resource *WorkOrderReportContractResource*.

Example:

```

POST /api/mytenant/myorganization/maintenancemanagementcore/
workorderreportcontracts

```

Body:

```

{
    "Description": "Consumo ao abrigo do contrato",
    "ReportDate": "2015-11-17T11:40:49.1534494+00:00",
    "IsPrinted": null,
    "StockIssueEmitted": true,
    "ExchangeRate": 1,
    "ExchangeRateDate": "2015-11-17T00:00:00",
    "ReturnsProcessed": false,
    "WorkOrder": "OT0006",
    "Location": null,
    "Asset": "PORTATIL-0028",
    "WorkOrderTaskReportContracts": [
        {
            "Date": "2015-11-17T11:41:02.6735047+00:00",
            "Quantity": 1,
            "Remarks": "",
            "MaintenancePlan": "PMGEN-000002.1",
            "Warehouse": "A1",
            "Unit": "UN",

```

```

        "Task": "T-000003",
        "Item": "FILTO-AR",
    }
],
    "Company" : "Default"
}

```

- **PUT**

/api/mytenant/myorganization/maintenancemanagementcore/workorderreportcontracts/sequenceKey={sequenceKey}

This method allows you to update a Report. To that end, the http message should have the “body” filled in with the full structure of the resource *WorkOrderReportContractResource* (including the unchanged ones).

- **PUT**

/api/mytenant/myorganization/maintenancemanagementcore/workorderreportcontracts/ChangeWorkOrderReportContractExternalInfo

This method allows you to update a Report with the data ExternalId, ExternalVersion and IsExternallyManaged from several Contract Consumptions. To that end, the http message should have the “body” filled in with the full structure of the resource *ExternalInfoResource*.

Example:

Changing the data of the Work Order Contracts with the SequenceKey = “WOC1” and “WOC2”

```

PUT
/api/mytenant/myorganization/maintenancemanagementcore/
workorderreportconsumptions/ChangeWorkOrderReportContractExternalInfo
{
  "ExternalInfoLines" : [
    {
      "EntityKey" : "WOC1",
      "ExternalKey" : "externalKey1",
      "ExternalVersion" : "v1",
      "IsExternallyManaged" : "true"
    },
    {
      "EntityKey" : "WOC2",
      "ExternalKey" : "externalKey2",
      "ExternalVersion" : "v1",
      "IsExternallyManaged" : "true"
    }
  ]
}

```

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/workorderreportcontracts/sequenceKey={sequenceKey}**

This method allows you to delete a specific Report through its natural key.

Example:

Deleting the Report with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportcontracts/
sequenceKey=Sample001
```

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/workorderreportcontracts/{id}**

This method allows you to delete a specific Report through its unique label.

Example:

Deleting the Report with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportcontracts/
Sample001
```

Note: The following fields are recalculated in the POST and PUT methods, the values assigned in the body of the message are ignored:

- **WorkOrderReportContract:** fields Currency, ExchangeRate and ExchangeRateDate (the report is always made in the base currency). Field StockIssueEmitted;
- **WorkOrderTaskReportContract:** fields Price and TotalCost.

Reporting Other Costs in the Work Order

GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportothercosts	Gets all WorkOrderReportOtherCosts.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportothercosts?sequenceKey={sequenceKey}	Gets a Report Other Cost by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportothercosts/{id}	Gets a Report Other Cost by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportothercosts/search/version/{version}	Gets all WorkOrderReportOtherCosts with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportothercosts/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportothercosts?page={page}&pageSize={pageSize}	Gets all the WorkOrderReportOtherCosts using a paged return.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportothercosts/GetDeletedWorkOrderReportOtherCosts?version={version}	Gets the deleted work order report other costs with a version higher than the specified.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportothercosts	Creates a Report Other Cost.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportothercosts?sequenceKey={sequenceKey}	Updates a Report Other Cost.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportothercosts/{id}	Updates a Report Other Cost by it's unique identifier.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportothercosts/ChangeWorkOrderReportOtherCostExternalInfo	Changes the work order report other cost external information.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportothercosts?sequenceKey={sequenceKey}	Deletes a specific Report Other Cost.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/workorderreportothercosts/{id}	Deletes a specific Report Other Cost by it's unique identifier.

Figure 27 - Example of the methods exposed by the Other Costs Report of the Work Order service

This service makes it possible to obtain and manipulate Work Order Other Costs Reports in Valuekeep. You can find an analysis of the methods made available by the service below:

- **GET /api/{tenant}/{organization}/maintenancemanagementcore/workorderreportothercosts**

This method allows you to get the full list of existing Reports.

Example:

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportothercosts
```

- **GET /api/{tenant}/{organization}/maintenancemanagementcore/workorderreportothercosts/search/version/{version}**

This method allows you to obtain the list of Reports which exist from a version onwards, the minimum value of the version is: "AAAAAAAAAAAA=".

Example:

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderreportothercosts
/search/version/AAAAAAAAAAAA=
```

- **GET**

/api/{tenant}/{organization}/maintenancemanagementcore/workorderreporttothercosts/{id}

Obtaining the details of the Report with the Id. "9X136A57-EAC1-4057-8054-0FAA6X2FF15E"

Example:

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderreporttothercosts/
9X136A57-EAC1-4057-8054-0FAA6X2FF15E
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorderreporttothercosts/sequenceKey={sequenceKey}**

This method allows you to obtain the details of a specific Report through its natural key.

Example:

Obtaining the details of the Report with the natural key "Sample001"

```
GET /api/mytenant/myorganization/maintenancemanagementcore/
workorderreporttothercosts/sequenceKey=Sample001
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorderreporttothercosts/search/version?version=AAAAAAAAAAAA=**

This method allows you to obtain the list of Reports from a version onwards. The minimum value of the version is: "AAAAAAAAAAAA=".

Example:

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderreporttotherco
sts/search/version?version=AAAAAAAAAAAA=
```

Notes: This method also returns the cost center key of the work order and the report notes in the following format:

```
[Work Order] - [Asset/Location/Asset Route/Location Route]
[Work Order Description]
[Remarks on each line of the report]
```

- **GET /api/mytenant/myorganization/maintenancemanagementcore/workorderreporttothercosts?page={page}&pageSize={pageSize}**

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 Reports:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/
workorderreporttothercosts?page=1&pageSize=20
```

- **GET** /api/mytenant/myorganization/maintenancemanagementcore/workorderreporttothercosts?field={field}&value={value}

This method allows you to search for a specific field of the Report.

Example:

Obtaining all Reports created by the User "Carlos".

```
GET /api/mytenant/myorganization/maintenancemanagementcore/
workorderreporttothercosts?field=CreatedBy&value=Carlos
```

- **GET** /api/{tenant}/{organization}/maintenancemanagementcore/workorderreporttothercosts/GetDeletedWorkOrderReportOtherCosts?version= AAAAAAAAAA=

This method allows you to obtain the list of Reports deleted from a version onwards, the minimum value of the version is: "AAAAAAAAA=".

Example:

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderreporttothercosts
/GetDeletedWorkOrderReportOtherCosts?version=AAAAAAAAA=
```

Notes: This method also returns the cost center key of the work order and the report notes in the following format:

```
[Work Order] - [Asset/Location/Asset Route/Location Route]
[Work Order Description]
[Remarks on each line of the report]
```

- **POST** /api/mytenant/myorganization/maintenancemanagementcore/workorderreporttothercosts

This method allows you to create a new Other Costs Report. To that end, the http message should have the "body" filled in with the structure of the resource *WorkOrderReportOtherCostsResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/
workorderreporttothercosts
Body:
{
  "SequenceKey": "Sample001",
  "Description": "Sample001",
}
```

- **PUT** /api/mytenant/myorganization/maintenancemanagementcore/workorderreporttothercosts/sequenceKey={sequenceKey}

This method allows you to update a Report. To that end, the http message should have the "body" filled in with the full structure of the resource *WorkOrderReportOtherCostsResource* (including the unchanged ones).

- **PUT**

/api/mytenant/myorganization/maintenancemanagementcore/workorderreporttothercosts/ChangeWorkOrderReportOtherCostExternalInfo

This method allows you to update a Report with the data ExternalId, ExternalVersion and IsExternallyManaged from several Other Costs. To that end, the http message should have the "body" filled in with the full structure of the resource *ExternalInfoResource*.

Example:

Changing the data of the Work Order Other Costs with the SequenceKey = "WOOC1" and "WOOC2"

```
PUT
/api/mytenant/myorganization/maintenancemanagementcore/
workorderreporttothercosts/ChangeWorkOrderReportOtherCostExternalInfo
{
  "ExternalInfoLines" : [
    {
      "EntityKey" : "WOOC1",
      "ExternalKey" : "externalKey1",
      "ExternalVersion" : "v1",
      "IsExternallyManaged" : "true"
    },
    {
      "EntityKey" : "WOOC2",
      "ExternalKey" : "externalKey2",
      "ExternalVersion" : "v1",
      "IsExternallyManaged" : "true"
    }
  ]
}
```

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/workorderreporttothercosts /sequenceKey={sequenceKey}**

This method allows you to delete a specific Report through its natural key.

Example:

Deleting the Report with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/workorderreporttothercosts
/sequenceKey=Sample001
```

- **DELETE /api/mytenant/myorganization/maintenancemanagementcore/workorderreporttothercosts/{id}**

This method allows you to delete a specific Report through its unique label.

Example:

Deleting the Report with the natural key *Sample001*:

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/workorderreporttothercosts
/Sample001
```

Note: The following fields are recalculated in the POST and PUT methods, the values assigned in the body of the message are ignored:

- **WorkOrderReportOtherCost:** fields ExchangeRate and ExchangeRateDate.
- **WorkOrderTaskReportOtherCost:** fields MaxValue and Total.

Inventory Management

Items

Items : Handles Item operations.

Show/Hide

List Operations

Expand Operations

Raw

GET	/api/materialscore/items	Gets all Items.
GET	/api/materialscore/items/{itemKey}	Gets a Item by the specified key.
GET	/api/materialscore/items?page={page}&pageSize={pageSize}	Gets all the Items using a paged return.
GET	/api/materialscore/items?field={field}&value={value}	Gets all Items containing the search criteria on the specified field.
POST	/api/materialscore/items	Creates a Item.
PUT	/api/materialscore/items/{itemKey}	Updates a Item.
DELETE	/api/materialscore/items/{itemKey}	Deletes a specific Item.

Figure 28 - Example of the methods exposed by the Items service

The items service makes it possible to obtain and manipulate Items. You can find an analysis of the methods made available by the service below:

○ **GET /api/materialscore/items**

This method allows you to get the full list of existing items.

Example:

```
GET /api/materialscore/items
```

○ **GET /api/materialscore/items/{itemKey}**

This method allows you to obtain the details of a specific item through its natural key.

Example:

Obtaining the details of the item with the natural key "A0001"

```
GET /api/materialscore/items/a0001
```

○ **GET /api/materialscore/items?page={page}&pagesize={pagesize}**

This method allows you to obtain a paged list.

The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 items:

```
GET /api/materialscore/items?page=1&pagesize=20
```

- **GET /api/materialscore/items?field={field}&value={value}**

This method allows you to search for a specific field of the item.

Example:

Obtaining all the items of the “aspectto” brand.

```
GET /api/materialscore/items?field=brand&value=aspectto
```

- **POST /api/materialscore/items**

This method allows you to create a new item. To that end, the http message should have the “body” filled in with the structure of the resource *ItemResource*.

Example:

```
POST /api/materialscore/items
Body:
{
  "ItemKey": "Z0001",
  "Description": "KING Chair",
  "KeyWord": "KING",
  "Barcode": "0000000000001",
  "Weight": 20,
  "Volume": 1,
  "ItemType": 1,
  "Assortment": "CHAIRS",
  "Brand": "ASPECTTO",
  "Rank": 1,
  "BaseUnit": "UN",
  "WeightUnit": "UN",
  "VolumeUnit": "UN",
  "OriginCountry": "PT",
  "ItemTaxSchema": "IVA-TN",
}
```

- **PUT /api/materialscore/items/{itemKey}**

This method allows you to update an item. To that end, the http message should have the “body” filled in with the full structure of the resource *ItemResource* (including the unchanged ones).

- **DELETE /api/materialscore/items/{itemKey}**

This method allows you to delete a specific item through its natural key.

Example:

Deleting the item with the natural key Z0001:

```
DELETE /api/materialscore/items/Z0001
```

Parties

Parties : Handles Party operations.

Show/Hide

List Operations

Expand Operations

Raw

GET	/api/patterns/parties	Gets all Parties.
GET	/api/patterns/parties/{partyKey}	Gets a Party by the specified key.
GET	/api/patterns/parties?page={page}&pageSize={pageSize}	Gets all the Parties using a paged return.
GET	/api/patterns/parties?field={field}&value={value}	Gets all Parties containing the search criteria on the specified field.
POST	/api/patterns/parties	Creates a Party.
PUT	/api/patterns/parties/{partyKey}	Updates a Party.
DELETE	/api/patterns/parties/{partyKey}	Deletes a specific Party.

Figure 29 - Example of the methods exposed by the Parties service

The parties service makes it possible to obtain and manipulate Parties. The parties can be customers, suppliers, employees, etc. You can find an analysis of the methods made available by the service below:

- **GET /api/patterns/parties**

This method allows you to get the full list of existing parties.

Example:

```
GET /api/patterns/parties
```

- **GET /api/patterns/parties/{partyKey}**

This method allows you to obtain the details of a specific party through its natural key.

Example:

Obtaining the details of the “sofrio” party:

```
GET /api/patterns/parties/sofrio
```

- **GET /api/patterns/parties?field={field}&value={value}**

This method allows you to search for a specific field of the party.

Example:

Obtaining the details of the party with the tax ID (companytaxid) 123456789

```
GET api/patterns/parties?field=companytaxid&value=123456789
```

- **GET /api/patterns/parties?page={page}&pageSize={pagesize}**

This method allows you to obtain a paged list. The *page* parameter indicates which is the desired page and the *pageSize* indicates the number of records per page. The *page* parameter is always greater than or equal to 1.

Example:

Obtaining the first page of 20 parties:

```
GET /api/patterns/parties?page=1&pagesize=20
```

- **POST /api/patterns/parties**

This method allows you to create a new party. To that end, the http message should have the “body” filled in with the structure of the resource *PartyResource*.

Example:

```
POST /api/patterns/parties
Body:
{
  "PaymentTerm": "30D",
  "PaymentMean": "CHQ",
  "DeliveryTerm": "V.VIAT",
  "DunningMethod": "STD",
  "PartyKey": "CustomerPartyA8",
  "SearchTerm": "CustomerPartyA8",
  "Name": "CustomerPartyA8, Lda",
  "CompanyTaxID": "123456789",
  "ElectronicMail": "geral@CustomerPartyA8.com",
  "Telephone": "211234567",
  "TeleFax": "211234568",
  "Mobile": "951234567",
  "WebsiteURL": "www.CustomerPartyA8.pt",
  "IsCustomer": true,
  "Currency": "EUR",
  "Country": "PT",
  "Addresses": [
    {
      "AddressKey": "CustomerPartyA8_HQ",
      "StreetName": "AV. of nowhere",
      "BuildingNumber": "888",
      "PostalZone": "1001",
      "CityName": "LISBON",
      "ContactName": "Joaquim",
      "ContactTitle": "Mr",
      "FormattedAddress": "Mr. Joaquim\r\nAV. of nowhere, 17\r\n9987 Lisboa\r\n",
      "IsHeadOffice": true,
      "IsPostalAddress": true,
      "IsExpeditionTarget": true,
      "IsLiftingSource": true,
      "IsMainAddress": true,
      "Party": "CustomerPartyA8",
      "Country": "PT",
    }
  ],
}
```


- **PUT /api/patterns/parties/{partyKey}**

This method allows you to update a party. To that end, the http message should have the “body” filled in with the full structure of the resource *PartyResource* (including the unchanged ones).

- **DELETE /api/patterns/parties/{partyKey}**

This method allows you to delete a specific party through its natural key.

Example:

Deleting the customer with the natural key *PartyA8*

```
DELETE /api/patterns/parties/PartyA8
```

Taxes

GET	/api/{tenant}/{organization}/taxescore/taxtypecodes	Gets all TaxTypeCodes.
GET	/api/{tenant}/{organization}/taxescore/taxtypecodes?taxCodeKey={taxCodeKey}	Gets a Tax Type Code by the specified key.
GET	/api/{tenant}/{organization}/taxescore/taxtypecodes/{id}	Gets a Tax Type Code by the resource unique identifier.
GET	/api/{tenant}/{organization}/taxescore/taxtypecodes/search/version/{version}	Gets all TaxTypeCodes with a version higher than the specified.
GET	/api/{tenant}/{organization}/taxescore/taxtypecodes/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/taxescore/taxtypecodes?page={page}&pageSize={pageSize}	Gets all the TaxTypeCodes using a paged return.
POST	/api/{tenant}/{organization}/taxescore/taxtypecodes	Creates a Tax Type Code.
POST	/api/{tenant}/{organization}/taxescore/taxtypecodes/CreateTaxTypeCode?taxCodeKey={taxCodeKey}&validFrom={validFrom}&validTo={validTo}&taxValue={taxValue}¤cyKey={currencyKey}&taxClassKey={taxClassKey}&taxSubClassKey={taxSubClassKey}&description={description}	Creates the tax type code.
PUT	/api/{tenant}/{organization}/taxescore/taxtypecodes?taxCodeKey={taxCodeKey}	Updates a Tax Type Code.
PUT	/api/{tenant}/{organization}/taxescore/taxtypecodes/{id}	Updates a Tax Type Code by it's unique identifier.
PUT	/api/{tenant}/{organization}/taxescore/taxtypecodes/UpdateTaxTypeCode?taxCodeKey={taxCodeKey}&validFrom={validFrom}&validTo={validTo}&taxValue={taxValue}¤cyKey={currencyKey}&taxClassKey={taxClassKey}&taxSubClassKey={taxSubClassKey}&description={description}	Updates the tax type code.
DELETE	/api/{tenant}/{organization}/taxescore/taxtypecodes?taxCodeKey={taxCodeKey}	Deletes a specific Tax Type Code.
DELETE	/api/{tenant}/{organization}/taxescore/taxtypecodes/{id}	Deletes a specific Tax Type Code by it's unique identifier.

Figure 30 - Example of the methods exposed by the Taxes service

The taxes service makes it possible to obtain and manipulate Taxes:

- **POST**

/api/{tenant}/{organization}/taxescore/taxtypecodes/CreateTaxTypeCode?taxCodeKey={taxCodeKey}&validFrom={validFrom}&validTo={validTo}&taxValue={taxValue}¤cyKey={currencyKey}&taxClassKey={taxClassKey}&taxSubClassKey={taxSubClassKey}&description={description}

This method allows you to create a new tax. It is necessary to set the natural key of the tax (e.g. “VAT”), the expiration date with the validFrom and validTo parameters, the value of the tax through the taxValue parameter, the natural key in CurrencyKey (e.g. “EUR”), the natural key of the class and subclass of the tax through the taxClassKey (e.g. “VAT”) and taxSubClassKey (e.g. “EXE”), respectively, as well as a description of the tax. In case the class or the subclass of the tax is not passed by parameters, they are created.

- PUT

/api/{tenant}/{organization}/taxscore/taxtypecodes/CreateTaxTypeCode?taxCodeKey={taxCodeKey}&validFrom={validFrom}&validTo={validTo}&taxValue={taxValue}¤cyKey={currencyKey}&taxClassKey={taxClassKey}&taxSubClassKey={taxSubClassKey}&description={description}

This method allows you to change a tax. It is necessary to set the natural key of the tax (e.g. "VAT") you want to change, the expiration date with the validFrom and validTo parameters, the value of the tax through the taxValue parameter, the natural key in CurrencyKey (e.g. "EUR"), the natural key of the class and subclass of the tax through the taxClassKey (e.g. "VAT") and taxSubClassKey (e.g. "EXE"), respectively, as well as a description of the tax. In case the tax class or subclass passed by the parameter does not exist, it is created.

Extended Items

GET	/api/{tenant}/{organization}/maintenancemanagementcore/itemextendeds	Gets all ItemExtendeds.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/itemextendeds?itemKey={itemKey}	Gets a Item Extended by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/itemextendeds/{id}	Gets a Item Extended by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/itemextendeds/search/version/{version}	Gets all ItemExtendeds with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/itemextendeds/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/itemextendeds?page={page}&pageSize={pageSize}	Gets all the ItemExtendeds using a paged return.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/itemextendeds	Creates a Item Extended.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/itemextendeds?itemKey={itemKey}	Updates a Item Extended.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/itemextendeds/{id}	Updates a Item Extended by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/itemextendeds?itemKey={itemKey}	Deletes a specific Item Extended.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/itemextendeds/{id}	Deletes a specific Item Extended by it's unique identifier.

Figure 31 - Example of the methods exposed by the Extended Items service

The items service makes it possible to obtain and manipulate Items and the respective purchase items. You can find an analysis of the methods made available by the service below:

- POST /api/{tenant}/{organization}/maintenancemanagementcore/itemextendeds

This method allows you to create a new item and the respective purchase item. To that end, the http message should have the "body" filled in with the structure of the resource *ItemExtendedResource*.

Example:

```
POST /api/mytenant/myorganization/maintenancemanagementcore/itemextendeds
Body:
{
  "ItemKey": "ART02",
  "Description": "QUEEN Chair",
  "ComplementaryDescription": null,
  "ItemType": 1,
  "Keyword": "QUEEN",
```

```

"Barcode": "5601234100012",
"Weight": 500,
"Volume": 1,
"InventoryPostings": true,
"ModelName": "",
"StockBalance": 0,
"MaxStock": null,
"MinStock": null,
"InventoryBalance": 0,
"CalculatedUnitCost": 0,
"LastUnitCost": 0,
"Remarks": "",
"IsSales": true,
"IsPurchases": true,
"ExternalId": null,
"ExternalVersion": null,
"IsExternallyManaged": false,
"LastPrice": 0,
"Rank": 1,
"BrandModelNaturalKey": null,
"BrandNaturalKey": "ASPECTTO",
"AssortmentNaturalKey": "Chairs",
"BaseUnitNaturalKey": "UN",
"WeightUnitNaturalKey": "UN",
"VolumeUnitNaturalKey": "UN",
"OriginCountryNaturalKey": "PT",
"ItemTaxSchemaNaturalKey": "IVA-TN",
"ItemWHTaxSchemaNaturalKey": "",
"ItemSubTypeNaturalKey": "3",
"UnitNaturalKey": "UN",
"LastSupplierPNaturalKey": "S.V.M."
}

```

○ PUT

/api/{tenant}/{organization}/maintenancemanagementcore/itemextendeds?ItemKey={ItemKey}

This method allows you to update a new item and the respective purchase item. To that end, the http message should have the “body” filled in with the full structure of the resource *ItemExtendedResource* (including the unchanged ones).

Example:

```

PUT
/api/mytenant/myorganization/maintenancemanagementcore/itemextendeds?ItemKey={ItemKey}
Body:
{
  "ItemKey": "ART02",
  "Description": "QUEEN Chair",
  "ComplementaryDescription": null,
  "ItemType": 1,

```

```

"Keyword": "QUEEN",
"Barcode": "5601234100012",
"Weight": 500,
"Volume": 1,
"InventoryPostings": true,
"ModelName": "",
"StockBalance": 0,
"MaxStock": null,
"MinStock": null,
"InventoryBalance": 0,
"CalculatedUnitCost": 0,
"LastUnitCost": 0,
"Remarks": "",
"IsSales": true,
"IsPurchases": true,
"ExternalId": null,
"ExternalVersion": null,
"IsExternallyManaged": false,
"LastPrice": 0,
"Rank": 1,
"BrandModelNaturalKey": null,
"BrandNaturalKey": "ASPECTTO",
"AssortmentNaturalKey": "Chairs",
"BaseUnitNaturalKey": "UN",
"WeightUnitNaturalKey": "UN",
"VolumeUnitNaturalKey": "UN",
"OriginCountryNaturalKey": "PT",
"ItemTaxSchemaNaturalKey": "IVA-TN",
"ItemWHTaxSchemaNaturalKey": "",
"ItemSubTypeNaturalKey": "3",
"UnitNaturalKey": "UN",
"LastSupplierPNaturalKey": "S.V.M."
}

```

○ **PUT /api/{tenant}/{organization}/maintenancemanagementcore/itemextendeds/{ID}**

This method allows you to update a new item and the respective purchase item. To that end, the http message should have the “body” filled in with the full structure of the resource *ItemExtendedResource* (including the unchanged ones).

Example:

```

PUT /api/mytenant/myorganization/maintenancemanagementcore/itemextendeds/8747fda4-
1cfb-49f3-96df-cd38acbf0a33
Body:
{
  "Description": "QUEEN Chair",
  "ComplementaryDescription": null,
  "ItemType": 1,
  "Keyword": "QUEEN",
  "Barcode": "5601234100012",

```

```

"Weight": 500,
"Volume": 1,
"InventoryPostings": true,
"ModelName": "",
"StockBalance": 0,
"MaxStock": null,
"MinStock": null,
"InventoryBalance": 0,
"CalculatedUnitCost": 0,
"LastUnitCost": 0,
"Remarks": "",
"IsSales": true,
"IsPurchases": true,
"ExternalId": null,
"ExternalVersion": null,
"IsExternallyManaged": false,
"LastPrice": 0,
"Rank": 1,
"BrandModelNaturalKey": null,
"BrandNaturalKey": "ASPECTTO",
"AssortmentNaturalKey": "Chairs",
"BaseUnitNaturalKey": "UN",
"WeightUnitNaturalKey": "UN",
"VolumeUnitNaturalKey": "UN",
"OriginCountryNaturalKey": "PT",
"ItemTaxSchemaNaturalKey": "IVA-TN",
"ItemWHTaxSchemaNaturalKey": "",
"ItemSubTypeNaturalKey": "3",
"UnitNaturalKey": "UN",
"LastSupplierPNaturalKey": "S.V.M."
}

```

- **GET /api/{tenant}/{organization}/maintenancemanagementcore/itemextendeds**

This method allows you to get the full list of existing items and the respective purchase items;

Example:

```
GET /api/mytenant/myorganization/maintenancemanagementcore/itemextendeds
```

- **GET /api/{tenant}/{organization}/
maintenancemanagementcore/itemextendeds?itemKey={itemKey}**

This method allows you to obtain the details of a specific item and the respective purchase item through its natural key.

Example:

Obtaining the details of the item with the natural key "A0001"

```
GET /api/mytenant/myorganization/maintenancemanagementcore/itemextendeds?ItemKey=A0001
```

- **DELETE /api/maintenancemanagementcore/itemextendeds?itemKey={itemKey}**

This method allows you to delete an item and purchase item through its natural key.

Example:

Deleting the item and the purchase item with the natural key A0001

```
DELETE
/api/mytenant/myorganization/maintenancemanagementcore/itemextendeds?itemKey=A0001
```

Extended Items Entry

GET	/api/{tenant}/{organization}/maintenancemanagementcore/itementryextendeds	Gets all ItemEntryExtendeds.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/itementryextendeds?key={key}	Gets a Item Entry Extended by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/itementryextendeds/{id}	Gets a Item Entry Extended by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/itementryextendeds/search/version/{version}	Gets all ItemEntryExtendeds with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/itementryextendeds/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/itementryextendeds?page={page}&pageSize={pageSize}	Gets all the ItemEntryExtendeds using a paged return.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/itementryextendeds	Creates a Item Entry Extended.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/itementryextendeds?key={key}	Updates a Item Entry Extended.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/itementryextendeds/{id}	Updates a Item Entry Extended by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/itementryextendeds?key={key}	Deletes a specific Item Entry Extended.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/itementryextendeds/{id}	Deletes a specific Item Entry Extended by it's unique identifier.

Figure 32 - Example of the methods exposed by the Extended Items Entry service.

NOTE: Only the POST and PUT methods work for creating and updating stock entries.

The extended items entry service allows you to create an items entry with an internal request associated.

POST /api/{tenant}/{organization}/ maintenancemanagementcore/itementryextendeds

This method allows you to create a new item entry with the information of the internal request.

Example:

```
POST api/mytenant/myorganization/maintenancemanagementcore/itementryextendeds
Body:
{
  "ItemEntryExtendedKey": "string",
  "DocumentTypeNaturalKey": "string",
  "SerieNaturalKey": "string",
  "SerieNumber": "int",
  "DocumentDate": "DateTime",
  "WarehouseNaturalKey": "string",
  "TotalValue": "decimal?",
  "PostingDate": "DateTime?",
  "Remarks": "string",
  "Note": "string",
  "IsPrinted": "bool?",
```

```

"CompanyNaturalKey": "string",
"ExternalId": "string",
"ExternalVersion": "string",
"IsExternallyManaged": "bool",
"InternalRequestNaturalKey": "string",
"ItemEntryLineExtendeds": [
  {
    "LineNumber": "int",
    "ItemNaturalKey": "string",
    "Description": "string",
    "Quantity": "decimal",
    "UnitNaturalKey": "string",
    "UnitPrice": "decimal?",
    "Value": "decimal?",
    "InventoryPostingNaturalKey": "string",
    "BaseUnitNaturalKey": "string",
    "ConversionFactor": "decimal?",
    "SchemaEntityNaturalKey": "string",
  }
],
}

```

- **PUT /api/{tenant}/{organization}/maintenancemanagementcore/itementryextendeds/ChangeItemEntryPrice**

This method allows you to update the price of an Item Entry. If there is a return of an Item Entry while updating its price, the return will be updated with the price at the moment of its respective Item Entry.

Example:

```

POST
/api/{tenant}/{organization}/maintenancemanagementcore/itementryextendeds/ChangeItemEntryPrice

Body:
{
  "ExternalId": "S.SS.2015.1",
  "ItemEntryPriceLines": [
    {
      "UpdatedPrice": "10.5",
      "ItemNaturalKey": "AIR-FILTER",
      "WarehouseNaturalKey": "A1",
      "Quantity": "25",
      "UnitNaturalKey": "UN"
    },
    {
      "UpdatedPrice": "20.5",
      "ItemNaturalKey": "AIR-CONDITIONING",
      "WarehouseNaturalKey": "A1",
      "Quantity": "5",
      "UnitNaturalKey": "UN"
    },
  ]
}

```

```

        "UpdatedPrice": "3",
        "ItemNaturalKey": "WATER-PUMP",
        "WarehouseNaturalKey": "A1",
        "Quantity": "1",
        "UnitNaturalKey": "UN"
    },
]
}

```

- **PUT**
/api/{tenant}/{organization}/maintenancemanagementcore/itementryextendeds/ChangeItemEntryPriceById

This method allows you to update the price of an Item Entry. If there is a return of an Item Entry while updating its price, the return will be updated with the price at the moment of its respective Item Entry.

Example:

```

POST
/api/{tenant}/{organization}/maintenancemanagementcore/itementryextendeds/ChangeItemEntryPriceById

Body:
{
  "ExternalId": "S.SS.2016.1",
  "ItemEntryPriceLineByIds": [
    {
      "UpdatedPrice": "25",
      "ItemEntryLineId": "60AE632D-FCEC-E511-9444-000C291C8A48"
    },
    {
      "UpdatedPrice": "11",
      "ItemEntryLineId": "34EBBAC4-FDF7-E511-9448-000C291C8A48"
    },
    {
      "UpdatedPrice": "19.91",
      "ItemEntryLineId": "CD197D9A-5396-42B9-B8F2-4F96A0093062"
    }
  ]
}

```


Extended Items Issue

ItemIssueExtendeds : Handles Item Issue Extended operations.

Show/Hide

List Operations

Expand Operations

Raw

GET	/api/{tenant}/{organization}/maintenancemanagementcore/itemissueextendeds	Gets all ItemIssueExtendeds.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/itemissueextendeds?key={key}	Gets a Item Issue Extended by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/itemissueextendeds/{id}	Gets a Item Issue Extended by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/itemissueextendeds/search/version/{version}	Gets all ItemIssueExtendeds with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/itemissueextendeds/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/itemissueextendeds?page={page}&pageSize={pageSize}	Gets all the ItemIssueExtendeds using a paged return.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/itemissueextendeds	Creates a Item Issue Extended.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/itemissueextendeds?key={key}	Updates a Item Issue Extended.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/itemissueextendeds/{id}	Updates a Item Issue Extended by it's unique identifier.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/itemissueextendeds?key={key}	Deletes a specific Item Issue Extended.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/itemissueextendeds/{id}	Deletes a specific Item Issue Extended by it's unique identifier.

Figure 33 - Example of the methods exposed by the Extended Items Issue service.

NOTE: Only the POST method works for creating stock issues.

The extended items issue service allows you to create an items issue with an internal request associated:

- **POST /api/{tenant}/{organization}/ maintenancemanagementcore/itemissueextendeds**

This method allows you to create a new item issue with the information of the internal requests.

Example:

```
POST /api/{tenant}/{organization}/maintenancemanagementcore/itemissueextendeds
```

Body:

```
{
  "ItemIssueExtendedKey": "string",
  "DocumentTypeNaturalKey": "string",
  "SerieNaturalKey": "string",
  "SerieNumber": "int",
  "DocumentDate": "DateTime",
  "WarehouseNaturalKey": "string",
  "TotalValue": "decimal?",
  "PostingDate": "DateTime?",
  "Remarks": "string",
  "Note": "string",
  "IsPrinted": "bool?",
  "CompanyNaturalKey": "string",
  "ExternalId": "string",
  "ExternalVersion": "string",
```

```

    "IsExternallyManaged": "bool",
    "ItemIssueLineExtendeds": [
      {
        "LineNumber": "int",
        "ItemNaturalKey": "string",
        "Description": "string",
        "Quantity": "decimal",
        "UnitNaturalKey": "string",
        "UnitPrice": "decimal?",
        "Value": "decimal?",
        "InventoryPostingNaturalKey": "string",
        "BaseUnitNaturalKey": "string",
        "ConversionFactor": "decimal?",
        "SchemaEntityNaturalKey": "string",
      }
    ],
    "IREExtendedLines": [
      {
        "InternalRequestNaturalKey": "string"
      }
    ]
  }

```

- **PUT /api/mytenant/myorganization/maintenancemanagementcore/itemissueextendeds/ChangeItemIssueExternalInfo**

This method allows you to update a Report with the data ExternalId, ExternalVersion and IsExternallyManaged from several item issues. To that end, the http message should have the “body” filled in with the full structure of the resource *ExternalInfoResource*.

Example:

Change the data of the stock issues with the NaturalKey = “SS.2015.1” and “SS.2015.2”

```

PUT
/api/mytenant/myorganization/maintenancemanagementcore/
workorderreportexpenses/ChangeWorkOrderReportExpenseExternalInfo

{
  "ExternalInfoLines" : [
    {
      "EntityKey" : "SS.2015.1",
      "ExternalKey" : "externalKey1",
      "ExternalVersion" : "v1",
      "IsExternallyManaged" : "true"
    },
    {
      "EntityKey" : "SS.2015.2",
      "ExternalKey" : "externalKey2",
      "ExternalVersion" : "v1",

```

```
    "IsExternallyManaged" : "true"  
  }  
]  
}
```

- **GET**

**/api/{tenant}/{organization}/maintenancemanagementcore/itemissueextendeds/GetItemIssues
WithReturns?version={version}**

This method allows you to obtain the list of item issues which are considered as returns (have the internal request information filled in), from a version onwards, the minimum value of the version is:
"AAAAAAAAAAAA=";

Example:

```
GET  
/api/mytenant/myorganization/maintenancemanagementcore/itemissueextendeds/GetItemIssuesWithReturns?version=AAAAAAA=
```

Purchases Management

Supplier

SupplierParties : Handles Supplier Party operations.

Show/Hide | List Operations | Expand Operations | Raw

GET	/api/{tenant}/{organization}/purchasescore/supplierparties	Gets all SupplierParties.
GET	/api/{tenant}/{organization}/purchasescore/supplierparties?partyKey={partyKey}	Gets a Supplier Party by the specified key.
GET	/api/{tenant}/{organization}/purchasescore/supplierparties/{id}	Gets a Supplier Party by the resource unique identifier.
GET	/api/{tenant}/{organization}/purchasescore/supplierparties/search/version/{version}	Gets all SupplierParties with a version higher than the specified.
GET	/api/{tenant}/{organization}/purchasescore/supplierparties/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/purchasescore/supplierparties?page={page}&pageSize={pageSize}	Gets all the SupplierParties using a paged return.
GET	/api/{tenant}/{organization}/purchasescore/supplierparties/GetFullByKey?partyKey={partyKey}	Gets a Customer Party by the specified key.
POST	/api/{tenant}/{organization}/purchasescore/supplierparties	Creates a Supplier Party.
POST	/api/{tenant}/{organization}/purchasescore/supplierparties/CreateFull	Creates a Supplier Party and Party.
PUT	/api/{tenant}/{organization}/purchasescore/supplierparties?partyKey={partyKey}	Updates a Supplier Party.
PUT	/api/{tenant}/{organization}/purchasescore/supplierparties/{id}	Updates a Supplier Party by it's unique identifier.
PUT	/api/{tenant}/{organization}/purchasescore/supplierparties/UpdateSupplierFullParty?partyKey={partyKey}	Updates a Party and Customer Party.
DELETE	/api/{tenant}/{organization}/purchasescore/supplierparties?partyKey={partyKey}	Deletes a specific Supplier Party.
DELETE	/api/{tenant}/{organization}/purchasescore/supplierparties/{id}	Deletes a specific Supplier Party by it's unique identifier.
DELETE	/api/{tenant}/{organization}/purchasescore/supplierparties/DeleteSupplierFullParty?partyKey={partyKey}	Deletes a specific Supplier Party and Party.

Figure 34 - Example of the methods exposed by the Suppliers service

The suppliers service allows you to obtain and manipulate Suppliers along with the associated data of the party. You can find an analysis of the methods made available by the service below:

- **GET /api/purchasescore/supplierparties/GetFullByKey?partyKey={partyKey}**

This method allows you to obtain the details of a specific party through its natural key.

Example:

```
GET
api/default/default/purchasescore/supplierparties/GetFullByKey?partyKey={partyKey}
```

- **POST /api/salescore/customerparties/CreateFull**

This method allows you to create a new Party entity and a new SupplierParty entity. To that end, the http message should have the "body" filled in with the structure of the resource *SupplierPartyFullResource*.

Example:

```
POST api/default/default/purchasescore/supplierparties/CreateFull.
Body:
{
  "SearchTerm": "Api",
```

```
"Name": "Api Teste XPTO",
>Name2": null,
"CompanyTaxID": "",
"ElectronicMail": null,
"Telephone": null,
"TeleFax": null,
"Mobile": null,
"WebsiteURL": null,
>Note": null,
>Type": null,
"IsCustomer": null,
"IsSupplier": true,
"IsFinancialInstitution": null,
"IsEmployee": null,
"IsTaxParty": null,
"FinancialBalance": 0,
"Remarks": null,
"EquityCapital": null,
"RegistrationOffice": null,
"RegistrationNumber": null,
"SocialSecurityNumber": null,
"StreetName": "Street API teste",
"BuildingNumber": "",
"PostalZone": "4830-755  ",
"CityName": "API city",
>ContactName": "",
>ContactTitle": "",
"Username": null,
"IsSalesRepresentative": null,
"IsProspect": null,
"Currency": "EUR",
"Country": "PT",
"PostalZone2": null,
"SettlementDiscountPercent": null,
"OpenOrderAmount": 0,
"OpenGoodsReceiptAmount": 0,
"OpenReturnsAmount": 0,
"PartyKey": "SP001",
"SupplierGroup": "GERAL",
"PaymentTerm": "PP",
"SupplierItemPrices": [],
"DeliveryTerm": "N/A",
"SupplierStatus": 1,
"PartyTaxSchema": "IVA-RN-MN",
"PartyWithholdingTaxSchema": "IRS",
"PaymentMean": "CHQ",
"AccountingSupplierParty": "0001",
>UserFields": [],
```

```
"Id": "64a53340-9eed-e411-8278-0021863035d8",  
"IsActive": true,  
"IsDeleted": false,  
"Version": "AAAAAAALE/Y="
```

```
}
```

- **PUT `api/default/default/salescore/customerparties?partyKey=partyKey008`.**

This method allows you to change a new Party entity and the SupplierParty entity associated with the party. To that end, the http message should have the “body” filled in with the structure of the resource *SupplierPartyFullResource*.

Example:

Edit the customer party and the party with the natural key *partyKey008*

```
PUT  
api/default/default/purchasescore/supplierparties/UpdateSupplierFullParty?partyK  
ey={partyKey}
```

- **DELETE `api/purchasescore/supplierparties/DeleteSupplierFullParty?partyKey={partyKey}`**

This method allows you to delete a given Party entity and the SupplierParty entity associated with the Party entity through its natural key.

Example:

Deleting the supplier with the natural key *partyKey08*

```
DELETE  
api/default/default/purchasescore/supplierparties/DeleteSupplierFullParty?partyK  
ey={partyKey}
```

Sales Management

Customer

CustomerParties : Handles Customer Party operations.			Show/Hide	List Operations	Expand Operations	Raw
GET	/api/{tenant}/{organization}/salescore/customerparties	Gets all CustomerParties.				
GET	/api/{tenant}/{organization}/salescore/customerparties?partyKey={partyKey}	Gets a Customer Party by the specified key.				
GET	/api/{tenant}/{organization}/salescore/customerparties/{id}	Gets a Customer Party by the resource unique identifier.				
GET	/api/{tenant}/{organization}/salescore/customerparties/search/version/{version}	Gets all CustomerParties with a version higher than the specified.				
GET	/api/{tenant}/{organization}/salescore/customerparties/search?field={field}&value={value}	No Documentation Found.				
GET	/api/{tenant}/{organization}/salescore/customerparties?page={page}&pageSize={pageSize}	Gets all the CustomerParties using a paged return.				
GET	/api/{tenant}/{organization}/salescore/customerparties/GetFullByKey?partyKey={partyKey}	Gets a Customer Party by the specified key.				
POST	/api/{tenant}/{organization}/salescore/customerparties	Creates a Customer Party.				
POST	/api/{tenant}/{organization}/salescore/customerparties/CreateFull	Creates a Customer Party.				
PUT	/api/{tenant}/{organization}/salescore/customerparties?partyKey={partyKey}	Updates a Customer Party.				
PUT	/api/{tenant}/{organization}/salescore/customerparties/{id}	Updates a Customer Party by it's unique identifier.				
PUT	/api/{tenant}/{organization}/salescore/customerparties/UpdateCustomerFullParty?partyKey={partyKey}	Updates a Party and Customer Party.				
DELETE	/api/{tenant}/{organization}/salescore/customerparties?partyKey={partyKey}	Deletes a specific Customer Party.				
DELETE	/api/{tenant}/{organization}/salescore/customerparties/{id}	Deletes a specific Customer Party by it's unique identifier.				
DELETE	/api/{tenant}/{organization}/salescore/customerparties/DeleteCustomerFullParty?partyKey={partyKey}	Deletes a specific Customer Party.				

Figure 35 - Example of the methods exposed by the Customers service

The customers service allows you to obtain and manipulate Customers along with the associated data of the party. You can find an analysis of the methods made available by the service below:

- **GET /api/ salescore/customerparties/GetFullByKey?partyKey={partyKey}**

This method allows you to obtain the details of a specific party through the natural key

Example:

```
GET /api/salescore/customerparties/GetFullByKey?partyKey=P0001
```

- **POST /api/ salescore/customerparties/CreateFull**

This method allows you to create a new Party entity and a new CustomerParty entity. To that end, the http message should have the “body” filled in with the structure of the resource *CustomerPartyFullResource*.

Example:

```
POST /api/salescore/customerparties/CreateFull
Body:
{
  "SearchTerm": "Api 005",
  "Name": "Api Teste 005",
  "Name2": null,
```

```
"CompanyTaxID": null,
"ElectronicMail": null,
"Telephone": null,
"TeleFax": null,
"Mobile": null,
"WebsiteURL": null,
>Note": null,
>Type": null,
"IsCustomer": true,
"IsSupplier": null,
"IsFinancialInstitution": null,
"IsEmployee": null,
"IsTaxParty": null,
"FinancialBalance": 0,
"Remarks": null,
"EquityCapital": null,
"RegistrationOffice": null,
"RegistrationNumber": null,
"SocialSecurityNumber": null,
"StreetName": "Street Name",
"BuildingNumber": "",
"PostalZone": "4470 -345  ",
"CityName": "Maia",
"ContactName": "",
"ContactTitle": "",
"Username": null,
"IsSalesRepresentative": null,
"IsProspect": null,
"Currency": "EUR",
"Country": "PT",
"PostalZone2": null,
"SettlementDiscountPercent": 50,
"CreditLimit": true,
"CreditLimitValue": 1000,
"CreditLimitDueDays": true,
"NumberDueDays": 30,
"CreditLimitDueDaysValue": 500,
"OpenOrderAmount": 0,
"OpenGoodsIssueAmount": 0,
"OpenReturnsAmount": 0,
"IndiscriminatedCustomerParty": true,
"EndCustomer": true,
"PartyKey": "CP01",
"CustomerGroup": "CF",
"PaymentTerm": "PP",
"DeliveryTerm": "TRANSP",
"PriceList": "OUTLET",
"CustomerStatus": 2,
```



```
"PartyTaxSchema": "IVA-RN-ME",
"PartyWithholdingTaxSchema": "RES",
"PaymentMean": "NUM",
"DunningMethod": "STD",
"AccountingSchema": 2,
"AccountingCustomerParty": "0004",
"SalesRepresentative": "0011"
}
```

- **PUT**

api/default/default/salescore/customerparties/UpdateCustomerFullParty?partyKey={partyKey}

.

This method allows you to change a new Party entity and the CustomerParty entity associated with the party. To that end, the http message should have the "body" filled in with the structure of the resource *CustomerPartyFullResource* as well as the partyKey.

Example:

Edit the customer party and the party with the natural key *partyKey008*

```
PUT
api/default/default/salescore/customerparties/UpdateCustomerFullParty?partyKey={
partyKey}.
```

- **DELETE**

/api/patterns/salescore/customerparties/DeleteCustomerFullParty?partyKey={partyKey}

This method allows you to delete a given Party entity and the CustomerParty entity associated with the Party entity through its natural key.

Example:

Deleting the customer with the natural key *partyKey008*

```
DELETE
api/default/default/salescore/customerparties/DeleteCustomerFullParty?partyKey={
partyKey}
```

Administration

Exchange Rates

ExchangeRatesDocuments : Handles Exchange Rates Document operations.

			Show/Hide	List Operations	Expand Operations	Raw
GET	/api/{tenant}/{organization}/patterns/exchangeratesdocuments	Gets all ExchangeRatesDocuments.				
GET	/api/{tenant}/{organization}/patterns/exchangeratesdocuments?number={number}	Gets a Exchange Rates Document by the specified key.				
GET	/api/{tenant}/{organization}/patterns/exchangeratesdocuments/{id}	Gets a Exchange Rates Document by the resource unique identifier.				
GET	/api/{tenant}/{organization}/patterns/exchangeratesdocuments/search/version/{version}	Gets all ExchangeRatesDocuments with a version higher than the specified.				
GET	/api/{tenant}/{organization}/patterns/exchangeratesdocuments/search?field={field}&value={value}	No Documentation Found.				
GET	/api/{tenant}/{organization}/patterns/exchangeratesdocuments?page={page}&pageSize={pageSize}	Gets all the ExchangeRatesDocuments using a paged return.				
POST	/api/{tenant}/{organization}/patterns/exchangeratesdocuments	Creates a Exchange Rates Document.				
POST	/api/{tenant}/{organization}/patterns/exchangeratesdocuments/CreateExchangeRateDocument?validFrom={validFrom}&fromCurrencyKey={fromCurrencyKey}&toCurrencyKey={toCurrencyKey}&conversionFactor={conversionFactor}	Creates the exchange rate document.				
PUT	/api/{tenant}/{organization}/patterns/exchangeratesdocuments?number={number}	Updates a Exchange Rates Document.				
PUT	/api/{tenant}/{organization}/patterns/exchangeratesdocuments/{id}	Updates a Exchange Rates Document by it's unique identifier.				
DELETE	/api/{tenant}/{organization}/patterns/exchangeratesdocuments?number={number}	Deletes a specific Exchange Rates Document.				
DELETE	/api/{tenant}/{organization}/patterns/exchangeratesdocuments/{id}	Deletes a specific Exchange Rates Document by it's unique identifier.				

Figure 36 - Example of the methods exposed by the Exchange Rates service

The exchange rates service makes it possible to obtain and manipulate Exchange Rates:

- **POST**
/api/{tenant}/{organization}/patterns/exchangeratesdocuments/CreateExchangeRateDocument?validFrom={validFrom}&fromCurrencyKey={fromCurrencyKey}&toCurrencyKey={toCurrencyKey}&conversionFactor={conversionFactor}

This method allows you to create a new exchange rate document. It is necessary to identify the exchange rate document's validity (validFrom) and the conversion factor (conversionFactor) between the currencies (fromCurrencyKey and toCurrencyKey) by indicating the respective natural keys.

Employees

GET	/api/{tenant}/{organization}/humancapitalmanagementcore/employees	Gets all Employees.
GET	/api/{tenant}/{organization}/humancapitalmanagementcore/employees?partyKey={partyKey}	Gets a Employee by the specified key.
GET	/api/{tenant}/{organization}/humancapitalmanagementcore/employees/{id}	Gets a Employee by the resource unique identifier.
GET	/api/{tenant}/{organization}/humancapitalmanagementcore/employees/search/version/{version}	Gets all Employees with a version higher than the specified.
GET	/api/{tenant}/{organization}/humancapitalmanagementcore/employees/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/humancapitalmanagementcore/employees?page={page}&pageSize={pageSize}	Gets all the Employees using a paged return.
GET	/api/{tenant}/{organization}/humancapitalmanagementcore/employees/GetFullEmployeeParties	Gets the full employee parties.
GET	/api/{tenant}/{organization}/humancapitalmanagementcore/employees/GetFullEmployeePartiesByVersion?version={version}	Gets the full employee parties by version.
GET	/api/{tenant}/{organization}/humancapitalmanagementcore/employees/GetFullEmployeePartyByKey?partyKey={partyKey}	Gets the full employee party by key.
POST	/api/{tenant}/{organization}/humancapitalmanagementcore/employees	Creates a Employee.
POST	/api/{tenant}/{organization}/humancapitalmanagementcore/employees/CreateFullEmployeeParty	Creates the full employee party.
PUT	/api/{tenant}/{organization}/humancapitalmanagementcore/employees?partyKey={partyKey}	Updates a Employee.
PUT	/api/{tenant}/{organization}/humancapitalmanagementcore/employees/{id}	Updates a Employee by it's unique identifier.
PUT	/api/{tenant}/{organization}/humancapitalmanagementcore/employees/UpdateFullEmployeeParty	Updates the specified employee party resource.
DELETE	/api/{tenant}/{organization}/humancapitalmanagementcore/employees?partyKey={partyKey}	Deletes a specific Employee.
DELETE	/api/{tenant}/{organization}/humancapitalmanagementcore/employees/{id}	Deletes a specific Employee by it's unique identifier.
DELETE	/api/{tenant}/{organization}/humancapitalmanagementcore/employees/DeleteFullEmployeePartyByKey?partyKey={partyKey}	Deletes the full employee party by key.

Figure 37 - Example of the methods exposed by the Employees service

The employees service allows you to obtain and manipulate Employees along with the associated data of the Party. You can find an analysis of the methods made available by the service below:

- **GET**
/api/mytenant/myorganization/humancapitalmanagementcore/employees/GetFullEmployeeParties

This method allows you to get the full list of existing parties.

Example:

```
GET
/api/mytenant/myorganization/humancapitalmanagementcore/employees/GetFullEmployeeParties
```

- **GET**
/api/mytenant/myorganization/humancapitalmanagementcore/employees/GetFullEmployeePartiesByVersion?version={version}

This method allows you to obtain the full list of existing parties from a version onwards. The minimum value of the version is "AAAAAAAAAA".

Example:

```
GET
/api/mytenant/myorganization/humancapitalmanagementcore/employees/GetFullEmployeePartiesByVersion?version=AAAAAAA=
```

- **GET**
/api/mytenant/myorganization/humancapitalmanagementcore/employees/GetFullEmployeePartyByKey?partyKey={partyKey}

This method allows you to obtain the existing entity through its PartyKey.

Example:

```
GET
/api/mytenant/myorganization/humancapitalmanagementcore/employees/GetFullEmployeePartyByKey?partyKey=0001
```

- **POST**
/api/mytenant/myorganization/humancapitalmanagementcore/employees/CreateFullEmployeeParty

This method allows you to create a new Employee entity and a new Party entity. To that end, the http message should have the “body” filled in with the structure of the resource *EmployeePartyResource* which can be found in this example.

Example:

```
POST
/api/mytenant/myorganization/humancapitalmanagementcore/employees/CreateFullEmployeeParty
Body:
{
  "Employee": {
    "ExternalReferenceID": "",
    "IsInternalEmployee": true,
    "PhoneNumber": "",
    "Extension": "",
    "ContractStartDate": null,
    "ContractEndDate": null,
    "CostperHour": 0,
    "VacationDays": null,
    "Notes": "",
    "ExchangeRate": 1,
    "ExchangeRateDate": "2015-04-30T14:40:02.013",
    "PartyKey": "",
    "EmployeeExpertises": [],
    "EmployeeSkills": [],
    "EmployeeQualifications": [],
    "Calendar": null,
    "TimeSchedule": null,
    "Company": null,
    "Currency": "EUR",
  },
  "Party": {
```

```
"PartyKey": "0002",
"SearchTerm": "Antonio",
"Name": "Antonio Ferreira",
"Name2": "",
"CompanyTaxID": "",
"ElectronicMail": "",
"Telephone": "",
"TeleFax": "",
"Mobile": "",
"WebsiteURL": "",
"Note": "",
"Type": "Party",
"IsCustomer": null,
"IsSupplier": null,
"IsFinancialInstitution": null,
"IsEmployee": true,
"IsTaxParty": null,
"FinancialBalance": 0,
"Remarks": "",
"EquityCapital": "",
"RegistrationOffice": "",
"RegistrationNumber": "",
"SocialSecurityNumber": "",
"StreetName": "",
"BuildingNumber": "",
"PostalZone": "",
"CityName": "",
"ContactName": "",
"ContactTitle": "",
"Username": "",
"IsSalesRepresentative": null,
"IsProspect": null,
"ExternalId": "",
"ExternalVersion": "",
"IsExternallyManaged": false,
"Currency": "EUR",
"Country": "PT",
"UserFields": [],
},
"Addresses": [],
"Contacts": []
}
```

- **PUT**
api/mytenant/myorganization/humancapitalmanagementcore/employees/UpdateFullEmployeeParty

This method allows you to change a Party entity and the Employee entity associated with that Party. To that end, the http message should have the “body” filled in with the structure of the resource *EmployeePartyResource* seen above, and the PartyKey attribute of the Party entity must be filled in.

- **DELETE**
/api/mytenant/myorganization/employees/DeleteFullEmployeePartyByKey?partyKey={partyKey}

This method allows you to delete an existing Party entity and the Employee entity associated with the Party entity through its natural key.

Example:

Deleting the Employee with the natural key 0001:

```
DELETE
api/mytenant/myorganization/humancapitalmanagementcore/employees/DeleteFullEmployeePartyByKey?partyKey=0001
```

Internal Request Management

Extended Internal Requests

GET	/api/{tenant}/{organization}/maintenancemanagementcore/internalrequestextendeds	Gets all InternalRequestExtendeds.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/internalrequestextendeds?serieNaturalKey={serieNaturalKey}&documentTypeNaturalKey={documentTypeNaturalKey}&seriesNumber={seriesNumber}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/internalrequestextendeds/{id}	Gets a Internal Request Extended by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/internalrequestextendeds/search/version/{version}	Gets all InternalRequestExtendeds with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/internalrequestextendeds/search?field={field}&value={value}	No Documentation Found.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/internalrequestextendeds?page={page}&pageSize={pageSize}	Gets all the InternalRequestExtendeds using a paged return.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/internalrequestextendeds/GetInternalRequestStates	Gets the Internal Request states.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/internalrequestextendeds/GetInternalRequestState?internalRequestKey={internalRequestKey}	Gets the Internal Request states.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/internalrequestextendeds/GetNotIntegratedInternalRequests	Gets the not integrated internal requests.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/internalrequestextendeds/GetClosedInternalRequests?version={version}	Gets the closed internal requests with a version higher than the specified.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/internalrequestextendeds	Creates a Internal Request Extended.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/internalrequestextendeds?serieNaturalKey={serieNaturalKey}&documentTypeNaturalKey={documentTypeNaturalKey}&seriesNumber={seriesNumber}	No Documentation Found.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/internalrequestextendeds/{id}	Updates a Internal Request Extended by it's unique identifier.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/internalrequestextendeds/ChangeInternalRequestState	Changes the state of the Internal Request.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/internalrequestextendeds/ChangeInternalRequestExternalInfo	Changes the internal request external information.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/internalrequestextendeds?serieNaturalKey={serieNaturalKey}&documentTypeNaturalKey={documentTypeNaturalKey}&seriesNumber={seriesNumber}	No Documentation Found.
DELETE	/api/{tenant}/{organization}/maintenancemanagementcore/internalrequestextendeds/{id}	Deletes a specific Internal Request Extended by it's unique identifier.

Figure 38 - Example of the methods exposed by the Extended Internal Requests service

NOTE: The methods described below are the only ones available.

This service makes it possible to obtain and manipulate Valuekeep's Internal Requests. You can find an analysis of the methods made available by the service below:

- **GET**
`/api/{tenant}/{organization}/maintenancemanagementcore/InternalRequestExtendedss/GetInternalRequestState?internalRequestKey={internalRequestKey}`
This method allows you to obtain the state of a specific party through the natural key.

Example:

Obtaining the state of the Report with the internalRequestKey = "PI.2015.4"

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/InternalRequestExtendedss
/GetInternalRequestState?internalRequestKey=PI.2015.4
```

- **GET**
/api/{tenant}/{organization}/maintenancemanagementcore/InternalRequestExtendedss/GetInternalRequestStates

This method allows you to obtain the possible states of the Internal Request.

- **GET /api/{tenant}/{organization}/maintenancemanagementcore/InternalRequestExtendedss/GetNotIntegratedInternalRequests**

This method allows you to obtain all internal requests in the Approved state which have not been integrated by an external application.

- **GET /api/{tenant}/{organization}/maintenancemanagementcore/InternalRequestExtendedss/GetClosedInternalRequests?version={version}**

This method allows you to obtain the list of internal requests in the Closed state from a version onwards, the minimum value of the version is: "AAAAAAAAAA=".

Example:

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/InternalRequestExtendedss/
GetInternalRequestState?verion= AAAAAAAAAAA=
```

- **POST /api/{tenant}/{organization}/maintenancemanagementcore/internalrequestextendeds**

This method allows you to create a new Internal Request.

Example:

```
POST
/api/{tenant}/{organization}/maintenancemanagementcore/internalrequestextendeds
Body:
{
  "SerieNaturalKey": "2015",
  "DocumentTypeNaturalKey": "PI",
  "SeriesNumber": "0",
  "DocumentDate": "2015-11-01T00:00:00",
  "Description": "Necessidade de ferramentas",
  "Remarks": "Substituição por ferramentas desgastadas",
  "IsPrinted": false,
  "Motive": "",
  "SourceDoc": "",
  "ExternalId": "",
  "ExternalVersion": "",
  "IsExternallyManaged": false,
  "SchemaEntStateNaturalKey": "001.PI",
  "CompanyNaturalKey": "",
  "WarehouseNaturalKey": "A1",
  "RequestedByNaturalKey": "0002",
  "InternalRequestNaturalKey": "",
  "CostCenterNaturalKey": ""
```



```

    "WorkOrderKey": "OT0003",
    "DocumentLines": [
      {
        "LineNumber": 1,
        "Description": "Chave Bocas 9mm",
        "Quantity": "1",
        "RequestDate": "2015-11-01T00:00:00",
        "ConversionFactor": "1",
        "Note": "",
        "OpenQuantity": "1",
        "DeliveryDate": "2015-11-15T00:00:00",
        "OriginalQuantity": "1",
        "Motive": "",
        "BaseUnitNaturalKey": "UN",
        "SchemaEntStateNaturalKey": "001.PI",
        "PurchaseItemNaturalKey": "CHAVE BOCAS 9MM",
        "UnitNaturalKey": "UN"
      }
    ]
  }
}

```

Note: The following field CostCenterKey is ignored in the POST and PUT methods.

- **PUT**

/api/mytenant/myorganization/maintenancemanagementcore/InternalRequestExtendeds/ChangeInternalRequestState "

This method allows you to change the state of the internal request. To that end, it is necessary to fill in the InternalRequestStateResource.

Example:

Changing the state of the internal request "PI.2015.2"

```

PUT
{
  "InternalRequestKey" : "PI.2015.2",
  "State" : "004.PI",
  "Motive" : "ASD"
}

```

NOTE: The "Motive" field should only be filled in when we reject an internal request, in all other cases it gets ignored.

- **PUT**

/api/mytenant/myorganization/maintenancemanagementcore/InternalRequestExtendeds/ChangeInternalRequestExternalInfo

This method allows you to fill in the ExternalId, ExternalVersion and IsExternallyManaged data from several internal requests. To that end, it is necessary to provide a resource with the internal requests we want to change and the respective data.

Example:

Changing the data of the internal requests = "PI.2015.1" and "PI.2015.2"

```

PUT
/api/mytenant/myorganization/maintenancemanagementcore/InternalRequestExtendeds/
ChangeInternalRequestExternalInfo

```

```

{
  "ExternalInfoLines" : [
    {
      "EntityKey" : "PI.2015.1",
      "ExternalKey" : "externalKey1",
      "ExternalVersion" : "v1",
      "IsExternallyManaged" : "true"
    },
    {
      "EntityKey" : "PI.2015.2",
      "ExternalKey" : "externalKey1",
      "ExternalVersion" : "v1",
      "IsExternallyManaged" : "true"
    }
  ]
}

```

Configuration Tables

Maintenance setup

GET	/api/{tenant}/{organization}/maintenancemanagementcore/maintenancesetups	Gets all MaintenanceSetups.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/maintenancesetups?key={key}	Gets a Maintenance Setup by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/maintenancesetups/{id}	Gets a Maintenance Setup by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/maintenancesetups/search/version/{version}	Gets all MaintenanceSetups with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/maintenancesetups/search/version?version={version}	Gets all MaintenanceSetups with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/maintenancesetups/search?field={field}&value={value}	Gets all MaintenanceSetups containing the search criteria on the specified field.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/maintenancesetups?page={page}&pageSize={pageSize}	Gets all the MaintenanceSetups using a paged return.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/maintenancesetups/GetCurrencies	Gets the currencies.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/maintenancesetups/GetIncludeAllAssetsAndLocations	Gets the include all assets and locations parameter value.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/maintenancesetups	Creates a Maintenance Setup.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/maintenancesetups?key={key}	Updates a Maintenance Setup.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/maintenancesetups/{id}	Updates a Maintenance Setup by it's unique identifier.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/maintenancesetups/UpdateCostCenters	Updates the cost centers.

Figure 39 - Example of the methods exposed by the Maintenance Setup service.

The maintenance setup service allows you to view its configuration. It is not possible to create new configurations or change the existing ones via WebAPI.

- **GET**
/api/{tenant}/{organization}/maintenancemanagementcore/maintenancesetups/GetIncludeAllAssetsAndLocations

This method allows you to obtain the current state of the field “Include all Assets and Locations” referring to the Work Report application.

Example:

Response

```
{
  "IncludeAllAssetsAndLocations": false
}
```

Base Currency and Reporting Currency set in Valuekeep

- **GET /api/{tenant}/{organization}/maintenancemanagementcore/maintenancesetups/GetCurrencies**
This method allows you to obtain the base currency and reporting currency set in Valuekeep.

Integration

Integration Log

GET	/api/{tenant}/{organization}/maintenancemanagementcore/integrationerrors	Gets all IntegrationErrors.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/integrationerrors?errorKey={errorKey}	Gets a Integration Error by the specified key.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/integrationerrors/{id}	Gets a Integration Error by the resource unique identifier.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/integrationerrors/search/version/{version}	Gets all IntegrationErrors with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/integrationerrors/search/version?version={version}	Gets all IntegrationErrors with a version higher than the specified.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/integrationerrors/search?field={field}&value={value}	Gets all IntegrationErrors containing the search criteria on the specified field.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/integrationerrors?page={page}&pageSize={pageSize}	Gets all the IntegrationErrors using a paged return.
GET	/api/{tenant}/{organization}/maintenancemanagementcore/integrationerrors/GetProcessedIntegrationErrorsByVersionHigherThan?version={version}	Gets the processed integration errors by version higher than.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/integrationerrors	Creates a Integration Error.
POST	/api/{tenant}/{organization}/maintenancemanagementcore/integrationerrors/CreateIntegrationErrors	No Documentation Found.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/integrationerrors?errorKey={errorKey}	Updates a Integration Error.
PUT	/api/{tenant}/{organization}/maintenancemanagementcore/integrationerrors/{id}	Updates a Integration Error by it's unique identifier.

Figura 40 - Example of the methods exposed by the Integration Log service.

The integration log service allows you to view and create events related to the Work Report integration process.

- **GET**

/api/{tenant}/{organization}/maintenancemanagementcore/integrationerrors/GetProcessedIntegrationErrorsByVersionHigherThan?version={version}

This method allows you to obtain the list of integration events that have been previously processed, from a version onwards. The minimum value of the version is: "AAAAAAAAAAAA=";

Example:

```
GET
/api/mytenant/myorganization/maintenancemanagementcore/workorderexecutions/
GetProcessedIntegrationErrorsByVersionHigherThan?version=AAAAAAAAAAAA=
```

Note: Version field could be left blank.

- **POST**

/api/{tenant}/{organization}/maintenancemanagementcore/integrationerrors/CreateIntegrationErrors

This method allows you to create a set of integration events.

Example:

```
POST
/api/{tenant}/{organization}/maintenancemanagementcore/integrationerrors/CreateIntegrationErrors

Body:
[
  {
    "ErrorSource": "Service",
    "IntegrationEventType" : "IntegrationError",
    "Description": "Unknown or expired access token",
    "Detail": "Technical data; stack trace; ...",
    "Date": "2015-12-01T09:05:01",
    "IsProcessed": "false",
    "IntegrationErrorActions": [
      {
        "Description": "Retry login",
        "URL": "http://universeapi.valuekeep.com/"
      }
    ]
  },
  {
    "ErrorSource": "User",
    "User": "tec1",
    "IntegrationEventType" : "IntegrationEvent",
    "Description": "User tec1 logged on.",
    "Detail": "",
    "Date": "2015-12-01T09:05:03",
    "IsProcessed": "false",
    "IntegrationErrorActions": []
  }
]
```

The allowed values for the event source and type are as follows:

ErrorSource

- User (1)
- Import (2)
- Export (3)
- Service (4)

IntegrationEventType

- IntegrationError (1)
- IntegrationEvent (2)

Entity Attachment Service

The entityattachment service makes it possible to download and upload files to entities that support attachments (e.g. WorkOrders, Assets, etc.). You can find an analysis of the methods made available by the service below:

Upload

- **POST /api/{tenant}/{organization}/{module}/{entity}/{id}/attachments**
It allows to upload a file to a specific entity.

Exemplo:

- POST /api/{tenant}/{organization}/MaintenanceManagementCore /Assets/{id}/attachments
- **Content-Type:** multipart/form-data;
 - **Content-Disposition:** form-data; name="description"; description="Description example."
 - **Content-Disposition:** form-data; name="notes"; description="Notes"
 - **Content-Disposition:** form-data; name="file"; filename="test.txt"
 - **Content-Type:** text/plain (It depends on file extension. Eg. Txt is a text/plain type; PDF is an application/pdf; etc.)

Example is in the code example section ([link](#)).

Download

- **GET /api/{tenant}/{organization}/{module}/{entity}/{id}/attachments**
This method allows you to get the full list of existing attachments associated with a specific entity (id);.
- **GET /api/{tenant}/{organization}/{module}/{entity}/{id}/attachments/{ attachmentId}**
Downloads the file (attachmentid) associated with a specific entity.

Code examples

Login

To create an HTTP client which can be used in requests to the Web API, see the following code:

```
using (HttpClient client = new HttpClient())
{
    // initialize HttpClient, ask for json format response.

    client.BaseAddress = new Uri("http://localhost:60326");
    client.DefaultRequestHeaders.Accept.Clear();
    client.DefaultRequestHeaders.Accept.Add(new
MediaTypeWithQualityHeaderValue("application/json"));

    // Login
    Console.WriteLine("Attempting Login");
    string accessToken = await Login(client);
}
```

The following routine exemplifies the login process.

```
private static async Task<string> Login(HttpClient client)
{
    try
    {
        HttpResponseMessage loginResponse = await client.PostAsJsonAsync(
            "api/login",
            new { userName = "admin", Password = "aa" });

        // valid the response
        if (!loginResponse.IsSuccessStatusCode)
        {
            throw new Exception(loginResponse.Content.ToString());
        }

        // get content data
        string loginResult = await
((StreamContent)loginResponse.Content).ReadAsStringAsync();

        // extract access token
        dynamic data = JObject.Parse(loginResult);
        string accessToken =
((dynamic)data).AuthorizedModules[0].AuthorizationToken.ToString();

        return accessToken;
    }
    catch (Exception ex)
```

```

    {
        throw new Exception("Login failed.", ex);
    }
}

```

Access Levels

The following classes will be used in this section's routines:

```

namespace PRIMAVERA.EAMWebAPIClient
{
    /// <summary>
    /// The AccessLevelAssetEmployeeInfo class.
    /// </summary>
    public class AccessLevelAssetEmployeeInfo
    {
        /// <summary>
        /// Gets or sets the access level key.
        /// </summary>
        /// <value>
        /// The access level key.
        /// </value>
        public string AccessLevelKey { get; set; }

        /// <summary>
        /// Gets or sets the access level description.
        /// </summary>
        /// <value>
        /// The access level description.
        /// </value>
        public string AccessLevelDescription { get; set; }

        /// <summary>
        /// Gets or sets the asset key.
        /// </summary>
        /// <value>
        /// The asset key.
        /// </value>
        public string AssetKey { get; set; }

        /// <summary>
        /// Gets or sets the asset description.
        /// </summary>
        /// <value>
        /// The asset description.
        /// </value>
        public string AssetDescription { get; set; }

        /// <summary>

```

```

        /// Gets or sets the asset is up time.
        /// </summary>
        /// <value>
        /// The asset is up time.
        /// </value>
        public bool? AssetIsUpTime { get; set; }

        /// <summary>
        /// Gets or sets the employee key.
        /// </summary>
        /// <value>
        /// The employee key.
        /// </value>
        public string EmployeeKey { get; set; }

        /// <summary>
        /// Gets or sets the name of the employee.
        /// </summary>
        /// <value>
        /// The name of the employee.
        /// </value>
        public string EmployeeName { get; set; }

        /// <summary>
        /// Gets or sets the version.
        /// </summary>
        /// <value>
        /// The version.
        /// </value>
        public byte[] Version { get; set; }
    }
}

namespace PRIMAVERA.EAMWebAPIClient
{
    /// <summary>
    /// The AccessLevelLocationEmployeeInfo class.
    /// </summary>
    public class AccessLevelLocationEmployeeInfo
    {
        /// <summary>
        /// Gets or sets the access level key.
        /// </summary>
        /// <value>
        /// The access level key.
        /// </value>
        public string AccessLevelKey { get; set; }

        /// <summary>

```



```
    /// Gets or sets the access level description.
    /// </summary>
    /// <value>
    /// The access level description.
    /// </value>
    public string AccessLevelDescription { get; set; }
```

```
    /// <summary>
    /// Gets or sets the location key.
    /// </summary>
    /// <value>
    /// The location key.
    /// </value>
    public string LocationKey { get; set; }
```

```
    /// <summary>
    /// Gets or sets the location description.
    /// </summary>
    /// <value>
    /// The location description.
    /// </value>
    public string LocationDescription { get; set; }
```

```
    /// <summary>
    /// Gets or sets the location is up time.
    /// </summary>
    /// <value>
    /// The location is up time.
    /// </value>
    public bool? LocationIsUpTime { get; set; }
```

```
    /// <summary>
    /// Gets or sets the employee key.
    /// </summary>
    /// <value>
    /// The employee key.
    /// </value>
    public string EmployeeKey { get; set; }
```

```
    /// <summary>
    /// Gets or sets the name of the employee.
    /// </summary>
    /// <value>
    /// The name of the employee.
    /// </value>
    public string EmployeeName { get; set; }
```

```
    /// <summary>
```

```

        /// Gets or sets the version.
        /// </summary>
        /// <value>
        /// The version.
        /// </value>
        public byte[] Version { get; set; }
    }
}

```

The following code shows you how to obtain information about the access levels of the Employees over the Assets:

```

private static async Task<IEnumerable<AccessLevelAssetEmployeeInfo>>
GetAccessLevelAssetEmployeeInfos(string version ,HttpClient client, string
accessToken)
{
    IEnumerable<AccessLevelAssetEmployeeInfo> entities = null;

    // create request message
    HttpRequestMessage getEntityMessage = new
HttpRequestMessage(HttpMethod.Get, FormatApiUri(client) +
"/GetAccessLevelAssetEmployeeInfos?version="+version );

    // add authorization token header
    getEntityMessage.AddAuthorizationHeader(accessToken);

    // request api
    HttpResponseMessage response = await
client.SendAsync(getEntityMessage);
    if (response.IsSuccessStatusCode)
    {
        IEnumerable<AccessLevelAssetEmployeeInfo> responseEntities = new
List<AccessLevelAssetEmployeeInfo>();

        entities = await
response.Content.ReadAsAsync<IEnumerable<AccessLevelAssetEmployeeInfo>>();
    }
    else
    {
        // request failed
        string result = await
((StreamContent)response.Content).ReadAsStringAsync();

        Console.WriteLine(string.Concat("GetAccessLevelAssetEmployeeInfos failed.\n",
result));
    }

    return entities;
}

```

The following routine shows you how to obtain information about the access levels of the Employees over the Locations.

```
private static async Task<IEnumerable<AccessLevelLocationEmployeeInfo>>
GetAccessLevelLocationEmployeeInfos(string version,HttpClient client, string
accessToken)
{
    IEnumerable<AccessLevelLocationEmployeeInfo> entities = null;

    // create request message
    HttpRequestMessage getEntityMessage = new
HttpRequestMessage(HttpMethod.Get, FormatApiUri(client) +
"/GetAccessLevelLocationEmployeeInfos?version="+version );

    // add authorization token header
    getEntityMessage.AddAuthorizationHeader(accessToken);

    // request api
    HttpResponseMessage response = await
client.SendAsync(getEntityMessage);
    if (response.IsSuccessStatusCode)
    {
        IEnumerable<AccessLevelLocationEmployeeInfo> responseEntities =
new List<AccessLevelLocationEmployeeInfo>();

        entities = await
response.Content.ReadAsAsync<IEnumerable<AccessLevelLocationEmployeeInfo>>();
    }
    else
    {
        // request failed
        string result = await
((StreamContent)response.Content).ReadAsStringAsync();

        Console.WriteLine(string.Concat("GetAccessLevelLocationEmployeeInfos failed.\n",
result));
    }

    return entities;
}
```

Asset Routes

The following classes will be used in this section's routines:

```
/// <summary>
/// The AssetRoute resource.
/// </summary>
public class AssetRoute
{
    #region Public Properties

    /// <summary>
    /// Route unique identifier
    /// </summary>
    /// <remarks>
    /// This property belongs to the entity natural key.
    /// This property value is required.
    /// </remarks>
    public string AssetRouteKey { get; set; }

    /// <summary>
    /// Route description
    /// </summary>
    /// <remarks>
    /// This property corresponds to the entity description.
    /// </remarks>
    public string Description { get; set; }

    /// <summary>
    /// Route Notes
    /// </summary>
    public string Notes { get; set; }

    /// <summary>
    /// Asset Route Compositions
    /// </summary>
    public ICollection<AssetRouteComposition> AssetRouteCompositions { get; set; }

    /// <summary>
    /// Family
    /// </summary>
    public string AssetFamily { get; set; }

    /// <summary>
    /// Brand
    /// </summary>
    public string Brand { get; set; }

    /// <summary>
```

```

        /// Location
        /// </summary>
        /// <remarks>
        /// This property value is required.
        /// </remarks>
        public string Location { get; set; }

        /// <summary>
        /// Select the route cost center
        /// </summary>
        public string CostCenter { get; set; }

        /// <summary>
        /// Model
        /// </summary>
        public string BrandModel { get; set; }

        #endregion Public Properties
    }
    /// <summary>
    /// The AssetRouteComposition resource.
    /// </summary>
    public class AssetRouteComposition
    {
        #region Public Properties

        /// <summary>
        /// Select the asset to add to the route
        /// </summary>
        /// <remarks>
        /// This property value is required.
        /// </remarks>
        public string Asset { get; set; }

        #endregion Public Properties
    }
}

```

The following routine exemplifies the process of creating an Asset Route:

```

private static async Task<Uri> CreateAssetRoute(HttpClient client, string
accessToken)
{
    var entityToCreate = GetAssetRouteSample();

    // create request message
    HttpRequestMessage postEntityMessage = new
HttpRequestMessage(HttpMethod.Post, FormatApiUri(client));

    // add authorization token header
    postEntityMessage.AddAuthorizationHeader(accessToken);
}

```

```

        // convert entity to json notation
        string entityRequest = entityToCreate.JsonSerialize();

        // add json to message body
        postEntityMessage.Content = new StringContent(entityRequest,
Encoding.UTF8, "application/json");

        // send POST to api for entity creation
        using (HttpResponseMessage response = await
client.SendAsync(postEntityMessage))
        {
            if (response.IsSuccessStatusCode)
            {
                // success: entity created.
                Uri resourceUri = response.Headers.Location;
                return resourceUri;
            }
            else
            {
                // request failed

                string result = await
((StreamContent)response.Content).ReadAsStringAsync();
                Console.WriteLine(string.Concat("POST AssetRoute failed.\n",
result));

                throw new Exception("Unable to create new resource.");
            }
        }
    }
}

```

Obtaining an Asset Route is exemplified in the following routine.

```

private static async Task<AssetRoute> GetAssetRoute(HttpClient client, string
accessToken, Uri resourceUri)
{
    AssetRoute entity = null;

    // create request message
    HttpRequestMessage getEntityMessage = new
HttpRequestMessage(HttpMethod.Get, resourceUri);

    // add authorization token header
    getEntityMessage.AddAuthorizationHeader(accessToken);

    // request api
    HttpResponseMessage response = await
client.SendAsync(getEntityMessage);
    if (response.IsSuccessStatusCode)
    {

```

```

        IEnumerable<AssetRoute> responseEntities = new
List<AssetRoute>();

        entity = await response.Content.ReadAsAsync<AssetRoute>();
    }
    else
    {
        // request failed
        string result = await
((StreamContent)response.Content).ReadAsStringAsync();
        Console.WriteLine(string.Concat("GetAssetRoute failed.\n",
result));
    }

    return entity;
}

```

To remove an Asset Route, the code to be used could be the following:

```

private static async Task DeleteAssetRoute(AssetRoute entity, HttpClient client,
string accessToken)
{
    // create request message
    HttpRequestMessage postEntityMessage = new
HttpRequestMessage(HttpMethod.Delete, FormatApiUri(client) + "?assetRouteKey=" +
entity.AssetId);

    // add authorization token header
    postEntityMessage.AddAuthorizationHeader(accessToken);

    // send DELETE to api for entity deletion
    using (HttpResponseMessage response = await
client.SendAsync(postEntityMessage))
    {
        if (response.IsSuccessStatusCode)
        {
            // success: entity deleted.
            Uri resourceUri = response.Headers.Location;
        }
        else
        {
            // request failed
            string result = await
((StreamContent)response.Content).ReadAsStringAsync();
            Console.WriteLine(string.Concat("DELETE Asset Route
failed.\n", result));

            throw new Exception("Unable to delete AssetRoute.");
        }
    }
}

```

Location Routes

The following classes will be used in this section's routines:

```
/// <summary>
/// The LocationRoute resource.
/// </summary>
public class LocationRoute
{
    #region Public Properties

    /// <summary>
    /// Location Route
    /// </summary>
    /// <remarks>
    /// This property belongs to the entity natural key.
    /// This property value is required.
    /// </remarks>
    public string LocationRouteKey { get; set; }

    /// <summary>
    /// Description
    /// </summary>
    /// <remarks>
    /// This property corresponds to the entity description.
    /// </remarks>
    public string Description { get; set; }

    /// <summary>
    /// Remarks
    /// </summary>
    public string Notes { get; set; }

    /// <summary>
    /// Location
    /// </summary>
    /// <remarks>
    /// This property value is required.
    /// </remarks>
    public string Location { get; set; }

    /// <summary>
    /// Location Route Compositions
    /// </summary>
    public ICollection<LocationRouteComposition> LocationRouteCompositions {
get; set; }
```



```

        /// <summary>
        /// Select the route cost center
        /// </summary>
        public string CostCenter { get; set; }

    #endregion
}
/// <summary>
/// The LocationRouteComnposition resource.
/// </summary>
public class LocationRouteComposition
{
    #region Public Properties

        /// <summary>
        /// Location
        /// </summary>
        /// <remarks>
        /// This property value is required.
        /// </remarks>
        public string Location { get; set; }

    #endregion
}

```

The following routine exemplifies the process of creating a Location Route:

```

private static async Task<Uri> CreateLocationRoute(HttpClient client, string
accessToken)
{
    var entityToCreate = GetLocationRouteSample();

    // create request message
    HttpRequestMessage postEntityMessage = new
HttpRequestMessage(HttpMethod.Post, FormatApiUri(client));

    // add authorization token header
    postEntityMessage.AddAuthorizationHeader(accessToken);

    // convert entity to json notation
    string entityRequest = entityToCreate.JsonSerialize();

    // add json to message body
    postEntityMessage.Content = new StringContent(entityRequest,
Encoding.UTF8, "application/json");

    // send POST to api for entity creation
    using (HttpResponseMessage response = await
client.SendAsync(postEntityMessage))
    {

```

```

        if (response.IsSuccessStatusCode)
        {
            // success: entity created.
            Uri resourceUri = response.Headers.Location;
            return resourceUri;
        }
        else
        {
            // request failed

            string result = await
            ((StreamContent)response.Content).ReadAsStringAsync();
            Console.WriteLine(string.Concat("POST LocationRoute
            failed.\n", result));

            throw new Exception("Unable to create new resource.");
        }
    }
}

```

Obtaining a Location Route is exemplified in the following routine.

```

    private static async Task<LocationRoute> GetLocationRoute(HttpClient
    client, string accessToken, Uri resourceUri)
    {
        LocationRoute entity = null;

        // create request message
        HttpRequestMessage getEntityMessage = new
        HttpRequestMessage(HttpMethod.Get, resourceUri);

        // add authorization token header
        getEntityMessage.AddAuthorizationHeader(accessToken);

        // request api
        HttpResponseMessage response = await
        client.SendAsync(getEntityMessage);
        if (response.IsSuccessStatusCode)
        {
            IEnumerable<LocationRoute> responseEntities = new
            List<LocationRoute>();

            entity = await response.Content.ReadAsAsync<LocationRoute>();
        }
        else
        {
            // request failed
            string result = await
            ((StreamContent)response.Content).ReadAsStringAsync();
            Console.WriteLine(string.Concat("GetLocationRoute failed.\n",
            result));
        }
    }
}

```

```

    }

    return entity;
}

```

To remove an Asset Route, the code to be used could be the following:

```

private static async Task DeleteLocationRoute(LocationRoute entity, HttpClient
client, string accessToken)
{
    // create request message
    HttpRequestMessage postEntityMessage = new
HttpRequestMessage(HttpMethod.Delete, FormatApiUri(client) +
"?locationRouteKey=" + entity.LocationRouteKey);

    // add authorization token header
    postEntityMessage.AddAuthorizationHeader(accessToken);

    // send DELETE to api for entity deletion
    using (HttpResponseMessage response = await
client.SendAsync(postEntityMessage))
    {
        if (response.IsSuccessStatusCode)
        {
            // success: entity deleted.
            Uri resourceUri = response.Headers.Location;
        }
        else
        {
            // request failed
            string result = await
((StreamContent)response.Content).ReadAsStringAsync();
            Console.WriteLine(string.Concat("DELETE Location Route
failed.\n", result));

            throw new Exception("Unable to delete LocationRoute.");
        }
    }
}

```

Work Centers

The following classes will be used in this section's routines:

```
/// <summary>
/// The WorkCenterAssetEmployeeInfo class.
/// </summary>
public class WorkCenterAssetEmployeeInfo
{
    /// <summary>
    /// Gets or sets the work center key.
    /// </summary>
    /// <value>
    /// The work center key.
    /// </value>
    public string WorkCenterKey { get; set; }

    /// <summary>
    /// Gets or sets the work center description.
    /// </summary>
    /// <value>
    /// The work center description.
    /// </value>
    public string WorkCenterDescription { get; set; }

    /// <summary>
    /// Gets or sets the asset key.
    /// </summary>
    /// <value>
    /// The asset key.
    /// </value>
    public string AssetKey { get; set; }

    /// <summary>
    /// Gets or sets the asset description.
    /// </summary>
    /// <value>
    /// The asset description.
    /// </value>
    public string AssetDescription { get; set; }

    /// <summary>
    /// Gets or sets the employee key.
    /// </summary>
    /// <value>
    /// The employee key.
    /// </value>
    public string EmployeeKey { get; set; }

    /// <summary>
```

```

    /// Gets or sets the name of the employee.
    /// </summary>
    /// <value>
    /// The name of the employee.
    /// </value>
    public string EmployeeName { get; set; }

    /// <summary>
    /// Gets or sets the version.
    /// </summary>
    /// <value>
    /// The version.
    /// </value>
    public byte[] Version { get; set; }
}
/// <summary>
/// The WorkCenterLocationEmployeeInfo class.
/// </summary>
public class WorkCenterLocationEmployeeInfo
{
    /// <summary>
    /// Gets or sets the work center key.
    /// </summary>
    /// <value>
    /// The work center key.
    /// </value>
    public string WorkCenterKey { get; set; }

    /// <summary>
    /// Gets or sets the work center description.
    /// </summary>
    /// <value>
    /// The work center description.
    /// </value>
    public string WorkCenterDescription { get; set; }

    /// <summary>
    /// Gets or sets the location key.
    /// </summary>
    /// <value>
    /// The location key.
    /// </value>
    public string LocationKey { get; set; }

    /// <summary>
    /// Gets or sets the location description.
    /// </summary>
    /// <value>

```

```

        /// The location description.
        /// </value>
        public string LocationDescription { get; set; }

        /// <summary>
        /// Gets or sets the employee key.
        /// </summary>
        /// <value>
        /// The employee key.
        /// </value>
        public string EmployeeKey { get; set; }

        /// <summary>
        /// Gets or sets the name of the employee.
        /// </summary>
        /// <value>
        /// The name of the employee.
        /// </value>
        public string EmployeeName { get; set; }

        /// <summary>
        /// Gets or sets the version.
        /// </summary>
        /// <value>
        /// The version.
        /// </value>
        public byte[] Version { get; set; }
    }

```

To obtain a summary of the access information while relating Work Centers, Employees and Assets, use the following routine:

```

private static async Task<IEnumerable<WorkCenterAssetEmployeeInfo>>
GetWorkCenterAssetEmployeeInfos(string version, HttpClient client, string
accessToken)
{
    IEnumerable<WorkCenterAssetEmployeeInfo> entities = null;

    // create request message
    HttpRequestMessage getEntityMessage = new
HttpRequestMessage(HttpMethod.Get, FormatApiUri(client) +
"/GetWorkCenterAssetEmployeeInfos?version="+version );

    // add authorization token header
    getEntityMessage.AddAuthorizationHeader(accessToken);

    // request api
    HttpResponseMessage response = await
client.SendAsync(getEntityMessage);

    if (response.IsSuccessStatusCode)

```

```

    {
        IEnumerable<WorkCenterAssetEmployeeInfo> responseEntities = new
List<WorkCenterAssetEmployeeInfo>();

        entities = await
response.Content.ReadAsAsync<IEnumerable<WorkCenterAssetEmployeeInfo>>();
    }
    else
    {
        // request failed
        string result = await
((StreamContent)response.Content).ReadAsStringAsync();
        Console.WriteLine(string.Concat("GetWorkCenterAssetEmployeeInfos
failed.\n", result));
    }

    return entities;
}

```

To obtain a summary of the access information while relating Work Centers, Employees and Locations, use the following routine:

```

private static async Task<IEnumerable<WorkCenterLocationEmployeeInfo>>
GetWorkCenterLocationEmployeeInfos(string version,HttpClient client, string
accessToken)
{
    IEnumerable<WorkCenterLocationEmployeeInfo> entities = null;

    // create request message
    HttpRequestMessage getEntityMessage = new
HttpRequestMessage(HttpMethod.Get, FormatApiUri(client) +
"/GetWorkCenterLocationEmployeeInfos?version="+version );

    // add authorization token header
    getEntityMessage.AddAuthorizationHeader(accessToken);

    // request api
    HttpResponseMessage response = await
client.SendAsync(getEntityMessage);
    if (response.IsSuccessStatusCode)
    {
        IEnumerable<WorkCenterLocationEmployeeInfo> responseEntities =
new List<WorkCenterLocationEmployeeInfo>();

        entities = await
response.Content.ReadAsAsync<IEnumerable<WorkCenterLocationEmployeeInfo>>();
    }
    else
    {
        // request failed
        string result = await
((StreamContent)response.Content).ReadAsStringAsync();
    }
}

```

```

Console.WriteLine(string.Concat("GetWorkCenterLocationEmployeeInfos failed.\n",
result));
    }

    return entities;
}

```

Work Orders

The following classes will be used in this section's routines:

```

/// <summary>
/// The SchemaEntityStateInfo class.
/// </summary>
public class SchemaEntityStateInfo
{
    /// <summary>
    /// Gets or sets the state key.
    /// </summary>
    /// <value>
    /// The state key.
    /// </value>
    public string StateKey { get; set; }

    /// <summary>
    /// Gets or sets the description.
    /// </summary>
    /// <value>
    /// The description.
    /// </value>
    public string Description { get; set; }

    /// <summary>
    /// Gets or sets a value indicating whether this instance is available
    in mobile.
    /// </summary>
    /// <value>
    /// <c>true</c> if this instance is available in mobile; otherwise,
    <c>false</c>.
    /// </value>
    public bool IsAvailableInMobile { get; set; }

    /// <summary>
    /// Gets or sets a value indicating whether this instance is initial
    state.
    /// </summary>
    /// <value>

```



```

        /// <c>true</c> if this instance is initial state; otherwise,
        <c>false</c>.
    /// </value>
    public bool IsInitialState { get; set; }

    /// <summary>
    /// Gets or sets the is locked for changes.
    /// </summary>
    /// <value>
    /// The is locked for changes.
    /// </value>
    public bool? IsLockedForChanges { get; set; }

    /// <summary>
    /// Gets or sets the version.
    /// </summary>
    /// <value>
    /// The version.
    /// </value>
    public byte[] Version { get; set; }
}
/// <summary>
/// The SchemaEntityStateTransitionInfo class.
/// </summary>
public class SchemaEntityStateTransitionInfo
{
    /// <summary>
    /// Gets or sets the schema entity state key.
    /// </summary>
    /// <value>
    /// The schema entity state key.
    /// </value>
    public string SchemaEntityStateKey { get; set; }

    /// <summary>
    /// Gets or sets the target entity state key.
    /// </summary>
    /// <value>
    /// The target entity state key.
    /// </value>
    public string TargetEntityStateKey { get; set; }

    /// <summary>
    /// Gets or sets a value indicating whether this instance is internal.
    /// </summary>
    /// <value>
    /// <c>true</c> if this instance is internal; otherwise, <c>false</c>.
    /// </value>
    public bool IsInternal { get; set; }
}

```

```

        /// <summary>
        /// Gets or sets a value indicating whether [need justification].
        /// </summary>
        /// <value>
        /// <c>true</c> if [need justification]; otherwise, <c>false</c>.
        /// </value>
        public bool NeedJustification { get; set; }

        /// <summary>
        /// Gets or sets the version.
        /// </summary>
        /// <value>
        /// The version.
        /// </value>
        public byte[] Version { get; set; }
    }

    /// <summary>
    /// The WorkOrder Resource
    /// </summary>
    public class WorkOrder
    {
        #region Public Properties

        /// <summary>
        /// Work order unique identifier
        /// </summary>
        /// <remarks>
        /// This property belongs to the entity natural key.
        /// This property value is required.
        /// </remarks>
        public string WorkOrderKey { get; set; }

        /// <summary>
        /// Work order description
        /// </summary>
        /// <remarks>
        /// This property corresponds to the entity description.
        /// </remarks>
        public string Description { get; set; }

        /// <summary>
        /// Date when the work order is expected to start
        /// </summary>
        public DateTime? Date { get; set; }

        /// <summary>
        /// Date when the work order was opened

```

```

    /// </summary>
    public DateTime? OpeningDate { get; set; }

    /// <summary>
    /// Estimated date to finish the work order
    /// </summary>
    public DateTime? EstimatedEndDate { get; set; }

    /// <summary>
    /// Date when the work order was closed technically
    /// </summary>
    public DateTime? EndDate { get; set; }

    /// <summary>
    /// Date when the work order was closed administratively
    /// </summary>
    public DateTime? ClosingDate { get; set; }

    /// <summary>
    /// Work order remarks
    /// </summary>
    public string Remarks { get; set; }

    /// <summary>
    /// Asset Under Warranty
    /// </summary>
    public bool? AssetUnderWarranty { get; set; }

    /// <summary>
    /// Original Date
    /// </summary>
    public DateTime? OriginalDate { get; set; }

    /// <summary>
    /// Is Printed
    /// </summary>
    public bool? IsPrinted { get; set; }

    /// <summary>
    /// Work Order state
    /// </summary>
    /// <remarks>
    /// This property value is required.
    /// </remarks>
    public string SchemaEntityState { get; set; }

    /// <summary>
    /// Defines the work order type

```

```

    /// </summary>
    /// <remarks>
    /// This property value is required.
    /// </remarks>
    public string WorkOrderType { get; set; }

    /// <summary>
    /// WorkCenter responsible for this work order
    /// </summary>
    public string WorkCenter { get; set; }

    /// <summary>
    /// Responsible person for this work order
    /// </summary>
    public string Responsible { get; set; }

    /// <summary>
    /// Asset where the work order will occur.
    /// </summary>
    public string Asset { get; set; }

    /// <summary>
    /// Location where the work order will occur.
    /// </summary>
    public string Location { get; set; }

    /// <summary>
    /// Asset route where the work will occur
    /// </summary>
    public string AssetRoute { get; set; }

    /// <summary>
    /// Location route where the work will occur
    /// </summary>
    public string LocationRoute { get; set; }

    /// <summary>
    /// Work order cost center
    /// </summary>
    public string CostCenter { get; set; }

    /// <summary>
    /// Work order defined priority
    /// </summary>
    public string PriorityLevel { get; set; }

    /// <summary>
    /// Condition to perform the work order tasks

```

```
    /// </summary>
    public string OperatingCondition { get; set; }

    /// <summary>
    /// Issue
    /// </summary>
    public string Issue { get; set; }

    /// <summary>
    /// Location Shutdown
    /// </summary>
    public string LocationShutdown { get; set; }

    /// <summary>
    /// Asset Shutdown
    /// </summary>
    public string AssetShutdown { get; set; }

    /// <summary>
    /// Failure
    /// </summary>
    public string Failure { get; set; }

    /// <summary>
    /// Supplier
    /// </summary>
    public string SupplierContract { get; set; }

    /// <summary>
    /// Customer Contract
    /// </summary>
    public string CustomerContract { get; set; }

    /// <summary>
    /// Warranty
    /// </summary>
    public string WarrantyContract { get; set; }

    /// <summary>
    /// Provider responsible to perform the work order tasks
    /// </summary>
    public string Provider { get; set; }

    /// <summary>
    /// Client where the work order will be performed
    /// </summary>
    public string Client { get; set; }
```

```

    /// <summary>
    /// Work Order Plans
    /// </summary>
    public ICollection<WorkOrderPlan> WorkOrderPlans { get; set; }

    /// <summary>
    /// Task of the Work Order Plans
    /// </summary>
    public ICollection<WorkOrderPlanTask> WorkOrderPlanTasks { get; set; }

    #endregion Public Properties
}

/// <summary>
/// The WorkOrderPlan Resource
/// </summary>
public class WorkOrderPlan
{
    #region Public Properties

    /// <summary>
    /// Maintenance Plan
    /// </summary>
    /// <remarks>
    /// This property value is required.
    /// </remarks>
    public string MaintenancePlan { get; set; }

    #endregion Public Properties
}

/// <summary>
/// The WorkOrderPlanTask Resource
/// </summary>
public class WorkOrderPlanTask
{
    #region Public Properties

    /// <summary>
    /// Task of the W0 Plan
    /// </summary>
    /// <remarks>
    /// This property belongs to the entity natural key.
    /// This property value is required.
    /// </remarks>
    public Guid WorkOrderPlanTaskKey { get; set; }

    /// <summary>
    /// Id. of the Work Order Plan

```

```

        /// </summary>
        /// <remarks>
        /// This property value is required.
        /// </remarks>
        public Guid WorkOrderPlanId { get; set; }

        /// <summary>
        /// Description
        /// </summary>
        public string TaskDescription { get; set; }

        /// <summary>
        /// Expected Date
        /// </summary>
        public DateTime? PredictedDate { get; set; }

        /// <summary>
        /// Maintenance Plan Task
        /// </summary>
        /// <remarks>
        /// This property value is required.
        /// </remarks>
        public string MaintenancePlanTask { get; set; }

        /// <summary>
        /// Task
        /// </summary>
        /// <remarks>
        /// This property value is required.
        /// </remarks>
        public string Task { get; set; }

        #endregion Public Properties
    }
    /// <summary>
    /// The WorkOrderPlanTasksInfo class.
    /// </summary>
    public class WorkOrderPlanTasksInfo
    {
        /// <summary>
        /// Gets or sets the work order key.
        /// </summary>
        /// <value>
        /// The work order key.
        /// </value>
        public string WorkOrderKey { get; set; }

        /// <summary>

```

```

    /// Gets or sets the work order description.
    /// </summary>
    /// <value>
    /// The work order description.
    /// </value>
    public string WorkOrderDescription { get; set; }

    /// <summary>
    /// Gets or sets the maintenance plan key.
    /// </summary>
    /// <value>
    /// The maintenance plan key.
    /// </value>
    public string MaintenancePlanKey { get; set; }

    /// <summary>
    /// Gets or sets the maintenance plan description.
    /// </summary>
    /// <value>
    /// The maintenance plan description.
    /// </value>
    public string MaintenancePlanDescription { get; set; }

    /// <summary>
    /// Gets or sets the task key.
    /// </summary>
    /// <value>
    /// The task key.
    /// </value>
    public string TaskKey { get; set; }

    /// <summary>
    /// Gets or sets the task description.
    /// </summary>
    /// <value>
    /// The task description.
    /// </value>
    public string TaskDescription { get; set; }

    /// <summary>
    /// Gets or sets the employee key.
    /// </summary>
    /// <value>
    /// The employee key.
    /// </value>
    public string EmployeeKey { get; set; }

    /// <summary>

```



```

        /// Gets or sets the name of the employee.
        /// </summary>
        /// <value>
        /// The name of the employee.
        /// </value>
        public string EmployeeName { get; set; }

        /// <summary>
        /// Gets or sets the version.
        /// </summary>
        /// <value>
        /// The version.
        /// </value>
        public byte[] Version { get; set; }
    }

```

The following routine creates a Work Order for an Asset.

```

private static async Task<Uri> CreateWorkOrderForAsset(string assetKey,
List<string> maintenancePlanKeys, HttpClient client, string accessToken)
{
    string key = IdGenerator.GetNextIdentifier();

    // create request message
    HttpRequestMessage postEntityMessage = new
HttpRequestMessage(HttpMethod.Post, FormatApiUri(client) +
"/CreateWorkOrderForAsset?key=" + key + "&description=OT%20" + key +
"&workOrderType=GN&assetKey=" + assetKey);

    // add authorization token header
    postEntityMessage.AddAuthorizationHeader(accessToken);

    // convert entity to json notation
    string entityRequest = maintenancePlanKeys.JsonSerialize();

    // add json to message body
    postEntityMessage.Content = new StringContent(entityRequest,
Encoding.UTF8, "application/json");

    // send POST to api for entity creation
    using (HttpResponseMessage response = await
client.SendAsync(postEntityMessage))
    {
        if (response.IsSuccessStatusCode)
        {
            // success: entity created.
            Uri resourceUri = response.Headers.Location;
            return resourceUri;
        }
        else
    }
}

```

```

        {
            // request failed

            string result = await
            ((StreamContent)response.Content).ReadAsStringAsync();
            Console.WriteLine(string.Concat("POST WorkOrder failed.\n",
            result));

            throw new Exception("Unable to create new resource.");
        }
    }
}

```

The following routine creates a Work Order for an Asset Route.

```

    private static async Task<Uri> CreateWorkOrderForAssetRoute(string
    assetRouteKey, List<string> maintenancePlanKeys, HttpClient client, string
    accessToken)
    {
        string key = IdGenerator.GetNextIdentifier();

        // create request message
        HttpRequestMessage postEntityMessage = new
        HttpRequestMessage(HttpMethod.Post, FormatApiUri(client) +
        "/CreateWorkOrderForAssetRoute?key=" + key + "&description=OT%20" + key +
        "&workOrderType=GN&assetRouteKey=" + assetRouteKey);

        // add authorization token header
        postEntityMessage.AddAuthorizationHeader(accessToken);

        // convert entity to json notation
        string entityRequest = maintenancePlanKeys.JsonSerialize();

        // add json to message body
        postEntityMessage.Content = new StringContent(entityRequest,
        Encoding.UTF8, "application/json");

        // send POST to api for entity creation
        using (HttpResponseMessage response = await
        client.SendAsync(postEntityMessage))
        {
            if (response.IsSuccessStatusCode)
            {
                // success: entity created.
                Uri resourceUri = response.Headers.Location;
                return resourceUri;
            }
            else
            {
                // request failed
            }
        }
    }
}

```

```

        string result = await
((StreamContent)response.Content).ReadAsStringAsync();

        Console.WriteLine(string.Concat("POST WorkOrder failed.\n",
result));

        throw new Exception("Unable to create new resource.");
    }
}
}

```

The following routine creates a Work Order for a Location.

```

private static async Task<Uri> CreateWorkOrderForLocation(string
locationKey, List<string> maintenancePlanKeys ,HttpClient client, string
accessToken)
{
    string key = IdGenerator.GetNextIdentifier();

    // create request message
    HttpRequestMessage postEntityMessage = new
HttpRequestMessage(HttpMethod.Post, FormatApiUri(client) +
"/CreateWorkOrderForLocation?key=" + key + "&description=OT%20" + key +
"&workOrderType=GN&locationKey=" + locationKey);

    // add authorization token header
    postEntityMessage.AddAuthorizationHeader(accessToken);

    // convert entity to json notation
    string entityRequest = maintenancePlanKeys.JsonSerialize();

    // add json to message body
    postEntityMessage.Content = new StringContent(entityRequest,
Encoding.UTF8, "application/json");

    // send POST to api for entity creation
    using (HttpResponseMessage response = await
client.SendAsync(postEntityMessage))
    {
        if (response.IsSuccessStatusCode)
        {
            // success: entity created.
            Uri resourceUri = response.Headers.Location;
            return resourceUri;
        }
        else
        {
            // request failed

            string result = await
((StreamContent)response.Content).ReadAsStringAsync();

```

```

        Console.WriteLine(string.Concat("POST WorkOrder failed.\n",
result));

        throw new Exception("Unable to create new resource.");
    }
}
}

```

The following routine creates a Work Order for a Location Route.

```

    private static async Task<Uri> CreateWorkOrderForLocationRoute(string
locationRouteKey, List<string> maintenancePlanKeys, HttpClient client, string
accessToken)
    {
        string key = IdGenerator.GetNextIdentifier();

        // create request message
        HttpRequestMessage postEntityMessage = new
HttpRequestMessage(HttpMethod.Post, FormatApiUri(client) +
"/CreateWorkOrderForLocationRoute?key=" + key + "&description=OT%20" + key +
"&workOrderType=GN&locationRouteKey=" + locationRouteKey);

        // add authorization token header
        postEntityMessage.AddAuthorizationHeader(accessToken);

        // convert entity to json notation
        string entityRequest = maintenancePlanKeys.JsonSerialize();

        // add json to message body
        postEntityMessage.Content = new StringContent(entityRequest,
Encoding.UTF8, "application/json");

        // send POST to api for entity creation
        using (HttpResponseMessage response = await
client.SendAsync(postEntityMessage))
        {
            if (response.IsSuccessStatusCode)
            {
                // success: entity created.
                Uri resourceUri = response.Headers.Location;
                return resourceUri;
            }
            else
            {
                // request failed

                string result = await
((StreamContent)response.Content).ReadAsStringAsync();
                Console.WriteLine(string.Concat("POST WorkOrder failed.\n",
result));
            }
        }
    }
}

```

```

        throw new Exception("Unable to create new resource.");
    }
}
}

```

In the following example, a specific Work Order is requested.

```

private static async Task<WorkOrder> GetWorkOrder(HttpClient client, string
accessToken, Uri resourceUri)
{
    WorkOrder entity = null;

    // create request message
    HttpRequestMessage getEntityMessage = new
HttpRequestMessage(HttpMethod.Get, resourceUri);

    // add authorization token header
    getEntityMessage.AddAuthorizationHeader(accessToken);

    // request api
    HttpResponseMessage response = await
client.SendAsync(getEntityMessage);
    if (response.IsSuccessStatusCode)
    {
        IEnumerable<WorkOrder> responseEntities = new List<WorkOrder>();

        entity = await response.Content.ReadAsAsync<WorkOrder>();
    }
    else
    {
        // request failed
        string result = await
((StreamContent)response.Content).ReadAsStringAsync();
        Console.WriteLine(string.Concat("GetWorkOrder failed.\n",
result));
    }

    return entity;
}

```

All the Work Orders in a specific State are returned in this routine.

```

private static async Task<IEnumerable<WorkOrder>> GetWorkOrdersByState(string
stateKey, string version, HttpClient client, string accessToken)
{
    IEnumerable<WorkOrder> entities = null;

    // create request message
    HttpRequestMessage getEntityMessage = new
HttpRequestMessage(HttpMethod.Get, FormatApiUri(client) +
"/GetWorkOrdersByState?stateKey=" + stateKey + "&version="+version);
}

```

```

        // add authorization token header
        getEntityMessage.AddAuthorizationHeader(accessToken);

        // request api
        HttpResponseMessage response = await
client.SendAsync(getEntityMessage);
        if (response.IsSuccessStatusCode)
        {
            IEnumerable<WorkOrder> responseEntities = new List<WorkOrder>();

            entities = await
response.Content.ReadAsAsync<IEnumerable<WorkOrder>>();
        }
        else
        {
            // request failed
            string result = await
((StreamContent)response.Content).ReadAsStringAsync();
            Console.WriteLine(string.Concat("GetWorkOrdersByState
failed.\n", result));
        }

        return entities;
    }

```

The next example allows you to obtain information on the tasks of a Work Order.

```

    private static async Task<IEnumerable<WorkOrderPlanTasksInfo>>
GetWorkOrderPlanTasksInfos(string version, HttpClient client, string
accessToken)
    {
        IEnumerable<WorkOrderPlanTasksInfo> entities = null;

        // create request message
        HttpRequestMessage getEntityMessage = new
HttpRequestMessage(HttpMethod.Get, FormatApiUri(client) +
"/GetWorkOrderPlanTasksInfos?version="+version);

        // add authorization token header
        getEntityMessage.AddAuthorizationHeader(accessToken);

        // request api
        HttpResponseMessage response = await
client.SendAsync(getEntityMessage);
        if (response.IsSuccessStatusCode)
        {
            IEnumerable<WorkOrderPlanTasksInfo> responseEntities = new
List<WorkOrderPlanTasksInfo>();

```

```

        entities = await
response.Content.ReadAsAsync<IEnumerable<WorkOrderPlanTasksInfo>>();
    }
    else
    {
        // request failed
        string result = await
((StreamContent)response.Content).ReadAsStringAsync();
        Console.WriteLine(string.Concat("GetWorkOrderPlanTasksInfos
failed.\n", result));
    }

    return entities;
}

```

The following routine returns all the States associated with the Work Orders.

```

private static async Task<IEnumerable<SchemaEntityStateInfo>>
GetWorkOrderStates(string version, HttpClient client, string accessToken)
{
    IEnumerable<SchemaEntityStateInfo> entities = null;

    // create request message
    HttpRequestMessage getEntityMessage = new
HttpRequestMessage(HttpMethod.Get, FormatApiUri(client) +
"/GetWorkOrderStates?version="+version);

    // add authorization token header
    getEntityMessage.AddAuthorizationHeader(accessToken);

    // request api
    HttpResponseMessage response = await
client.SendAsync(getEntityMessage);
    if (response.IsSuccessStatusCode)
    {
        IEnumerable<SchemaEntityStateInfo> responseEntities = new
List<SchemaEntityStateInfo>();

        entities = await
response.Content.ReadAsAsync<IEnumerable<SchemaEntityStateInfo>>();
    }
    else
    {
        // request failed
        string result = await
((StreamContent)response.Content).ReadAsStringAsync();
        Console.WriteLine(string.Concat("GetWorkOrderStates failed.\n",
result));
    }

    return entities;
}

```

```
}
```

The following routine returns all State Transitions associated with the Work Orders.

```
private static async Task<IEnumerable<SchemaEntityStateTransitionInfo>>
GetWorkOrderStateTransitions(string version, HttpClient client, string
accessToken)
{
    IEnumerable<SchemaEntityStateTransitionInfo> entities = null;

    // create request message
    HttpRequestMessage getEntityMessage = new
HttpRequestMessage(HttpMethod.Get, FormatApiUri(client) +
"/GetWorkOrderStateTransitions?version="+version);

    // add authorization token header
    getEntityMessage.AddAuthorizationHeader(accessToken);

    // request api
    HttpResponseMessage response = await
client.SendAsync(getEntityMessage);
    if (response.IsSuccessStatusCode)
    {
        IEnumerable<SchemaEntityStateTransitionInfo> responseEntities =
new List<SchemaEntityStateTransitionInfo>();

        entities = await
response.Content.ReadAsAsync<IEnumerable<SchemaEntityStateTransitionInfo>>();
    }
    else
    {
        // request failed
        string result = await
((StreamContent)response.Content).ReadAsStringAsync();
        Console.WriteLine(string.Concat("GetWorkOrderStateTransitions
failed.\n", result));
    }

    return entities;
}
```

This routine allows you to change the state of a Work Order.

```
private static async Task ChangeWorkOrderState(WorkOrder entity, string
stateKey, HttpClient client, string accessToken)
{
    // create request message
    HttpRequestMessage postEntityMessage = new
HttpRequestMessage(HttpMethod.Put, FormatApiUri(client) +
"/ChangeWorkOrderState?workOrderKey=" + entity.WorkOrderKey + "&newStateKey=" +
stateKey + "&reason=NOP");

    // add authorization token header
```



```

        postEntityMessage.AddAuthorizationHeader(accessToken);

        // convert entity to json notation
        string entityRequest = entity.JsonSerialize();

        // add json to message body
        postEntityMessage.Content = new StringContent(entityRequest,
            Encoding.UTF8, "application/json");

        // send PUT to api for entity update
        using (HttpResponseMessage response = await
            client.SendAsync(postEntityMessage))
        {
            if (response.IsSuccessStatusCode)
            {
                // success: entity updated.
                Uri resourceUri = response.Headers.Location;
            }
            else
            {
                // request failed
                string result = await
                    ((StreamContent)response.Content).ReadAsStringAsync();
                Console.WriteLine(string.Concat("PUT WorkOrder failed.\n",
                    result));

                throw new Exception("Unable to update WorkOrder state.");
            }
        }
    }
}

```

To remove a Work Order, see the following example.

```

    private static async Task DeleteWorkOrder(WorkOrder entity, HttpClient client,
        string accessToken)
    {
        // create request message
        HttpRequestMessage postEntityMessage = new
            HttpRequestMessage(HttpMethod.Delete, FormatApiUri(client) + "?workOrderKey=" +
                entity.WorkOrderKey);

        // add authorization token header
        postEntityMessage.AddAuthorizationHeader(accessToken);

        // send DELETE to api for entity deletion
        using (HttpResponseMessage response = await
            client.SendAsync(postEntityMessage))
        {
            if (response.IsSuccessStatusCode)
            {
                // success: entity deleted.
            }
        }
    }
}

```

```

        Uri resourceUri = response.Headers.Location;
    }
    else
    {
        // request failed
        string result = await
            ((StreamContent)response.Content).ReadAsStringAsync();
        Console.WriteLine(string.Concat("DELETE WorkOrder
failed.\n", result));

        throw new Exception("Unable to delete WorkOrder.");
    }
}
}
}

```

Tasks Assignment

The following classes will be used in this section's routines:

```

/// <summary>
/// The WorkOrderTaskAssignment class.
/// </summary>
public class WorkOrderTaskAssignment
{
    #region Public Properties

    /// <summary>
    /// Assignment of the WO Task
    /// </summary>
    /// <remarks>
    /// This property belongs to the entity natural key.
    /// This property value is required.
    /// </remarks>
    public Guid WorkOrderTaskAssignmentKey { get; set; }

    /// <summary>
    /// Task of the Work Order Plan
    /// </summary>
    public string WorkOrderPlanTask { get; set; }

    /// <summary>
    /// Specialty
    /// </summary>
    public string Expertise { get; set; }

    /// <summary>
    /// Employee
    /// </summary>
    public string Employee { get; set; }
}

```

```
#endregion
```

```
}
```

To assign a specific Task of a Work Order to an Employee, see the following example:

```
private static async Task<WorkOrderExecution> AssignWorkOrderTask(string
workOrderKey, string maintenancePlanKey, string taskKey, string expertiseKey,
string employeeKey, HttpClient client, string accessToken)
{
    WorkOrderExecution entity;

    // create request message
    HttpRequestMessage postEntityMessage = new
HttpRequestMessage(HttpMethod.Put, FormatApiUri(client) +
"/AssignWorkOrderTask?workOrderKey=" + workOrderKey + "&maintenancePlanKey=" +
maintenancePlanKey + "&taskKey=" + taskKey + "&expertiseKey=" + expertiseKey +
"&employeeKey=" + employeeKey);

    // add authorization token header
    postEntityMessage.AddAuthorizationHeader(accessToken);

    // send PUT to api for entity update
    using (HttpResponseMessage response = await
client.SendAsync(postEntityMessage))
    {
        if (response.IsSuccessStatusCode)
        {
            // success: entity updated
            entity = await
response.Content.ReadAsAsync<WorkOrderExecution>();
        }
        else
        {
            // request failed
            string result = await
((StreamContent)response.Content).ReadAsStringAsync();
            Console.WriteLine(string.Concat("PUT AssignWorkOrderTask
failed.\n", result));

            throw new Exception("Unable to assign the WorkOrder task");
        }
    }

    return entity;
}
```

Tasks Execution

The following classes will be used in this section's routines:

```

/// <summary>
    /// The WorkOrderExecution class.
    /// </summary>
    public class WorkOrderExecution
    {
        #region Public Properties

        /// <summary>
        /// Work Order
        /// </summary>
        /// <remarks>
        /// This property belongs to the entity natural key.
        /// This property value is required.
        /// </remarks>
        public string WorkOrder { get; set; }

        /// <summary>
        /// Executions of the WO Task
        /// </summary>
        public ICollection<WorkOrderTaskExecution> WorkOrderTaskExecutions {
get; set; }

        #endregion
    }

    /// <summary>
    /// The WorkOrderTaskExecution class.
    /// </summary>
    public class WorkOrderTaskExecution
    {
        #region Public Properties

        /// <summary>
        /// Is Started
        /// </summary>
        public bool? IsStarted { get; set; }

        /// <summary>
        /// Start Date
        /// </summary>
        public DateTime? StartDate { get; set; }

        /// <summary>
        /// Is Completed
        /// </summary>
        public bool? IsCompleted { get; set; }

        /// <summary>

```

```

    /// End Date
    /// </summary>
    public DateTime? EndDate { get; set; }

    /// <summary>
    /// Location
    /// </summary>
    public string Location { get; set; }

    /// <summary>
    /// Asset
    /// </summary>
    public string Asset { get; set; }

    /// <summary>
    /// Task of the Work Order Plan
    /// </summary>
    /// <remarks>
    /// This property value is required.
    /// </remarks>
    public string WorkOrderPlanTask { get; set; }

    /// <summary>
    /// Employee
    /// </summary>
    /// <remarks>
    /// This property value is required.
    /// </remarks>
    public string Employee { get; set; }

    /// <summary>
    /// Work Order
    /// </summary>
    public string WorkOrder { get; set; }

    #endregion
}

```

The following routine starts a specific Task of a Work Order.

```

/// <summary>
/// Starts the work order task.
/// </summary>
/// <param name="workOrderKey">The work order key.</param>
/// <param name="maintenancePlanKey">The maintenance plan key.</param>
/// <param name="taskKey">The task key.</param>
/// <param name="employeeKey">The employee key.</param>
/// <param name="locationKey">The location key.</param>
/// <param name="assetKey">The asset key.</param>

```

```

        /// <param name="startDate">The start date.</param>
        /// <param name="client">The client.</param>
        /// <param name="accessToken">The access token.</param>
        /// <returns></returns>
        /// <exception cref="System.Exception">Unable to start the WorkOrder
task</exception>

        private static async Task<WorkOrderExecution> StartWorkOrderTask(string
workOrderKey, string maintenancePlanKey, string taskKey, string employeeKey,
string locationKey, string assetKey, string startDate, HttpClient client, string
accessToken)
        {
            WorkOrderExecution entity;

            // create request message
            HttpRequestMessage postEntityMessage = new
HttpRequestMessage(HttpMethod.Put, FormatApiUri(client) +
"/StartWorkOrderTask?workOrderKey=" + workOrderKey + "&maintenancePlanKey=" +
maintenancePlanKey + "&taskKey=" + taskKey + "&employeeKey=" + employeeKey +
"&locationKey=" + locationKey + "&assetKey=" + assetKey + "&startDate=" +
startDate);

            // add authorization token header
            postEntityMessage.AddAuthorizationHeader(accessToken);

            // send PUT to api for entity update
            using (HttpResponseMessage response = await
client.SendAsync(postEntityMessage))
            {
                if (response.IsSuccessStatusCode)
                {
                    // success: entity updated
                    entity = await
response.Content.ReadAsAsync<WorkOrderExecution>();
                }
                else
                {
                    // request failed
                    string result = await
((StreamContent)response.Content).ReadAsStringAsync();
                    Console.WriteLine(string.Concat("PUT StartWorkOrderTask
failed.\n", result));

                    throw new Exception("Unable to start the WorkOrder task");
                }

                return entity;
            }
        }
    }

```

To complete a specific Task, see the following example:

```

    /// <summary>
        /// Completes the work order task.
    /// </summary>
    /// <param name="workOrderKey">The work order key.</param>
    /// <param name="maintenancePlanKey">The maintenance plan key.</param>
    /// <param name="taskKey">The task key.</param>
    /// <param name="employeeKey">The employee key.</param>
    /// <param name="locationKey">The location key.</param>
    /// <param name="assetKey">The asset key.</param>
    /// <param name="endDate">The end date.</param>
    /// <param name="client">The client.</param>
    /// <param name="accessToken">The access token.</param>
    /// <returns></returns>
    /// <exception cref="System.Exception">Unable to end the WorkOrder
task</exception>
    private static async Task<WorkOrderExecution>
CompleteWorkOrderTask(string workOrderKey, string maintenancePlanKey, string
taskKey, string employeeKey, string locationKey, string assetKey, string
endDate, HttpClient client, string accessToken)
    {
        WorkOrderExecution entity;

        // create request message
        HttpRequestMessage postEntityMessage = new
HttpRequestMessage(HttpMethod.Put, FormatApiUri(client) +
"/CompleteWorkOrderTask?workOrderKey=" + workOrderKey + "&maintenancePlanKey=" +
maintenancePlanKey + "&taskKey=" + taskKey + "&employeeKey=" + employeeKey +
"&locationKey=" + locationKey + "&assetKey=" + assetKey + "&endDate=" +
endDate);

        // add authorization token header
        postEntityMessage.AddAuthorizationHeader(accessToken);

        // send PUT to api for entity update
        using (HttpResponseMessage response = await
client.SendAsync(postEntityMessage))
        {
            if (response.IsSuccessStatusCode)
            {
                // success: entity updated.
                entity = await
response.Content.ReadAsAsync<WorkOrderExecution>();
            }
            else
            {
                // request failed
                string result = await
((StreamContent)response.Content).ReadAsStringAsync();
                Console.WriteLine(string.Concat("PUT CompleteWorkOrderTask
failed.\n", result));
            }
        }
    }
}

```

```

        throw new Exception("Unable to end the WorkOrder task");
    }

    return entity;
}

}

```

Labor Report

The following classes will be used in this section's routines:

```

/// <summary>
/// The WorkOrderReportLabor class.
/// </summary>
public class WorkOrderReportLabor
{
    #region Public Properties

    /// <summary>
    /// Document sequence.
    /// </summary>
    /// <remarks>
    /// This property belongs to the entity natural
key.

    /// This property value is required.
    /// </remarks>
    public string SequenceKey { get; set; }

    /// <summary>
    /// The labor report description.
    /// </summary>
    /// <remarks>
    /// This property corresponds to the entity
description.

    /// </remarks>
    public string Description { get; set; }

    /// <summary>
    /// Date
    /// </summary>
    /// <remarks>
    /// This property value is required.
    /// </remarks>
    public DateTime ReportDate { get; set; }

    /// <summary>
    /// Is Printed
    /// </summary>

```



```

        public bool? IsPrinted { get; set; }

        /// <summary>
        /// Exchange Rate
        /// </summary>
        /// <remarks>
        /// This property value is required.
        /// </remarks>
        public decimal ExchangeRate { get; set; }

        /// <summary>
        /// Exchange Rate Date
        /// </summary>
        public DateTime? ExchangeRateDate { get; set; }

        /// <summary>
        /// Work Order
        /// </summary>
        /// <remarks>
        /// This property value is required.
        /// </remarks>
        public string WorkOrder { get; set; }

        /// <summary>
        /// Location
        /// </summary>
        public string Location { get; set; }

        /// <summary>
        /// Asset
        /// </summary>
        public string Asset { get; set; }

        /// <summary>
        /// Report Labor
        /// </summary>
        public ICollection<WorkOrderTaskReportLabor>
WorkOrderTaskReportLabors { get; set; }

        /// <summary>
        /// Currency
        /// </summary>
        /// <remarks>
        /// This property value is required.
        /// </remarks>
        public string Currency { get; set; }

        #endregion

```

```

    }

    /// <summary>
    /// The WorkOrderTaskReportLabor class.
    /// </summary>
    public class WorkOrderTaskReportLabor
    {
        #region Public Properties

        /// <summary>
        /// Start Date
        /// </summary>
        /// <remarks>
        /// This property value is required.
        /// </remarks>
        public DateTime StartDate { get; set; }

        /// <summary>
        /// End Date
        /// </summary>
        /// <remarks>
        /// This property value is required.
        /// </remarks>
        public DateTime EndDate { get; set; }

        /// <summary>
        /// Hours
        /// </summary>
        /// <remarks>
        /// This property value is required.
        /// </remarks>
        public decimal Hours { get; set; }

        /// <summary>
        /// Total
        /// </summary>
        /// <remarks>
        /// This property value is required.
        /// </remarks>
        /// <summary>
        public decimal? Total { get; set; }

        /// <summary>
        /// Remarks
        /// </summary>
        public string Remarks { get; set; }

        /// <summary>
        /// Task of the Work Order Plan

```

```

        /// </summary>
        public string WorkOrderPlanTask { get; set; }

        /// <summary>
        /// Maintenance Plan
        /// </summary>
        public string MaintenancePlan { get; set; }

        /// <summary>
        /// Employee
        /// </summary>
        public string Employee { get; set; }

        /// <summary>
        /// Specialty
        /// </summary>
        public string Expertise { get; set; }

        /// <summary>
        /// Skill
        /// </summary>
        public string Skill { get; set; }

        /// <summary>
        /// Task
        /// </summary>
        public string Task { get; set; }

        #endregion
    }

```

To report the consumption of Labor of an Employee in a specific Task of a Work Order, see the following example:

```

        private static async Task<WorkOrderReportLabor>
ReportLaborForWorkOrderTask(string workOrderKey, string maintenancePlanKey,
string taskKey, string employeeKey, HttpClient client, string accessToken)
        {
            WorkOrderReportLabor entity;

            // create request message
            HttpRequestMessage postEntityMessage = new
HttpRequestMessage(HttpMethod.Put, FormatApiUri(client) +
"/ReportLaborForWorkOrderTask?workOrderKey=" + workOrderKey +
"&maintenancePlanKey=" + maintenancePlanKey + "&taskKey=" + taskKey +
"&employeeKey=" + employeeKey);

            // add authorization token header
            postEntityMessage.AddAuthorizationHeader(accessToken);

```

```

        // convert entity to json notation
        //string entityRequest = entity.JsonSerialize();

        // add json to message body
        //postEntityMessage.Content = new StringContent(entityRequest,
        Encoding.UTF8, "application/json");

        // send PUT to api for entity update
        using (HttpResponseMessage response = await
        client.SendAsync(postEntityMessage))
        {
            if (response.IsSuccessStatusCode)
            {
                // success: entity updated.

                entity = await
                response.Content.ReadAsAsync<WorkOrderReportLabor>();
            }
            else
            {
                // request failed
                string result = await
                ((StreamContent)response.Content).ReadAsStringAsync();
                Console.WriteLine(string.Concat("PUT WorkOrderReportLabor
                failed.\n", result));

                throw new Exception("Unable to update WorkOrderReportLabor
                state.");
            }
            return entity;
        }
    }
}

```

Entity Attachment

```

/// <summary>
/// Uploads the attachment.
/// </summary>
/// <param name="apiUriFormat">The API URI format.</param>
/// <param name="id">The entity identifier.</param>
/// <param name="fileName">Name of the file.</param>
/// <param name="attachmentDescription">The attachment
description.</param>
/// <param name="attachmentNotes">The attachment notes.</param>
/// <param name="client">The client.</param>
/// <param name="accessToken">The access token.</param>
/// <param name="tenantKey">The tenant key.</param>

```

```

    /// <param name="organizationKey">The organization key.</param>
    /// <returns>
    /// The attachment resource.
    /// </returns>
    /// <exception cref="Exception">UploadAttachment failed.</exception>
    /// <exception cref="System.Exception">UploadAttachment
failed.</exception>
    public static async Task<Attachment> UploadAttachment(string
apiUriFormat, string id, string fileName, string attachmentDescription,
string attachmentNotes, HttpClient client, string accessToken, string
tenantKey, string organizationKey)
    {
        var baseUrl = TestHelper.FormatApiUri(client, apiUriFormat,
tenantKey, organizationKey);

        Attachment responseAttachment = new Attachment();

        // create request message
        HttpRequestMessage postAttachmentMessage = new
HttpRequestMessage(HttpMethod.Post, baseUrl + "/" + id + "/attachments");

        var fileContent = new FileStream(fileName, FileMode.Open);

        var content = new MultipartFormDataContent();

        content.Add(new StringContent(attachmentDescription),
"\description\");
        content.Add(new StringContent(attachmentNotes), "\notes\");

        var streamContent = new StreamContent(fileContent);
        streamContent.Headers.TryAddWithoutValidation("Content-Type",
MimeMapping.GetMimeMapping(fileName));

        content.Add(streamContent, "\file\"); string.Format("\{0}\");
Path.GetFileName(fileName));

        // add authorization token header
        postAttachmentMessage.Content = content;
        postAttachmentMessage.AddAuthorizationHeader(accessToken);
        postAttachmentMessage.Headers.TryAddWithoutValidation("Accept",
"/");

        // request api
        HttpResponseMessage response = await
client.SendAsync(postAttachmentMessage);

        if (response.IsSuccessStatusCode)
        {
            responseAttachment.Id = Guid.Empty;
            responseAttachment.FileName = fileName;
            responseAttachment.Description = attachmentDescription;
            responseAttachment.Notes = attachmentNotes;
        }
        else
        {
            // request failed

```

```
        string result = await
        ((StreamContent)response.Content).ReadAsStringAsync();
        Console.WriteLine(string.Concat("UploadAttachment failed.\n",
result));
        throw new Exception("UploadAttachment failed.");
    }

    return responseAttachment;
}
```