

Version 25



Trapeze

Enterprise Asset Management

API

Reference Guide

Trade Secret | February 2026

© 2025 Trapeze Software ULC, its subsidiaries and affiliates (collectively “TSU”). All rights reserved.

Trapeze Software ULC (“TSU”) Proprietary and Confidential: Information contained in this document is proprietary to TSU and its subsidiaries and may be used or disclosed only with written permission from TSU. This guide, or any part thereof, may not be reproduced without the prior written permission of TSU. The recipient acknowledges and agrees that disclosure of this document to recipient, and its use, are subject to the terms and conditions specified in their relevant software license agreement, software maintenance agreement, and/or non-disclosure agreement (“Governing Agreement”) under which this document was disclosed. This document is for internal use only in conjunction with TSU products. This document may not be modified in any way.

All other trademarks, registered trademarks, product names and company names or logos mentioned are the property of their respective owners.

Except as may be provided in the Governing Agreement, TSU and its affiliates, subsidiaries, officers, directors, employees, and agents provide the information contained in this manual on an “as-is” basis and do not make any express or implied warranties or representations with respect to such information including, without limitation, warranties as to non-infringement, reliability, fitness for a particular purpose, usefulness, completeness, accuracy or currentness. TSU shall not in any circumstances be liable to any person for any special, incidental, indirect or consequential damages, including without limitation, damages resulting from use of or reliance on information presented herein, or loss of profits or revenues or costs of replacement goods, even if informed in advance of the possibility of such damages.

TSU reserves the right, to be exercised at its sole discretion and without notice, to change, modify or adapt the contents of this edition in order to accurately reflect any future upgrades to the TSU proprietary software and/or hardware. In the event of such changes, TSU expects, but does not represent or guarantee, that this current edition will continue to remain reasonably accurate insofar as it describes the basic functions of the TSU proprietary software and/or hardware. Furthermore, based on your system settings, certain functionality, such as application screens, may not function exactly as shown or described in this guide.

Technical Support

Trapeze provides several ways to connect with the Customer Care team. Please provide detailed information to the representative. If you are reporting an issue by e-mail, include any error messages, screen shots of your problem, and steps to replicate the issue.

Customer Care is available Monday through Friday, 8:00 a.m. to 8:00 p.m., Eastern Standard Time.

Toll-Free Telephone: 1-877-411-8727

E-mail: cc@trapezegroup.com

FTP Site: <https://ftps.trapezegroup.com/>

Web Site: <https://collaborate.trapezegroup.com>

The website provides access to Trapeze Collaborate which offers resources to implement and operate the software, search knowledge, participate in communities, log/track service requests, sign-up for release alerts, view product health recommendations, and download updates or documentation. For secure access to Collaborate, contact Customer Care through the website or by calling the number above.

Product Compatibility

Refer to the latest [Product Compatibility](#) information on Trapeze Collaborate for requirements including operating system information, databases, Crystal reports, network protocol, HTTP servers and browsers that are supported.

API – Reference Guide

Version 25.x

February 2026

Contents

1. Overview.....	2
Installation or Upgrade	2
License	2
2. Auto Generated API Documents and Test Page	4
Authentication and Sample Request.....	4
Code Sample	6
Query Parameters & Filters	7
Common Query Parameters.....	7
Filtering.....	7
Paging	8
Count Records.....	8
Application Security and Screen Rights	8
Web Service security	8
Application security	9
3. Details by API Package	10
Assets.....	10
Notes.....	12
Core Organization	12
Fuel and Fluids.....	13
Incident Management.....	14
Operations.....	15
Consist Management	16
Mobile	16
Parts Inventory	16
Work Management.....	18

1. Overview

Trapeze EAM offers a published REST API for communicating with Trapeze EAM business objects. This API allows agencies to build their own integrations and tools leveraging Trapeze EAM business logic in a supported manner.

REST APIs are a collection of Endpoints that can be called by another program. The types of methods commonly made available include:

- GET (retrieve data)
- PUT (update existing data)
- POST (insert or create new data)
- DELETE (delete existing data)

The communication protocol is either HTTP or HTTPS. Authentication generates a token for a limited time. A valid Trapeze EAM user ID and password is required for all transactions. The defined user must have appropriate rights to the functionality that the API is calling for the transaction to be successful.

Installation or Upgrade

API requires installation and configuration of Data Services.

Data Services is configured separately from Web Modules.

If any configuration changes have been made to Web Modules such as database connection, application server connection, API host or port, or for any settings that are shared between the API and Web Modules, then you must also make the same changes to the API service.

 Go to the folder where the API Service runs and launch Configurator.exe to make the changes.

License

A separate license is required to use API. Currently, all API endpoints are split into six licensable modules based on functionality:

- Assets
- Core Organization
- Incident Management

- Parts Inventory
- Purchasing
- Work Management

Each module requires a separate license key. Licenses can be configured through the License Keys screen in Enterprise Portal using Module ID: 156.

The screenshot displays the 'License Keys' interface in 'ADMIN MODE'. At the top, there are buttons for Search, Reload, and Sort, along with dropdown menus for 'Select Filter' and 'Select Sort'. To the right are buttons for New, Copy, Delete, and Edit. Below these is a table with columns: Row #, License ID, and License Key. The table contains one row with Row # 1 and License ID 156. Below the table, the 'Basic Info' tab is selected, showing a 'License key' input field.

Row #	License ID	License Key
1	156	

Basic Info

License key

2. Auto Generated API Documents and Test Page

The API documents and test page contains a list of all currently available endpoints with the ability to test them. For a valid product install running Data Services, the documentation and test web page is accessible at:

`http[s]://[machine name]:[INTEGRATION services url port]/docs/api/index`

Data Services API

Getting Started

STEP 1: AUTHENTICATE VIA LOGIN

To generate a token, go to the login fields on the main page.

Enter your username and password. If a token is not already active, the system will call Data Services which will return a token and allow you to login and continue.

STEP 2: MAKE YOUR FIRST API REQUEST

You can test API calls using the "Try It Out" feature. Navigate to an API endpoint from the list below and click "Try It Out". If a token is not currently active, you will be prompted to enter your user name and password to generate the token.

Account Show/Hide List Operations Expand Operations

DELETE	/api/v1/Accounts
GET	/api/v1/Accounts
POST	/api/v1/Accounts
PUT	/api/v1/Accounts
GET	/api/v1/AccountTasks
DELETE	/api/v1/Accounts/{AccountId}

Authentication and Sample Request

Authenticating with the Data Services API will return an access bearer token that expires after 30 days, at which point the client will be required to request another bearer token from the API.

The following example can be used when authenticating against API.

1. Post to:

`http[s]://[machine name]:[Integration Services url port]/api/v1/token`

Authorization:

Type: Basic Auth

Username: api-user

Password: api-user-secret

Headers:

Content-Type: application/json
Body:
Empty

2. A basic authorization token will be returned in the header:

Header:
Authorization: Basic bW9iaWxlLXVzZXI6bW9iaWxlLXVzZXItc2VjcmV0

3. Post to:

http[s]://[machine name]:[Integration Services url port]/connect/token/

Authorization:

Type: No Auth

Headers:

Content-Type: application/x-www-form-urlencoded

Authorization: Basic bW9iaWxlLXVzZXI6bW9iaWxlLXVzZXItc2VjcmV0
(authorization token returned in step 2)

Body:

Format: x-www-form-urlencoded

Keys/Values:

grant type: password

username: EAM_USER

password: EAM_USER_PASSWORD

scope: api

4. A JSON with EAM access bearer token will be returned:

```
"access_token":"9b3eeb60eb3f8c130ad78e28b206506c",  
"expires_in":2592000,  
"token_type":"Bearer"
```

The “expires_in” token returns how many seconds until the access token will expire.

5. After acquiring the access token, use it to call one of the endpoints. Refer to the auto generated documents for available endpoints and sample input/output. For example:

GET

http[s]://[machine name]:[Data Services API port]
port]/api/v1/PartsCrossReferences

or http[s]://[machine name]:[INTEGRATION service]
port]/api/v1/PartsCrossReferences?CrossReferencePartID=Sensor1&CrossRefere
nceManufacturerID=Bosch

Authorization:

Type: No Auth

Headers:

Content-Type: application/json

Authorization: Bearer 9b3eeb60eb3f8c130ad78e28b206506c (bearer access token acquired step 4)

Code Sample

```
//Sample Auth using IdentityModel library
using (var httpClient = new HttpClient())
using (var req = new PasswordTokenRequest
{
    Address = IDENTITY_SERVICE_URL + "/connect/token", // Integration Services URL
    ClientId = "api-user",
    ClientSecret = "api-user-secret",
    UserName = "test",
    Password = "test",
    Scope = "api"
})
{
    // Attempt to get a user token
    TokenResponse token = await httpClient.RequestPasswordTokenAsync(req);

    if (token.HttpStatusCode == HttpStatusCode.OK && token.AccessToken != null)
    {
        //Set Bearer token on http client that will be used to make requests
        Client.SetBearerToken(token.AccessToken);
    }
    else
        //handle error while retrieving token
}

//Sample request
using (var reqMsg = new HttpRequestMessage
{
    RequestUri = new Uri(GetUrl(req.Endpoint)), // Format request URL
    Method = req.Method, //POST/GET/PUT
    Content = req.BodyString.NotWS() ? new StringContent(req.BodyString,
System.Text.Encoding.UTF8, "application/json") : null // JSON body
})
{
```

```
using (var rawResponse = await Client.SendAsync(reqMsg,
HttpCompletionOption.ResponseHeadersRead, req.TokenSource == null ? token :
req.TokenSource.Token)) //send request passing in cancellation token as needed
{
    //response code here
}
}
```

Query Parameters & Filters

Common Query Parameters

Query Parameters	
\$countrecords	Returns a count of records
\$exclude	Exclude specific columns
\$filter	Complex filtering with logical operators
\$nested	Nested property access; set to false to only include base data
\$orderby	Sorting
\$select	Specify columns to be returned
\$skip	Pagination offset
\$top	Limit results (max 250)

Filtering

Any request can include URI filtering. In the following example, parts are filtered from PartsCrossReferences endpoint to only contain data where CrossReferencePartID=Sensor1 and CrossReferenceManufacturerID=Bosch:

```
http[s]://[machine name]:[INTEGRATION service url
port]/api/v1/PartsCrossReferences?CrossReferencePartID=Sensor1&CrossReferenceManu
facturerID=Bosch
```

DateTimes and Date Filtering (API v1)

API Requests expect datetimes in UTC time.

API Responses return datetimes in UTC time.

API Requests that filter date/times, the filter expects the datetime to be a single string (with no spaces). For example, "2024-02-16T18:40:25" or "2024/02/16" or "2024-02-16", etc.

\$filter flag can be used to create date range filters. Only two dates can be specified to perform a date range filter. For example:

- Looking for records where date is greater than: parameter: \$filter value: "DateTime gt 2024/01/28"
- Looking for records where date/time is less than or equal: parameter: \$filter value: "DateTime le 2024/01/28T18:40:25"
- Looking for records between date/time values: parameter: \$filter value: "DateTime ge 2024/01/28T18:40:25 and DateTime le 2024/02/28T19:01:22"1
- Supported operators
 - eq → equal to
 - ge → greater than or equal to
 - gt → greater than
 - le → less than or equal to
 - lt → less than

Paging

\$skip can be used to skip pages of data if you use paging in your requests. In the following example, \$skip will skip the first page of data and return you to the next page.

```
http[s]://[machine name]:[INTEGRATION service url  
port]/api/v1/PartsCrossReferences?$skip=1
```

Count Records

To count records in API, users need to use \$countrecords = true instead of the \$countrecords parameter.

Application Security and Screen Rights

Web Service security

IdentityServer library is used for authentication and authorization. IdentityServer is an officially certified implementation of OpenID Connect.

Application security

Currently, all application server security and validation rules are followed during POST and PUT requests. All GET request are processed directly through the database and therefore none of the application server security rules apply (standard web service security procedures are still followed). For this reason, access to certain GET endpoints that contain sensitive information has been disabled.

3. Details by API Package

Trapeze EAM APIs are grouped into functional area packages which can be subscribed to individually. Once four packages are subscribed to, the subscription moves to an “enterprise” level subscription which includes all current and future Trapeze EAM API packages.

Assets

Assets - API Package Details	
AssetCategories	GET, POST, PUT, DELETE
Assets	GET, POST
AssetsLinear	GET
AssetFiles	GET, PUT, POST, DELETE
AssetAdditionalInfo	GET
AssetTemplates	GET
AssetIndividualPMs	GET
DateNextDue	
FuelOverride	
FuelUsedSinceLastPM	
LastCommercialUniqueID	
LastPMVendorID	
LifeMeter1LastPerformed	
LifeMeter1LastStart	
LifeMeter2LastPerformed	
LifeMeter2LastStart	
Meter1Override	
Meter2Override	
NumberOfTimeUnits	
PMService	
TimeUnit	
AssetModel	GET, PUT, POST
AssetTypes	GET
Asset/Warranties	GET, POST, PUT, DELETE
Attributes	GET, POST, PUT
AttributeAssetCategories	GET, POST, PUT

Assets - API Package Details	
ConditionRatings	GET
ConditionRatingAssetCategories	GET
EquipmentAssignmentHistory	GET
EquipmentClasses	GET
EquipmentRentalRates	GET
EquipmentComponents	GET
ComponentAssignments	GET
EquipmentConditionRatings	GET
EquipmentID	POST, PUT, DELETE
IndividualIPMs	
EquipmentManufacturers	GET, PUT, POST, DELETE
EquipmentManufacturerModels	GET
EquipmentMeterReadings	GET, POST, PUT
EquipmentMeterReadings/Latest	GET
EquipmentSubsystems	GET
EquipmentSubsystemDetails	GET, PUT, POST, DELETE
EquipmentTypes	GET, PUT, POST, DELETE
EquipmentTypeAttributes	GET
EquipmentTypeAttributesMain	GET, PUT, POST
EquipmentUsage	GET, POST
EquipmentUsageMeterReadings	POST
EquipmentUsageTripUsage	POST
LifeCycleStatus	GET
Lines	GET, PUT, POST, DELETE
Markers	GET, POST, PUT, DELETE
Segments	GET, POST, PUT, DELETE
Subsystems	GET
SubsystemProperties	GET

Notes

When retrieving coordinates for point (non-linear) assets from the API, coordinates will be looked up in the following order, stopping as soon as some coordinates are found. If no coordinates are found, then the asset will not be drawn on the map.

1. Last known position of the asset, whether from AVL or manual latitude and longitude entry or from dragging the asset's icon on the map.
2. Coordinates of the asset's Station Location (from Asset Primary > Locations tab > Station Location)
3. Coordinates of the asset's Stored Location (from Asset Primary > Locations tab > Stored Location)

Core Organization

Core Organization - API Package Details	
Account	
Accounts	GET, PUT, POST, DELETE
AccountTasks	GET
Addresses	GET, PUT, POST, DELETE
DepartmentAttribute	GET, PUT, POST
DepartmentAttributeMain	GET, PUT, POST, DELETE
Departments	GET, PUT, POST
Employees	GET
EmployeeLocations	GET
EmployeeClassifications	GET
EmployeeClassificationMain	GET, PUT, POST, DELETE
Employess.ClockIn.StartStop	PUT
Locations	GET
LocationRollups	GET
LocationAttributesMain	GET, PUT, POST, DELETE
LocationAttributes	GET, POST, PUT, DELETE
MAXQueueStaging	GET, PUT, POST
Messages	GET, PUT

Core Organization - API Package Details	
Operators	GET
OperatorEmployees	GET
Operator Departments	GET
Users	GET, POST, PUT
Users.UserID	GET, DELETE
UserGroups	GET, PUT, POST, DELETE
UserUserGroups	GET
Util	GET
ValidateUser	POST
Vendors	GET
VendorContracts	GET
VendorContractLines	GET
VendorContractFiles	GET
VendorContractNotes	GET
VendorLocations	GET

Fuel and Fluids

Fuel and Fluids – API Package Details	
FluidProductTypes	GET
FuelFluidTanks	GET
FuelingSites	GET
FuelPumps	GET
FuelTypes	GET
InternalFuelTickets	GET, POST

Incident Management

Incident Management - Screen Data

IncidentManagement	
Incidents	GET, POST, PUT, DELETE
IncidentFiles	GET, POST, PUT, DELETE
IncidentNotification	GET, POST, PUT, DELETE
IncidentNotes	GET
IncidentInjuries	GET, POST, DELETE
IncidentManagementAttributes	GET, POST, DELETE
Incident Equipment	GET, POST, PUT, DELETE
Incident Procedures	GET, POST, PUT, DELETE

Incident Management - Setup Screen

IncidentManagementSetup	
IncidentManagementSetupAttributes	GET, POST, PUT, DELETE
IncidentCauses	GET, POST, PUT, DELETE
IncidentContactTypes	GET, POST, PUT, DELETE
IncidentDelayTypes	GET, POST, PUT, DELETE
IncidentDelayCauses	GET
IncidentInjuryTypes	GET, POST, PUT, DELETE
IncidentPriorities	GET, POST, PUT, DELETE
IncidentReportingClassifications	GET, POST, PUT, DELETE
IncidentResponsibilityTypes	GET, POST, PUT, DELETE
IncidentStatuses	GET, POST, PUT, DELETE
IncidentTripPortions	GET, POST, PUT, DELETE
IncidentWeather	GET, POST, PUT, DELETE

Operations

Operations - Setup Screen	
Operations Crew Runs	GET, POST, PUT, DELETE
Operations Holidays	GET, POST, PUT, DELETE
Operations Holidays Dates	GET, POST, PUT, DELETE
Operations Peak Types	GET, POST, PUT, DELETE
Operations Routes	GET, POST, PUT, DELETE
Operations Routes / Operations Route Stops	GET, POST, PUT, DELETE
Operations Schedule Types	GET, POST, PUT, DELETE
Operations Setup Options	GET
Operations Stops	GET, POST, PUT, DELETE

Operations – Passenger Counts Screen	
Operations Passenger Counts	GET
Operations Trip Counts	GET

Operations – Schedules Screen	
Operations Schedules	GET, POST, PUT, DELETE
Operations Schedules Crew Runs	GET, POST, PUT, DELETE
Operations Schedules Stops	GET, POST, PUT, DELETE

Operations – Equipment Cycles Screen	
Equipment Cycles	GET, POST, PUT, DELETE
Equipment Cycles Schedules	GET, POST, PUT, DELETE

Operations – Trips Screen

Operations Trips	GET, POST, PUT, DELETE
Operations Trips Consist Equipment	GET, POST, PUT, DELETE
Operations Trips Crew	GET, POST, PUT, DELETE
Operations Trips Stops	GET, PUT

Consist Management**Consist Management Screen**

Consist Management	GET, POST, PUT, DELETE
Consist Management Equipment	GET, POST, PUT, DELETE

Mobile**Mobile-Related Endpoints**

Card Layouts	
CardLayouts	GET
CardLayoutAssetCategories	GET
CardLayoutFields	GET

Parts Inventory**Parts Inventory - API Package Details**

Allocation and Assignment	GET
Bins	GET, PUT, POST, DELETE
DockReceipts	GET, PUT, POST, DELETE
DockReceiptsLines	GET, PUT, POST, DELETE
DockReceiptFiles	GET, PUT, POST, DELETE
EnterprisePurchaseOrders	GET, PUT, POST, DELETE
EnterprisePurchaseOrderLines	GET, PUT, POST, DELETE
EnterprisePurchasingStatuses	GET, PUT, POST, DELETE

Parts Inventory - API Package Details	
EnterprisePurchaseOptions	GET
StockTransfers	GET, PUT, POST, DELETE
StockTransferLines	GET, PUT, POST, DELETE
EnterprisePurchaseReceipts	GET, PUT, POST, DELETE
EnterprisePurchaseReceiptSerials	GET, PUT, POST
EPReceiptWorkOrders	GET
EnterpriPurchaseWorkOrders	GET
InventoryCount	GET, PUT, POST
InventoryCountLines	GET
InventoryCountNotes	GET, POST
PartsLocation	GET
PartManufacturers	GET
PartRequests	GET, POST
PartRequestLines	GET, PUT, POST
Parts	GET, POST
PartClassifications	GET, PUT, POST
PartsCrossReferences	GET *Does not require separate license key when already licensed for parts catalog web module. If licensed for Parts Catalog web module, the part-catalog scope can be used with the following credentials to access the PartsCrossReferences endpoint: User: part-catalog-user Password: part-catalog-secret
PartsLocationInformation	GET
PartsSerial	GET
Printing	
PrintedLabels	GET
PrintedMessages	GET

Parts Inventory - API Package Details	
PutAwayCards	GET, POST
PutAwayCardSerials	GET
PutAwayCardID	PUT
PutAwayCardID/SerialNumbers	GET, POST, DELETE
PutAwayCardID/Movements	GET, POST
UnitsofMeasure	GET

Work Management

Work Management- API Package Details	
ClassTasks	GET
ClassTaskPmTasks	GET
ClassPmServiceDetails	GET
Class IndividualPmPrograms	GET
ClassTaskParts	GET
ClassTaskTests	GET
ClassTaskModel	GET
ClassTaskWacModelAdapter	GET
Commercial Postings	GET, POST
Crews	GET, PUT, POST, DELETE
CrewEmployees	GET
EmployeeLabor	GET, PUT
EmployeeModel	GET
FailCauses	GET
FailCauseCodes	GET
FailCauseSymptoms	GET
FailCasueAssetCategories	GET
FailCauseManufacturers	GET
ManufacturerCauseCodes	GET
Labor	GET, POST
DirectLabor	POST

Work Management- API Package Details	
MWO	
MultiUnitWorkOrders	GET, PUT, POST
MultiUnitWorkOrderNotes	GET, POST
MultiUnitWorkOrderTasks	GET, POST, PUT, DELETE
PartIssues	GET, POST
DirectIssues	GET, POST
Positions	GET
Priorities	GET
PriorityAssetCategories	GET
RepairReasons	GET, POST, PUT, DELETE
RepairSites	GET
ReturnReasons	GET, POST, PUT, DELETE
ServiceRequests	GET, POST, PUT
ServiceRequestFiles	GET, POST, PUT, DELETE
ServiceRequestNotes	GET, POST
ServiceRequestTasks	GET, POST, PUT, DELETE
ServiceRequestAttributes	GET, POST, PUT
ServiceRequestAssigned	GET
ServiceRequestModel	GET, POST, PUT
ServiceStatuses	GET Can use a filter parameter called \$filterbyassets to take a list of assets when requesting symptoms or service statuses. Records are only returned if the symptom or service status supports at least one asset's asset category from the provided list of assets.
StandardJobs	GET, POST, PUT, DELETE
StandardJobTasks	GET, POST, PUT, DELETE
StandardJobManufacturers	GET, POST, PUT, DELETE
StandardJobEquipmentClasses	GET, POST, PUT, DELETE

Work Management- API Package Details	
Symptoms	GET, POST, PUT, DELETE
Related Symptoms	GET
SymptomAssetCategories	GET
SymptomTasks	GET
	Can use a filter parameter called \$filterbyassets to take a list of assets when requesting symptoms or service statuses. Records are only returned if the symptom or service status supports at least one asset's asset category from the provided list of assets.
Tasks	
TaskAssetCategories	GET
TaskLocations	GET
TaskAppointments	GET
TaskAssignments	GET
TaskAssignmentEmployees	GET
TaskAssignmentModel	GET, POST, PUT, DELETE
TaskWorkAccomplishedCodes	GET
Tests	
TestResults	GET, POST, PUT
TestResultLines	GET, PUT
TestTypes	GET
TestTypeLines	GET
TestElements	GET
TestElementSymptoms	GET
TestTypeEquipment	GET
TestTypeEquipmentTypes	GET
TestTypeEquipmentClasses	GET
TestResultNotes	GET
TestResultFiles	GET, POST, PUT DELETE
TimeCodes	GET
WorkAccomplishedCodes	GET
WorkAccomplishedCodeCategories	GET
WorkAccomplishedCodeProdCategories	GET

Work Management- API Package Details	
WorkAssignmentProfiles	GET, PUT, POST, DELETE
WorkAssignmentProfileIDTasks	GET, POST, DELETE
WorkAssignmentProfileIDTaskID	DELETE
WorkAssignmentProfileIDAssets	GET, POST
WorkAssignmentProfileIDAssetsEquipmentID	DELETE
WorkClasses	GET
WorkDelayCodes	GET
WorkDelays	GET, PUT, POST
WorkOrderAttributes	GET, PUT, POST, DELETE
WorkOrderAttributesMain	GET, PUT, POST
WorkOrderAttributesMainRowID	DELETE
WorkOrderAttributesLocationYearNumberRowID	DELETE

Work Management- API Package Details	
WorkOrders	GET, POST
WorkOrderStandardJobs	GET, PUT, POST, DELETE
WorkOrdersAssigned	GET
LinearWorkOrders	GET
WorkOrderTasks	GET, PUT, POST, DELETE
LinearWorkOrderTasks	GET
WorkOrderTaskModel	GET, PUT, POST
WorkOrderModel	GET, PUT, POST
WorkOrderNotes	GET
LinearWorkOrderNotes	GET
WorkOrderFiles	GET
LinearWorkOrderFiles	GET, PUT
WorkOrderLocationYearNumber	PUT
Tasks	POST
PMTasks	POST
PMTasksRelatedPM	DELETE
TasksTaskID	DELETE
TestsTestResultID	DELETE
TestTypeResultID	DELETE
PMTasksTaskID	PUT
Files	GET, POST, PUT, DELETE
Notes	GET
Delays	PUT
Messages	GET
WorkPositions	GET, POST
WorkPositionCoordinates	GET, PUT, POST, DELETE