TopTeam Requirements REST API Guide

Revised: February 12, 2026

Contents

Getting Started with TopTeam Requirements REST API

TopTeam Requirements REST API provides programmatic access to the platform's core capabilities, enabling seamless integration with external tools for automation, reporting, and lifecycle management.

What is an API

An **Application Programming Interface (API)** is a set of rules and protocols that allow software applications to communicate with each other. APIs expose specific data and functionality, enabling integration between different systems and platforms. For instance, a project management tool might use an API to allow external systems to create tasks or fetch reports automatically.

Understanding REST APIs

TopTeam Requirements uses **REST (Representational State Transfer)** APIs, which are designed around standard HTTP methods like `GET`, `POST`, `PUT`, and `DELETE`. REST APIs are lightweight, stateless, and accessible over HTTP/S, making them ideal for web-based integrations. These APIs typically work with JSON-formatted request and response bodies, enabling seamless interoperability across diverse technology stacks.
Key benefits of REST APIs in *TopTeam* include:

- Platform-agnostic access using any programming language or tool
- Use of standard HTTP methods for CRUD operations
- Clear URI structures for addressing resources
- Lightweight data exchange using JSON

TopTeam REST APIs follow a versioned URI format and standardized response structure. For more details, see the next **Structure of REST API Unique Resource Identifier (URI)** section.

Structure of REST API Unique Resource Identifier (URI)

TopTeam Requirements REST APIs provide access to resources (data entities) via URI paths. To use REST API, clients must send HTTP or HTTPS requests and parse the JSON responses returned by the API server. Standard HTTP methods are used: GET, POST, PUT, and DELETE.

URI Format

TopTeam REST API resources follow this URI structure:

```
http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/resource-name
```

API Version

Each REST API endpoint includes a version number that determines the supported query parameters and response structure. If the structure or parameters change in a future release, a new version will be introduced. Older versions will continue to be supported for backward compatibility.

All endpoints in this guide use version **2**, so use 2 in the `<api_version>` placeholder.

```
http://myCompany.com/rest/ttmrestsrv.dll/2/projects
```

URI Path Convention

URIs may be **static** or **dynamic**:

- **Static URIs** never change:
  `http://myCompany.com/rest/ttmrestsrv.dll/2/projects`
- **Dynamic URIs** include path variables in angle brackets (<>), which must be replaced with actual values:
  `http://myCompany.com/rest/ttmrestsrv.dll/2/projects/<projectId>/teamMembers`

In the above example, replace `<projectId>` with the `recordId` of the project you want to query.

Common Response Structure

All API responses include a `success` field:

- If `true`, the response will include the requested data in a `response` field.
- If `false`, the `error` field will contain diagnostic details.

Success response

```
{
  "success": true,//Indicates that the server processed the request
successfully
  "response": <Here will come the json representation of resource
accessed>
}
```

Example

```
{
  "success": true,
  "response": {
    "children": [
      {
        "name": "Video Rental System",
        "recordId": 58336474,
        "id": "PRJ-100",
        "isLoggedInUserTeamMember": true,
        "isTemplateProject": false,
        "isPublished": false
      }
    ]
  }
}
```

Failure response

```
{
  "success": false, //Indicates that the server was unable to
process the request successfully
  "error": {
    "requestedurl": "<The requested URI>", //The resource being
accessed
    "error": "<Error message>", //Error message
    "type": "<Error Type>", //This can be etError, etInfo for error
and information respectively
```

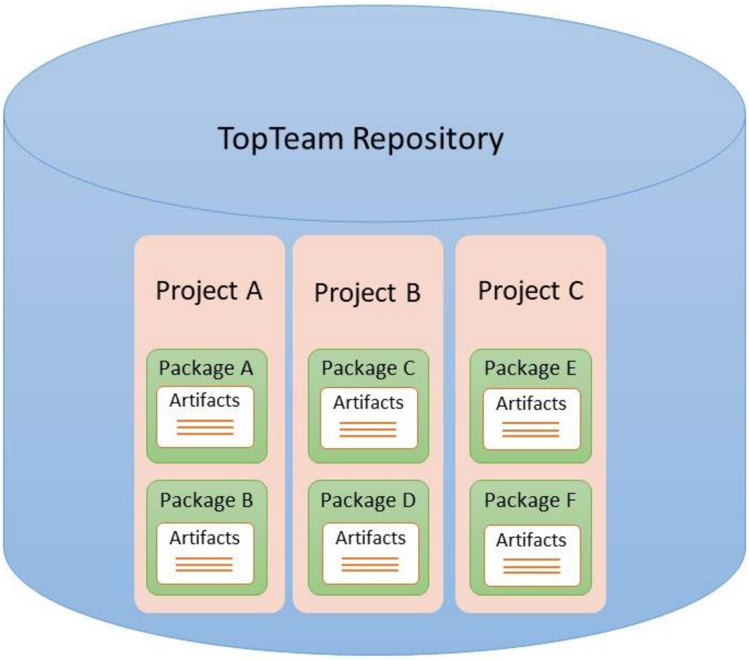
```
    "code": <Error code>,//There are different error codes according the type of error
    "debugMsg": "<Debug message if any>",
    "solution": "<Reason or solution of the error>",
    "stack": "<Method call stack>"
  }
}
```

Example

```
{
  "success": false,
  "error": {
    "requestedurl": "mycompany.com:/2/projects/34234",
    "error": "Unable to fetch project PRJ-1002. Either you are not a team member of this project or project does not exist.",
    "type": "etError",
    "code": 404,
    "stack": "",
    "solution": ""
  }
}
```

Understanding TopTeam Repository Structure

TopTeam organizes data using a hierarchy of **Projects, Packages, Records**, and **Fields**. Understanding this structure is essential for constructing meaningful API queries.



Projects

A ***Project* is a topmost container** in the *TopTeam* repository. **All artifacts belong to a *Project*.** You have to create a project before you can start adding artifacts to the repository. Each project has a unique `recordId` and can contain multiple packages, artifacts, and associated configurations.

Sample API Usages

- **Fetch List of Projects:**
  `GET /projects?view=tsList`
  → Returns a flat list of all projects accessible to the logged-in user
  See Fetch List of Projects

- **Fetch Project Details:**
  `GET /projects/{projectId}`
  → Returns detailed metadata for a given project.
  See Fetch Project Details

Packages

Packages are containers to store and organize artifacts within a project. All artifacts in a project must be added to one of the packages of the project.
Packages can be nested hierarchically. They allow structured organization of artifacts (e.g., *Use Cases*, *Requirements*, *User Stories*).
There are two ways packages are added to a project:

- **Automatically Created Packages**: When a record type is included in a project, a corresponding package (with the same name as the record type) is automatically created.
- **User-Defined Packages**: Users can manually create packages and configure them to accept **one or more record types**. This allows flexibility in organizing artifacts across different record types within the same package.

Sample API Usages

- **Fetch Single Package:**
  `GET http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/packages/{packageId}`
  → Returns the details of a single package, including its artifacts and sub-packages.

- **Fetch Packages:**
  `GET http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/packages`
  → Returns all packages in the repository along with their hierarchical structure.

  See Fetch Packages

- **Create Package:**
  `POST http://myCompany.com/rest/ttmrestsrv.dll/2/packages`
  → Creates a new package under the specified project and parent.
  See Create Package

Artifacts

An *Artifact (or record)* is a unit of information such as requirements, user story, issue, change request, etc.
Each artifact:

- Created inside a package.
- Belongs to a specific **record type**, which defines the set of fields and properties it supports.
- Exists within a **package**, which acts as its logical container.
- Has a unique `recordId` and is version-controlled via `recordRevisionNumber`.

Artifacts can be organized hierarchically, where:

- An artifact may be the **child** of another artifact or of a package.
- A **parent artifact** can contain one or more **groups**, each of which contains related child artifacts.

This structure enables fine-grained control over how artifacts are visually and functionally grouped, both in the UI and when querying via the API.

Sample API Usages

- **Create Artifact:**
  `POST /projects/{projectId}/{recordTypeIdPrefix}`
  → Creates a new artifact of the specified record type in the given project, under a specified package or parent artifact.
  See Create an Artifact

- **Update Artifact:**
  `PUT /repositoryObjects/byid/{recordId}`
  → Updates field values of the specified artifact using its latest revision number.
  See Edit Fields of an Artifact or Package

- **Fetch Artifact:**
  `GET /projects/{projectId}/{recordTypeIdPrefix}{recordId}`
  → Returns one artifact by ID.

- **Fetch Multiple Artifacts:**
  `GET /projects/{projectId}/{recordTypeIdPrefix}`
  → Returns a list of artifacts of that record type within the project.
  See Endpoints to Fetch List of Artifacts in Hierarchical Structure

Record Types

Record Types are fundamental to *TopTeam*. Each artifact added to the repository project must be of certain **Record Type**. A *Record type* is an object used to manage different types of artifacts in TopTeam such as *Product Requirements, User Stories, Use Cases, Test Cases*, etc.
For a project to use a record type, it must be **explicitly included** in that project's configuration. This allows each project to work with only the relevant set of record types.

Sample API Usages

- **Fetch All Record Types in the Repository**
  `GET /metadata/recordTypes`
  → Returns a complete list of all record types defined in the repository, including their metadata such as name, ID prefix, hierarchy support, and status.
  See Fetch Record Types

Record Type Configuration

Each record type is associated with configuration data that controls how artifacts of that type behave in a project.
This configuration includes:

- Fields (what data the artifact stores)
- States (how the artifact progresses through a lifecycle)
- Traceability rules (which types it can link to)

Fields

Fields define the structure of data captured in artifacts of a record type. These may include text fields (e.g., Name), rich text fields (e.g., Description), date fields, and List-of-Values (LOV) fields (e.g., Priority).

Sample API Usage

- **Fetch Record Type Fields**
  GET /metadata/{recordTypeIdPrefix}/fields
  → Returns all fields (system and custom) for the specified record type prefix, including metadata such as caption, type, LOV values, and whether the field is mandatory.
  See: **Fetch Record Type Fields**

  Example:
  GET /metadata/UC/fields
  (returns all fields for the "Use Case" record type)

  See Fetch Record Type Fields

States

States define the workflow or lifecycle an artifact follows. For example, an artifact might move through states such as Draft → In Review → Approved → Completed.

Sample API Usage

- **Fetch Record Type States**

  GET /recordTypes/{recordTypeId}/states
  → Returns all states configured for the given record type.
  See Fetch Record Type States

Traceability

Traceability defines which types of records this record type can link to. For instance, a "Requirement" might trace to a "Test Case" or "Design".

Sample API Usage

- **Fetch Record Type Traceability Rules**

  GET /metadata/{recordTypeIdPrefix}/linkingrules
  → Returns allowed traceability relationships for the given record type, including the record type and link type IDs for each direction.
  See Fetch Record Type Traceability Rules

Example:

GET /metadata/UC/linkingrules

(returns incoming/outgoing link rules for "Use Case")

Important Fields of all API Objects

Field Name	Type	Description
recordId	Resource Id	Unique Id of an artifact (record)/resource. **This Id needs to be sent to perform any operation on an artifact.** This is also required when you want to fetch details of an artifact.
recordRevisionNumber	Number	Number of times an artifact has been updated. **You need to send this value along with *recordId* to perform any edit or delete operation on an artifact.** If the *recordRevisionNumber* value provided by you doesn't match the value in the database, the system will return an error. This happens when an artifact is updated in between getting the artifact details via the GET method and sending the edit or DELETE method for it. In such a case, you need to fetch the artifact detail again and send the latest value of this field.
name	String	Name of an artifact.
createdDate	String	Date and time when an artifact was created. The string is in ISO 8601 format. e.g. 2023-08-04T10:13:29.726Z This value is in UTC time zone. You need to convert this into a local time zone if required.

lastUpdateDate	String	Date and time when an artifact was last updated. The string is in ISO 8601 format. e.g. 2023-08-04T10:13:29.726Z This value is in UTC time zone. You need to convert this into a local time zone if required.
createdByUserID	Resource Id	*recordId* of the user account that has created the artifact.
lastUpdatedByUserId	Resource Id	*recordId* of the user account that has modified the artifact.

Important Fields of all Repository Artifact Objects

Apart from the fields mentioned in the previous section, the following are the additional fields common to all repository artifacts/issues.

Field Name	**Type**	**Description**
Id/disp.Id	String	Display Id of a repository artifact that a user sees on the user interface. This ID is just for display purposes only. **This cannot be used to perform any query or operation via REST API.**
versionId/ recordVersionId	Resource Id	Unique id of a version of an artifact. This needs to be sent along with recordId to query details of a version of an artifact.
ownerRecordId	Resource Id	*recordId* of the package of an artifact, i.e., the unique identifier of the package to which an artifact belongs.
recordParentId / parentRecordId	Resource Id	*recordId* of the parent artifact of an artifact.
projectRecordId	Resource Id	*recordId* of the project to which an artifact belongs

recordTypeId	Resource Id	*recordId* of the record type of an artifact.
displaySequence	number	Display sequence number of an artifact. This determines the sequence of the artifacts to show them in an ordered structure.
reuseType	string	This field will be present only when a record is reused. This field Indicates reuse type of the record. Possible values are: • ReusedAsInstance • ReusedAsReference
reuseStatus	String	This field indicates whether the reused record is synchronized with its source. Possible values are: • UptoDate – record is synchronized with its source. • OutDated – there are changes in the source of the reused record which are not synchronized with this record.
reuseSourceRecordId	Resource Id	*recordId* of the source record
reuseScope	String	This field indicates how the record is reused. Possible values are: • Individual – Record is reused individually • Package – Record is reused along with its package.
reuseUpdateMethod	String	This indicates how the record will be synchronized with its source record. Possible values are:

		• Manual – The record will be manually synchronized by a user. • Automatic – TopTeam will automatically synchronize the record as and when its source is changed.
checkedOut	Boolean	True value indicates that the record is checkedOut.
checkedOutByUserId	Resource Id	This field gives the recordId of the user who has checked out the record. This field is applicable only when *checkedOut* is True.

Basic CRUD Operations

TopTeam REST API supports standard Create, Read, Update, and Delete (CRUD) operations:

Create a Record

```
POST /projects/{projectId}/{recordTypePrefix}
```

Example

```
{
    "newRecordOwnerId": 2220200815,
    "fields": {
      "Name": "Create Customer Account",
      "MoSCoW": "Must Have"
    }
}
```

Read a Record

```
GET /repositoryObjects/byId/{recordId}
```

Update a Record

```
PUT /repositoryObjects/byId/{recordId}
```

Example

```
{
  "recordRevisionNumber": 1,
  "fields": {
    "MoSCoW": "Should Have"
  }
}
```

Delete a Record

`DELETE /repositoryObjects/byId/{recordId}?recordRevisionNumber={revision}`

Advanced Artifact Operations

Comments & Artifact Linking

Comments

You can fetch, add, update, or delete comments on any artifact.

- **Add a Comment**:
  `POST /repositoryObjects/{recordId}/comments`
- **Fetch Comments**:
  `GET /repositoryObjects/{recordId}/comments`

Linking Artifacts

Artifacts can be linked to each other using traceability relationships.

- **Fetch Links**:
  `GET /repositoryObjects/{recordId}/links`
- **Add a Link**:
  `POST /repositoryObjects/{recordId}/links`

Querying in Baseline and Non-Baseline Modes

TopTeam supports two modes for retrieving records:

Non-Baseline Mode

- Retrieves the **latest version** of all artifacts.
- Default behavior if `tsBaselineId` is not specified.

Baseline Mode

Retrieves artifact states as they were at the time a **baseline snapshot** was captured. Requires passing a baseline ID via `tsBaselineId`.

Endpoint:

`GET /byRecordType/REP?tsGenListViewParams={...}`

Example:

```
{
    "tsProjects": "376523234",
  "tsBaselineId": "7458798944",
  "tsDisplayType": "dtTree",
  "tsEtpIds": "170"
}
```

Authentication

Authentication using API Key

You can generate and use personal API keys to authenticate requests with *TopTeam REST API*. You can generate the API Keys through *TopTeam* interface. For more information, see Using API Keys.

Specify API key in Request:
The API key can be included in the request's *Authorization Header*. The *Authorization Header* follows the Basic authentication scheme. Here, the API Key, along with an empty username, is encoded in Base64 format.

```
Authorization: Basic base64(username:APIKey)
```

Replace `username` with an empty string and `APIKey` with your actual API Key.

Example in JavaScript:

```
const request = require('request');

// Define the URL you want to request
const url = ‘http://myCompany.com/rest/ttmrestsrv.dll/2/projects’;

// Define your api key
const accessToken = 'your_api_key';

// Set up the request options
const options = {
    url: url,
    headers: {
        'Authorization': 'Basic ' + Buffer.from(':' +
accessToken).toString('base64')
    }
};

// Make the HTTP request
request(options, function(error, response, body) {
    if (!error && response.statusCode === 200) {
        console.log('Response body:', body);
    } else {
        console.error('Error:', error);
        console.error('Response status code:', response.statusCode);
    }
});
```

NOTE

Ensure that you include an authorization header in every request to authenticate with *TopTeam REST API*.

Authentication using Username and Password

1. Login
2. Logout
3. Check session

Login

Purpose	Authenticate and create a new session for a user in *TopTeam*
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/login
Request Method	**POST**

Request Body:

Field	Mandatory	Type	Allowed Values	Description
username	Yes	String		Specify the login Name of the User.
password	Yes	String		Password of the user account.
dbinstance	Yes	String	User Data	Name of a database instance. Always use "User Data".

Example:

POST http://myCompany.com/rest/ttmrestsrv.dll/2/login

```
{
  "username": "Steve Project Manager",
  "password": "Steve@321",
  "dbinstance": "User Data"
}
```

Response:

```
{
    "success": true,
    "response": {
        "userId": 4512547,
        "userName": "Steve Project Manager",
        "sessionId": "{87A5A534-FE34-4978-AEDC-D71F6F9DB205}",
        }
}
```

NOTE	After successful login, the REST server sets the session ID of the authenticated session in a cookie in response. This cookie must be sent in all subsequent endpoints to authenticate a request.

Logout

Purpose	Terminate existing session for a logged-in user and invalidate the session cookie.
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/logout
Request Method	**GET**

Request Body: Not Required

Example:
GET http://myCompany.com/rest/ttmrestsrv.dll/2/logout

Response:

```
{
    "success": true,
    "response": ""
}
```

Check session

Purpose	Returns information about the authenticated user's session
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/checkSession
Request Method	**GET**

Request Body: Not Required

Example
GET http://myCompany.com/rest/ttmrestsrv.dll/2/checksession

Response:

Expected JSON when user session exists:

```
{
    "success": true,
    "response": {
        "userId": 2220169265,
        "userName": "Steve Project Manager",
        "sessionId": "{D35E9E84-BC47-44D7-B9B1-849772279556}"
    }
}
```

Expected JSON when user session does not exist:

```
{
    "success": false,
    "error": {
        "requestedurl": "",
        "error": "Session cookie does not exist.",
        "type": "etError",
        "code": 0,
        "stack": "",
        "solution": ""
    }
}
```

Project Endpoints

Fetch List of Projects

Purpose	Returns a list of projects accessible to a logged-in user
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/projects
Request Method	**GET**

Supported Query Parameters:

Parameter Name	Mandatory	Allowed Values	Default Value	Description
view	No	tsList tsTree	tsList	The view param determines the structure of response. Use *tsList* to get projects in a non-hierarchical structure. Use *tsTree* to get projects in a hierarchical structure.

Request Body: Not Required

Example

1. **Projects in List Format**

GET http://myCompany.com/rest/ttmrestsrv.dll/2/projects?view=tsList

Response:

```
{
    "success": true,
    "response": {
        "projects": [
            {
                "recordId": 6484175747,
                "id": "PRJ-14284",
                "isPublished": false,
                "isTemplateProject": false,
                "managerId": 0,
                "name": "Vehicle Mounted Surveillance System",
                "parentRecordId": 648493163,
                "recordRevisionNumber": 2,
                "projectCode": "VMSS",
                 "primaryComponentRecordTypeId": 0,
                "primaryComponentRecordId": 0,
                "isLoggedInUserTeamMember": true,
                "projectType": "Component Project"
            },
            {
                "name": "Sample Projects",
                "recordId": 648440165,
                "id": "PRJ-11128",
                "isPublished": false,
                "isTemplateProject": false,
                "managerId": 0,
                "parentRecordId": 0,
                "recordRevisionNumber": 9,
                "projectCode": "SPRJ",
                "primaryComponentRecordTypeId": 0,
                "primaryComponentRecordId": 0,
                "isLoggedInUserTeamMember": true,
                "projectType": "Regular Project"
            }
        ]
    }
}
```

NOTE	The *parentRecordId* field of a project refers to the parent of a Project. If parentRecordId is 0, it means the project is at the root.

2. Projects in Hierarchical Format:

GET http://myCompany.com/rest/ttmrestsrv.dll/2/projects?view=tsTree

Response:

```
{
    "success": true,
    "response": {
        "projects": [
            {
                "name": "Global Library",
                "recordId": 6484137781,
                "id": "PRJ-13947",
                "isPublished": false,
                "isTemplateProject": false,
                "managerId": 0,
                "parentRecordId": 0,
                "recordRevisionNumber": 3,
                "projectCode": "GLIB",
                "primaryComponentRecordTypeId": 0,
                "primaryComponentRecordId": 0,
                "isLoggedInUserTeamMember": true,
                "projectType": "Library",
                "children": [
                    {
                        "recordId": 6484140509,
                        "id": "PRJ-13973",
                        "isPublished": false,
                        "isTemplateProject": true,
                        "managerId": 0,
                        "name": "FMEA Template",
                        "parentRecordId": 6484137781,
                        "recordRevisionNumber": 6,
                        "projectCode": "FMEAT",
                        "primaryComponentRecordTypeId": 2500,
                        "primaryComponentRecordId": 6542124541242,
                        "isLoggedInUserTeamMember": true,
                        "projectType": "Regular Project",
                        "children": []
                    }

                ]
            }
      ]
    }
}
```

Explanation of Response Fields

Field Name	Type	Description
name	String	Name of the project
recordId	Resource Id	Unique Id of the project. **You need to use this in endpoints that require recordId of project/projectId.**
parentRecordId	Resource Id	*recordId* of the parent project, i.e., the project to which this project is a child.

recordRevisionNumber	Number	Number of times an artifact has been updated.
id	String	Display Id of the project that a user sees on the user interface. This Id is just for display purposes only. **This cannot be used to perform any query or operation via REST API.**
isPublished	Boolean	Determines if project is published. Returns True if project is published. Returns false if project is not published.
isTemplateProject	Boolean	Determines if it is a project template or not.
managerId	Resource Id	recordId of manager.
isLoggedInUserTeamMember	Boolean	Determines if logged-in user is a team member or not.
projectCode	String	Unique code for the Project. Main purpose of the *projectCode* code is to easily search or identify the project when working with a large number of projects in a repository.
projectType	String	This indicates which type of project it is. The following are the different project types: • Component Project • Component Catalog

		- Project Group - Library - Regular Project
primaryComponentRecordTypeId	Resource Id	recordId of the primary component record type of the project.
primaryComponentRecordId	Resource Id	recordId of the primary component of the project.
children	Array of Project object	Array of child project object. This field only comes in response when Project List is in hierarchical format.

Fetch Project Details

Purpose	Returns details of a project
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/projects/*<projectId>*
Request Method	**GET**

Dynamic Path Variables

Variable Name	Type	Description
projectId	Resource Id	*recordId of the project*

Request Body: Not Required

Example:

GET http://myCompany.com/rest/ttmrestsrv.dll/2/projects/376523234

Response:

```
{
    "success": true,
    "response": {
        "name": "Kitchen-Sink",
        "recordId": 376523234,
        "id": "PRJ-10123",
        "isPublished": false,
        "isTemplateProject": false,
        "managerId": 0,
        "description": "",
        "parentRecordId": 648440165,
        "path": "Sample Projects\\Kitchen-Sink",
        "recordRevisionNumber": 46,
        "projectCode":"KSPRJ",
        "isLoggedInUserTeamMember": true,
        "projectType":"Regular Project"
    }
}
```

Explanation of Response Fields

Field Name	Type	Description
name	String	Name of the project
recordId	Resource Id	Unique Id of the project. **You need to use this in endpoints that require recordId of project/projectId.**
parentRecordId	Resource Id	*recordId* of the parent project, i.e., the project to which this project is a child.
recordRevisionNumber	Number	Number of times an artifact has been updated.
id	String	Display Id of the project that a user sees on the user interface. This ID is just for display purposes only. **This cannot be used to perform any query or operation via REST API.**
description	String	Specify description of the project in plain text.

isPublished	Boolean	Determines if project is published. Returns True if project is published. Returns false if project is not published.
isTemplateProject	Boolean	Determines if project is a project template or not. Return True if project is project template.
managerId	Resource Id	recordId of manager.
path	String	Determines the path of the project.
isLoggedInUserTeamMember	Boolean	Determines if logged-in user is a team member or not.

Fetch Record Types Included in a Project

Purpose	Returns list of record types included in a project
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/projects/*<projectId>*/recordTypes
Request Method	**GET**

Dynamic Path Variables

Variable Name	Type	Description
projectId	Resource Id	*recordId of the project*

Request Body: Not Required

Example:
GET http://myCompany.com/rest/ttmrestsrv.dll/2/projects/376565234/recordTypes

Response:

```
{
    "success": true,
    "response": {
        "projectRecordId": 376523234,
        "recordTypes": [
            {
                "recordId": 180,
                "parentRecordId": 110,
                "singularName": "Actor",
                "pluralName": "Actors",
                "canCreateRecord": true
            },
            {
                "recordId": 1200,
                "parentRecordId": 1190,
                "singularName": "Approval Request",
                "pluralName": "Approval Requests",
                "canCreateRecord": true
            },
            {
                "recordId": 1090,
                "parentRecordId": 1610,
                "singularName": "Business Process",
                "pluralName": "Business Processes",
                "canCreateRecord": true
            }
        ]
    }
}
```

Explanation of Fields

Field Name	Type	Description
recordId	Resource Id	recordId of record type.
parentRecordId	Resource Id	recordId of parent record type.
singularName	String	Singular name of record type.
pluralName	String	Plural name of record type.
canCreateRecord	Boolean	Specifies if a user can create artifact of this type under a project.

Fetch Team Members of a Project

Purpose	Returns list of active team members of a specified project along with roles granted to team members on that project. The list will contain internal as well as external team member details.
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/projects/ *<projectId>*/teamMembers
Request Method	**GET**

Dynamic Path Variables

Variable Name	Type	Description
projectId	Resource Id	*recordId* of the project

Request Body: Not Required

Example:

GET http://myCompany.com/rest/ttmrestsrv.dll/2/projects/376523234/teamMembers

Response:

```
{
    "success": true,
    "response": {
        "projectRecordId": 376523234,
        "teamMembers": [
            {
                "userId": 2220169265,
                "displayName": "Steve Project Manager",
                "userName": " Steve Project Manager",
                "isExternal": false,
                "roles": []
            },
            {
                "userId": 2220169271,
                "displayName": "Chris Team Lead",
                "userName": "Chris Team Lead",
                "isExternal": false,
                "roles": [
                    {
                        "recordId": 4550,
                        "name": "Business Analyst"
                    }
```

```
                              ]
                         }
                    ]
          }
}
```

Explanation of Fields

Field Name	Type	Description
userId	Resource Id	recordId of the user.
username	String	Login name of the user. Used for login authentication only.
displayName	String	Display the name of the user that appears in all *TopTeam* interfaces.
isExternal	Boolean	Defines user as author user or non-author user. Non-author users can be collaborators or viewers.

roles	Array of Role Object	An array of role objects which contains the roles granted to the user
		
		Structure of the role array:
		"roles":[
		 {
		 "recordId":<recordId of role>,
		  "name":"<name of the role>"
		 }
		].

Role Object

Field Name	Type	Description
recordId	Resource Id	*recordId* of the roles that are assigned to the user.
name	String	Name of the role assigned to the user.

Fetch List of Baselines in a Project

Purpose	Returns list of baselines in a project
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/projects/*<projectId>*/baselines
Request Method	**GET**

Dynamic Path Variables

Variable Name	Type	Description
projectId	Resource Id	*recordId* of the project

Request Body: Not Required

Example:
GET http://myCompany.com/rest/ttmrestsrv.dll/2/projects/376565234/baselines

Response:

```
{
    "success": true,
    "response": {
        "projectRecordId": "376565234",
        "baselines": [
            {
                "recordId": 6484150277,
                "name": "Production Release 2.1",

                "isLocked": false,
                "description": "",
                "createdByUserId": 2,
                "createDate": "2022-11-23T11:37:40.150Z",
                "recordRevisionNumber": 1,
                "isCurrentBaseline": false,
                "isSystemCreated": false,
                "baselineDate": "2022-11-23T11:37:40.150Z"

            },
            {
                "recordId": 6484147591,
                "name": "Beta Release 2.0",
                "isLocked": false,
                "description": "",
                "createdByUserId": 2,
                "createDate": "2022-11-23T11:07:53.453Z",
                "recordRevisionNumber": 1,
                "isCurrentBaseline": false,
                "isSystemCreated": false,
                "baselineDate": "2022-11-23T11:07:53.453Z"

            }
        ]
    }
}
```

Explanation of Fields

Field Name	Type	Description
recordId	Resource Id	*recordId* of a baseline.
recordRevisionNumber	Number	Number of times baseline has been updated.
name	String	Name of a baseline.
description	String	Description of a baseline.
createdByUserId	Resource Id	*recordId* of the user who created the baseline.
createDate	String	Date and time when a baseline is created. The string is in ISO 8601 format.
isCurrentBaseline	Boolean	Determines if a baseline is current or not. True value specifies that the baseline is a current baseline and that all the artifacts in it are of the current version.
isSystemCreated	Boolean	Determines if a baseline is created by the system or not.
baselineDate	string	Date and time for which a baseline was captured. In *TopTeam*, you can create a baseline for the current date and time or for a previous date.

		When you create a baseline for a previous date, the *baselineDate* will be the date for which this baseline was created, whereas the *createDate* will have the date and time on which the baseline is created. The date string is in ISO 8601 format.

Fetch Project Permissions

Purpose	Returns permissions of logged-in users on a project
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/projects/*<projectId>*/permissions
Request Method	**GET**

Dynamic Path Variables

Variable Name	Type	Description
projectId	Resource Id	*recordId* of the project

Request Body: Not Required

Example:
GET http://myCompany.com/rest/ttmrestsrv.dll/2/projects/376565234/permissions

Response:
```
{
    "success": true,
    "response": {
        "projectRecordId": 376565234,
        "permissions": {
            "isSecurityEnabled": false,
            "canAddRootProject": false,
            "canAddChildProject": false,
            "canAddRemoveRecordType": false,
            "canDelete": false,
            "canModify": false,
            "canAddRemoveTraceRelations": false,
            "canRankBacklog": false,
            "canAddRecordToBacklog": false,
            "canRemoveRecordFromBacklog": false,
            "canAddRemoveTeamMember": false,
            "canChangeProjectManager": false,
            "canCreateBranch": false,
            "canGrantRevokePrivileges": false,
            "canUpdateRecordType": false,
            "canMergeBranch": false,
            "canReuseRecord": false,
            "canUpdateReusedRecord": false,
            "canDelinkReusedRecord": false
        }
    }
}
```

Project Artifact Endpoints

1. Endpoints to Fetch List of Artifacts in Hierarchical Structure
2. Endpoints to Fetch details of single artifact
3. Artifact Endpoints
4. Package Endpoints
5. Diagram Field Endpoints
6. File Endpoints
7. Use Case Endpoints
8. Link Endpoints
9. Traceability Endpoints
10. Comment Endpoints
11. Attachment Endpoints
12. Linked Issue Endpoints
13. Version Endpoints
14. Audit Log Endpoints

Endpoints to Fetch List of Artifacts in Hierarchical Structure

Purpose	Returns a list of artifacts of a particular record type in a project, package, or parent artifact
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/ByRecordType/*<recordTypeIdPrefix>*
Request Method	**GET**

Dynamic Path Variables

Variable Name	Type	Description
recordTypeIdPrefix	String	*idPrefix of recordType* the record type that you want to fetch, such as 'UC', 'PREQ', and more.

NOTE	To fetch artifacts of **all record types or multiple record types use 'REP'** as the recordTypeIdPrefix. You can then use the field tsEtpIds in *tsGenListViewParams* query parameter to specify *recordId* of multiple record types. This is explained later in this document.

Supported Query Parameters:

Parameter Name	Mandatory	Allowed Values	Default Value	Description
nameMaxLength	No	Number	-	Some of the record types support names up to 4000 characters. Use this parameter to limit the length of an artifact name to be sent in the response. For example, `nameMaxLength=239`

				By default, full text of artifact name is sent in the response.
tsGenListViewParams	Yes	JSON Object	-	Click here to see all the fields of tsGenListViewParams object
recordIds	No	Comma text *recordId* of Artifact		Use this parameter to fetch artifacts of specific artifact Ids. Example: recordIds=6484947331, 64849947341, 12457845
tsClientTimeDiffInSeconds	No	number		Time difference in seconds between UTC and your local time zone. The value should be negative if local time is behind UTC. This parameter is mandatory when you want to filter artifacts based on Create On, Modified On fields such as "Created On" "IS TODAY", and more.

Request Body: Not Required

Example:
GET http://myCompany.com/rest/ttmrestsrv.dll/2/byRecordType/REP?tsGenListViewParams={ "ts
DisplayType": "dtTree", "tsProjects": "376523234", "tsParentId": 648497219, "tsEtpIds":"1060, 180", "fields" :"name, MoSCoW "}

Response:

```
{
    "resultContentType": "json",
    "success": true,
    "response": {
        "children": [
            {
                "fields": {
                    "name": "Manage Customer",
                    "MoSCoW": "Not Prioritized"
                },
                "recordId": 648497219,
                "versionId": 648497220,
                "recordTypeId": 1170,
                "id": "GROUP-13210",
                "ownerRecordId": 376524202,
                "recordRevisionNumber": 1,
                "recordParentId": 376524202,
                "isPublished": false,
                "createdByUserId": 2,
                 "checkedOut":false,
                 "lastUpdatedByUserId": 2,
               "webURL":"http://myCompany.com/#record?id=648497219",
                "filterSatisfied": false,
                "isOwnerCollection": true,
                "ownerRecordTypeId": 1370,
                "ownerRecordTypeIndicator": "CLL",
                "etpDispIdPrefix": "GROUP",
                "children": [
                    {

                        "fields": {
                            "name": "Upload new Videos",
                            "MoSCoW": "Not Prioritized"
                        },
                        "recordId": 3592322901,
                        "versionId": 376525608,
                        "recordTypeId": 1060,
                        "id": "STORY-10395",
                        "ownerRecordId": 376524202,
                        "recordRevisionNumber": 1,
                        "recordParentId": 648497219,
                        "reuseStatus": "UptoDate",
                        "reuseType": "ReusedAsInstance",
                        "reuseSourceRecordId": 64849999946946,
                        "reuseScope": "Individual",
                        "reuseUpdateMethod": "Automatic",
                        "isPublished": false,
                        "createdByUserId": 2,
                        "checkedOut": false,
                        "lastUpdatedByUserId":2,
                      "webURL":http://myCompany.com/#record?id=35923
                      22901",
                        "filterSatisfied": true,
                        "isOwnerCollection": true,
                        "ownerRecordTypeId": 1370,
                        "ownerRecordTypeIndicator": "CLL",
                        "etpDispIdPrefix": "STORY",
                        "children": []
                      }
                ]
            }
        ]
    }
}
```

Explanation of Response Fields

Field Name	Type	Description
children	Array	Array of child artifact object.

For an explanation of the common fields in the response, refer to Important fields of all Repository Artifacts / Issues. Any fields not explained are intended for internal use and should be ignored.

Fields of tsGenListViewParams Object

Field	Mandatory	Type	Allowed Values	Description
tsDisplayType	No	String	dtTree, dtList	Use *dtTree* to get response in hierarchical format. Use *dtList* to get a response in a non-hierarchical format.
tsFilterType	No	String	ftDataFilter ftFilterCondition	*ftDataFilter*: Use this to filter artifacts using *TopTeam Saved Filters*.

				ftFilterCondition: Use this to specify filter criteria in Text format. Example: `"State" = "Approved"`
tsFilterValue	No	Resource Id / String	recordId of a saved filter *OR* Filter condition in text format	The value and datatype of this field depend on the field *tsFilterType*. Refer to Filter artifacts using saved filters. Refer to Filter artifacts using filter criteria in text format.
tsProjects	Yes	Resource Id	recordId of a Project	Use this parameter to fetch artifacts of a specific project.
tsBaselineId	No	String	recordId of Baseline	Specify the *recordId* of a baseline if you want to fetch artifacts from a particular baseline. This field is optional. If not passed, the endpoint will return the current version of artifacts. Example: "tsBaselineId":"7458798944"

				Refer to Get list of baselines in a project get the *recordId* of a baseline.
tsParentId	No	ID	recordId of a parent artifact	Specify this field if you want to fetch child artifacts of a particular package or artifact. In the case of a hierarchical format response, it will fetch all the artifacts under that parent till the last artifact of that hierarchy. In case of a non-hierarchical response, it will fetch only artifacts that have this parent ID.
tsEtpIds	No	String	Comma text *recordId* of recordTypes	Use this parameter to fetch the artifacts of one or more record types. Example: "170, 1060, 640" You can get *recordId* of record types using the endpoint Fetch Project record types.
fields	No	String	Comma text fields	Specify a comma-separated list of fields that you want to fetch. This can be used to retrieve only a subset of fields. For example, fields=name,MoSCoW

Get Hierarchy of Packages in a Project

The REST API endpoint to fetch artifacts can be used to get a hierarchy of packages in a project.

URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/byRecordType/REP?tsGenListViewParams={"tsDisplayType":"dtTree","tsProjects":"*<projectID>*","tsEtpIds": "*<packageRecordTypeIds>*"}
Request Method	**GET**

Request Body: Not Required

Example:
GET http://myCompany.com/rest/ttmrestsrv.dll/2/byRecordType/REP?tsGenListViewParams ={"tsDisplayType": "dtTree","tsProjects": "**37652323**", "tsEtpIds" **: "1370, 1300, 2180, 2530, 2490, 2500, 2510"**}

The above endpoint fetches all package types of artifacts by specifying *tsEtpIds* field of *tsGenListViewParams* object. In the above example, 1370, 1300, 2180, 2530, 2490, 2500, 2510 are the recordTypeId of *Packages, OneView Documents, Components, Blocks, Functional Components, Physical Components,* and *Software Components,* respectively.

To know the package record types defined in your repository, refer to the section How to find list of Package Record Types in my repository.

Get Artifacts of a Package in Hierarchical Format

To fetch the contents of a package, set the *recordId* of the package in *tsParentId* field of *tsGenListViewParams* object.

URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/byRecordType/REP?tsGenListViewParams={"tsDisplayType":"dtTree"tsProjects":"*<projectId>*","tsParentId": *<packageId>*}

Request Method	**GET**

Request Body: Not Required

Example:-
GET http://myCompany.com/rest/ttmrestsrv.dll/2/byRecordType/REP?tsGenListViewParams={ "tsDisplayType": "dtTree", "tsProjects": "**37656523234**","tsParentId**": 37656523234** }

Get Artifacts from all Projects Where the User is a Team Member

URL	http://myCompany.com/rest/ttmrestsrv.dll/2/byRecordType/REP
Request Method	**GET**

Fetches all artifacts from projects where the user is a team member. If scope fields such as tsProjects or tsParentId are not provided, the system defaults to fetching artifacts without scope constraints.

Request Body: Not Required

Example:
GET
http://myCompany.com/rest/ttmrestsrv.dll/2/byRecordType/REP?tsGenListViewParams={ "tsEtpIds":"170, 180”}

Get Artifacts Based on a Filter Condition

To fetch artifacts based on a filter condition, you need to specify *tsFilterType* and *tsFilterValue* in *tsGenListViewParams*.

Filter Artifacts using Saved Filters

URL	http://myCompany.com/rest/ttmrestsrv.dll/2/byRecordType/REP?tsGenListViewParams={"tsDisplayType": "dtTree", "tsProjects": "*<recordID of Project>*", "tsParentId": <*recordID of Parent / Package*>, "tsEtpIds":"*<recordTypeIDs>*", "tsFilterType":"ftDataFilter", "tsFilterValue": *<recordId of saved filter>*}
Request Method	**GET**

If *tsFilterType* is *ftDataFilter,* specify the *recordId* of a saved filter in tsFilterValue. Save Filter can be created via List or Tree Editor of TopTeam Application.

You can use Get Filters endpoint to get *recordId* of a saved filter.

Request Body: Not Required

Example:-
GET
http://myCompany.com/rest/ttmrestsrv.dll/2/byRecordType/REP?tsGenListViewParams={
"tsDisplayType": "dtTree", "tsProjects": "746779873", "tsParentId": 746779879,
"tsEtpIds":"170, 180", "tsFilterType":"**ftDataFilter**", "tsFilterValue": **4124578** }

Filter Artifacts using Filter Criteria in Text Format

URL	http://myCompany.com/rest/ttmrestsrv.dll/2/byRecordType/REP?tsGenListViewParams={ "tsDisplayType": "dtTree", "tsProjects": "746779873", "tsParentId": 746779879, "tsEtpIds":"170, 180", "tsFilterType":"ftFilterCondition", "tsFilterValue": *< filter conditions in text format>*}
Request Method	**GET**

If *tsFilterType* is *ftFilterCondition,* specify the filter condition as text in *tsFilterValue*.

For more information on how to build filter conditions in text, refer to filter-syntax.

Note	Although the article for filter-syntax describes how to build filter conditions for DocProcessor templates, these rules also apply to REST API.

Request Body: Not Required

Example:-

GET
http://myCompany.com/rest/ttmrestsrv.dll/2/byRecordType/REP?tsGenListViewParams={ "tsDisplayType": "dtTree", "tsProjects": "746779873", "tsParentId": 746779879, "tsEtpIds":"170, 180", "tsFilterType":"**ftFilterCondition**", "tsFilterValue": " **\"State\" = \"Approved\"** "}

Note	When specifying a filter condition as text, you need to encode the filter condition text to build a valid JSON object. Encoding means you need to prefix each Double Quote (**“**) inside the filter condition text with the character **\\.** For example, filter condition "state" = "Approved" and "Points" > "10" should be encoded as \"state\" = \"Approved\" and \"Points\" > \"10\"

Filter Artifacts without providing Scope Fields

URL	http://myCompany.com/rest/ttmrestsrv.dll/2/byRecordType/REP
Request Method	**GET**

Fetches all artifacts from projects where the user is a team member, based on the provided filter condition. If scope fields such as tsProjects or tsParentId are not provided, the system defaults to fetching artifacts without scope constraints.

Request Body: Not Required

Example:

GET

http://myCompany.com/rest/ttmrestsrv.dll/2/byRecordType/REP?tsGenListViewParams**=**{ "tsEtpIds":"170, 180”, "tsFilterType":"**ftDataFilter**", "tsFilterValue": **4124578** }

Get Artifacts for Selected recordIds

To fetch the artifacts for selected *recordIds*, set the recordId of the selected artifacts in *recordIds* query param.

URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/byRecordType/REP?tsGenListViewParams**=**{"tsDisplayType":"dtList"tsProjects":"*<projectId>*"}&recordIds=*<recordId of the selected artifacts>*
Request Method	**GET**

Request Body: Not Required

Example:

GET http://myCompany.com/rest/ttmrestsrv.dll/2/byRecordType/REP?tsGenListViewParams**=**{ "tsDisplayType": "dtTree”, "tsProjects": "**37656523234**"}&recordIds**=64849999947331, 64849999947341**

TIP	**For Better Performance of *Fetch List of Artifacts in Hierarchical Structure* Endpoints** If you have over 1000 artifacts in a project, do not fetch all artifacts in a single request. Try to fetch them by individual packages. Example: First, fetch a list of packages of a project, then fetch a list of artifacts of packages you are interested in.

Endpoint to Fetch Details of Single Artifact

Purpose	Returns details of a single artifact
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/byId/*<recordId>*
Request Method	**GET**

Dynamic Path Variables:

Variable Name	Type	Description
recordId	Resource Id	*recordId* of an artifact

Supported Query Parameters:

Parameter	Mandatory	Allowed Values	Description
fields	No	comma-separated field names	Specify a comma-separated list of fields that you want to fetch. This can be used to retrieve only a subset of fields. For example, `fields=id,name` If you don't specify this parameter, the server will return all fields of the artifact.
specialFields	No	tsWebURL	Set the *specialFields* as *tsWebURL* to get the fully qualified *TopTeam URL* to access the artifact. specialFields=tsWebURL

Request Body: Not Required

Examples:

GET
http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byId/2220202574?fields=id,name,MoSCow,Description

Sample Response:

```
{
    "success": true,
    "response": {
        "id": "UC-14800",
        "recordId": 2220202574,
        "name": "deliver pizza via Zomato",
        "nodeType": "ntRecord",
        "projectRecordId": 376523234,
        "parentRecordId": 376524218,
        "ownerRecordId": 376524218,
        "recordVersionId": 2220202711,
        "reuseType": "ReusedAsReference",
        "reuseStatus": "UptoDate",
        "reuseSourceRecordId": 64849999941478,
        "reuseScope": "Individual",
        "reuseUpdateMethod": "Automatic",
        "checkedOut": true,
        "checkedOutByUserId": 8193171620,
        "recordTypeId": 170,
        "recordRevisionNumber": 2,
        "fields": {
            "id": "UC-14800",
            "name": "deliver pizza via Zomato",
            "MoSCow": "",
            "Description": "Delivering pizza via Zomato"
        }
    }
}
```

Explanation of Response Fields

Field Name	Type	Description
fields	Fields Object	Fields JSON Object

For an explanation of the common fields in the response, refer to Important fields of all Repository Artifacts / Issues. Any fields not explained are intended for internal use and should be ignored.

Artifact Endpoints

Create an Artifact

Purpose	Create an artifact of a record type
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/projects/*<projectID>*/*<recordTypeIdPrefix>*
Request Method	**POST**

Dynamic Path Variables:

Variable Name	**Type**	**Description**
projectId	Resource Id	*recordId* of the project
recordTypeIdPrefix	String	*idPrefix of recordType* the record type that you want to create. e.g. 'UC', 'PREQ', etc.

Request Body:

Field	**Mandatory**	**Type**	**Allowed Values**	**Description**
newRecordOwnerId	Yes	Resource Id		*recordId* of the package in which you want to create this artifact.
newRecordLocation	No	String	tsLastChild tsSiblingAfter	tsLastChild: Creates the artifact as the last child of the parent artifact mentioned in

				newRecordParentId field. tsSiblingAfter: Creates the artifact after the artifact mentioned in *newRecordParentId* field
newRecordParentId	Yes	Resource Id		If *newRecordLocation* is tsLastChild, specify the *recordId* of the parent artifact. If *newRecordLocation* is *tsSiblingAfter*, specify the *recordId* of the artifact below which the new record should be created.
fields	Yes	[Fields Object]()		Specify Fields JSON Object containing a list of fields and their value. You must provide values of all the fields that are configured as mandatory for a record type. **Typically, only the Name field is mandatory**. However, your administrator may have configured more

				fields to be mandatory.

NOTE	
	You cannot specify a value for the *State* field while creating an artifact. Use the *Edit Artifact* endpoint to set the *State* field if you want to specify a value for *State* after creating the artifact.

Examples:

Create new record under a Package

POST http://myCompany.com/rest/ttmrestsrv.dll/2/projects/376523234/STORY

```
{
  "newRecordOwnerId":2220200815,
  "fields":{
      "Name":"Create customer account",
      "Description":"Customer register with valid credit card
      details.",
      "MoSCoW":"Could Have"
      }
}
```

Create artifact as a child of another artifact

POST http://myCompany.com/rest/ttmrestsrv.dll/2/projects/376523234/STORY

```
{
  "newRecordOwnerId":2220200815, //recordID of Package
  "newRecordLocation": "tsLastChild",
  "newRecordParentId":2220202324, //recordId of parent artifact
  "fields":{
    "Name":"Create customer account",
    "Description":"Customer register with valid credit card
details.",
      "MoSCoW":"Could Have"
      }
}
```

Create artifact below or as a sibling of another artifact

POST http://myCompany.com/rest/ttmrestsrv.dll/2/projects/376523234/STORY

```
{
  "newRecordOwnerId":2220200815,
  "newRecordLocation": "tsSiblingAfter",
  "newRecordParentId":2220202324,
  "fields":{
    "Name":"Create Customer Account",
    "Description":"Customer register with valid credit card 
details.",
      "MoSCoW":"Could Have"
   }
}
```

Response:

```
{
    "success": true,
    "response": {
        "name": "Create Customer Account",
        "recordId": 2220200821,
        "id": "UC-14781",
        "projectRecordId": 376523234,
        "parentRecordId": 2220202324,
        "ownerRecordId": 2220200815,
        "recordVersionId": 2220200822,
        "recordTypeId": 170,
        "recordRevisionNumber": 1,
        "fields": {
            "Checked-out": false,
            "Checked-out By": null,
            "Check-out Date": null,
            "Created By": "Steve Tester",
            "Created On": "2023-08-04T10:13:29.726Z",
            "Description": "Customer register with valid credit card 
details.",
            "Disp.Id": "UC-14781",
            "ID": "UC-14781",
            "Is Suspect": false,
            "Locked": false,
            "Modified By": "Steve Tester",
            "Modified On": "2023-08-04T10:13:29.726Z",
            "MoSCoW": "Could&#160;Have",
            "Multi Value 1": "",
            "Name": "New&#160;customer&#160;registration",
            "Notes": "",
            "Owner": "",
            "Post Condition": "<<Some rich text data>>",
            "Pre Condition": "",
            "Project": "Kitchen-Sink",
            "Record Type": "Use&#160;Case",
            "Ref. ID": null,
            "State": "Draft",
            "Trigger": "",
            "Unique ID": 2220200821,
            "Version": "1.00"
        }
    }
}
```

Explanation of Response Fields

For an explanation of the common fields in the response, refer to Important fields of all Repository Artifacts / Issues. Any fields not explained are intended for internal use and should be ignored.

Field Name	Type	Description
fields	Fields Object	Fields JSON Object

Create Multiple Artifacts

Purpose	Create Bulk artifacts
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/bulk
Request Method	**POST**

Request Body:

Field	Mandatory	Type	Allowed Values	Description
destinationRecordId	Yes	Resource Id		Specify the recordId of record below which you want to create the records.

position	No	String	LastChild SiblingAfter	**lastChild:** This option appends the new records as the last children of the specified record. **siblingAfter:** This option places the new records immediately after a specified record at the same hierarchical level, effectively adding them as sequential peers. Default position will be *LastChild.*
records	Yes	[List of records]()		Detail of the records that needs to be created

Records JSON Object

Field	**Mandatory**	**Type**	**Allowed Values**	**Description**
recordTypeId	Yes	ResourceId		recordId for the record type that needs to be created
fields	Yes	[Fields Object]()		List the fields and their corresponding values that needs be set when creating new records.

```
 POST http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryOjects/bulk
{
  "destinationRecordId":376524202,
 "position": " LastChild",
  "records": [
    {
      "recordTypeId": 1060,
      "fields": {
        "name": "Create Licence",
        "description": "Create License for the customer",
        "Moscow": "Must Have"
      },
      "children": [
        {
          "recordTypeId": 1060,
          "fields": {
            "name": "Create Driving Licence",
            "description": "Create Driving Licence for 18 yr old",           "Moscow": "Must Have"
          }
        },
        {
          "recordTypeId": 1060,
          "fields": {
            "name": "Create Pan Card",
            "description": "Create Pan Card for Income",
            "Moscow": "Must Have"
          }
          }
      ]
    }
  ]
}
```

Response:

```
{
    "resultContentType": "json",
    "success": true,
    "response": {
        "children": [
            {
                "id": "STORY-15379",
                "recordId": 7229217850,
                "name": "Create Licence",
                "projectRecordId": 376523234,
                "parentRecordId": 376524202,
                "ownerRecordId": 376524202,
                "recordVersionId": 7229217851,
                "recordTypeId": 1060,
                "recordRevisionNumber": 1,
                "fields": {
                    "Acceptance Criteria": "",
                    "Business Value": null,                "Checked-out": false,
                    "Checked-out By": null,
                    "Check-out Date": null,
                    "Created By": "Steve&#160;Tester",
                    "Created On": "2023,10,30,2,34,8,304",
                    "Description": "",
                    "Disp.Id": "STORY-15379",
                    "ID": "STORY-15379",
                    "Is Suspect": false,
                    "IsBoolean": true,
                    "Locked": false,
                    "Modified By": "Steve&#160;Tester",
                    "Modified On": "2023,10,30,2,34,8,304",
                    "MoSCoW": "Must&#160;Have",
                    "Name": "Create&#160;Licence",
                    "Notes": "",
                    "Opportunity Value": null,
                    "Owner": "",
                    "Points": null,
                    "Project": "Kitchen-Sink",
                    "Record Type": "User&#160;Story",
            "Ref. ID": null,
            "State": "Draft",
            "Story": "\r\n<p style=\" text-align: left; textindent: 0px; padding: 0px 0px 0px 0px; margin: 0px 0px 0px 0px;\">",
            "Unique ID": 7229217850,
            "Urgency": null,
            "Version": "1.00",
            "Weighted Value": 0
        },
        "children": [
            {
                "id": "STORY-15380",
                "recordId": 7229217856,
                "name": "Create Driving Licence",
                "projectRecordId": 376523234,
                "parentRecordId": 7229217850,
                "ownerRecordId": 376524202,
                "recordVersionId": 7229217857,
                "recordTypeId": 1060,
                "recordRevisionNumber": 1,
                "fields": {
                    "Acceptance Criteria": "",
                    "Business Value": null,
                     "Checked-out": false,
                    "Checked-out By": null,
                    "Check-out Date": null,
                    "Created By": "Steve&#160;Tester",
                    "Created On": "2023,10,30,2,34,9,493",
                     "Description": "Create Driving Licence f or aboce 18 yr old",
                    "Disp.Id": "STORY-15380",
                    "ID": "STORY-15380",
                    "Is Suspect": false,
                    "IsBoolean": false,
                    "Locked": false,
                    "Modified By": "Steve&#160;Tester",
                    "Modified On": "2023,10,30,2,34,9,493",
                    "MoSCoW": "Must&#160;Have",
                        "Name": "Create&#160;Driving&#160;Licence",
                        "Notes": "",
                        "Opportunity Value": null,
                        "Owner": "",
                        "Points": null,
                        "Project": "Kitchen-Sink",
                        "Record Type": "User&#160;Story",
                        "Ref. ID": null,
                        "State": "Draft",
                        "Story": "\r\n<p style=\" color: transparent; text-decoration: none; ;\">: </span></p>\r\n",
                        "Unique ID": 7229217856,
                        "Urgency": null,
                        "Version": "1.00",
                        "Weighted Value": 0
                    }
                },
                {
                    "id": "STORY-15381",
                    "recordId": 7229217862,
                    "name": "Create Pan Card",
                    "projectRecordId": 376523234,
                    "parentRecordId": 7229217850,
                    "ownerRecordId": 376524202,
                    "recordVersionId": 7229217863,
                    "recordTypeId": 1060,
                    "recordRevisionNumber": 1,
                    "fields": {
                        "Acceptance Criteria": "",
                        "Business Value": null,
                         "Checked-out": false,
                        "Checked-out By": null,
                        "Check-out Date": null,
                        "Created By": "Steve&#160;Tester",
                        "Created On": "2023,10,30,2,34,9,914",
                        "Description": "Create Pan Card for Income",
                        "Disp.Id": "STORY-15381",
                            "ID": "STORY-15381",
                            "Is Suspect": false,
                            "IsBoolean": false,
                            "Locked": false,
                            "Modified By": "Steve&#160;Tester",
                            "Modified On": "2023,10,30,2,34,9,914",
                            "MoSCoW": "Must&#160;Have",
                            "Name": "Create&#160;Pan&#160;Card",
                            "Notes": "",
                            "Opportunity Value": null,
                            "Owner": "",
                            "Points": null,
                            "Project": "Kitchen-Sink",
                            "Record Type": "User&#160;Story",
                            "Ref. ID": null,
                            "State": "Draft",
                            "Story": "\r\n<p style=\" text-align: left;,
                            "Unique ID": 7229217862,
                            "Urgency": null,
                            "Version": "1.00",
                            "Weighted Value": 0
                        }
                    }
                ]
            }
        ]
    }
}
```

Explanation of Response Fields
For an explanation of the common fields in the response, refer to Important fields of all Repository Artifacts / Issues. Any fields not explained are intended for internal use and should be ignored.

Field Name	Type	Description
fields	Fields Object	Fields Object

Update Artifact

Edit Fields of an Artifact or Package

Purpose	Modify fields of an artifact or Package
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/byid/*<recordId>*
Request Method	**PUT**

Dynamic Path Variables:

Variable Name	Type	Description
recordId	Resource Id	*recordId* of the artifact that you want to modify.

Request Body:

Field	Mandatory	Type	Allowed Values	Description
recordRevisionNumber	Yes	Number		Number of times an artifact has been updated.
fields	Yes	Fields Object		Specify JSON Object containing a list of fields and their value. **You need to specify the values of only those fields that you want to modify.**

Example 1:

PUT http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byid/2220200821

```
{
  "recordRevisionNumber": 6,
  "fields": {
    "Description": "Videos returned by Customers are entered into the
system.",
    "MoSCoW":"Must Have",
    "state": "accepeted"
  }
}
```

Response:

```
{
    "success": true,
    "response": {
        "name": "New customer registration",
        "recordId": 2220200821,
        "id": "UC-14781",
        "projectRecordId": 376523234,
        "parentRecordId": 2220200815,
        "ownerRecordId": 2220200815,
        "recordVersionId": 2220200822,
        "recordTypeId": 170,
        "recordRevisionNumber": 1,
        "fields": {
            "Checked-out": false,
            "Checked-out By": null,
            "Check-out Date": null,
            "Created By": "Steve Tester",
            "Created On": "2023-08-04T10:13:29.727Z",
            "Description": "Videos returned by Customers are entered
into the system.",
            "Disp.Id": "UC-14781",
            "MoSCoW": "MustCould&#160;Have",
            "Multi Value 1": "ba,ca",
            "State": "accepetedAccepted",
            "ID": "UC-14781",
            "Is Suspect": false,
            "Locked": false,
            "Modified By": "Steve Tester",
            "Modified On": "2023-08-04T10:13:29.727Z",
            "MoSCoW": "Not Prioritized",
            "Multi Value 1": "",
            "Name": "New customer registration",
            "Notes": "",
            "Owner": "",
            "Post Condition": ”<some rich text data>",
            "Pre Condition": "",
            "Project": "Kitchen-Sink",
            "Record Type": "Use Case",
            "Ref. ID": null,
            "State": "Draft",
            "Trigger": "",
            "Unique ID": 2220200821,
            "Version": "1.01"
        }
    }
}
```

Explanation of Response Fields

Field Name	Type	Description
fields	Fields Object	Fields JSON Object

For an explanation of the common fields in the response, refer to Important fields of all Repository Artifacts / Issues. Any fields not explained are intended for internal use and should be ignored.

Example 2:

When the user tries to edit an artifact with incorrect recordId

PUT http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byid/2220200821

```
{
  "recordRevisionNumber": 4,
  "fields": {
    "Description": "Videos returned by Customers are entered into
the system.",
    "MoSCoW": "Could Have",
    "Multi Value 1":"ca,ba",
    "state":"accepted"
  }
}
```

Response for Failed Edit Operation:

```
{
    "success": false,
    "response": "",
    "error": {
        "requestedurl": "localhost:2673/2/repositoryObjects/byId/222
0200821",
        "error": "Unable to access the record.",
        "type": "etInfo",
        "code": 404,
        "stack": "",
        "solution": "It may have been deleted or you do not have per
mission to view it."
    }
}
```

Field for failed Edit Operation

Field Name	Type	Description
success	Boolean	Indicates whether the operation on the artifact was successful. False value indicates failure.
error	Object	Refer Error Object This field is sent when success is false.

Edit Fields of Multiple Artifacts or Packages

Purpose	Modify multiple records in a single request
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/projects/*<projectId>*/REP
Request Method	**PUT**

Dynamic Path Variables:

Variable Name	Type	Description
projectId	Resource Id	*recordId* of the project

Supported Query Parameters:

Parameter Name	Mandatory	Allowed Values	Description
fields	No	Comma-separated field list	List of fields that you want to be returned in the response. This can be used to retrieve only a subset of fields. e.g. `fields=id,name` If this parameter is not specified, values of all fields will be returned in the response.
nameMaxLength	No	Number	The **nameMaxLength** parameter specifies the maximum length of an artifact name in a response. e.g. `nameMaxLength=239`

Request Body:

Array of Artifact Object

Fields of Artifact Object

Field	Mandatory	Type	Allowed Values	Description
recordId	Yes	Resource Id		*recordId* of an artifact that you want to edit.
recordRevisionNumber	Yes	Number		Number of times an artifact has been updated.
projectRecordId	Yes	Resource Id		Specify the *recordId* of a project.
fields	Yes	Fields Object		Specify JSON Object containing list of fields and their value. **You need to specify the values of only those**

				fields that you want to modify.

Example 1:

PUT http://myCompany.com/rest/ttmrestsrv.dll/2/projects/376523234/REP

```
[
  {
    "recordId": 2220199511,
    "projectRecordId":376523234,
    "recordRevisionNumber": 3,
    "fields": {
        "Description": "This User Story has been updated",
        "MoSCoW":"Should Have"
    }
  },
  {
    "recordId": 2220199517,
    "projectRecordId":376523234,
    "recordRevisionNumber": 5,
    "fields": {
        "Description": "This User Story has been updated",
        "MoSCoW":"Would Have",
        "State":"Completed"
    }
  }
]
```

Response:

```
{
    "success": true,
    "response": [
        {
            "success": true,
            "name": "Add 1",
            "recordId": 2220199511,
            "id": "STORY-14754",
            "nodeType": "ntRecord",
            "projectRecordId": 376523234,
            "parentRecordId": 376524202,
            "ownerRecordId": 376524202,
            "recordVersionId": 2220203624,
            "recordTypeId": 1060,
            "recordRevisionNumber": 4,
            "fields": {
                "Acceptance Criteria": "",
                "Business Value": null,
                "Checked-out": false,
                "Checked-out By": null,
                "Check-out Date": null,
                "Created By": "u1",
                "Created On": "2023-07-26T19:59:30.777Z",
                "Description": "This User Story has been updated",
                "Disp.Id": "STORY-14754",
                "ID": "STORY-14754",
                "Is Suspect": false,
                "Locked": false,
                "Modified By": "u1",
                "Modified On": "2023-08-14T21:04:13.409Z",
                "MoSCoW": "Should&#160;Have",
                "Name": "Add&#160;1",
                "Notes": "Add 1",
                "Opportunity Value": null,
                "Owner": "",
                "Points": 12,
                "Project": "Kitchen-Sink",
                "Record Type": "User&#160;Story",
                "Ref. ID": null,
                "Rich Text 1": "",
                "State": "Draft",
                "Story": "\r\nAs a: \r\nI want to: \r\nSo that: ",
                "Unique ID": 2220199511,
                "Urgency": null,
                "Version": "1.01",
                "Weighted Value": 0
            }
        },
        {
            "success": true,
            "name": "Add 2",
            "recordId": 2220199517,
            "id": "STORY-14755",
            "nodeType": "ntRecord",
            "projectRecordId": 376523234,
            "parentRecordId": 376524202,
            "ownerRecordId": 376524202,
            "recordVersionId": 2220203621,
            "recordTypeId": 1060,
            "recordRevisionNumber": 6,
            "fields": {
                "Acceptance Criteria": "",
                "Business Value": null,
                "Checked-out": false,
                "Checked-out By": null,
                "Check-out Date": null,
                "Created By": "u1",
                "Created On": "2023-07-26T19:59:31.473Z",
                "Description": "This User Story has been updated",
                "Disp.Id": "STORY-14755",
                "ID": "STORY-14755",
                "Is Suspect": false,
                "Locked": false,
                "Modified By": "u1",
                "Modified On": "2023-08-14T21:04:13.625Z",
                "MoSCoW": "Would Have",
                "Name": "Add&#160;2",
                "Notes": "This for Log level 1",
                "Opportunity Value": null,
                "Owner": "",
                "Points": null,
                "Project": "Kitchen-Sink",
                "Record Type": "User&#160;Story",
                "Ref. ID": null,
                "Rich Text 1": "",
                "State": "Completed",
                "Story": "\r\nAs a: \r\nI want to: \r\nSo that: ",
                "Unique ID": 2220199517,
                "Urgency": null,
                "Version": "1.01",
                "Weighted Value": 0
            }
        }
    ]
}
```

Explanation of Response Fields

Response object contains an array of artifact objects. Each artifact object contains the following fields.

Field Name	Type	Description
success	Boolean	Indicates whether operation on the artifact was successful. False value indicates failure.
error	Object	Refer Error Object This field is sent when success is false.
fields	Fields Object	Fields JSON Object

For an explanation of the common fields in the response, refer to Important fields of all Repository Artifacts / Issues. Any fields not explained are intended for internal use and should be ignored.

Example 2:

When the user tries to edit an artifact with incorrect recordId:

PUT http://myCompany.com/rest/ttmrestsrv.dll/2/projects/376523234/REP

```
[
  {
    "recordId": 376525513,
    "projectRecordId":376523234,
    "recordRevisionNumber": 8,
    "fields": {
        "Description": "This User Story has been updated",
        "MoSCoW":" Must Have"
    }
  },
  {
    "recordId": 376525573,
    "projectRecordId":376523234,
    "recordRevisionNumber": 4,
    "fields": {
        "Description": "This Multiple Type record",
        "MoSCoW":"Must Have",
        "State":"Completed"
    }
  }
]
```

Response:

```
{
    "resultContentType": "json",
    "success": true,
    "response": [
        {
            "recordId": 376525513,
            "success": false,
            "error": "Cannot perform this operation on 'Register 
Customer [STORY-10380]'.",
            "tsLogMessage": "Cannot perform this operation on 
'Register Customer [STORY-10380]'.",
            "tsErrSolution": "This record is Locked.\n\rUnlock the 
record to edit it.",
            "type": "etInfo",
            "errorCode": 0
        },
        {
            "success": true,
            "id": "STORY-10392",
            "recordId": 376525573,
            "name": "Reserve Videos",
            "nodeType": "ntRecord",
            "projectRecordId": 376523234,
            "parentRecordId": 648497219,
            "ownerRecordId": 376524202,
            "recordVersionId": 3180187050,
            "recordTypeId": 1060,
            "recordRevisionNumber": 5,
            "fields": {
                "Acceptance Criteria": "\r\nSearch Video by Name",
                "Business Value": 5,
                "Checked-out": false,
                "Checked-out By": null,
                "Check-out Date": null,
                "Created By": "Admin",
                "Created On": "2020-05-06T13:17:26.990Z",
                "Description": "This Multiple Type record",
                "Disp.Id": "STORY-10392",
                "ID": "STORY-10392",
                "Is Suspect": false,
                "Locked": false,
                "Modified By": "jay",
                "Modified On": "2023-08-19T09:24:20.470Z",
                "MoSCoW": "Must Have",
                "Name": "Reserve&#160;Videos",
                "Notes": "",
                "Opportunity Value": 5,
                "Owner": "",
                "Points": 5,
                "Project": "Kitchen-Sink",
                "Record Type": "User&#160;Story",
                "Ref. ID": null,
                "Relational Multi": null,
                "Relational Single": null,
                "State": "Completed",
                "Story": “<Some rich text data>",
                "Unique ID": 376525573,
                "Urgency": 5,
                "Version": "1.01",
                "Weighted Value": 3
            }
        }
    ]
}
```

Fields for Failed Bulk Edit Operation:

Field Name	Type	Description
success	Boolean	Indicates whether operation on an artifact was successful. False value indicates failure.
error	String	Error message.
tsLogMessage	String	Log message (internal use).
tsErrSolution	String	Reason or solution of the error
type	String	Type of error. Possible values are: etInfo, etError, or etWarning.
errorCode	Number	Error code of the error.

For an explanation of the common fields in the response, refer to Important fields of all Repository Artifacts / Issues. Any fields not explained are intended for internal use and should be ignored.

Lock an *Artifact* or *Package*

Purpose	Lock an artifact to disable further modification of it.
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/projects/*<projectId >*/*<recordTypeIdPrefix>*/*<recordId>*/lockRecord
Request Method	**PUT**

Dynamic Path Variables:

Variable Name	Type	Description
projectid	Resource Id	*recordId* of the project
recordTypeIdPrefix	String	*idPrefix of recordType* the record type that you want to edit. e.g. 'UC', 'PREQ', etc
recordId	Resource Id	*recordId* of the artifact

Request Body:

Field	Mandatory	Type	Allowed Values	Description
recordRevisionNumber	Yes	Number		Number of times an artifact has been updated.

Example:

PUT

http://myCompany.com/rest/ttmrestsrv.dll/2/projects/376523234/UC/2220200821/lockRecord

```
{
    "recordRevisionNumber": 1
}
```

Response:

```
{
    "success": true,
    "response": {
        "name": "Login to ATM",
        "recordId": 2220200821,
        "id": "UC-14781",
        "nodeType": "ntRecord",
        "projectRecordId": 376523234,
        "parentRecordId": 2220200815,
        "ownerRecordId": 2220200815,
        "recordVersionId": 2220200838,
        "recordTypeId": 170,
        "recordRevisionNumber": 3,
        "fields": {
            "Checked-out": false,
            "Checked-out By": null,
            "Check-out Date": null,
            "Created By": "Steve Tester",
            "Created On": "2023-08-04T10:13:29.727Z",
            "Description": "Editing 1",
            "Disp.Id": "UC-14781",
            "ID": "UC-14781",
            "Is Suspect": false,
            "Locked": true,
            "Modified By": "Steve Tester",
            "Modified On": "2023-08-04T10:29:19.889Z",
            "MoSCoW": "Not&#160;Prioritized",
            "Multi Value 1": "",
            "Name": "Login&#160;to&#160;ATM",
            "Notes": "",
            "Owner": "",
            "Post Condition": "<<some rich text data>>",
            "Pre Condition": "",
            "Project": "Kitchen-Sink",
            "Record Type": "Use&#160;Case",
            "Ref. ID": null,
            "State": "Draft",
            "Trigger": "",
            "Unique ID": 2220200821,
            "Version": "1.01"
        }
    }
}
```

For an explanation of the common fields in the response, refer to Important fields of all Repository Artifacts / Issues. Any fields not explained are intended for internal use and should be ignored.

Unlock *Artifact* or *Package*

Purpose	Remove the lock on an artifact to enable its modification
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/projects/*<projectId>*/ *<recordTypeIdPrefix>*/*<recordId>*/unlockRecord
Request Method	**PUT**

Dynamic Path Variables:

Variable Name	**Type**	**Description**
projectid	Resource Id	*recordId* of the project
recordTypeIdPrefix	String	*idPrefix of recordType* the record type that you want to fetch. e.g. 'UC', 'PREQ', and more.
recordId	Resource Id	*recordId* of the artifact

Request Body:

Field	**Mandatory**	**Type**	**Allowed Values**	**Description**
recordRevisionNumber	Yes	Number		Number of times an artifact has been updated.

Example:

PUT

http://myCompany.com/rest/ttmrestsrv.dll/2/projects/376523234/UC/2220200821/unlockRecord

```
{
"recordRevisionNumber": 2
}
```

Response:

```
{
    "success": true,
    "response": {
        "name": "Login to ATM",
        "recordId": 2220200821,
        "id": "UC-14781",
        "projectRecordId": 376523234,
        "parentRecordId": 2220200815,
        "ownerRecordId": 2220200815,
        "recordVersionId": 2220200838,
        "recordTypeId": 170,
        "recordRevisionNumber": 4,
        "fields": {
            "Checked-out": false,
            "Checked-out By": null,
            "Check-out Date": null,
            "Created By": "Steve Tester",
            "Created On": "2023-08-04T10:13:29.727Z",
            "Description": "Editing 1",
            "Disp.Id": "UC-14781",
            "ID": "UC-14781",
            "Is Suspect": false,
            "Locked": false,
            "Modified By": "Steve Tester",
            "Modified On": "2023-08-04T10:30:58.137Z",
            "MoSCoW": "Not&#160;Prioritized",
            "Multi Value 1": "",
            "Name": "Login&#160;to&#160;ATM",
            "Notes": "",
            "Owner": "",
            "Post Condition": "<Some rich text Data>",
            "Pre Condition": "",
            "Project": "Kitchen-Sink",
            "Record Type": "Use&#160;Case",
            "Ref. ID": null,
            "State": "Draft",
            "Trigger": "",
            "Unique ID": 2220200821,
            "Version": "1.01"
        }
    }
}
```

For an explanation of the common fields in the response, refer to Important fields of all Repository Artifacts / Issues. Any fields not explained are intended for internal use and should be ignored.

Delete Artifact

1. Delete an Artifact or Package
2. Delete Multiple Artifacts

Delete an Artifact or a Package

Purpose	Delete an artifact or a package.
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/byid/*<recordId>*
Request Method	**DELETE**

Dynamic Path Variables:

Variable Name	Type	Description
recordId	Resource Id	*recordId* of the artifact

Request Body:

Field	Mandatory	Type	Allowed Values	Description
recordRevisionNumber	Yes	Number		Number of times an artifact has been updated.

Example 1:

DELETE http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byId/2220200821

```
{
"recordRevisionNumber" : 2
}
```

Response:

```
{
    "success": true,
    "response": {
        "name": "Login to ATM1edit 1",
        "recordId": 2220200821,
        "id": "UC-14781",
        "projectRecordId": 376523234,
        "parentRecordId": 2220200815,
        "ownerRecordId": 2220200815,
        "recordVersionId": 2220200838,
        "recordTypeId": 170,
        "recordRevisionNumber": 4
    }
}
```

For an explanation of the common fields in the response, refer to Important fields of all Repository Artifacts / Issues. Any fields not explained are intended for internal use and should be ignored.

Example 2:

When you sent incorrect recordId for delete operation:

DELETE http://myCompany.com/rest/ttmrestsrv.dll/2/ repositoryObjects/byId/2220200

Response :

```
{
    "success": false,
    "response": "",
    "error": {
        "requestedurl":"": "localhost:2673/2/repositoryObjects/byId/
2220200",
        "error": "Unable to access the record.",
        "type": "etInfo",
        "code": 404,
        "stack": "",
        "solution": "It may have been deleted or you do not have per
mission to view it."
    }
}
```

Fields for failed Delete Operation

Field Name	Type	Description
success	Boolean	Indicates whether operation on an artifact was successful. False value indicates failure.
error	Object	Refer Error Object This field is sent when success is false.

Delete Multiple Artifacts or Packages

Purpose	Delete multiple artifacts or packages
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/projects/*<project Id>*/REP
Request Method	**DELETE**

Dynamic Path Variables:

Variable Name	Type	Description
projectid	Resource Id	*recordId* of the project
recordTypeIdPrefix	String	*idPrefix of recordType* the record type that you want to fetch. e.g., 'UC', 'PREQ', and more.

Request Body:

Array of Artifact Object

Fields of Artifact Object

Field	Mandatory	Type	Allowed Values	Description
recordId	Yes	Resource Id		*recordId* of an artifact that you want to edit.
recordRevisionNumber	Yes	Number		Number of times an artifact has been updated.

Example 1:

DELETE http://myCompany.com/rest/ttmrestsrv.dll/2/projects/376523234/REP

```
[
  {
    "recordId":1498854,
    "recordRevisionNumber":1
  },
  {
    "recordId":1498858,
    "recordRevisionNumber":1
  }
]
```

Response:

```
{
    "success": true,
    "response": {
        "children": [
            {
                "recordId": 2220192922,
                "recordRevisionNumber": 2,
                "success": true,
                "type": "etInfo",
                "tsLogMessage": "Record has been deleted successfull
y."
            },
            {
                "recordId": 376526350,
                "recordRevisionNumber": 10,
                "success": true,
                "type": "etInfo",
                "tsLogMessage": "Record has been deleted successfull
y."
            }
        ]
    }
}
```

For an explanation of the common fields in the response, refer to Important fields of all Repository Artifacts / Issues. Any fields not explained are intended for internal use and should be ignored.

Example 2:

When you send an already deleted recordId in request.

DELETE http://myCompany.com/rest/ttmrestsrv.dll/2/projects/376523234/REP

```
[
  {
    "recordId":2220198861,
    "recordRevisionNumber":1
  },
  {
    "recordId":376524424,
    "recordRevisionNumber":1
  }
]
```

Response:

```
{
    "resultContentType": "json",
    "success": true,
    "response": {
        "children": [
            {
                "recordId": 2220198861,
                "recordRevisionNumber": 1,
                "success": false,
                "error": "Cannot perform this operation.",
                "tsLogMessage": "Cannot perform this operation.",
                "tsErrSolution": "The record could not be opened for
 one of the following reasons:\r\n- The record does not exist.\r\n-
You do not have team member privileges in this project.",
                "type": "etInfo",
                "errorCode": 0
            },
            {
                "recordId": 376524424,
                        "recordRevisionNumber": 1,
                        "success": true,
                        "type": "etInfo",
                        "tsLogMessage": "Record has been deleted successfull
y."
                  }
            ]
      }
}
```

FieldsField for failed Delete Operation:

Field Name	Type	Description
success	Boolean	Indicates whether operation on an artifact was successful. False value indicates failure.
error	String	Error message.
tsLogMessage	String	Log message.
tsErrSolution	String	Reason or solution of the error.
type	String	Type of error. Possible values are: etInfo, etError, or etWarning.
errorCode	Number	Error code of the error.

For an explanation of the common fields in the response, refer to Important fields of all Repository Artifacts / Issues. Any fields not explained are intended for internal use and should be ignored.

Copy or Reuse Artifacts

Purpose	Copy or reuse a set of artifacts.
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/copyRecords
Request Method	**POST**

Request Body

Field	Mandatory	Type	Allowed Values	Description
operation	Yes	String	Copy, Reuse	Specifies action that needs to be performed on selected artifacts. **Copy**: Creates a copy of the selected artifacts. **Reuse**: Reuses the selected artifacts at the provided copy location. If an invalid value is provided, the system defaults to 'Copy'.
recordsToCopy	Yes	Array of records		Contains the details of artifacts you want to copy or reuse.
copyDestination	Yes	copy_location_obj		Specifies the destination where artifacts will be copied or reused.
traceLinksCopyOptions	No	Trace link Copy Option		Determines how the system copies the links associated with the artifacts specified in *recordsToCopy*. Applicable only when the *action* is *Copy*.
reuseType	Yes	String	ReuseAsInstance, ReuseAsReference	Indicates the type of reuse when an artifact is reused from a source artifact.

				ReuseAsReference: Creates a reference for selected artifacts, serving as a shortcut to the source. Any changes made to the reference will be reflected in the source. This option is not subject to review and is particularly useful for breaking down higher-level requirements from another package into lower-level ones in your package. **ReuseAsInstance**: Creates independent instances of selected artifacts, which are separate from their source. Fields of these instances synchronize with the source based on the chosen update method. Instances can be delinked for project-specific changes. Unlike references, instances can have their own comments and links, and they can be sent for review. This option is ideal for reusing

				artifacts from a shared library project.
reuseUpdateMethod	No	String	Manual, Automatic	determines the synchronization method for the reused records with their source. **Manual**: You must explicitly execute the *Update From Source* option to synchronize changes from the source artifact. **Automatic:** The system will automatically synchronize changes from the source artifact.
fields	No	string	Comma text fields	Specifies which fields of the copied artifacts should be returned in the response of this end point. If not specified, the system will return the basic artifact information.

Copy Record Json Object

Field	**Mandatory**	**Type**	**Allowed Values**	**Description**
recordId	Yes	Resource Id		Specify the *recordId* of source which you want to copy.

copyChildren	No	Boolean	true false	Specify if you want copy the child artifacts or not. Default values will be 'false'.

Copy Destination Json Object

Field	**Mandatory**	**Type**	**Allowed Values**	**Description**
recordId	Yes	Resource Id		Specify the *recordId* of artifact below which you want to copy artifacts.
position	No	String	lastChild, siblingAfter	**lastChild:** This option appends the copied artifacts as the last children of the specified artifact. **siblingAfter:** This option places the copied artifacts immediately after a specified artifact at the same hierarchical level, effectively adding them as sequential peers.

Trace Links Copy Options Json Object

Field	Mandatory	Type	Allowed Values	Description
copyOutboundLinks	No	Boolean	true false	This option copies the links from the selected artifacts to external artifacts. It replicates the outward connections of the chosen artifacts, preserving the links that extend from the selected group to other artifacts in the system. Default value is 'false'.
copyIntraLinks	No	Boolean	true false	This option copies all the links existing between the selected artifacts. Default value is 'false'.

Example for copy records

POST http://myCompany.com/rest/ttmrestsrv.dll/2/copyRecords

```
{
  "operation": "copy",
   "recordsToCopy": [
    {
      "recordId": 376524210,
      "copyChildren": true
   },
  "copyDestination":{
      "recordId" : 376523234,
      " position":"tsLastChild"
  },
  "traceLinksCopyOptions":{
       "copyOutboundLinks" : false,
      "copyIntraLinks":true
   },
  "fields": "description",

  ]
}
```

Copy Records Response

```
{
    "resultContentType": "json",
    "success": true,
    "response": {
        "failedRecords": [],
        "recordsCopied": {
           "records": [
                {
                    "name": "Actors-Copy10",
                    "recordId": 7229284806,
                    "versionId": 7229284807,
                    "recordTypeId": 1370,
                    "id": "PKG-22686",
                    "ownerRecordId": 376523234,
                    "recordRevisionNumber": 1,
                    "recordParentId": 376523234,
                    "isPublished": false,
                    "createdByUserId": 7229184850,
                    "lastUpdatedByUserId": 7229184850,
                    "fields": {},
                    "filterSatisfied": true,
                    "isOwnerCollection": true,
                    "ownerRecordTypeId": 370,
                    "ownerRecordTypeIndicator": "PRJ",
                    "etpDispIdPrefix": "PKG",
                    "children": [
                        {
                            "name": "Customer",
                            "recordId": 7229284821,
                            "versionId": 7229284822,
                            "recordTypeId": 180,
                            "id": "ACTR-22687",
                            "ownerRecordId": 7229284806,
                            "recordRevisionNumber": 1,
                            "recordParentId": 7229284806,
                            "isPublished": false,
                            "createdByUserId": 7229184850,
                            "lastUpdatedByUserId": 7229184850,
                            "fields": {},
                            "filterSatisfied": true,
                            "isOwnerCollection": true,
                            "ownerRecordTypeId": 1370,
                            "ownerRecordTypeIndicator": "CLL",
                            "etpDispIdPrefix": "ACTR",
                            "children": []
                        }
                    ]
                }
            ]
         }
     }
}
```

Example for reuse records

POST http://myCompany.com/rest/ttmrestsrv.dll/2/copyRecords

```
{
  "operation": "Reuse",
  "fields": "description",
  "reuseType" : "ReuseAsInstance",
  "reuseUpdateMethod":"Automatic",
  "copyDestination":{
      "recordId" : 1340673,
      "position":"lastChild"
  },
  "recordsToCopy": [
    {
      "recordId": 1340679,
      "copyChildren": false
    }
  ]
}
```

Refer to Request Body of copy/reuse records to know explanation of the fields in the request.

Response for reuse records

```
{
    "resultContentType": "json",
    "success": true,
    "response": {
        "failedRecords": [],
        "recordsCopied": {
            "records": [
                {
                    "name": "Add  multiple Customers",
                    "recordId": 3592324632,
                    "versionId": 1364864,
                    "recordTypeId": 790,
                    "id": "FREQ-2884",
                    "ownerRecordId": 1329371,
                    "recordRevisionNumber": 1,
                    "recordParentId": 1340673,
                    "reuseStatus": "UptoDate",
                    "reuseType": "ReusedAsInstance",
                    "reuseSourceRecordId": 1340679,
                    "reuseScope": "Individual",
                    "reuseUpdateMethod": "Automatic",
                    "isPublished": false,
                    "createdByUserId": 132322,
                    "checkedOut": false,
                    "lastUpdatedByUserId": 132328,
                    "fields": {
                        "description": "<Description>"
                    },
                    "filterSatisfied": true,
                    "isOwnerCollection": true,
                    "ownerRecordTypeId": 1370,
                    "ownerRecordTypeIndicator": "CLL",
                    "etpDispIdPrefix": "FREQ",
                    "children": [],
                    "leaf": true,
                    "index": 1
                }
            ]
        }
    }
}
```

Failed Response when user tries to reuse records in incorrect package structure

```
{
    "resultContentType": "json",
    "success": true,
    "response": {
        "hierarchyResequenced": false,
        "failedRecords": [
            {
                "recordId": 1340679,
                "success": false,
                "name": "Add  multiple Customers",
                "tsErrSolution": "",
                "type": "etError",
                "errorCode": 500
            }
        ],
        "recordsCopied": {}
    }
}
```

Refer to Important fields of all Repository Artifacts / Issues to know explanation of the fields in the response. Fields that are not explained are for internal usage and should be ignored.

Fetch Artifact Permissions

Purpose	Use this to find the permissions that a logged-in user has on an artifact.
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/byId/*<recordId>*/permissions
Request Method	**GET**

Dynamic Path Variables:

Variable Name	Type	Description
recordId	Resource Id	*recordId* of the artifact

Request Body: Not Required

Example :

GET
http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byId/376526667/permissions

Response:

```
{
    "success": true,
    "response": {
        "canAddRemoveAttachments": true,
        "canAddRemoveComments": true,
        "canCheckIn": false,
        "canCheckOut": true,
        "canConvertRecordType": true,
        "canCreateRecord": true,
        "canDeleteRecord": true,
        "canForceCheckIn": false,
        "canLockUnLockRecord": true,
        "canModifyRecord": true,
        "canAddRemoveDiscussion": true,
        "canAddRemoveTraceability": true,
        "canCancelCheckOut": false,
        "canLockRecord": true,
        "canUnlockRecord": false,
        "canAddToBaseline": false,
        "canForceLock": false,
        "canAddToRelease": false,
        "canSendNotification": true,
        "canMoveToAnotherProject": true,
        "canIncVersion": true,
        "canMove": true,
        "canChangeState": true,
        "canChangeOwner": true,
        "canChangeParent": true,
        "canAddFolder": true,
        "canEditFolder": true,
        "canDeleteFolder": true,
        "canAddRequirementType": false,
        "canAddRemoveLinks": true,
        "canRankBacklog": true,
        "canAddRecordToBacklog": true,
        "canRemoveRecordFromBacklog": true
    }
}
```

Fetch Allowed Values for Fields of Artifact

Purpose	Use this to know values allowed in the fields that accept a limited set of values, e.g., *List field, Multi-Value Field, User Field, Team Member Field, Project Field*, and *State Field*.
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/projects/*<projectId>*/*<recordTypeIdPrefix>*/*<recordId>*/fields/allowedValues
Request Method	**GET**

Dynamic Path Variables:

Variable Name	Type	Description
projectid	Resource Id	*recordId* of the project
recordTypeIdPrefix	String	*idPrefix of recordType* the record type for which you want to fetch allowed values, such as 'UC', 'PREQ', and more
recordId	Resource Id	*recordId* of the artifact

Supported Query Parameters:

Parameter Name	Mandatory	Allowed Values	Default Value	Description
fields	No	comma-separated list of fields	All	The *fields* param specifies which fields' allowed values are to be fetched. All fields are fetched if this param is not passed. For Example, `fields=Complexity, Functional Area`

Request Body: Not Required

Example:

GET

http://myCompany.com/rest/ttmrestsrv.dll/2/projects/376523234/UC/2220199464/fields/allowedValues

Response:

```
{
    "success": true,
    "response": {
        "projectRecordId": 376523234,
        "recordId": 2220199464,
        "allowedValues": {
            "MoSCoW": [
                {
                    "recordId": 1,
                    "value": "Must Have",
                    "isSelected": false,
                    "imageIndex": 0
                },
                {
                    "recordId": 2,
                    "value": "Should Have",
                    "isSelected": false,
                    "imageIndex": 1
                },
                {
                    "recordId": 3,
                    "value": "Could Have",
                    "isSelected": false,
                    "imageIndex": 2
                },
                {
                    "recordId": 4,
                    "value": "Won't Have",
                    "isSelected": false,
                    "imageIndex": 3
                },
                {
                    "recordId": 1880,
                    "value": "Not Prioritized",
                    "isSelected": true,
                    "imageIndex": -1
                }
            ],
            "State": [
                {
                    "recordId": 4892,
                    "value": "Accepted",
                    "isSelected": false,
                    "imageIndex": 15
                },
                {
                    "recordId": 4866,
                    "value": "Approved",
                    "isSelected": false,
                    "imageIndex": 16
                },
                {
                    "recordId": 4894,
                    "value": "Completed",
                    "isSelected": false,
                    "imageIndex": 15
                }
             ]
         }
   }
}
```

Package Endpoints

1. Fetch Packages
2. Fetch Packages By Record Type
3. Create Package
4. Fetch Package Configuration Detail
5. Edit Package Configuration

Fetch Packages

Purpose	Fetch packages from projects with specified filters.
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/projects/<projectId>/packages
Request Method	**GET**

Supported Query Parameters:

Parameter Name	Mandatory	Allowed Values	Default Value	Description
recordTypes	NO	recordTypeIds	None	Fetch all packages that include the specified record type IDs as included record types.
packageTypes	NO	recordTypeIds	None	Record Type Ids of packages that are to be fetched.

Request Body: Not Required

Example: GET

http://myCompany.com/rest/ttmrestsrv.dll/2/projects/729212320174?packageTypes=1370&recordTypes=1060

Response:

```
{
    "resultContentType": "json",
    "success": true,
    "response": {
        "Packages": [
            {
                "recordId": 729212320359,
                "name": "User Stories",
                "projectId": 729212320174,
                "recordParentId": 729212320174,
                "recordOwnerId": 729212320174,
                "recordTypeId": 1370,
                "recordRevisionNumber": 1,
                "packageCode": "",
                "packageStructureId": 0
            },
            {
                "recordId": 729212607575,
                "name": "User Stories for Project",
                "projectId": 729212320174,
                "parentId": 729212320174,
                "ownerId": 729212320174,
                "recordTypeId": 1370,
                "recordRevisionNumber": 3,
                "packageCode": "",
                "packageStructureId": 0
            }

        ]
    }
}
```

Explanation of Response Fields:

Field Name	Type	Description
packageStructureId	ResourceId	Details of the package structure applied to a package.
packageCode	String	Unique code for the package. The main purpose of the *packageCode* is to easily search or identify the package when working with a large number of packages in the repository.

Refer to Important fields of all Repository Artifacts / Issues to know explanation of the fields in the response. Fields that are not explained are for internal usage and should be ignored.

Fetch Packages By Record Type

Purpose	Fetch packages from projects with specified record types included.
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/projects/<projectId>/packagesForRecordType
Request Method	**GET**

Supported Query Parameters:

Parameter Name	Mandatory	Allowed Values	Default Value	Description
recordTypes	No	recordTypeIds	None	Fetch all packages that include the specified record type IDs as included record types.

Request Body: Not Required

Example: GET

http://myCompany.com/rest/ttmrestsrv.dll/2/projects/729212320174/packagesForRecordType?recordTypes=1060,170

Response:

```
{
    "resultContentType": "json",
    "success": true,
    "response": [
        {
            "recordTypeId": 170,
            "packages": [
                729213025453,
                729213025425
            ]
        },
        {
            "recordTypeId": 1060,
            "packages": [
                729213025425,
                729212607575,
                729213029675,
                729212320359,
                729212607604
            ]
        }
    ]
}
```

Explanation of Response Fields:

Field Name	Type	Description
recordTypeId	Resource Id	*recordId* of record types whose packages are fetched.

packages	Resource Id	List of record IDs of the packages contain the specified record type.

Create Package

Purpose	Create a new package artifact in *TopTeam* repository
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/packages
Request Method	**POST**

Request Body:

Field	**Mandatory**	**Type**	**Allowed Values**	**Description**
recordTypeId	No	Resource Id		Specify the *recordId* of a package record type that you want to create. Refer to the section [How to find list of Package Record Types in my repository]() for more details.
name	Yes	String		Provide package name.
packageCode	No	String		Provide package code.
includedRecordTypes	Yes	Array of RecordType objects		Provide list of record type objects as an array. Structure of Record Type Object JSON { “recordId” : <recordId of record type that you want to include>

				}
				
				E.g.
				"includedRecordTypes": [
				{
				"recordId": 170
				},
				{
				"recordId": 180
				}
				]

description	No	String		Specify description of the package in HTML format or in plain text.
newRecordOwnerId	Yes	Resource Id		*recordId* of the package in which you want to create this package. If you want to create the package directly under the Project, specify the *recordId* of the Project.
newRecordProjectId	Yes	Resource Id		*recordId* of the Project.
newRecordLocation	Yes	String	tsLastChild tsSiblingAfter	**tsLastChild**: Creates the package as the last child of the parent package. **tsSiblingAfter**: Creates the package after the artifact mentioned in *newRecordReferenceId* field
newRecordReferenceId	Yes	Resource Id		Specify the *recordId* of the artifact in parent package below or

				sibling of which you want to create the new package.

Example:

POST http://myCompany.com/rest/ttmrestsrv.dll/2/packages

```
{
  "recordTypeId": 1370,
  "name": "User Story for Rental Video Games",
  "description": "Package for Use Cases and Actors",
  "newRecordOwnerId": 376523234,
  "packageCode": "USFRVG",
  "newRecordProjectId":376523234,
  "newRecordLocation": "tsLastChild",
  "newRecordReferenceId": 376523234,
  "packageStructureId": 5713,
    "includedRecordTypes": [
    {
      "recordId": 170
    },
    {
      "recordId": 180
    }
  ]
}
```

NOTE	URI and JSON are formatted to improve readability. You may need to encode it for valid JSON.

Response:

```
{
    "resultContentType": "json",
    "success": true,
    "response":{
        "name":
"User&#160;Story&#160;for&#160;Rental&#160;Video&#160;Games",
        "recordId": 3592330157,
        "id": "PKG-43425",
        "forEtpId": 0,
        "projectRecordId": 376523234,
        "parentRecordId": 376523234,
        "ownerRecordId": 376523234,
        "recordVersionId": 3592330156,
        "recordTypeId": 1370,
        "recordRevisionNumber": 1,
        "packageCode": "USFRVG",
        "createdByUserId": 3592196196,
        "createdByUserName": "Steve Tester",
        "createDate": "2024-01-12T10:34:03.747Z",
        "id2": "PKG-43425",
        "lastUpdatedByUserId": 3592196196,
        "lastUpdatedByUserName": "Steve Tester",
        "lastUpdateDate": "2024-01-12T10:34:03.747Z",
        "description": "Package for Use Cases and Actors",
        "displaySequence": 904800000,
        "recordVersionDisp": "1",
        "packageStructure": {
            "packageStructureId": 5713,
            "apiEndpoint":"http://myCompany.com/rest/ttmrestsrv.dll/
2/packageStructure/5713"
        },
        "imageIndex": -1,
        "includedRecordTypes": [
            {
                "recordId": 170,
                "canCreateRecordInPackage": true,
                "canReuseRecordInPackage": true
            },
            {
                "recordId": 180,
                "canCreateRecordInPackage": true,
                "canReuseRecordInPackage": true
            }
        ]
    }
}
```

Explanation of Response Fields

Refer to Important fields of all Repository Artifacts / Issues to know explanation of the fields in the response. Fields that are not explained are for internal usage and should be ignored.

Fetch Package Configuration Detail

Purpose	Fetch the package configuration details.
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/packageConfiguration
Request Method	**GET**

Supported Query Parameters:

Parameter Name	Mandatory	Allowed Values	Default Value	Description
recordId	yes			*recordId* of a package

Request Body: Not Required

Example : GET

http://myCompany.com/rest/ttmrestsrv.dll/2/packageConfiguration?recordId=3592197269

Response:

```
{
    "resultContentType": "json",
    "success": true,
    "response": {
        "name": "Objective",
        "recordId": 3592197269,
        "id": "PKG-43335",
        "forEtpId": 180,
        "projectRecordId": 376523234,
        "parentRecordId": 376523234,
        "ownerRecordId": 376523234,
        "recordVersionId": 3592197270,
        "recordTypeId": 1370,
        "recordRevisionNumber": 16,
        "packageCode": "fsdf",
        "createdByUserId": 3592196196,
        "createdByUserName": "Steve Tester",
        "createDate": "2023-12-16T09:41:00.973Z",
        "id2": "fsdf-PKG-43335",
        "lastUpdatedByUserId": 3592196196,
        "lastUpdatedByUserName": "Steve Tester",
        "lastUpdateDate": "2024-01-12T09:53:36.497Z",
        "description": "",
        "displaySequence": 904500000,
        "recordVersionDisp": "1",
        "imageIndex": -1,
        "packageStructure": {
            "packageStructureId": 5713,
           "apiEndpoint":"http://myCompany.com/rest/ttmrestsrv.dll
           /2/packageStructure/5713"
        },
        "includedRecordTypes": [
            {
                "recordId": 180,
                "canCreateRecordInPackage": true,
                "canReuseRecordInPackage": true
            },
            {
                "recordId": 2310,
                "canCreateRecordInPackage": false,
                "canReuseRecordInPackage": false
            },
            {
                "recordId": 13227728,
                "canCreateRecordInPackage": true,
                "canReuseRecordInPackage": true
            }
        ],
        "qaPackageConfig": {
            "recordTypes": [
                "940"
            ]
        }
    }
}
```

Explanation of Response Fields

Field Name	Type	Description
packageStructure	packageStrcuture Object	Details of the package structure applied to a package.

recordRevisionNumber	Number	The number of times an artifact has been updated.
packageCode	String	Unique code for the package. The main purpose of the *packageCode* is to easily search or identify the package when working with a large number of packages in the repository.
includedRecordTypes	Array of Include Record Type Object	Contains the list of record types included in the package.
qaPackageConfig	Json Object	Contains the QA record types included in the package.

Package Structure Object

Field Name	Type	Description
packageStrcutureId	Resource Id	Unique ID of the package structure applied to a package.
apiEndpoint	Resource url	Resource URL to access the details of the package structure. Use this URL directly to fetch the details of a package structure such as name, included record types, permissions, etc.

Object of included recordtype

Field Name	Type	Description
recordId	Resource Id	*recordId* of a record type.
canCreateRecordInPackage	Boolean	Specifies whether an artifact can be created in this package.
canReuseRecordInPackage	Boolean	Specifies whether an artifact can be reused within the package.

Response Object of QA Configuration

Field Name	Type	Description
recordTypes	Resource Id	List of *recordIds* of *recordTypes* which are included in QA Package.

Refer to Important fields of all Repository Artifacts / Issues for explanations of the fields in the response. Fields that are not explained are for internal usage and should be ignored.

Edit Package Configuration

Purpose	Edit package configuration
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/packageConfiguration
Request Method	**PUT**

Supported Query Parameters:

Parameter Name	Mandatory	Allowed Values	Default Value	Description
recordId	yes			*recordId* of an artifact

Request Body:

Field	Mandatory	Type	Allowed Values	Description
recordRevisionNumber	Yes	Number		The number of times a package has been updated.
name	No	String		Provides the new package name if you intend to modify the existing package name.
description	No	String		Provides the updated package description if you intend to modify the existing package description.
packageStructureId	No	ResourceId		Specifies the *recordId* of the package structure that

				you want to apply to package.
qaPackageConfig	No	Json Object		Contains the QA record types that needs to be either included or removed from the package.
includedRecordTypes	No	Array of Object which contained recordId of recordtype		Contains the record types that needs to be included in the package.

Request Json Object for QA Configuration.

Field	Mandatory	Type	Allowed Values	Description
addRecordTypes	No	Array of recordType Id		Contains the QA *recordTypeIds* which you want to include in package.
removeRecordTypes	No	Array of recordType Id		Contains the QA recordTypeIds which you want to remove from package.

Example : PUT

http://myCompany.com/rest/ttmrestsrv.dll/2/packageConfiguration?recordId=3592197269

```
{
  "name" : "User Story for Video Player",
  "recordRevisionNumber": 32,
  "packageStructureId":5713,
  "qaPackageConfig": {
      "addRecordTypes": [
      890, 910],
      "removeRecordTypes": [950
        ]
  },
  "includedRecordTypes":[
    {"recordId" : 1060},
      {"recordId" : 180}
  ]
}
```

Response:

```
{
    "resultContentType": "json",
    "success": true,
    "response": {
        "name": "User Story for Video Player",
        "recordId": 3592197269,
        "id": "PKG-43335",
        "forEtpId": 180,
        "projectRecordId": 376523234,
        "parentRecordId": 376523234,
        "ownerRecordId": 376523234,
        "recordVersionId": 3592197270,
        "recordTypeId": 1370,
        "recordRevisionNumber": 33,
        "packageCode": "fsdf",
        "createdByUserId": 3592196196,
        "createdByUserName": "Steve Tester",
        "createDate": "2023-12-16T09:41:00.973Z",
        "id2": "fsdf-PKG-43335",
        "lastUpdatedByUserId": 3592196196,
        "lastUpdatedByUserName": "Steve Tester",
        "lastUpdateDate": "2024-01-12T09:53:36.497Z",
        "description": "",
        "displaySequence": 904500000,
        "recordVersionDisp": "1",
        "imageIndex": -1,
        "packageStructure": {
            "packageStructureId": 5713,
              "apiEndpoint":"http://myCompany.com/rest/ttmrestsrv.dll
/2/packageStructure/5713"
        },
        "includedRecordTypes": [
            {
                "recordId": 1060,
                "canCreateRecordInPackage": true,
                "canReuseRecordInPackage": true
            },
            {
                "recordId": 180,
                "canCreateRecordInPackage": false,
                "canReuseRecordInPackage": false
            }
        ],
        "qaPackageConfig": {
            "recordTypes": [
                "890",
                "910"
            ]
        }
    }
}
```

Explanation of Response Fields: Refer to Fetch Package Configuration Detail response structure.

Diagram Field Endpoints

TopTeam has many record types that contain diagrams such as *UML Diagram, SysML Diagram, Business Process Diagram*, etc. This section describes the endpoints that help you to get information from the *Diagram* field of record types that support diagramming.

Get Image of a Diagram

Purpose	Export diagram in a diagram field as an image
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/byid/*<recordId>*/fields/*<diagramFieldName>*/image
Request Method	**GET**

Dynamic Path Variables:

Variable Name	Type	Description
recordId	Resource Id	*recordId* of the artifact

diagramFieldName	String	Specify the caption given to a *Diagram* field of the concerned record type. Generally, the caption is “Diagram”.

Supported Query Parameters:

Parameter Name	Mandatory	Allowed Values	Default Value	Description
tsRecordId	Yes	Resource Id	-	This is a mandatory param for fetching diagram as an image for a specific version.

Request Body: Not Required

Example :

GET http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byId/376526667/fields/diagram/image

Response Content-Type:

application/octet-stream

Response Stream:

The server sends the image file as a content stream in the response. The image is in jpeg format. The name of the image file is given in the response header “Content-Disposition”.

File Endpoints

Download File

Purpose	Downloads document associated with the artifact
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version> /repositoryObjects/byId/*<fileRecordId>*/file
Request Method	**GET**

Dynamic Path Variables:

Variable Name	Type	Description
fileRecordId	Resource Id	*recordId of the File artifact.*

Request Body: Not Required

Example:

GET http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byId/376526667/file

Response Content-Type:

application/octet-stream

Response Stream:

The server sends the file as a content stream in the response. The image is in jpeg format. The name of the image file is given in the response header "Content-Disposition".

Use Case Endpoints

1. Fetch Use Case Actors
2. Download Activity Diagram

Fetch Use Case Actors

Purpose	Get a list of actors linked to *a Use Case*
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/byid/*<recordId>*/useCaseActors
Request Method	**GET**

Dynamic Path Variables:

Variable Name	Type	Description
recordId	Resource Id	*recordId* of the Use Case

Request Body: Not Required

Example:

GET

http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byid/376526034/useCaseActors

Response:

```
{
  "success": true,
  "response": {
    "useCaseActors": [
      {
        "name": "Store Clerk",
        "recordId": 14999999997872
        "id": "ACTR-753",
        "projectId": "PRJ-711",
        "isSupportingActor": false,
        "isPrimaryActor": true,
      },
      {
        "name": "Store Manager",
        "recordId": 14999999997876
        "id": "ACTR-757",
        "projectId": "PRJ-711",
        "isSupportingActor": false,
        "isPrimaryActor": true,
      }
    ]
  }
}
```

Download Activity Diagram

Purpose	Download *Activity Diagram* as an image
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/byId/*<recordId>*/fields/Activity Diagram/image
Request Method	**GET**

Dynamic Path Variables:

Variable Name	**Type**	**Description**
recordId	Resource Id	*recordId* of Use

Request Body: Not Required

Example:

GET
http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byId/648443197/fields/Activity Diagram/image

Response Content-Type:

application/octet-stream

Response Stream:

The server sends the image file as a content stream in the response. The image is in jpeg format. The name of the image file is given in the response header "Content-Disposition".

Link Endpoints

1. Fetch Links for an Artifact
2. Add Links for an Artifact
3. Delete Links for an Artifact
4. Make Suspect Link – Clear Suspect Link

Fetch Links for an Artifact

Purpose	Get links of an artifact
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/byid/*<recordId>*/links
Request Method	**GET**

Dynamic Path Variables:

Variable Name	**Type**	**Description**
recordId	Resource Id	*recordId* of the artifact

Supported Query Parameters:

Parameter Name	Mandatory	Allowed Values	Default Value	Description
baselineId	No	Resource Id	-	Specify the *recordId* of a baseline if you want to fetch links of an artifact from a particular baseline. This field is optional. If not passed, the endpoint will return the links of current version of an artifact. Example: baselineId=6484150277 To get the *recordId* of a baseline, refer to Get list of baselines in a project.

Request Body: Not Required

Example:
GET http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byid/2220200176/links

Response:

```
{
    "success": true,
    "response": {
        "projectRecordId": 376523234,
        "recordId": 2220200176,
        "links": [
            {
                "name": "Store&#160;Clerk",
                "recordId": 2220203316,
                "id": "ACTR-10183",
                "isSuspect": false,
                "recordRevisionNumber": 1,
                "linkTypeId": 220,
                "isReverse": false,
                "isSystem": true,
                "linkedRecordId":6484157845,
                "projectRecordId": 6484152941,
                "recordTypeId": 180,
                "permissions": {
                    "canDelete": true,
                    "canSuspect": true,
                    "canClearSuspect": false
                }
            },
            {
                "name": "Customer",
                "recordId": 2220203212,
                "id": "ACTR-10185",
                "isSuspect": false,
                "recordRevisionNumber": 1,
                "linkTypeId": 220,
                "isReverse": false,
                "isSystem": true,
                "linkedRecordId":376524412,
                "projectRecordId": 376523234,
                "recordTypeId": 180,
                "permissions": {
                    "canDelete": true,
                    "canSuspect": true,
                    "canClearSuspect": false
                }
            }
        ]
    }
}
```

Example with baselineId parameter:

GET http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byid/2220200176/links?baselineId=6484150277

Response:

```
{
    "success": true,
    "response": {
        "projectRecordId": 376523234,
        "recordId": 2220200176,
        "links": [
            {
                "name": "Store&#160;Clerk",
                "recordId": 2220203316,
                "id": "ACTR-10183",
                "isSuspect": false,
                "recordRevisionNumber": 1,
                "linkTypeId": 220,
                "isReverse": false,
                "isSystem": true,
                "linkedRecordId":6484157845
                "projectRecordId": 6484152941,
                "recordTypeId": 180,
                "permissions": {
                    "canDelete": false,
                    "canSuspect": false,
                    "canClearSuspect": false
                }
            }
        ]
    }
}
```

Explanation of Response Fields

Field Name	Type	Description
name	String	Name of the linked artifact
linkTypeId	Resource Id	recordId of the link type
isReverse	Boolean	Indicates whether the link is upstream or downstream direction. If true, link is in upstream direction. If false, link is in downstream direction.
recordId	Resource Id	*recordId* of link.
linkedRecordId	Resource Id	*recordId* of the linked artifact
projectRecordId	Resource Id	*recordId* of the project of the linked artifact.
isSuspect	Boolean	Indicates whether the link is suspect

permissions	JSON Object	Provides the permission that logged-in user has on an artifact.
isSystem	Boolean	Indicates whether the link is a system link. True value indicates that link is system link and cannot be *Edited* or *Deleted* by the user.

For an explanation of the common fields in the response, refer to Important fields of all Repository Artifacts / Issues. Any fields not explained are intended for internal use and should be ignored.

Add Links for an Artifact

Purpose	Add links for an artifact
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/byid/*<recordId>*/links
Request Method	**POST**

Dynamic Path Variables:

Variable Name	Type	Description
recordId	Resource Id	*recordId* of the artifact

Request Body:
Array of Link record Object

Fields of Link Record Object:

Field	Mandatory	Type	Allowed Values	Description
recordId	Yes	Resource Id		Specify the *recordId* of an artifact with which you want to create the link.

linkTypeId	Yes	Resource Id		Specify the *recordId* of a *Link Type* that you want to create between the artifact. To get the *recordId* of a *Link Type*, refer Link Types endpoint.
isReverse	Yes	Boolean	true / false	This flag is used to identify the direction of the link that is getting created, i.e., *Upstream* or *Downstream*. Specify the value *false* if you want to create the *Downstream* link. Specify the value *true* if you want to create the *Upstream* link.

Example:

POST http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byid/2220200176/links

```
[
  {
    "recordId": 2220199511,
    "linkTypeId":220,
    "isReverse": false
  },
  {
    "recordId": 2220199517,
    "linkTypeId":220,
    "isReverse": true
  }
]
```

Response:

```
{
    "resultContentType": "json",
    "success": true,
    "response": {
        "links": [
            {
              "success": true,
              "recordId": 7229190200,
              "linkTypeId": 220,
              "isSuspect": false,
              "isReverse": false,
              "isSystem": false,
              "recordRevisionNumber": 1,
              "linkedRecordId": 2220199511,
              "recordTypeId": 1060
            },
            {
              "success": true,
              "recordId": 7229190205,
              "linkTypeId": 220,
              "isSuspect": false,
              "isReverse": false,
              "isSystem": false,
              "recordRevisionNumber": 1,
              "linkedRecordId": 2220199517,
              "recordTypeId": 1090
            }
        ]
    }
}
```

Explanation of Response Fields

Field Name	Type	Description
success	Boolean	Indicates whether operation on an artifact was successful. False value indicates failure.
message	String	Determines the reason why operation on artifact get failed.
type	String	Type of error. Possible values are: etInfo, etError, or etWarning.

For an explanation of the common fields in the response, refer to Important fields of all Repository Artifacts / Issues. Any fields not explained are intended for internal use and should be ignored.

Delete Links for an Artifact

Purpose	Delete single/multiple links for an artifact
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/byid/*<recordId>*/links
Request Method	**DELETE**

Dynamic Path Variables:

Variable Name	Type	Description
recordId	Resource Id	*recordId* of the artifact

Request Body:

Array of Link Record Object

Fields of Link Record Object

Field	Mandatory	Type	Allowed Values	Description
recordId	Yes	Resource Id		*recordId* of a link that you want to delete. To get the *recordId* of a *Link Type*. Refer to artifacts Links Endpoint
recordRevisionNumber	Yes	number		Number of times an artifact has been updated.

Example:

DELETE

http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byId/376523234/links

```
[
    {
        "recordId":7229190205,
        "recordRevisionNumber":1
    },
    {
        "recordId":7229190200,
        "recordRevisionNumber":1
    }
]
```

Response:

```
{
    "resultContentType": "json",
    "success": true,
    "response": [
        {
            "recordId": 7229190200,
            "success": true
        },
        {
            "recordId": 7229190205,
            "success": true
        }
    ]
}
```

Explanation of Response Fields

Field Name	Type	Description
recordId	Resource Id	*recordId* of link that is deleted.
success	Boolean	Indicates whether link deletion was successful. False value indicates failure.
error	Object	Refer Error Object This field is sent when success is false.

Make Suspect Link – Clear Suspect Link

Purpose	Make supect link – Clear suspect link
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/byid/*<recordId>*/links
Request Method	**PUT**

Dynamic Path Variables:

Variable Name	Type	Description
recordId	Resource Id	*recordId* of the link

Request Body:

Field	Mandatory	Type	Allowed Values	Description
recordId	Yes	Resource Id		Specify the recordId of link which you want to edit.
isSuspect	Yes	Boolean	True,false	Please specify whether you want to make a suspect link or not.
recordRevisionNumber	Yes	number		The number of times a record has been updated.

Example: PUT

http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byId/7229194494/makelink suspect

```
[
   {
            "recordId":7229198033,
            "isSupect":true,
      "recordRevisionNumber":9     },
         {
            "recordId":7229198054,
            "isSupect":false,
        "recordRevisionNumber":9
    }

]
```

Response:

```
{
    "resultContentType": "json",
    "success": true,
    "response": [
        {
            "recordId": 7229198033,
            "recordRevisionNumber": 10,
            "canEdit": true,
                      "isSupect":true,
            "canDelete": true,
            "success": true
        },
           {
            "recordId": 7229198054,
            "recordRevisionNumber": 10,
            "canEdit": true,
                      "isSupect":false,
            "canDelete": true,
            "success": true
        }

    ]
}
```

Refer Response structure links to know explanation of the other fields in the response. Fields that are not explained are for internal usage and should be ignored.

Traceability Endpoints

Nested Traceability Query

Purpose	This endpoint retrieves linked records in hierarchical fashion up to a certain depth.
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/ /nestedTraceQuery
Request Method	**POST**

Request Body:

Field	Mandatory	Type	Allowed Values	Description
rootRecords	Yes	Root Records Query		Artifacts for which you want to query links.
queryDirection	Yes	String	upstream, downstream	Direction of Links that we need to fetch for *rootRecords* Default value is 'downstream' if not specified.
queryDepth	Yes	number	Between 1-15	Determines the depth of link traversal in the query. It sets how many levels deep the API should follow linked artifacts, enabling a more extensive search with higher values.
linkFilter	No	Link Filter Object		This filter specifies criteria for linked

				artifacts, applied at each level of the query.

Json Object for querying root records

Field	Mandatory	Type	Allowed Values	Description
queryMethod	Yes	String	ByLocation, ByRecordIds	Specifies how the root artifacts should be queried. **ByLocation**: Queries child artifacts of Package, Project, Oneview or another artifact. **ByRecordIds**: Queries a set of artifacts by *recordId*. Default method is 'ByLocation'.
recordLocationId	Yes (if queryMethod is *byLocation*)	ResourceId		Specifies the *recordId* whose child artifacts you want to query.
recordIds	Yes (if queryMethod is *ByRecordIds*)	String	Comma text recordIds of artifacts	Specifies the *recordIds* for which you want query links.
recordTypeIds	No	String	Comma text recordTypeIds	Specifies the *recordTypeIds* of the record types

				you want to query. This parameter is applicable only when *queryMethod* is *ByLocation*.

Json Object for Link Filter

Field	Mandatory	Type	Allowed Values	Description
linkTypeIds	No	String	Comma text *recordId* of link types.	Specifies *recordId* of the link types that you want to query.
linkedRecordProjectIds	No	String	Comma text *recordId* of projects.	Specifies *recordId* of the projects from which you want to query the linked artifacts. If not specified, the query fetches linked artifacts from all projects where the user is a team member.
linkedRecordTypeIds	No	String	Comma text *recordId* of record types.	This parameter helps you to filter linked artifacts based on their record type.

Example:

POST http://myCompany.com/rest/ttmrestsrv.dll/2/nestedTraceQuery

Example for querymethod ‘byLocation’:-

Request:

```
{
      "queryDirection":"upStream",
      "queryDepth": 5,
      "rootRecords":{
            "queryMethod": "byLocation",
            "recordLocationId": 7229262267 ,
            "recordTypeIds": "1060”
      },
      "linkFilter": {
            "linkTypeIds": "7229196230",
            "linkedRecordProjectIds":"6484151194,376520051",
            "linkedRecordTypeIds":"180"
      }
}
```

Response:

```
{
    "resultContentType": "json",
    "success": true,
    "response": {
        "rootRecordIds": [
            7229262267
        ],
        "records": [
            {
                "name": "ACT 2",
                "recordId": 7229262267,
                "recordTypeId": 180,
                "projectRecordId": 376523234,
                "isSuspect": false,
                "parentRecordId": 7229262207,
                "packageRecordId": 7229262207,
                "upStreamLinks": [
                    {
                        "linkId": 7229262272,
                        "linkedRecordId": 72299999922375,
                        "linkTypeId": 7229196230,
                        "isSuspect": false
                    },
                    {
                        "linkId": 7229262277,
                        "linkedRecordId": 72299999922367,
                        "linkTypeId": 7229196230,
                        "isSuspect": false
                    }
                ]
            }
        ]
    }
}
```

Example for querymethod 'byRecordIds':-

Request:

```
{
  "queryDirection": "downStream",
  "queryDepth": 2,
  "rootRecords": {
    "queryMethod": "byRecordIds",
    "recordIds": "1340679, 1340722"
  },
  "linkFilter": {
    "linkedRecordTypeIds": "890, 1340773",
    "linkTYpeIds": "220"
  }
}
```

Response :

```
{
    "resultContentType": "json",
    "success": true,
    "response": {
        "rootRecordIds": [
            1340679,
            1340722
        ],
        "records": [
            {
                "name": "Add  multiple Customers",
                "recordId": 1340679,
                "recordTypeId": 790,
                "projectRecordId": 132363,
                "isSuspect": false,
                "parentRecordId": 1340673,
                "packageRecordId": 1329371,
                "downStreamLinks": [
                    {
                        "linkId": 1341403,
                        "linkedRecordId": 1341399,
                        "linkTypeId": 220,
                        "isSuspect": false
                    }
                ]
            },
            {
                "name": "Manage Rentals",
                "recordId": 1340722,
                "recordTypeId": 790,
                "projectRecordId": 132363,
                "isSuspect": false,
                "parentRecordId": 1340667,
                "packageRecordId": 1329371,
                "downStreamLinks": [
                    {
                        "linkId": 3592323352,
                        "linkedRecordId": 13241797,
                        "linkTypeId": 220,
                        "isSuspect": false
                    }
                ]
            },
            {
                "name": "Customer UC1 - for TC generate from UC -
TestCase 1",
                "recordId": 13241797,
                "recordTypeId": 890,
                "projectRecordId": 132363,
                "isSuspect": false,
                "parentRecordId": 13241793,
                "packageRecordId": 3592195530,
                "downStreamLinks": []
            },
            {
                "name":  "The  system  shall  allow  users  to  add  new
Customers",
                "recordId": 1341399,
                "recordTypeId": 1340773,
                "projectRecordId": 132363,
                "isSuspect": false,
                "parentRecordId": 1343364,
                "packageRecordId": 1329371,
                "downStreamLinks": []
            }
        ]
    }
}
```

Comment Endpoints

1. Fetch Comments
2. Add Comment
3. Edit Comment
4. Delete Comment

Fetch Comments

Purpose	Get comments made on an artifact
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/byId/*<recordId>*/comments
Request Method	**GET**

Dynamic Path Variables:

Variable Name	Type	Description
recordId	Resource Id	*recordId* of the artifact

Request Body: Not Required

Example:

GET
http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byId/2220199429/Comments

Response:

```
{
    "success": true,
    "response": {
        "projectRecordId": 376523234,
        "recordId": 2220199429,
        "comments": [
            {
                "recordId": 2220199861,
                "level": 1,
                "comment": "This record is changed",
                "commentType": "tsCommentType_Comment",
                "createDate": "2023-07-27T11:37:35.717Z",
                "lastUpdateDate": "2023-07-27T11:37:35.717Z",
                "parentRecordId": -1,
                "recordRevisionNumber": 1,
                "person": "Steve Tester",
                "authorId": 2220169265,
                "lastUpdatedByUserId": 2220169265,
                "newState": "",
                "oldState": "",
                "isPublic": true,
                "isOpen": true,
                "masterRecordVersionId": 2220199430,
                 "children": [],
                "permissions": {
                    "canOpen": false,
                    "canClose": true,
                    "canModify": false,
                    "canDelete": true
                }
            },
            {
                "recordId": 2220199866,
                "level": 2,
                "comment": "This Record is edited",
                "commentType": "tsCommentType_Comment",
                "createDate": "2023-07-27T11:37:50.777Z",
                "lastUpdateDate": "2023-07-27T11:37:50.777Z",
                "parentRecordId": 2220199861,
                "recordRevisionNumber": 1,
                "person": "Steve Tester",
                "authorId": 2220169265,
                "lastUpdatedByUserId": 2220169265,
                "newState": "",
                "oldState": "",
                "isPublic": true,
                "isOpen": false,
                "masterRecordVersionId": 2220199430,
                 "children": [],
                "permissions": {
                    "canOpen": true,
                    "canClose": false,
                    "canModify": false,
                    "canDelete": false
                }
            }
        ]
    }
}
```

Fields of a fetch comments endpoint

Field Name	**Type**	**Description**
recordId	Resource Id	recordId of a comment.
parentRecordId	Resource Id	recordId of comment to which this comment is a reply.
comment	String	Comment given by a user.
isPublic	Boolean	True Indicates whether the comment is visible to collaborators and viewers. False indicates that the comment is visible only to author users.
isOpen	Boolean	Indicates whether the comment is in open or closed state.

person	String	Display name of the user who commented.
authorId	Resource Id	*recordId* of the user who commented.
recordRevisionNumber	Number	Number of times a comment has been updated.
permissions	JSON Object	Provides the permission that logged-in user has on an artifact.
masterRecordVersionId	Resource Id	Unique Id of an artifact version on which the comment was given. You can use this to query the details of the version using the endpoint Fetch version of an artifact.

For an explanation of the common fields in the response, refer to Important fields of all Repository Artifacts / Issues. Any fields not explained are intended for internal use and should be ignored.

Add Comment

Purpose	Add Comment to the artifact
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/byId/*<recordId>*/comments
Request Method	**POST**

Dynamic Path Variables:

Variable Name	**Type**	**Description**
recordId	Resource Id	*recordId* of the artifact

Request Body:

Field	**Mandatory**	**Type**	**Allowed Values**	**Description**

comment	Yes	String		Comment text that needs to be added
isPublic	No	Boolean	true false	Determines if comment is public or not.
parentRecordId	No	Resource Id		Specify the recordId of the parent comment to which user want to reply.
notifyTo	No	Object for UserIds	"userIds" Array	Object which contains the array of user ids to send the notification for user

Object for UserIds

Field	**Mandatory**	**Type**	**Allowed Values**	**Description**
userIds	No	Array of UserIds		It contains the list of userIds to who you want to notify.

Example: POST

http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byId/2220199429/Comments

```
{
  "comment": "Hello @Sagar",
  "isPublic": false,
  "parentRecordId": 7229213156,
   "notifyTo":{
         "userIds":[7229184850, 7229207751]
  }
}
```

Response:

```
{
    "resultContentType": "json",
    "success": true,
    "response": {
        "recordId": 7229213162,
       -"masterRecordVersionId": 7229184951,
        "comment": "Hello&#160;@Sagar",
        "recordRevisionNumber": 2,
        "isPublic": false,        "parentRecordId": 7229213156,
        "isOpen": false,
        "commentedByUserId": 7229184850,
        "contextText": "",
        "isContextualComment": false,
        "permissions": {
            "canDelete": true,
            "canModify": true,
            "canOpen": true,
            "canClose": false
        },
        "notifyTO": {
                "userIds": [
                    7229184850,
                    7229207751
                ]
        },
        "createDate": "2023-10-18T14:37:55.500Z",
        "lastUpdateDate": "2023-10-18T14:37:55.528Z"
    }
}
```

Refer Response Structure Comments to know explanation of the other fields in the response. Fields that are not explained are for internal usage and should be ignored.

Edit Comment

Purpose	Edit Comment
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/byId/*<recordId>*/comments /*<commentId>*
Request Method	**PUT**

Dynamic Path Variables:

Variable Name	**Type**	**Description**
recordId	Resource Id	*recordId* of the artifact
commentId	ResourceId	*recordId* of the comment

Request Body:

Field	**Mandatory**	**Type**	**Allowed Values**	**Description**
comment	Yes	String		Comment to the artifact
recordRevisionNumber	Yes	Number		The number of times a record has been updated.
operationType	Yes	String	'CloseComment' 'OpenComment' 'MakeCommentPublic' 'MakeCommentInternal' 'EditComment'	**CloseComment:**To Close the Comment use operationType 'CloseComment. **OpenComment :** To Open the Comment use operationType 'OpenComment'. **MakeCommentPublic:** To make comment public use operationType 'MakePublicComment'

				MakeCommentInternal: To make comment internal use operationType ‘MakeInternalComment’. **EditComment:** To edit comment note use operationType ‘EditComment’. For editing comment note if you send blank value in operationType it will perform edit comment operation but operationType is Mandtory.

Example:

```
PUT
http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byId/2220199429/Comments/7229197537
 {
  "comment": "Hello 456 @RavindraT 1234",
  "operationType":"EditComment",
  "recordRevisionNumber" : 3
}
```

Response:

```
{
   "resultContentType": "json",
   "success": true,
   "response": {
      "recordId": 7229197537,
      "masterRecordVersionId":7229194495,
      "comment": "Hello&#160;456&#160;@RavindraT&#160;1234",
      "createDate": "2023,9,15,7,13,52,130",
      "lastUpdatedDate": "2023,9,15,7,20,7,199",
      "recordRevisionNumber": 4,
      "isPublic": true,
      "parentRecordId": 7229197478,
      "isOpen": false,
      "commentedByUserId": 7229184850,
      "contextText": "",         "isContextualComment": false,
      "permissions": {
          "canDelete": true,
          "canModify": true,
          "canOpen": true,
          "canClose": false
      }
   }
}
```

Example for making comment Open: PUT http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byId/2220199429/Comments/7229197537

```
{
  "comment": "Hello 456 @RavindraT 1234",
  "operationType":"OpenComment",
  "recordRevisionNumber" : 5
}
```

Response:

```
{
    "resultContentType": "json",
    "success": true,
    "response": {
        "recordId": 7229197537,
        "masterRecordVersionId":7229194495,
        "comment": "Hello&#160;456&#160;@RavindraT&#160;1234",
        "createDate": "2023,9,15,7,13,52,130",
        "lastUpdatedDate": "2023,9,15,7,20,7,199",
        "recordRevisionNumber": 6,
        "isPublic": falsesss,
        "parentRecordId": 7229197478,
        "isOpen": true,
        "commentedByUserId": 7229184850,
        "contextText": "",         "isContextualComment": false,
        "permissions": {
            "canDelete": true,ss
            "canModify": true,
            "canOpen": true,
            "canClose": false
        }
    }
}
```

Example for making comment public: PUT http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byId/2220199429/Comments/7229197537

```
{
  "comment": "Hello 456 @RavindraT 1234",
  "operationType":"MakeCommentInternal",
  "recordRevisionNumber" : 4
}
```

Response:

```
{
    "resultContentType": "json",
    "success": true,
    "response": {
        "recordId": 7229197537,
        "masterRecordVersionId":7229194495,
        "comment": "Hello&#160;456&#160;@RavindraT&#160;1234",
        "createDate": "2023,9,15,7,13,52,130",
        "lastUpdatedDate": "2023,9,15,7,20,7,199",
        "recordRevisionNumber": 5,
        "isPublic": falsesss,
        "parentRecordId": 7229197478,
        "isOpen": false,
        "commentedByUserId": 7229184850,
        "contextText": "",
        "isContextualComment": false,
        "permissions": {
            "canDelete": true,
            "canModify": true,
            "canOpen": true,
            "canClose": false
        }
    }
}
```

Refer Response Structure Comments to know explanation of the other fields in the response. Fields that are not explained are for internal usage and should be ignored.

Delete Comment

Purpose	Delete Comment
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/byId/*<recordId>*/comments /*<commentId>*
Request Method	**DELETE**

Dynamic Path Variables:

Variable Name	**Type**	**Description**
recordId	Resource Id	*recordId* of the artifact
commentId	ResourceId	*recordId* of the comment

Request Body:

Field	**Mandatory**	**Type**	**Allowed Values**	**Description**

recordRevisionNumber	Yes	Number		The number of times a record has been updated.

Example: DELETE

http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byId/2220199429/Comments/7229197537

```
{
   "recordRevisionNumber": 4
}
```

Response:

```
{
   "resultContentType": "json",
   "success": true,
   "response": {
      "recordId": 7229213070,
      "message": "",
      "success": "tsResultType_Success"
   }
}
```

Refer Response Structure Comments to know explanation of the other fields in the response. Fields that are not explained are for internal usage and should be ignored.

Attachment Endpoints

1. Fetch Attachments Separately Along with the Record
2. Fetch Attachments
3. Download an Attachment

Fetch Attachments Separately Along with the Record

Attachments are now available as part of a record and can be retrieved by including the `Attachments` field in the list of visible fields.
To fetch the attachments of a record, specify `Attachments` in the `fields` parameter of the request.

Refer to Fetch Details of Single Artifact.

Examples:
GET
http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byId/2220199429?fields=Attachments

Sample Response:

```
{
    "success": true,
    "response": {
        "id": "UC-14800",
        "recordId": 2220202574,
        "name": "deliver pizza via Zomato",
        "nodeType": "ntRecord",
        "projectRecordId": 376523234,
        "parentRecordId": 376524218,
        "ownerRecordId": 376524218,
        "recordVersionId": 2220202711,
        "reuseType": "ReusedAsReference",
        "reuseStatus": "UptoDate",
        "reuseSourceRecordId": 64849999941478,
        "reuseScope": "Individual",
        "reuseUpdateMethod": "Automatic",
        "checkedOut": true,
        "checkedOutByUserId": 8193171620,
        "recordTypeId": 170,
        "recordRevisionNumber": 2,
        "fields": {
            "attachment": [
              {
               "fileId": "8EF403C2-1775-4907-94B5-91BFACEE69D8",
               "note": "This is new attachment",
               "fileName": "New Test.txt",
               "size": 20,
               "addedOn": "2023-07-27T12:59:07.067Z",
               "addedByUserId": 2220169265,
               "addedByUser": "Steve Project Manager",
               "recordRevisionNumber": 1
             },
             {
               "fileId": "8EF403C2-1775-4907-94B5-91BFACEE62R4",
               "note": "",
               "fileName": "Sample-Product-Requirements.xlsx",
               "size": 12338,
               "addedOn": "2023-08-15T22:04:09.463Z",
               "addedByUserId": 2220169265,
               "addedByUser": "Steve Project Manager",
               "recordRevisionNumber": 1
             }
           ]
        }
    }
}
```

For an explanation of the common fields in the response, refer to Important fields of all Repository Artifacts / Issues. For an explanation of the attachment fields in the response, refer to the Attachment Field JSON Object.

Any fields not explained are intended for internal use and should be ignored.

Fetch Attachments

Purpose	Get a list of attachments for an artifact
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/byId/*<recordId>*/attachments
Request Method	**GET**

Dynamic Path Variables:

Variable Name	Type	Description
recordId	Resource Id	*recordId* of the artifact

Request Body: Not Required

Example :

GET

http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byId/2220199429/attachments

Response:

```
{
    "success": true,
    "response": {
        "projectRecordId": 376523234,
        "recordId": 2220199429,
        "attachments": [
           {
              "fileId": "8EF403C2-1775-4907-94B5-91BFACEE69D8",
                 "fileName": "New Test.txt","note": "This    is    new
           attachment",
                 "size": 20,
                 "recordRevisionNumber": 1,
                 "addedByUser": "Steve Project Manager",
                 "addedByUserId": 2220169265,
                 "addedOn": "2023-07-27T12:59:07.067Z",

                "permissions": {
                    "canDelete": true,
                    "canModify": true
                }
            },
           {
                 "fileId": "8EF403C2-1775-4907-94B5-91BFACEE62R4",
                 "fileName": "Sample-Product-
           Requirements.xlsx","note": "",
                 "size": 12338,
                "recordRevisionNumber": 1,
                 "addedByUser": "Steve Project Manager",
                 "addedByUserId": 2220169265,
                 "addedOn": "2023-08-15T22:04:09.463Z",
                "permissions": {
                    "canDelete": true,
                    "canModify": true
                }
```

```
                    }
                ]
        }
}
```

For an explanation of the attachment fields in the response, refer to the Attachment Field JSON Object. Any fields not explained are intended for internal use and should be ignored.

Additional Fields in Attachment JSON Objects

Field Name	Type	Description
permissions	JSON Object	Provides the permission that logged-in user has on an artifact.

Download an Attachment

Purpose	Download file of an attachment
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/byId/*<recordId>*/attachments/*<attachmentId>*/file
Request Method	**PUT**

Dynamic Path Variables:

Variable Name	Type	Description
recordId	Resource Id	*recordId* of the artifact
fileId	Resource Id	*fileId* of an attachment

Request Body: Not Required

Example :

GET

http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byId/376526667/attachments/2220199911/file

Response Content-Type:

application/octet-stream

Response Stream:

The server sends the file as a content stream in the response. The image is in jpeg format. The name of the image file is given in the response header “Content-Disposition”.

Linked Issue Endpoints

Fetch Linked Issues of an Artifact

Purpose	Get issues linked to an artifact
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/byId/*<recordId>*/trackingItems
Request Method	**GET**

Dynamic Path Variables:

Variable Name	Type	Description
recordId	Resource Id	*recordId* of the artifact

Supported Query Parameters:

Parameter Name	Mandatory	Allowed Values	Default Value	Description
nameMaxLength	No	Number	-	The *nameMaxLength* param specifies the maximum length of an artifact name in response. For example `nameMaxLength=80`

Request Body: Not Required

Example :

GET

http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byId/6484106731/trackingItems

Response:

```
{
    "success": true,
    "response": {
        "projectRecordId": 376523234,
        "recordId": 6484106731,
        "issues": [
            {
                "title": "Need Estimated end date to resolve a compl
aint",
                "recordId": 2220203541,
                "id": "CR-14803",
                "recordRevisionNumber": 1,
                "state": "Submitted",
                "daysOpen": 1,
                "id2": "CR-14803"
            },
            {
                "title": "Not able to register Complaint",
                "recordId": 2220203522,
                "id": "ISS-14802",
                "recordRevisionNumber": 3,
                "state": "Submitted",
                "daysOpen": 1,
                "id2": "ISS-14802"
            }
        ]
    }
}
```

Explanation of Response Fields

Field Name	Type	Description
title	String	Name of an issue linked with an artifact.
recordId	Resource Id	recordId of an issue that is linked with an artifact.
id	String	Id of Issue that is displayed on the user interface.
id2	String	Calculated Id if the issue was generated using Custom Formatted ID settings.
state	String	Value in the *State* field of the issue.
recordRevisionNumber	Number	Number of times the issue has been updated.
daysOpen	Number	The number of days the issue has been in *Open* state.

To get more fields of a linked issue, use Fetch details of single artifact.

Version Endpoints

1. Fetch Versions of an Artifact
2. Fetch Version of an Artifact
3. Download Activity Diagram of a Version

Fetch Versions of an Artifact

Purpose	Fetch versions of an artifact
URL	http://<host>:<port>/rest/ttmrestsrv.dll/api_version/repositoryObjects/byId/*<recordId>*/versions
Request Method	**GET**

Dynamic Path Variables:

Variable Name	Type	Description
recordId	Resource Id	*recordId* of the artifact

Request Body: Not Required

Example :

GET
http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byId/376526599/versions

```
NA
```

Response:

```
{
    "success": true,
    "response": {
        "projectRecordId": 376523234,
        "recordId": 376526599,
        "versions": [
            {
                "recordId": 2220200014,
                "isCurrentVersion": true,
                "recordRevisionNumber": 4,
                "keep": false,
                "version": 1.03,
                "comment": "",
                "createDate": "2023-07-28T12:33:13.427Z",
                "createdByUserId": 2220169265,
                "isImportantVersion": true
            },
            {
                "recordId": 2220200011,
                "isCurrentVersion": false,
                "recordRevisionNumber": 3,
                "keep": false,
                "version": 1.02,
                "comment": "",
                "createDate": "2023-07-28T12:30:20.890Z",
                "createdByUserId": 2220169265,
                "isImportantVersion": false
            },
            {
                "recordId": 648496246,
                "isCurrentVersion": false,
                "recordRevisionNumber": 2,
                "keep": false,
                "version": 1.01,
                "comment": "",
                "createDate": "2022-05-26T08:34:13.850Z",
                "createdByUserId": 2,
                "isImportantVersion": true
            },
            {
                "recordId": 376526600,
                "isCurrentVersion": false,
                "recordRevisionNumber": 1,
                "keep": false,
                "version": 1,
                "comment": "",
                "createDate": "2020-05-07T10:56:03.277Z",
                "createdByUserId": 2,
                "isImportantVersion": true
            }
        ]
    }
}
```

Explanation of Response Fields

Field Name	Type	Description
recordId	Resource Id	Unique Id of a version. This needs to be sent along with *recordId* to query details of a version of an artifact.
isCurrentVersion	Boolean	Indicates whether the version is the latest or tip version of an artifact.
version	Number	Version number of the version. Usually, this is in decimal format. e.g., 1.01
comment	String	Note added by the user who created this version.
keep	Boolean	True indicates that the system should not allow the version to be purged by the auto-clean-up process.
createDate	String	Date and time when attachment was added. The string is in ISO 8601 format.
createdByUserId	Resource ID	*recordID* of the user who created the version.

Fetch Version of an Artifact

Purpose	Returns all fields of a version of an artifact
URL	http://host:port/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/byVersionId/*<versionId>*
Request Method	**GET**

Dynamic Path Variables:

Variable Name	Type	Description
versionId	Resource Id	Unique id of a version of an artifact

Supported Query Parameters:

Parameter Name	Mandatory	Allowed Values	Default Value	Description
tsRecordId	Yes	Resource Id	-	recordID of the artifact whose version you want to query.

Request Body: Not Required

Example:

GET http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byVersionId/2220174818?tsRecordId=2220169739

Response:

```
{
    "success": true,
    "response": {
        "name": "Use Case 1",
        "recordId": 2220169739,
        "id": "UC-14224",
        "nodeType": "ntRecord",
        "projectRecordId": 376523234,
        "parentRecordId": 376524218,
        "ownerRecordId": 376524218,
        "recordVersionId": 2220174818,
        "recordTypeId": 170,
        "recordRevisionNumber": 3,
        "fields": {
            "Checked-out": false,
            "Checked-out By": null,
            "Check-out Date": null,
            "Created By": "Steve Tester",
            "Created On": "2023-03-03T04:41:59.277Z",
            "Description": "abcd",
            "Disp.Id": "UC-14224",
            "ID": "UC-14224",
            "Is Suspect": false,
            "Locked": false,
            "Modified By": "Steve Tester",
            "Modified On": "2023-04-11T12:34:13.893Z",
            "MoSCoW": "Not Prioritized",
            "Multi Value 1": "",
            "Name": "Use Case 1",
            "Notes": "",
            "Owner": "",
            "Post Condition": ”<some rich text data>",
            "Pre Condition": "",
            "Project": "Kitchen-Sink",
            "Record Type": "Use Case",
            "Ref. ID": null,
            "State": "Draft",
            "Trigger": "",
            "Unique ID": 2220169739,
            "Version": "1.02"
        }
    }
}
```

Explanation of Response Fields

Field Name	Type	Description
Fields	Fields Object	Fields JSON Object

For an explanation of the common fields in the response, refer to Important fields of all Repository Artifacts / Issues. Any fields not explained are intended for internal use and should be ignored.

Download Activity Diagram of a Version

Purpose	Download *Activity Diagram* of a version of a *Use Case*
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects/byId//*<recordId>*/versions/*<versionNumber>*/fields/Activity Diagram/image
Request Method	**GET**

Dynamic Path Variables:

Variable Name	Type	Description
recordId	Resource Id	*recordId* of the artifact
versionNumber	Resource Id	*Version number* of an artifact

Request Body: Not Required

Example:

GET
http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byId/345125412/versions/1.01/fields/Activity Diagram/image

Response Content-Type:
application/octet-stream

Response Stream:

The server sends the image file as a content stream in the response. The image is in jpeg format. The name of the image file is given in the response header “Content-Disposition”.

Audit Log Endpoints

Fetch Audit Logs

Purpose	Returns audit logs of an artifact
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/repositoryObjects /byid/*<recordId>*/auditLogs
Request Method	**GET**

Dynamic Path Variables:

Variable Name	**Type**	**Description**
recordId	Resource Id	*recordId* of the artifact

Request Body: Not Required

Example :

GET http://myCompany.com/rest/ttmrestsrv.dll/2/repositoryObjects/byid/376526599/ auditLogs

Response:

```
{
    "success": true,
    "response": {
        "projectRecordId": 376523234,
        "recordId": 376526599,
        "auditLogs": [
            {
                "recordId": 2220204316,
                "recordVersion": "1.01",
                "recordVersionId": 2220204315,
                "action": "Modified",
                "actionLog": "Version incremented by the System.",
                "createdByUserId": 2220169265,
                "createDate": "2023-08-18T08:21:25.687Z",
                "changeLogs": [
                    {
                        "fieldName": "Description",
                        "oldValue": "",
                        "newValue": "<<Updated>>"
                    },
                    {
                        "fieldName": "MoSCoW",
                        "oldValue": "Not Prioritized",
                        "newValue": "Must Have"
                    },
                    {
                        "fieldName": "Multi Value 1",
                        "oldValue": "",
                        "newValue": "ba"
                    },
                    {
                        "fieldName": "State",
                        "oldValue": "Draft",
                        "newValue": "Approved"
                    }
                ]
            },
            {
                "recordId": 376526608,
                "recordVersion": "1",
                "recordVersionId": 376526600,
                "action": "Link Added",
                "actionLog": "[UC--10442] Search for Video  --
Primary Actors--  [ACTR-10185] Customer",
                "createdByUserId": 2,
                "createDate": "2020-05-07T10:48:35.840Z",
                "changeLogs": []
            }
        ],
    }
}
```

OneView Document Endpoints

Fetch Artifacts of OneView Document

Purpose	Fetch all the artifacts present in a *OneView Document* in a hierarchical format
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/oneview/*<oneviewId>*/oneviewDocumentMap
Request Method	**GET**

Dynamic Path Variables

Variable Name	Type	Description
oneviewId	Resource Id	*recordId* of the oneview.

Request Body: Not Required

Example :

GET

http://myCompany.com/rest/ttmrestsrv.dll/2/Oneview/648481372/oneviewDocumentMap

Response:

```
{
  "success": true,
  "response": {
    "children": [
      {
        "name": "OV&#160;Template&#160;5",
        "recordId": 2220180168,
        "dispId": "OneV-14448",
        "Children": [
           {
             "name": "Customer&#160;and&#160;Users&#160;Goals",
             "recordId": 14111067,
             "dispId": "SELS-940",
             "children": [
               {
                  "name": "Section&#160;1",
                  "recordId": 2220180187,
                  "dispId": "UNCPKG-14449",
                  "children": []
                 }
              ]
            }
         ]
      }
    ]
  }
}
```

Explanation of Important Fields

Field Name	Type	Description
children	Array of artifact object	Contains the artifact objects that are present in *OneView Docment*.

Artifact Object

Field Name	Type	Description
name	String	Name of an artifact.
recordId	Resource Id	recordId of an artifact in the *OneView Document*.
dispId	String	Calculated display Id of an artifact.
children	Array of artifact object	

Any fields not explained are intended for internal use and should be ignored.

To get more fields an artifact, use Fetch details of single artifact using the Id field in the response above**.**

Fetch OneView Baselines

Purpose	Fetch a list of baselines in a Oneview
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/oneview/*<oneviewId>*/baselines
Request Method	**GET**

Dynamic Path Variables

Variable Name	Type	Description
oneviewId	Resource Id	*recordId* of the oneview.

Request Body: Not Required

Example :

GET http://myCompany.com/rest/ttmrestsrv.dll/2/Oneview/7229188450/baselines

Response:

```
{
    "resultContentType": "json",
    "success": true,
    "response": {
        "baselines": [          {
                "recordId": 7229188901,
                "name": "Steve OV Baseline",
                "projectRecordId": 376523234,
                "createdByUserId": 7229184850,
                "createDate": "2023-09-06T08:16:08.430Z",
                "baselineDate": "2023-09-06T08:16:08.430Z",
                "baselineScope": "OneView",
                "baselineType": "Manual",
                "description": "",
                "recordRevisionNumber": 1
            },
            {
                "recordId": 7229188886,
                "name": "Ktichen Sink 6 Sept 1_30 PM",
                "projectRecordId": 376523234,
                "createdByUserId": 7229184850,
                "createDate": "2023-09-06T08:07:14.697Z",
                "baselineDate": "2023-09-06T08:07:14.697Z",
                "baselineScope": "Project",
                "baselineType": "",
                "description": "",
                "recordRevisionNumber": 1
            }
        ]
    }
}
```

Explanation of Fields

Field Name	**Type**	**Description**
recordId	Resource Id	*recordId* of a baseline.
recordRevisionNumber	Number	Number of times baseline has been updated.
name	String	Name of a baseline.
description	String	Description of a baseline.
projectRecordId	Resource Id	*recordId* of the project.
createdByUserId	Resource Id	*recordId* of the user who created the baseline.
createDate	String	Date and time when baseline is created. The string is in ISO 8601 format.
baselineDate	string	Date and time for which the baseline was captured. In *TopTeam*, you can create a baseline for the current date and time or for a previous date. When you create a baseline for a previous date the *baselineDate* will be the date for which this baseline was created, whereas, the *createDate* will have the date and time on which the baseline is created. The date string is in ISO 8601 format.

baselineScope	String	Determines whether the baseline is created from project, i.e., 'Project' or it is created from *OneView* i.e., '*OneView*'.
baselineType	String	Determines whether the baseline is created for *OneView Document* reviews, i.e., 'Review' OR it is created for OneView, i.e., 'Manual'

Any fields not explained are intended for internal use and should be ignored.

Test Management Endpoints

Fetch Test Run Contents

Purpose	Fetch the contents of a *Test Run*
URL	http://<host>:<port>/rest/ttmRestSrv.dll/<api_version>/testRunContents
Request Method	**GET**

Supported Query Parameters:

Parameter Name	Mandatory	Allowed Values	Default Value	Description
testRunIds	Yes	String (Comma separated *recordId* of Test Run record	-	Specify the *recordId* of a *Test Run* artifact for which contents need to be fetched. To fetch contents of multiple *Test Runs*, you need to specify the comma-separated *recordId* of *Test Run* artifact. Example: testRunIds=6484150277, 6484150275

				To get the *recordId* of *Test Run* artifacts, use Fetch List of Artifacts in Hierarchical Structure and pass the *tsEtpIds* param as *recordId* of the *Test Run* record type.
defectCount	No	Boolean	false	Specify this flag to true if you want to fetch defects count against the *Test Result*. Defect count will return the number of bugs reported against the *Test Result* while executing *Test Run*.

Request Body: Not Required

Example:

GET
http://myCompany.com/rest/ttmRESTSrv.dll/2/testRunContents?testRunIds=6484170511&defectCount=true

Response:

```
{
    "resultContentType": "json",
    "success": true,
    "response": {
        "7229190059": {
            "contents": [
                {
                    "recordId": 7229190064,
                    "parentRecordId": 7229190059,
                    "recordTypeId": 950,
                    "name": "Login to System - Customer enters
correct Credentials",
                    "testResult": "Passed"
,
                    "runDate": "2023,9,7,6,28,34,370",
                    "runBy": 7229184850,
                    "testCaseRecordId": 6484139219,
                    "dispId": "TC-10449",
                    "testCaseRecordTypeId": 890,
                    "displaySequence": 100000,
                    "recordRevisionNumber": 2,
                    "defectCount": 0
                },
                {
                    "recordId": 7229190065,
                    "parentRecordId": 7229190059,
                    "recordTypeId": 950,
                    "name": "Login to System - Incorrect credentials
entered for more than three attempts",
                    "testResult":"Failed",

                    "runDate": "2023,9,7,6,28,37,400",
                    "runBy": 7229184850,
                    "testCaseRecordId": 6484139216,
                    "dispId": "TC-10450",
                    "testCaseRecordTypeId": 890,
                    "displaySequence": 200000,
                    "recordRevisionNumber": 2,
                    "defectCount": 1
                },
                {
                    "recordId": 7229190066,
                    "parentRecordId": 7229190059,
                    "recordTypeId": 950,
                    "name": "Login to System - Correct credentials
entered in third attempt",
                    "testResult":"Passed",
                    "runDate": "2023,9,7,6,28,46,487",
                    "runBy": 7229184850,
                    "testCaseRecordId": 6484139222,
                    "dispId": "TC-10451",
                    "testCaseRecordTypeId": 890,
                    "displaySequence": 300000,
                    "recordRevisionNumber": 2,
                    "defectCount": 0
                }
            ]
        }
    }
}
```

Explanation of Response Fields

Field Name	Type	Description
recordId	Resource Id	Unique Id of a *Test Case Result* artifact.
parentRecordId	Resource Id	*recordId* of a parent artifact. This is used to build the parent-child hierarchy.
name	String	Name of the *Test Case Result* artifact.
testResult	String	This field indicates the result status of the *Test Run Result* artifact.
runDate	String	Date and Time when a *Test Case Result* was executed in *Test Run*.
runBy	String	*recordId* of the user who executed the *Test Case Result.*
testCaseRecordId	Resource Id	*recordId* of the actual *Test Case* for which this *Test Case Result* is getting executed.
testCaseRecordTypeId	Resource Id	*recordId* of the *Test Case* record type for which this *Test Case Result* is getting executed.
recordRevisionNumber	Number	Number of times an artifact has been updated.
displaySequence	Number	Display a sequence number of an *Test Case Result.* This can be used to sort the contents within its parent. Artifact with lower value should appear before artifact with higher value.
defectCount	Number	Number of defects reported for this *Test Case Result.*

Endpoints to Get System Configuration Data

1. System-Wide Master States
2. System-Wide Fields
3. Record Type Endpoints
4. Link Type Endpoints
5. User Endpoints

System-Wide Master States

Purpose	Get a list of master states defined in the system
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/metadata/masterStates
Request Method	**GET**

Request Body: Not Required

Example :

GET http://myCompany.com/rest/ttmrestsrv.dll/2/metadata/masterStates

Response:

```
{
    "resultContentType": "json",
    "success": true,
    "response": {
        "masterStates": [
            {
                "name": "New",
                "id": 1018,
                "considerAsOpen": true,
                "inactive": "N",
                "imageIndex": 16,
                "agileStateId": 1429,
                "recordRevisionNumber": 0,
                "displaySequence": 600,
                "createDate": "2001,1,1,0,0,0,0",
                "lastUpdateDate": "2001,1,1,0,0,0,0",
                "createdByUserName": "System",
                "lastUpdatedByUserName": "System",
                "isPrimary": false,
                "recordTypeId": 90,
                "description": "",
                "activityDescription": "",
                "isSystem": false
            },
            {
                "name": "Completed (inactive)",
                "id": 1019,
                "considerAsOpen": false,
                "inactive": "Y",
                "imageIndex": 15,
                "agileStateId": -1,
                "recordRevisionNumber": 1,
                "displaySequence": 3100,
                "createDate": "2001,1,1,0,0,0,0",
                "lastUpdateDate": "2001,1,1,0,0,0,0",
                "createdByUserName": "System",
                "lastUpdatedByUserName": "System",
                "isPrimary": false,
                "recordTypeId": 90,
                "description": "",
                "activityDescription": "",
                "isSystem": false
            }
        ]
    }
}
```

Explanation of Fields

Field Name	Type	Description
name	String	Name of the state.
id	Resource Id	*recordId* of the state.
considerAsOpen	Boolean	Indicates whether the state is *Open* or *Closed*.
description	String	Value in the *Default Description* field
activityDescriion	String	Value in the *Default Description* field.
inactive	Boolean	Indicates whether the state is inactive.
isSystem	Boolean	Indicates whether an artifact is a system artifact. System artifacts cannot be deleted.

For more information about these fields, refer to the **System-Wide Master States** module in **Administration** Section in *TopTeam*.

Any fields not explained are intended for internal use and should be ignored.

System-Wide Fields

Purpose	Returns the list of all fields, both system and custom, included in a record type
URL	http://<host><port>/rest/ttmrestsrv.dll/<api_version>/metadata/*REP*/fields
Request Method	**GET**

Request Body: Not Required

Example:

GET http://myCompany.com/rest/ttmrestsrv.dll/2/metadata/REP/fields

Response:

```
{
  "success": true,
  "response": {
    "fields": [
      {
        "caption": "MoSCoW",
        "recordId": 316,
        "dbColumnName": "ELM_PRIORITY",
        "isReadonly": false,
        "dataType": "tsFieldType_List",
        "size": 9,
        "required": false,
        "isSystem": true,
        "visible": true,
        "isMandatory": true,
        "dependentOnFieldId": 0,
        "dataSubType": "tsFieldSubType_None",
        "sysHidden": false,
        "allowedValues": [
          {
            "name": "Must Have",
            "recordId": 1,
            "description": "",
            "displaySequence": 1,
            "imageIndex": 0,
            "dependentOnFieldId": 0,
            "dependentOnValue": ""
          },
          {
            "name": "Should Have",
            "recordId": 2,
            "description": "",
            "displaySequence": 2,
            "imageIndex": 1,
            "dependentOnFieldId": 0,
            "dependentOnValue": ""
          },
          {
            "name": "Could Have",
            "recordId": 3,
            "description": "",
            "displaySequence": 3,
            "imageIndex": 2,
            "dependentOnFieldId": 0,
            "dependentOnValue": ""
          },
          {
            "name": "Not Prioritized",
            "recordId": 4,
            "name": "Won't Have",
            "description": "",
            "displaySequence": 4,
            "imageIndex": 3,
            "dependentOnFieldId": 0,
            "dependentOnValue": ""
          },
          {
            "name": "Not Prioritized",
            "recordId": 1880,
            "description": "",
            "displaySequence": 5,
            "imageIndex": -1,
            "dependentOnFieldId": 0,
            "dependentOnValue": ""
          }
        ]
      },
      {
        "caption": "Name",
        "recordId": 317,
        "dbColumnName": "ELM_NAME",
        "isReadonly": false,
        "dataType": "tsFieldType_Text",
        "size": 239,
        "required": false,
        "isSystem": true,
        "visible": true,
        "isMandatory": true,
        "dependentOnFieldId": 0,
        "dataSubType": "tsFieldSubType_None",
        "sysHidden": false,
        "allowedValues": []
      }
    ]
  }
}
```

Explanation of Response Fields

Field Name	Type	Description
caption	String	Display the name of a field.
recordId	Resource Id	recordId of the record type field.
isReadonly	Boolean	Indicate whether a user can edit a field. True value indicates that only the system can edit its value.
dataType	String	Type of the field. Refer to Fields of Value Object.
isMandatory	Boolean	Field is mandatory while creating an artifact.
visible	Boolean	Fields that are of internal usage to the system are hidden from users. Such fields have False as a value in this field.
dependentOnFieldId	Resource Id	*recordId* of the field on which this field is dependent. Applicable when a field is configured to be dependent on the value of another field.
allowedValues	Array of Value Object	List of values allowed in the field. Applicable when datatype is *tsfieldType_List, or tsfieldType_Multi-Value*

Fields of Value Object

Field Name	Type	Description
name	String	Display name of the value object.
recordId	Resource Id	*recordId* of the value object.
displaySequence	Number	Use to sort the values of a field.
description	String	Usage hint of the value object.
dependentOnFieldId	Resource Id	*recordId* of the field on which this value is dependent. Applicable when a field is configured to be dependent on the value of another field.

dependentOnValue	String	name of the parent value object of field given in *dependentOnFieldId* field.

Explanation of Values in dataType Field

Value	Field Type	Allowed Values
tsFieldType_Number	Number Field	Numeric non-decimal value.
tsFieldType_Double	Decimal Field	Numeric value in decimal format.
tsFieldType_Text	Text Field	Plain text.
tsFieldType_LargeText	Large Text	Plain text.
tsFieldType_RichText	Rich Text Field	Text in html format.
tsFieldType_Boolean	Boolean	True Or False.
tsFieldType_List	List Field	Single value name from the allowed list of values attached to the field.
tsFieldType_MultiValue	Multi-Value Field	Multiple comma-separated value names from the list of values attached to the field.
tsFieldType_User	User Field	Display the name of a user.
tsFieldType_ProjectTeamMember	Team Member Field	Display the name of the team member of the project of the concerned artifact.
tsFieldType_Project	Project Field	Name of a project.
tsFieldType_StateTransition	State Field	Name of a record type state.
tsFieldType_Date	Date Field	Date in ISO 8601 format without time zone.
tsFieldType_DateTime	Date Time Field	Date and Time in ISO 8601 format without time zone.
tsFieldType_Time	Time Field	Time in ISO 8601 format without time zone.
tsFieldType_Diagram	Diagram	Field contains diagram objects in JSON format. The schema is proprietary. DO NOT edit this field via REST API.

tsFieldType_Special	Custom	Field contains data of custom objects internal to the system. The data is in proprietary XML format. DO NOT edit this field via REST API.

Record Type Endpoints

1. Fetch Record Types
2. Fetch Record Type Fields
3. Fetch Record Type States
4. Fetch Traceability Rules of a Record Type
5. Fetch Saved Filters of a Record Type Fetch Record Types

Fetch Record Types

Purpose	Get a list of record types defined in the repository.
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/metadata/recordTypes
Request Method	**GET**

Request Body: Not Required

Example :

GET http://myCompany.com/rest/ttmrestsrv.dll/2/metadata/recordTypes

Response:

Response json in list view

```
{
    "success": true,
    "response": {
        "recordTypes": [
            {
                "id": 90, //"recordId" of record
                "name": "Repository Artifacts",
                "nodeType": "ntRecordType",
                "etpId": 90,
                "etpInd": "RO",
                "description": "This is the base Record Type for all
                                Repository objects",
                "idPrefix": "REP",
                "recordRevisionNumber": 15,
                "leaf": false,
                "singularName": "Repository Artifact",
                "displaySequence": 0,
                "considerAsBacklog": false,
                "parentId": 80,
                "imageIndex": 18,
                "isFoldered": false,
                "isHierarchical": false,
                "isOrdered": true,
                "isRepositoryObject": true,
                "isTrackingItem": false,
                "canCreateRecords": false,
                "isStateLess": false,
                "isInactive": false,
                "useEclOfEtpId": 90,
                "groupName": "",
                "isUnique": false,
                "initialStateId": 4867,
                "isPrimary": false
            },
            {
                "id": 100,
                "name": "Repository Artifacts",
                "nodeType": "ntRecordType",
                "etpId": 100,
                "etpInd": "RO",
                "description": "This is the base Record Type for all
                                Repository objects",
                "idPrefix": "REP",
                "recordRevisionNumber": 15,
                "leaf": false,
                "singularName": "Repository Artifact ",
                "displaySequence": 0,
                "considerAsBacklog": false,
                "parentId": 80,
                "imageIndex": 18,
                "isFoldered": false,
                "isHierarchical": false,
                "isOrdered": true,
                "isRepositoryObject": true,
                "isTrackingItem": false,
                "canCreateRecords": false,
                "isStateLess": false,
                "isInactive": false,
                "useEclOfEtpId": 90,
                "groupName": "",
                "isUnique": false,
                "initialStateId": 4867,
                "isPrimary": false
```

```
                    }
                ]
            }
}
```

Explanation of Important Fields

Field Name	Type	Description
name	String	Name of a record type.
id	Resource Id	*recordId* of a record type.
description	String	Value in the *Description* field.
idPrefix	String	Id Prefix to be used in the display Id of artifacts of the record type.
parentId	Resource Id	*recordId* of the parent record type.
isHierarchical	Boolean	This specifies whether the artifacts of this record type are hierarchical in nature. If this is true, the system allows the users to create a child artifact of an artifact of this type. NOTE: This is applicable only to single artifact type packages.
isInactive	Boolean	Indicates whether the record type is inactive.
recordRevisionNumber	Number	Number of times an artifact has been updated.

Any fields not explained are intended for internal use and should be ignored.

Fetch Record Type Fields

Purpose	Returns the list of all fields, both system and custom, included in an artifact type
URL	http://<host><port>/rest/ttmrestsrv.dll/<api_version>/metadata/*<recordTypeIdPrefix>*/fields
Request Method	**GET**

Dynamic Path Variables

Variable Name	Type	Description
recordTypeIdPrefix	String	idPrefix *of recordType* the record type for which you want to fetch fields, such as 'UC', 'PREQ', and more.

Request Body: Not Required

Example:

GET http://myCompany.com/rest/ttmrestsrv.dll/2/metadata/UC/fields

Response:

```
{
  "success": true,
  "response": {
    "fields": [
      {
        "caption": "MoSCoW",
        "recordId": 316,
        "dbColumnName": "ELM_PRIORITY",
        "isReadonly": false,
        "dataType": "tsFieldType_List",
        "size": 9,
        "required": false,
        "isSystem": true,
        "visible": true,
        "isMandatory": true,
        "dependentOnFieldId": 0,
        "dataSubType": "tsFieldSubType_None",
        "sysHidden": false,
        "allowedValues": [
          {
            "name": "Must Have",
            "recordId": 1,
            "description": "",
            "displaySequence": 1,
            "imageIndex": 0,
            "dependentOnFieldId": 0,
            "dependentOnValue": ""
          },
          {
            "name": "Should Have",
            "recordId": 2,
            "description": "",
            "displaySequence": 2,
            "imageIndex": 1,
            "dependentOnFieldId": 0,
            "dependentOnValue": ""
          },
          {
            "name": "Could Have",
            "recordId": 3,
            "description": "",
            "displaySequence": 3,
            "imageIndex": 2,
            "dependentOnFieldId": 0,
            "dependentOnValue": ""
          },
          {
            "name": "Not Prioritized",
            "recordId": 4,
            "name": "Won't Have",
            "description": "",
            "displaySequence": 4,
            "imageIndex": 3,
            "dependentOnFieldId": 0,
            "dependentOnValue": ""
          },
          {
            "name": "Not Prioritized",
            "recordId": 1880,
            "description": "",
            "displaySequence": 5,
            "imageIndex": -1,
            "dependentOnFieldId": 0,
            "dependentOnValue": ""
          }
        ]
      },
      {
        "caption": "Name",
        "recordId": 317,
        "dbColumnName": "ELM_NAME",
        "isReadonly": false,
        "dataType": "tsFieldType_Text",
        "size": 239,
        "required": false,
        "isSystem": true,
        "visible": true,
        "isMandatory": true,
        "dependentOnFieldId": 0,
        "dataSubType": "tsFieldSubType_None",
        "sysHidden": false,
        "allowedValues": []
      }
    ],
    "recordTypeId": 170
  }
}
```

Explanation of Response Fields

Refer to the section System Wide Fields

Fetch Record Type States

Purpose	Get a list of states that are included in a record type
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/metadata/*<recordTypeIdPrefix>*/states
Request Method	**GET**

Dynamic Path Variables

Variable Name	Type	Description
recordTypeIdPrefix	String	*idPrefix of recordType* the record type for which you want to fetch states, such as 'UC', 'PREQ', and more.

Request Body: Not Required

Example :

GET http://myCompany.com/rest/ttmrestsrv.dll/2/metadata/UC/states

NA

Response:

```
{
    "success": true,
    "response": {
        "states": [
            {
                "name": "Defined",
                "id": 4888,
                "considerAsOpen": true,
                "isDisabled": false,
                "imageIndex": 0,
                "recordRevisionNumber": 0,
                "description": "",
                "activityDescription": "",
                "lastUpdatedDate": "2001,1,1,0,0,0,0",
                "lastUpdatedByUserName": ""
            },
            {
                "name": "In Progress",
                "id": 4890,
                "considerAsOpen": true,
                "isDisabled": false,
                "imageIndex": 0,
                "recordRevisionNumber": 0,
                "description": "",
                "activityDescription": "",
                "lastUpdatedDate": "2001,1,1,0,0,0,0",
                "lastUpdatedByUserName": ""
            },
            {
                "name": "Accepted",
                "id": 4892,
                "considerAsOpen": false,
                "isDisabled": false,
                "imageIndex": 0,
                "recordRevisionNumber": 0,
                "description": "",
                "activityDescription": "",
                "lastUpdatedDate": "2001,1,1,0,0,0,0",
                "lastUpdatedByUserName": ""
            },
            {
                "name": "Completed",
                "id": 4894,
                "considerAsOpen": false,
                "isDisabled": false,
                "imageIndex": 0,
                "recordRevisionNumber": 0,
                "description": "",
                "activityDescription": "",
                "lastUpdatedDate": "2001,1,1,0,0,0,0",
                "lastUpdatedByUserName": ""
            },
            {
                "name": "Draft",
                "id": 4867,
                "considerAsOpen": true,
                "isDisabled": false,
                "imageIndex": 0,
                "recordRevisionNumber": 0,
                "description": "",
                "activityDescription": "",
                "lastUpdatedDate": "2001,1,1,0,0,0,0",
                "lastUpdatedByUserName": ""
            },
            {
                "name": "Ready for Review",
                "id": 4871,
                "considerAsOpen": true,
                "isDisabled": false,
                "imageIndex": 0,
                "recordRevisionNumber": 0,
                "description": "",
                "activityDescription": "",
                "lastUpdatedDate": "2001,1,1,0,0,0,0",
                "lastUpdatedByUserName": ""
            },
            {
                "name": "Approved",
                "id": 4866,
                "considerAsOpen": true,
                "isDisabled": false,
                "imageIndex": 0,
                "recordRevisionNumber": 0,
                "description": "",
                "activityDescription": "",
                "lastUpdatedDate": "2001,1,1,0,0,0,0",
                "lastUpdatedByUserName": ""
            }
        ]
    }
}
```

Explanation of Important Fields

Field Name	Type	Description
Name	String	Name of the state.
id	Resource Id	*recordId* of the state.
considerAsOpen	Boolean	Indicates whether the state is *Open* or *Closed*.
Description	String	Value in the *Description* field.

activityDescriion	String	Value in the *Description* field.
isDisabled	Boolean	Indicates whether the state has been disabled for further use.
recordRevisionNumber	Number	Number of times the artifact was updated.
lastUpdatedDate	string	Date and time when a state was last updated. This format is the internal format of *TopTeam*. This will be represented as: yyyy,mm,dd,hh,nn,ss,zzz Example: 2023,08,23,13,36,56,256

Refer to the **Record type Master States** module in *Administration* section of *TopTeam* for more information on these fields.

Any fields not explained are for internal usage of the system.

Fetch Traceability Rules of a Record Type

Purpose	Returns list of outgoing and incoming link types defined for a record type
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/metadata /*<recordTypeIdPrefix>*/linkingrules
Request Method	**GET**

Dynamic Path Variables

Variable Name	Type	Description
recordTypeIdPrefix	String	*idPrefix of recordType* the record type that you want to fetch, such as 'UC', 'PREQ', and more.

Request Body: Not Required

Example:

GET http://myCompany.com/rest/ttmrestsrv.dll/2/metadata/UC/linkingrules

Response:

```
{
    "success": true,
    "response": {
        "recordTypeId": 1060,
        "linkingRules": {
            "outgoing": [
                {
                    "recordTypeId": 890,
                    "linkTypeId": 220
                },
                {
                    "recordTypeId": 170,
                    "linkTypeId": 220
                }
            ],
            "incoming": [
                {
                    "recordTypeId": 1090,
                    "linkTypeId": 220
                },
                {
                    "recordTypeId": 1420,
                    "linkTypeId": 220
                },
                {
                    "recordTypeId": 1430,
                    "linkTypeId": 220
                },
                {
                    "recordTypeId": 130,
                    "linkTypeId": 220
                }
            ]
        }
    }
}
```

Explanation of Imporant Fields

Field Name	Type	Description
recordTypeId	Resource Id	*recordId* of record type for which you create linking rules.
outgoing	Array of Linking Rule Object	Contains downstream linking rules.

incoming	Array of Linking Rule Object	Contains upstream linking rules.

Linking Rule Fields

Field Name	Type	Description
linkTypeId	Resource Id	*recordId* of link type.
recordTypeId	Resource Id	*recordId* of linked record type.

Fetch Saved Filters of a Record Type

Purpose	Get saved filters defined for a record type. The filters can be created *via Artifacts Tree/Artifact List* editor. Presently REST API doesn't support creating or editing saved filters. You can use *recordID* returned by this endpoint in Fetch List of Artifacts in Hierarchical Structure to query artifacts based on filter criteria.
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/metadata/ *<recordTypeIdPrefix>*/treefilters
Request Method	**GET**

Dynamic Path Variables

Variable Name	Type	Description
recordTypeIdPrefix	String	*idPrefix of recordType* for which we want to fetch filters, such as 'UC', 'REQ', and more As each record type can have a different set of fields, the filters are also specific to a record type.

Request Body: Not Required

Example:

GET http://myCompany.com/rest/ttmrestsrv.dll/2/metadata/UC/treeFilters

Response:

```
{
    "success": true,
    "response": {
         "recordTypeId": 130,
        "permissions": {
            "canAdd": true,
            "canCreatePublicFilter": true
        },
        "treeFilters": [
            {
                "name": "Requirements not having downstream test cas
es",
                "recordId": 6173,
                "ownerId": 1,
                "isPublic": true,
                "description": "",
                "isAdhoc": false,
                "filterModeType": "tsFilterMode_ParentOriented",
                "highlightFilteredRecord": false,
                "recordRevisionNumber": 1
            },
            {
                "name": "Requirements not having downstream trace li
nks",
                "recordId": 6174,
                "ownerId": 1,
                "isPublic": true,
                "description": "",
                "isAdhoc": false,
                "filterModeType": "tsFilterMode_ParentOriented",
                "highlightFilteredRecord": false,
                "recordRevisionNumber": 1
            },
            {
                "name": "Requirements not having upstream trace link
s",
                "recordId": 6175,
                "ownerId": 1,
                "isPublic": true,
                "description": "",
                "isAdhoc": false,
                "filterModeType": "tsFilterMode_ParentOriented",
                "highlightFilteredRecord": false,
                "recordRevisionNumber": 1
            },
            {
                "name": "Requirements in Closed States",
                "recordId": 6176,
                "ownerId": 1,
                "isPublic": true,
                "description": "",
                "isAdhoc": false,
                "filterModeType": "tsFilterMode_ParentOriented",
                "highlightFilteredRecord": false,
                "recordRevisionNumber": 1
            },
            {
                "name": "Requirements in Open States",
                "recordId": 6177,
                "ownerId": 1,
                "isPublic": true,
                "description": "",
                "isAdhoc": false,
                "filterModeType": "tsFilterMode_ParentOriented",
                "highlightFilteredRecord": false,
                "recordRevisionNumber": 1
            }
        ]
    }
}
```

Explanation of Response Fields

Field Name	Type	Description
name	String	Name of the filter.
recordId	Resource Id	*recordId* of the filter.
ownerId	Resource Id	*recordId* of the user who created the filter.
description	String	Description of the filter.
isPublic	Boolean	Filter is shared or private. True value indicates that filter is shared.
recordRevisionNumber	Number	Number of times record type filter is updated
filterModeType	String	Indicates whether child artifacts of an artifact that doesn’t satisfy the filter should be queried. *tsFilterMode_ChildOriented*: Query child artifacts of parent

		artifacts that do not satisfy the filter. *tsFilterMode_ParentOriented*: Do not query child artifacts of parent artifacts that do not satisfy the filter.
highlightFilteredRecord	Boolean	Ignore it, as it is an internal field.
isAdhoc	Boolean	Indicates that filter is an ad hoc filter created by the logged-in user. Do NOT use the filters having true value in this field in REST API endpoints to fetch artifacts.

Link Type Endpoints

Fetch System Link Types

Purpose	Returns list of all link types defined in the system
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/metadata/linktypes
Request Method	**GET**

Request Body: Not Required

Example:

GET http://myCompany.com/rest/ttmrestsrv.dll/2/metadata/linktypes

Response:

```
{
    "success": true,
    "response": {
        "linkTypes": [
            {
                "forwardName": "Traces Into",
                "reverseName": "Traces From",
                "recordId": 220,
                "indicator": "TRC",
                "isSystem": true,
                "imageIndex": 5
            },
            {
                "forwardName": "Branched Into",
                "reverseName": "Branched From",
                "recordId": 410,
                "indicator": "BRH",
                "isSystem": true,
                "imageIndex": 22
            },
            {
                "forwardName": "Includes Into",
                "reverseName": "Included From",
                "recordId": 450,
                "indicator": "CNS",
                "isSystem": true,
                "imageIndex": -1
            },
            {
                "forwardName": "Represented By",
                "reverseName": "Represents",
                "recordId": 610,
                "indicator": "ASO",
                "isSystem": true,
                "imageIndex": -1
            }
        ]
    }
}
```

Explanation of Imporant Fields

Field Name	Type	Description
recordId	Resource Id	*recordId* of the link type.
forwardName	String	Downstream name of the link type.
reverseName	String	Upstream name of link type.

indicator	String	Internal code of link type. **Values** TRC: Link type is a trace link type. ASO: Link type is a non-trace link type. Other codes: Internal usage.
isSystem	Boolean	System link types that cannot be edited or deleted by users.

Any fields not explained are for internal usage of the system.

User Endpoints

Fetch Users

Purpose	Get a list of users in the system
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/metadata/users
Request Method	**GET**

Request Body: Not Required

Example:

GET http://myCompany.com/rest/ttmrestsrv.dll/2/metadata/users

Response:

```
{
    "success": true,
    "response": {
        "users": [
            {
                "userName": "SteveTester",
                "displayName": "Steve Tester",
                "recordId": 2220200411,
                "isExternal": true,
                "recordRevisionNumber": 1,
                "isInactive": false,
                "isLoginDisable": false,
                "authenticationType": "tsAuthenticationType_Native",
                "emailAddress1": {
                    "id": "ab@gmail.com",
                    "useForNotification": true
                },

            },
            {
                "userName": "clarkManager",
                "displayName": "Clark Manager",
                "recordId": 2220200416,
                "isExternal": false,
                "recordRevisionNumber": 1,
                "isInactive": false,
                "isLoginDisable": false,
                "authenticationType": "tsAuthenticationType_Native",
                "emailAddress1": {
                    "id": "ac@gmail.com",
                    "useForNotification": true
                },
            }
      ]
   }
}
```

Explanation of Response Fields

Field Name	Type	Description
recordId	Resource Id	*recordId* of a user
userName	String	Login name of a user. Used for login authentication only.
displayName	String	Display the name of the user that appears in all *TopTeam* interfaces.
isExternal	Boolean	Defines user as author user or non-author user. Collaborators and viewers are non-author users.
recordRevisionNumber	Number	Number of times a user has been updated.
authenticationType	String	Defines the authentication type of user.

		tsAuthenticationType_Native: User is authenticated via user name and password. **tsAuthenticationType_LDAP**: User is authenticated via Windows active directory. **tsAuthenticationType_3AUTH**: User is authenticated via third-party identity provider via SAML protocol.
emailAddress1	Object	Object contains the email address of a user.
isLoginDisable	Boolean	Indicates that the administrator or the system has temporarily disabled login due to multiple incorrect password attempts.
isInactive	Boolean	Indicates that the administrator has inactivated the user.

Any fields not explained are for internal usage of the system.

Packages Structure Endpoints

1. Get List of Package Structures
2. Get Package Structure Details

Get List of Package Structures

Purpose	Get list of package structures defined in the system
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/packageStructure
Request Method	**GET**

Request Body: Not Required

Example:

GET http://myCompany.com/rest/ttmrestsrv.dll/2/packageStructure

Response:

```
{
    "success": true,
    "response": {
        "packageStructure": [
            {
                "recordId": 5713,
                "parentRecordId": null,
                "ownedByUserId": 1,
                "recordTypeId": null,
                "name": "Business Requirements Model",
                "recordRevisionNumber": 1,
                "isPublic": false,
                "isHidden": false,
                "description": null,
                "isDefault": false,
                "isActive": true,
                "createDate": "2016-09-26",
                "createdByUserId": 1,
                "lastUpdateDate": "2016-09-26",
                "lastUpdatedByUserId": -1
            },
            {
                "recordId": 5710,
                "parentRecordId": null,
                "ownedByUserId": 1,
                "recordTypeId": null,
                "name": "Data Definition Model",
                "recordRevisionNumber": 2,
                "isPublic": false,
                "isHidden": true,
                "description": null,
                "isDefault": false,
                "isActive": false,
                "createDate": "2016-09-26",
                "createdByUserId": 1,
                "lastUpdateDate": "2016-09-26",
                "lastUpdatedByUserId": -1
            },
            {
                "recordId": 5712,
                "parentRecordId": null,
                "ownedByUserId": 1,
                "recordTypeId": null,
                "name": "User Story Model",
                "recordRevisionNumber": 2,
                "isPublic": false,
                "isHidden": false,
                "description": null,
                "isDefault": false,
                "isActive": true,
                "createDate": "2016-09-26",
                "createdByUserId": 1,
                "lastUpdateDate": "2016-09-26",
                "lastUpdatedByUserId": -1
            }
        ]
    }
}
```

Explanation of Response Fields

Field Name	Type	Description
recordId	Resource Id	*recordId* of the package structure. Use this to apply a package structure on a package in Create Package endpoint.
name	String	Name of the package structure.
description	String	Description of the package structure.
recordRevisionNumber	Number	Number of times the artifact is updated.
isActive	Boolean	Indicates whether the package structure is active or deprecated. False value indicates that this package structure is deprecated and should not be applied to a package.

Get Package Structure Details

Purpose	Fetches data for single package structure
URL	http://<host>:<port>/rest/ttmrestsrv.dll/<api_version>/packageStructure/*<packageStructureId>*
Request Method	**GET**

Dynamic Path Variables

Variable Name	Type	Description
packageStructureId	Resource Id	*recordId* of the packageStructure

Request Body: Not Required

Example:

GET http://myCompany.com/rest/ttmrestsrv.dll/2/packageStructure/5712

Response:

```
{
    "success": true,
    "response": {
        "packageStructure": [
            {
                "recordId": 5712,
                "parentRecordId": 0,
                "ownedByUserId": 1,
                "recordTypeId": 0,
                "name": "User Story Model",
                "recordRevisionNumber": 2,
                "isPublic": false,
                "isHidden": false,
                "description": "",
                "isDefault": false,
                "isActive": true,
                "createDate": "2016-09-26",
                "lastUpdateDate": "2016-09-26",
                "createdByUserId": 1,
                "lastUpdatedByUserId": -1,
                "recordTypes": [
                    {
                        "recordId": 590,
                        "name": "Business Rule",
                        "linkType": 0,
                        "isForwardLink": true,
                        "canCreateRecordInPackage": true,
                        "canReuseRecordInPackage": true,
                        "childElementTypes": []
                    },
                    {
                        "recordId": 1000,
                        "name": "Non-Functional Requirement",
                        "linkType": 0,
                        "isForwardLink": true,
                        "canCreateRecordInPackage": true,
                        "canReuseRecordInPackage": true,
                        "childElementTypes": []
                    },
                    {
                        "recordId": 1060,
                        "name": "User Story",
                        "linkType": 0,
                        "isForwardLink": true,
                        "canCreateRecordInPackage": true,
                        "canReuseRecordInPackage": true,
                        "childElementTypes": [
                            {
                                "recordId": 170,
                                "name": "Use Case",
                                "linkType": 220,
                                "isForwardLink": true,
                                "canCreateRecordInPackage": true,
                                "canReuseRecordInPackage": true,
                                "childElementTypes": []
                            }
                        ]
                    }
                ]
            }
        ]
    }
}
```

Explanation of Response Fields

Field Name	Type	Description
recordTypes	Array of Record Type Object	Contains the definition of the package structure.

Interpreting the package structure definition

Each item in the array depicts the record type allowed in the package structure where this package structure is applied. The array is hierarchical in structure. Each item contains child items in the field *childElementTypes*, which again is an array of the same object type.

The items in the root are the record types that are allowed (created or reused) at the root of a package. Items mentioned in the *childElementTypes* array are the record types that can be created or reused under an artifact of the record type.

Record Type Object

Field Name	Type	Description
recordId	Resource Id	*recordId* of an allowed record type.
name	String	Name of an allowed record type.
linkType	Resource Id	*recordId* of the link type. A link of this type will be automatically created by the system between an artifact of a given record type

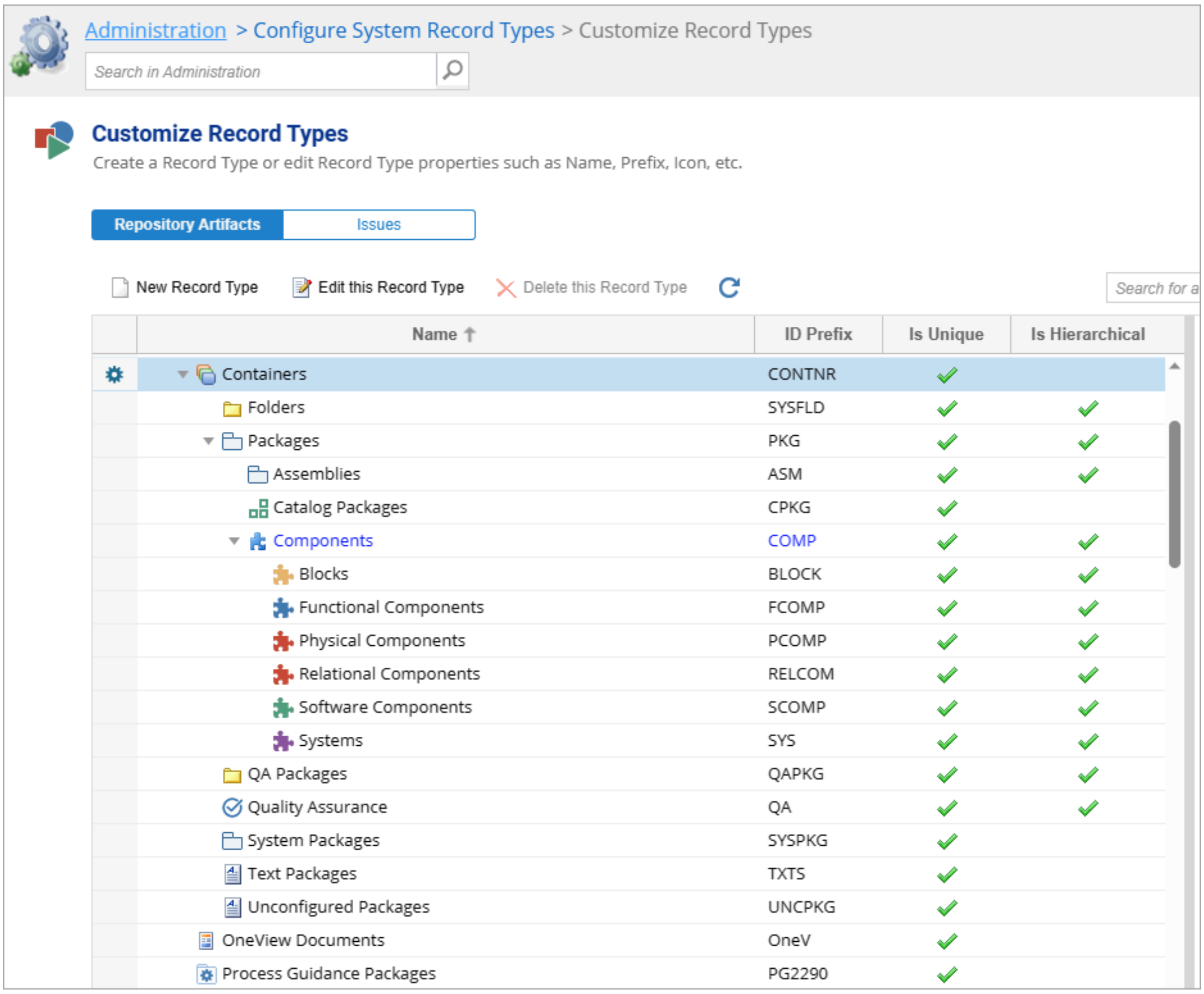
		and an artifact of the parent record type within the same package structure. The field is applicable only in the object added in a *childElementTypes* array. Value 0 indicates that no link shall be created.
isForwardLink	Boolean	Indicates the direction of a link between an artifact of a given record type and an artifact of the parent record type within the same package structure.
canCreateRecordInPackage	Boolean	Determines that a user can create an artifact of a given record type.
canReuseRecordInPackage	Boolean	Determines that a user can reuse artifacts of a given record type.
childElementTypes	Array of Record Type Object	What type of artifacts can be created as child artifacts of an artifact.

Appendix

How to find a list of Package Record Types in my repository

Packages are a basic container of *TopTeam*. There are multiple types of packages in *TopTeam*. E.g., *Packages, System Packages, Components,* and more. You can find the list of package record types in your repository by opening **Record Type Editor** in **Administration** section in *TopTeam*.

The list of record types that appear as child record types inside "Containers" are the package record types defined in your repository.



Administration > Configure System Record Types > Customize Record Types

Customize Record Types

Create a Record Type or edit Record Type properties such as Name, Prefix, Icon, etc.

Name	ID Prefix	Is Unique	Is Hierarchical
Containers	CONTNR	✓	
Folders	SYSFLD	✓	✓
Packages	PKG	✓	✓
Assemblies	ASM	✓	✓
Catalog Packages	CPKG	✓	
Components	COMP	✓	✓
Blocks	BLOCK	✓	✓
Functional Components	FCOMP	✓	✓
Physical Components	PCOMP	✓	✓
Relational Components	RELCOM	✓	✓
Software Components	SCOMP	✓	✓
Systems	SYS	✓	✓
QA Packages	QAPKG	✓	✓
Quality Assurance	QA	✓	✓
System Packages	SYSPKG	✓	
Text Packages	TXTS	✓	
Unconfigured Packages	UNCPKG	✓	
OneView Documents	OneV	✓	
Process Guidance Packages	PG2290	✓	

Object Definitions

Fields JSON Object

The fields of this object are dynamic. The structure of the object is of the following format:

```
{
    "<field1 Caption>":"<Field Value>",
    "<field2 Caption>":"<Field Value>",
    "<field3 Caption>":"<Field Value>",
    …..
    "<fieldN Caption>":"<Field Value>"
 }
```

The fields in the object depend on the field included in the concerned record type. To know the fields included in a record type, refer to Record Type Fields. For each field included in a record type a corresponding field is added to the *Fields* object. The name of the field of the object is the same as the caption of the corresponding field.

NOTE	The datatype of field value depends upon the type of the field. For more details, refer to the section JSON Data Type for Specifying Field Values.

Steps Fields JSON Object

The following table specifies the JSON data types to use when defining values for different types of *Steps* type fields.

Type of Steps Field	JSON Data Type	Remark
version	Number	Specifies the schema version of the test case steps template. It ensures compatibility with the defined structure.
column	Array of objects	Defines the column headers of the steps template. Each object represents a column containing and includes properties such as **id**, **name**, **width**, and **index**.
steps	Array of objects	Contains the row data corresponding to the columns. Each step object uses the column `id`s as keys, with their values representing the data for that column. Each step also includes metadata fields such as **Id**, **ResultUpdBy**, and **ResultUpdDt.**

The fields of this object is in the following format:

```
{
   "version": 2,
      "columns": [
        { "id": 100, "name": "Step No", "index": 0 },
        { "id": 110, "name": "Action", "index": 1 },
        { "id": 170, "name": "Test Data", "index": 2 },
        { "id": 120, "name": "Expected Result", "index": 3 },
        { "id": 130, "name": "Pass / Fail", "index": 4 },
        { "id": 140, "name": "Actual Result / Observation", "index":
5 }
      ],
      "steps": [
        {
          "stepNo": "1",
          "action": "Customer goes to Login Screen",
          "testData": "",
          "expectedResult": "System displays login and prompts for
credentials. The system shall use RSA 256 bit encryption for all
authentication tokens.",
          "actualResult": "System displays login screen and prompts
as expected.",
          "status": "Passed",
          "updatedBy": " Cynthia Tester ",
          "updatedAt": "Mon, 22 Sep 2025 10:10:25 GMT"
        },
        {
          "stepNo": "2",
          "action": "Customer enters Credentials",
          "testData": "",
          "expectedResult": "System displays that the password is
incorrect.",
          "actualResult": "Skipped – test not executed.",
          "status": "Skipped",
          "updatedBy": " Cynthia Tester ",
          "updatedAt": "Mon, 22 Sep 2025 10:13:02 GMT"
        },
        {
          "stepNo": "3",
          "action": "Incorrect password attempt is more than 3 times",
          "testData": "",
          "expectedResult": "System locks the account.",
          "actualResult": "Account was not locked after 3 incorrect
attempts.",
          "status": "Failed",
          "updatedBy": " Cynthia Tester ",
          "updatedAt": "Mon, 22 Sep 2025 10:13:13 GMT"
        }
      ]
}
```

NOTE	The specific fields included in the *steps* object depend on the *Steps* field defined for the corresponding record type.

JSON Data Type for Specifying Field Values

The following table specifies the JSON data type to use when defining values for different types of record type fields.

Type of Record Type Field	JSON Data Type	Remark
Number Field	Number	
Decimal Field	Number	
Text Field	String	You need to encode the string.
Rich Text Field	String	You need to encode the string.
List Field	String	Specify the name of a value.
Multi-Value Field	String	Specify the names of values separated by a comma.
User Field	String	Specify user name.
Team Member Field	String	Specify user name.
Project Field	String	Specify project name.
State Field	String	Specify state name.
Date Field	String	String ISO 8601 format without time and time zone. e.g. 2023-01-01 (YYYY-MM-DD)

Date Time Field	String	String ISO 8601 format without time zone. e.g. 2023-08-04T10:13:29.726Z (YYYY-MM-DDTHH:MM:SS) The letter 'T' is used to separate the date portion from the time portion and the letter 'Z' represents Coordinated Universal Time (UTC).
Time Field	String	String ISO 8601 format without date and time zone. e.g. 12:30:45 (12 hours, 30 minutes, 45 seconds) HH:MM:SS

URI Field JSON Object

Field Name	Type	Description
hint	String	A hint or description for the URL.
url	URL	The actual URL provided.
displayText	String	The text that should be displayed for the URI.

Attachment Field JSON Object

Field Name	Type	Description
fileName	String	Name of the attachment file, including its extension.
fileId	Resource Id	Unique ID of the attachment. The type is a string.

size	Number	File size of the attachment in bytes.
addedByUser	String	Name of the user who added the attachment.
addedOn	String	Date and time when the attachment was added. The value follows the ISO 8601 format.
note	String	Note associated with the attachment.
recordRevisionNumber	Number	Number of times the attachment has been updated.

Error JSON Object

Field Name	**Type**	**Description**
error	String	Error message.
type	String	Type of error. Possible values are *etInfo*, *etError*, or *etWarning*.
code	Number	Error code of the error.
solution	String	Reason or solution of the error.
stack	String	Call the stack of server in case of an internal error. This is sent when type field is Error.