Sopheon Accolade®

Accolade Data API Reference Guide

Version: 15.3

About Sopheon Accolade®

Document Name:	Accolade Data API Reference Guide
Document Version:	1
Software Version:	Sopheon Accolade 15.3
Document Date:	May 2023

Ownership of Software and Documentation

The Sopheon® software described in this documentation is furnished under a license agreement and may be used only in accordance with the terms of that license agreement.

Sopheon Corporation and its associated Sopheon Group companies, including its subsidiaries, its immediate holding company and its ultimate holding company (together, "Sopheon") have created and own all rights to the software and documentation. Licensees of the software have purchased a limited right to use the software in accordance with their license agreement.

Copyright Notice

Trademarks

No Warranty

The technical documentation is being delivered to you AS-IS, and Sopheon makes no warranty as to its accuracy or use. Any use of the technical documentation or the information contained therein is at the risk of the user. Documentation may include technical or other inaccuracies or typographical errors. Sopheon reserves the right to make changes without prior notice. In no circumstances will Sopheon, its agents or employees be liable for any special, consequential or indirect loss or damage arising from any use of or reliance on any materials in this documentation or in the software.

Patents

Aspects of Sopheon software are protected by U.S. Patents 5634051, 6632251, and 6526404; European Patent EP0914637; and by U.K. Patent GB2341252A.

Contents

About this Guide **7**

Accolade Web API Overview **9**

Accolade as a RESTful API 10

Accolade API Versioning 10

Accolade API Authentication and Access 10

API Authentication as an Application **11**

Creating and Managing Accolade Data API Keys 12

API Key Security Best Practices 13

API Key Authentication Setup 14

Option 1: API Key Authentication 14

Option 2: Bearer Authentication using an API key 14

Appendix A Accolade Web API Developer References **16**

Accolade Web API Design 17

Versioning 17

Restrictions and Rate Limits 17

Resources 18

Verbs 18

Status Codes 19

Payload 20

Actions and Functions 21

OData 21

Paging, Filtering, and Data Shaping 21

Alternative Keys 23

Caching 24

Concurrency 24

Cross-Origin Resource Sharing (CORS) 24

Security 24

Batch Support 25

Standardized Properties 27

Accolade Web API Navigation 29
Appendix B Data API Tables Reference 30
Data API Load Schedule Best Practices 31
User Data Tables 32
Users 32
User Access Log 33
User Functions 34
Security Data Tables 35
Security Users 35
Security Lists 36
Projects Data Tables 37
User Project Access 37
Project Security List Values by List 38
Project Details 39
Project History 41
Project Team Members 43
Project Stages and Gates 44
Project Gatekeepers 46
Project Deliverables 48
Project Activities 50
Project MS Project Tasks 52
Trended Metric History 54
Trended Project Metrics 55
Project Matrix Metrics 56
Project Metrics 57
Related Projects Data Table 58
Reference Tables Data Table 59
Resources Data Tables 60
Resource Pools 60
Resources 61
Resource Capacities 62
Resource Demands 63

Time Periods ... 65
Functions Data Table ... 66
Time Tracking Data Tables ... 67
Timesheets ... 67
Timesheet Rows ... 69
Timesheet Entries ... 71
User Scenario Access and Favorite Information Data Table ... 72
Scenarios Data Table ... 73
Scenario Editors Data Table ... 75
Scenario Projects Data Tables ... 76
Scenario User Project Access ... 76
Scenario Project Security List Values by List ... 77
Scenario Project Details ... 78
Scenario Project Stages and Gates ... 80
Scenario Project Matrix Metrics ... 82
Scenario Project Metrics ... 84
Scenario Resources Data Tables ... 85
Scenario Resource Pools ... 85
Scenario Resources ... 86
Scenario Resource Capacities ... 87
Scenario Resource Demands ... 89
Snapshots Data Table ... 91
Snapshot Projects Data Tables ... 92
Snapshot User Project Access ... 92
Snapshot Project Security List Values by List ... 93
Snapshot Project Details ... 94
Snapshot Project Stages and Gates ... 96
Snapshot Project Matrix Metrics ... 98
Snapshot Project Metrics ... 99
Snapshot Resources Data Tables ... 100
Snapshot Resource Pools ... 100
Snapshot Resources ... 101
Snapshot Resource Capacities ... 102

Snapshot Resource Demands 104
Custom Data Tables 106

About this Guide

Welcome to the *Sopheon Accolade Data API Reference Guide*. The Data API area allows Accolade users to retrieve Accolade data for integration into the BI application of their choice.

This guide contains instructions for creating and managing Accolade Data API keys to access Accolade information, as well as design and data tables information for BI data scientists and developers.

This information presented in this guide is intended for use with the Accolade Web API v. 2.5, to be used in conjunction with Sopheon Accolade v.15.3.

Note: Your company may use other components of the Accolade Web API resources for other purposes in your business. This guide discusses only the information you need to access the Data API resources contained within the Accolade Web API. For additional information about the Accolade Web API, please contact Sopheon Customer Support or refer to the Web API topics in the Accolade application main Help.

Assumptions

This guide assumes you are familiar with Accolade and with your company's BI application. For more information about Accolade, see the Accolade online Help available from within the main application.

Font Conventions

- This **bold font** is used for important words and the names of the items you need to identify.

  Create a SQL account named “Geneva”, and give this account the **VS_Write** database role.

- `This fixed-width font` is used for examples of code, paths, and URLS.

  `https//:your-server-name:port-number/`

- *This italic font* is used for document names.
- *An italic font* enclosed in brackets shows what information is displayed in this location when the information is changeable, rather than fixed.

  Process Document - Smart Excel <*version*>.xlt

- Blue text indicates a cross-reference link that you can click to take you to that location.

Icon Conventions

 - Indicates a tip to assist with Accolade configuration or management.

- Indicates an example use case to assist with Accolade component configuration.

Important! This is an important statement. Read it carefully before proceeding with an action.

Related Documentation

Sopheon Accolade Web API Reference Guide

Contacting Technical Publications

To send comments and suggestions regarding this document, send email to techpubs@sopheon.com.

Chapter 1

Accolade Web API Overview

An application programming interface (API) is an interface that enables interaction with other software. APIs let programs share information and influence each others' behavior through a "request and response" method of conversation. This conversation is similar to a conversation between people, but with defined rules about the type of communication allowed. APIs can be used both to retrieve and update data.

Accolade as a RESTful API

The Accolade Web API is a private RESTful web service, intended for use by company assets and by our customers. A RESTful API is based on representational state transfer (REST), an architectural style and approach to communications often used in web services development. A RESTful web application exposes information about itself in the form of information about its resources. It also enables the client to take actions on those resources, such as create new resources (i.e. create a new user) or change existing resources (i.e. edit a post).

Accolade's Web API conforms to the Richardson Maturity Model Level 2. It is implemented with ASP.Net Web API 2.2 with attribute routing, and follows OData v3 conventions. JSON is the media type that is supported for request and response payloads.

Accolade API Versioning

Accolade v.15.3 currently supports Accolade Web API version 2.5. The Accolade Web API is semantic versioned, and the version number is independent from the Accolade version. The version number has the MAJOR.MINOR format and is incremented as follows:

- **MAJOR** version when there are **incompatible** API changes.
- **MINOR** version when functionality is added in a **backwards-compatible** manner.

Accolade API Authentication and Access

Accolade has several areas within its API designed to retrieve the appropriate information based on the application. Before users can make requests, they need to have a method for authenticating the requests. The Accolade API has two methods for verifying requests:

- using authentication tokens to identify and verify the users.
- via a generated API key, which identifies the application making the API call.

Access to the Accolade Web API is available to any developer, and requests are authenticated via the user's Accolade credentials. The Web API areas can be used to create customized configuration solutions for users within your company, such as creating an advanced layout which uses functionality in a non-standard way, or retrieving the results of an Accolade online report for use in process configuration.

Access to the Accolade Data API requires the use of an API key, which can be created by users with the Data Analyst role and then shared with appropriate users within your organization. The Data API area can be used to retrieve project data and other data from the Accolade database, and display the Accolade information using your company's BI application.

Chapter 2

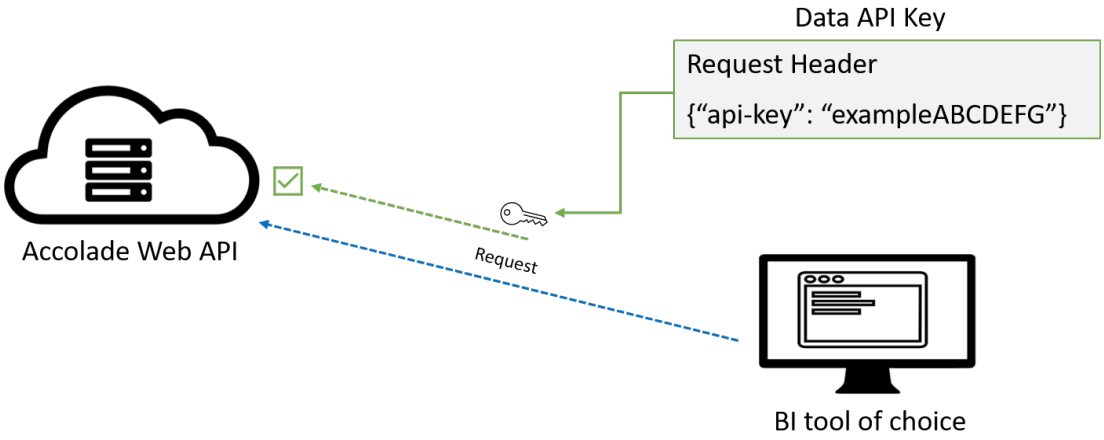
API Authentication as an Application

The Accolade Data API area requires the use of generated API keys for use in retrieving project data and other data from the Accolade database, and displaying the Accolade information using your company's BI tool of choice. These API keys can be created within Accolade and then shared with appropriate users within your organization.

API keys are used to identify the application making the call, and are designed to serve two main purposes:

- Application identification - to identify the application making the call to the API.
- Application authorization - to help check whether the application making the call has access to call it. It also checks whether the API in this application is enabled.

Note: While API keys identify the calling project, they don't identify the calling user. For instance, if you have created an application that is calling an API, an API key can identify the application that is making the call but not the identity of the person who is using the application.



Creating and Managing Accolade Data API Keys

Users with the Data Analyst role can create access keys that can be provided to developers for use to authenticate calls to the Accolade API when requesting Accolade information. API keys can be viewed and managed from the **System** menu, under **System > Data API**.

Important! Each API key is generated and revealed at the time of creation, and cannot be viewed a second time. If this key is lost or misplaced, you will need to delete the existing key and create a new one.

To create an Accolade API key:

1. From the **System** menu, select **System > Data API**.
2. Click **Add New** in the upper right corner of the page
3. Complete the following information to identify the key:

   Required fields display with **red** text and an asterisk * if the field is empty.

Field	Description
Name	Enter a name, up to 64 characters long, which identifies the API key.
Description	Enter a description of the purpose or nature of the API key.

4. Click **Create** to generate a new API key.

   After you click **Create** and generate the new API key, the key value is displayed. Click **Copy** to copy the key value a different location, as you will need it to access the Accolade Data API and will not be able to access it again once you exit the dialog.

5. Click **Done** to exit the dialog.

API Key Security Best Practices

API keys grant full access to your Accolade data, and should be protected the same way you would protect your password. In particular, there are a few common scenarios to keep in mind when working with API keys:

- Give each integration its own API key, and assign descriptions to each key so you know which key goes with which application. If a specific API key is compromised, you can disable that key without disabling access to all of your other integrations.
- Be careful not to expose the key to the public (such as in screenshots, videos, or help documentation). Remember that blurring your data isn't always enough. It's best to use "cut" functions in your graphics program to remove the data completely.
- If a key needs to be shared, generate a new key and label it accordingly so it can be disabled, if needed. Never email the API key, because it would allow access to your Accolade information if hackers were to compromise your email account.
- Do not embed, store, or expose your API key(s) within client-side code, including JavaScript, mobile applications, and native executables. A bad actor can decompile your application and obtain your API key(s).
- Delete unneeded API keys to minimize exposure to attacks.

Notes:

- To delete an existing API key, click  in the appropriate key row.

API Key Authentication Setup

Access to the Accolade Data API is available via a generated API key that is generated within Accolade, and can be used by any *in-network* authorized developer. Once the key is generated, your company's BI analysts have two different options for authentication setup.

Option 1: API Key Authentication

Analysts can use the API key directly to authenticate access to Accolade data. Using this method, each call to the API needs a Authorization header with API Key scheme to authenticate the application, as follows:

Request Example	Response
GET https://<server>/api/v2/data/GetTimePeriodsData Authorization: APIKey <API Key name>:<API Key value>	

Option 2: Bearer Authentication using an API key

Analysts can use the Bearer authentication method to authenticate access to Accolade data, to give additional security to calls being made by an application.

To request an API key-based access token:

Request Example	Response

POST https://<server>/Token
Content-Type: application/json
 	
{
"grant_type": "client_credentials",
"client_id": "<API key name>",
"client_secret": <API key value>"
}
	200 OK
	Content-Type: application/json; charset=utf-8
	
	{
	"access_token": "<token>",
	"token_type": "Bearer",
	"expires_in": "<seconds>",
	"refresh_token": "<token>"
	}

The access token expires in expires_in seconds. For refresh token, see the Refresh Tokens section below.

Refresh Tokens

If refresh tokens is enabled, you can use a refresh token to get a new access token without supplying the API key again. The refresh token expires after 365 days.

Request Example	Response

POST https://<server>/Token
Content-Type: application/json
 	
{
"grant_type": "refresh_token",
"refresh_token": "<refresh token>"
}
	200 OK
	Content-Type: application/json; charset=utf-8
	
	{
	"access_token": "<token>",
	"expires_in": "<seconds>",
	"refresh_token": "<token>"
	}

Note that the returned refresh token is a new token with a new expiration window.

To call the API with the Access Token:

Using this method, each call to the API needs an Authorization header with Bearer scheme to authenticate the application, as follows:

Request Example	Response
GET https://<server>/api/v2/data/GetTimePeriodsData Authorization: Bearer <access token>	

Notes:

- The content type for a Data API call can be either json or csv, depending on your use scenario.
- All API requests should be made over HTTPS. Calls made over plain HTTP could potentially expose information such as passwords or client secrets to other users within your network.
- The Accolade Web API technical documentation, which includes examples, can be accessed at *<server name>/help/apihelp*.
- In you are unfamiliar with working in APIs or require more technical support, contact Sopheon's Consulting group for more information on training opportunities or additional services.

Appendix A

Accolade Web API Developer References

This section contains developer information about the Accolade Web API resources.

Accolade Web API Design

The Accolade API is a private RESTful web service, intended for use by company assets and by our customers. Accolade's API conforms to the Richardson Maturity Model Level 2. It is implemented with ASP.NET Web API 2.2 with attribute routing, and follows OData v3 conventions.

Versioning

The Accolade Web API is semantic versioned, and the version number is independent from the Accolade version. The version number has the MAJOR.MINOR format and is incremented as follows:

- **MAJOR** version when there are **incompatible** API changes.
- **MINOR** version when functionality is added in a **backwards-compatible** manner.

Accolade v.15.3 currently supports Accolade Web API version 2.5.

Restrictions and Rate Limits

The only restrictions to the Accolade API are via authentication and authorization as follows:

- User authentication - verify that the API user making the call has the appropriate credentials.
- User authorization - a check whether the API user making the call has permission to make this kind of request.

Important! The Accolade Web API is sensitive to the user roles and rights that are assigned within Accolade. When creating API calls that will be made on behalf of an Accolade user, be mindful that the user must have the appropriate Accolade system roles and rights to access the data or perform the action included in the call. For example, Team Members do not have the appropriate rights to create projects, therefore it would not make sense to assign a POST call that creates projects for this user to execute.

Note: Roles and rights are not applicable to calls made to the Accolade Data API that are authenticated via a generated API key. See " API Authentication as an Application" on page 11 for more information.

Rate limits determine how frequently you can call a particular endpoint. While Accolade's API does not currently impose any rate limits, be mindful that API usage may affect the performance of Accolade; you may wish to schedule large updates or requests outside of peak usage times.

Resources

A resource is uniquely identified by its URI (or URL). Resources URIs must follow the following conventions:

- Resources are grouped in **areas**. Examples: `core, administration, configuration`.
- Resources are **nouns** that convey meaning. Examples: `/api/v2/core/projects, /api/v2/configuration/processmodels`.
- Resource names should be **pluralized**. Examples: `/api/v2/core/projects, /api/v2/administration/users`.
- A single resource is addressed with its **ID in parenthesis** after the resource name. Examples: `/api/v2/core/projects(1), /api/v2/administration/users(42)`.
- Alternatively, a resource may be addressed with its **name or system name** in single quotes. Examples: `/api/v2/core/projects('Project%201'), /api/v2/configuration/metrics('Costs')`.
- Resources represent **contained structure**. Structure is represented by **hierarchical resource paths**. Examples: `/api/v2/core/projects(1)/metrics, /api/v2/core/projects(1)/metrics(111)`.
- Filters, sorting, etc. aren't resources thus should be in the **query string** of the URL. Example: `/api/v2/core/projects?$orderby=Name`.

Resource IDs should be unique, static and remain the same over time.

Verbs

The following verbs (**HTTP Methods**) are supported:

HTTP Method	Request Payload	Sample URI	Response Payload
GET	-	`/api/v2/core/projects` `/api/v2/core/projects({projectID})`	Resource collection Single resource
POST	Single resource	`/api/v2/core/projects`	Single resource
PUT	Single resource	`/api/v2/core/projects({projectID})`	Single project
PATCH	Batch of partial resources Partial resource	`/api/v2/core/projects` `/api/v2/core/projects({projectID})`	Resource collection

HTTP Method	Request Payload	Sample URI	Response Payload
			Single resource
DELETE	-	`/api/v2/core/projects({projectID})`	-

The difference between **PUT** and **PATCH** is that **PUT** will update the whole resource, so for not supplied properties their default values will be used. **PATCH** only updates supplied properties.

Status Codes

Status Code	HTTP Methods	Comments
Level 200 - Success		
200 - OK	GET, PUT, PATCH	And POST for Actions.
201 - Created	POST	
204 - No content	DELETE	
Level 400 - Client Errors		
400 - Bad request	POST, PUT, PATCH	Invalid or corrupt request payload, the response body contains the error message.
401 - Unauthorized	All	No or invalid authentication details provided.
403 - Forbidden	All	Authenticated user doesn't have access to resource.
404 - Not found	GET, PUT, PATCH, DELETE	Resource not found.
405 - Method not allowed	All	HTTP method not allowed on resource.

Status Code	HTTP Methods	Comments
406 - Not acceptable	All	Media type not supported (response payload).
409 - Conflict	All	Caching or concurrency conflict (or resource already exists when trying to create it).
415 - Unsupported media type	POST, PUT, PATCH	Media type not supported (request payload).
Level 500 - Server Faults		
500 - Internal server error	All	Error will be logged in the Accolade log. No error message will be returned as this exposes implementation details.

Payload

Resource != Business Model != Entity Model

Resources are mapped to one or more entities or their subsets. For example, a metric can contain both project and process model related properties.

A entity can be represented by more than 1 resource, depending on their usages. For example, a metric can have a separate resource type for: process model, project, matrix, planning/roadmap, etc.

All resources will have standardized property names and standardized value types as follows:

- **Enum** values are represented as **strings** and not as their underlying numeric value.
- **Dates** as **'YYYY-MM-DD'** strings.
- **Timestamps** as **'YYYY-MM-DDThh:mm:ss'** strings.

Related resources will be represented with their **ID** and can be expanded (with the `$expand` option) to full resource objects to reduce server round trips; for example, the team leader ID on a project can be expanded to a user resource.

Only JSON (media type: **application/json**) will be supported for request and response payloads. JSON literals will be in **camel-case**. Null values will be **omitted** from the payload to reduce network bandwidth.

Actions and Functions

Besides resources, actions and functions can be exposed by the API; this should be used sparingly.

The difference between actions and functions is that **actions** can have **side effects**, and **functions** do **not**. Both actions and functions can return data.

Actions and function should be **named** like C# method names, in other words, the name should be a **verb phrase**. **Actions** should be invoked with a **POST** method, **functions** with a **GET** method.

Actions and functions can be global, for a resource collection or for a single resource.

Examples:

```
api/v2/administration/users/SetPushSubscription

api/v2/core/ExecuteQuery(123)

api/v2/core/projects(42)/phases(1)/gate/keepers(5})/EnterVote

api/v2/core/projects(42)/MigrateProject(111)
```

OData

A request returns a maximum of 50 items in a collection, for example `api/v2/Projects` returns 50. To avoid the limitation you can use the OData $top token and set the results count to any number you need in the form:

```
api/v2/Projects?$top=100000
```

OData parameters are passed as a regular query string. For example `http://domain.com/page?$top=100&$skip=10` will pass 2 parameters, `$top` and `$skip`. You can pass as many parameters as you like, separated with a &.

Paging, Filtering, and Data Shaping

The following options are supported on resources collections (`$expand` and `$select` are also applicable on a single resource).

Option	Descrip-tion	Format
$expand	Expands related entities inline.	$expand is a comma-separated list of related resources to be included in line with the retrieved resources. Nested resources can be expressed using a slash (‘/’), max depth is 10. **Example:** `/api/v2/core/projects(42)?$expand=Metrics,Phases/Stage/Deliverables`
$filter	Filters the results, based on a Boolean conditio n.	$filter is a Boolean expression to filter a collection of resources. The expression specified with the filter is evaluated for each resource and when the expression evaluates to true, the resource is included in the result. All OData v3 operators and functions are supported; see http://www.odata.org/documentation/odata-version-3-0/url-conventions/, section ’5.1.2. Filter System Query Option’. **Example:** `/api/v2/core/projects?$filter=Name+eq+'Ajax'` Nested $filter statements are not supported ($expand=Metrics($filter=DataType+eq+'String'))
$inlineco unt	Include the total count of matching entities in the respons e.	$inlinecount with the value allpages returns the total number of resources in the (filtered) collection in the all-pages-count HTTP header. **Example:** `/api/v2/core/projects?$inlinecount=allpages`
$orderby	Sorts the results.	$orderby is a comma-separated list of property order clauses for resources to be returned in either ascending order using asc or descending order using desc. If asc or desc not specified, then the resources will be ordered in ascending order. **Example:** `/api/v2/core/projects?$orderby=TeamLeader+asc,Name+desc` Nested $orderby statements are not supported

Option	Descrip-tion	Format
		($expand=Metrics($orderby=SystemName)).
$select	Selects which properties to include in the response.	$select is a comma-separated list of resource properties to be returned. Nested resources can be expressed using a slash ('/'), max depth is 10. **Example:** `/api/v2/core/projects(42)?$expand=Metrics&$select=Name,Code,Metrics/SystemName`
$skip	Skips the first in results.	$skip is a number that defines the number of resources to be skipped and not included from a (filtered) collection. **Example:** `/api/v2/core/projects?$skip=10`
$top	Returns only the first in the results.	$top is a number that defines the number of resources to be returned from a (filtered) collection. If not supplied, all resources in collection are returned. **Example:** `/api/v2/core/projects?$top=5`

As of Accolade v. 13.2, support has been added for some DTO's to allow for filtering by expanded properties. Example: /api/v2/core/projects?$expand=TeamLeader&$filter=TeamLeader/Name +eq+%27FirstName%20LastName%27

Notes:

If a resource or action returns tabular data, the format query string can be used to shape the tabular data: `List, Table` and `Table with Headers`:

- `api/v2/core/ExecuteQuery`
- `api/v2/core/RunReport`
- `api/v2/core/projects/matrices`
- `api/v2/configuration/referencetables`

Alternative Keys

The APIv2 resource ID arguments can have 3 types of values:

- `Int32`: resource(123)
- `String`: resource("name")
- `Enum`: resource(Flag)

Caching

Each response will define itself a cacheable or not using HTTP Caching (RFC 2616 and RFC 7234 standards). Caching is used to eliminate the number of requests (aka network-roundtrips; uses an expiration model) and to eliminate the need to send full responses (aka network bandwidth; uses a validation model).

The following cache types will be supported in the future:

- Client Cache (== Private Cache), lives on the client
- Gateway Cache (== Shared Cache), lives on the server
- Proxy Cache (== Shared Cache), lives on the network

The expiration model will be defined with the Cache-Control header. The validation model will be defined with the `Last-Modified` and `ETag` (strong and/or weak) headers. Only `no-cache` is supported for now.

Notes:

`api/v2/core/ExecuteQuery` and `api/v2/core/RunReport` already support `Last-Modified` and `ETag` to avoid re-executing queries and reports when paging is used (`$top` and `$skip`).

Concurrency

APIv2 will follow current Accolade practices with respect to concurrency: last save wins.

Cross-Origin Resource Sharing (CORS)

JSONP support is dropped in APIv2. Only CORS will be supported. The existing Accolade CORS infrastructure will be used.

Security

APIv2 follows all Accolade visibility, accessibility and manageability rules for resources and their underlying entities (role, access groups, security list, security profiles, team membership, etc.).

If APIv2 is called from within Accolade (i.e. Quick Grids), the user is already authenticated and the session and authentication cookies will be send with each request automatically.

If APIv2 is called from outside Accolade and Accolade is configured for Windows Integration authentication, APIv2 can be called without using an access token (providing that the HTTP request stack supports this).

If APIv2 is called from outside Accolade and Accolade is configured for LDAP or SSO (WS-Fed, SAMLp, OAuth2 and/or OpenID Connect), an access token should be requested. The access token should be supplied on each request as HTTP Authorization header with Bearer scheme (Authorization: Bearer <access token>).

Currently only the Resource Owner Password Credentials Grant flow is supported by Accolade. This flow requires the submission of username and password to obtain the access token (see /Token endpoint). Also, the OAuth2 offline scope (aka refresh tokens) is supported; this means a refresh token (that has a long expiration date) can be used to get an new access token without supplying the user credentials again.

Note: Currently there is no other means than deleting or de-activating the user in Accolade to revoke access and refresh tokens.

Batch Support

A batch request combines multiple APIv2 requests into a single POST request to the `/api/v2/$batch` endpoint. The payload should be multipart/mixed.

Request Example (only showing `GET` requests, but `POST, PUT, PATCH,` and `DELETE` requests can be included as well:

Request

```
POST http://<server>/api/v2/$batch HTTP/1.1
Content-Type: multipart/mixed; boundary="batch_e5b6e99a-61b3-4369-9331-c87803c7089a"
Host: <host>
Content-Length: 409
Expect: 100-continue
--batch_e5b6e99a-61b3-4369-9331-c87803c7089a
Content-Type: application/http; msgtype=request
GET /api/v2/administration/users(1) HTTP/1.1
Host: <host>

--batch_e5b6e99a-61b3-4369-9331-c87803c7089a
Content-Type: application/http; msgtype=request
GET /api/v2/core/projects HTTP/1.1
Host: <host>
```

Request

```
--batch_e5b6e99a-61b3-4369-9331-c87803c7089a--
```

Response Example (only showing `GET` requests, but `POST`, `PUT`, `PATCH`, and `DELETE` requests can be included as well:

Response

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 507
Content-Type: multipart/mixed; boundary="2d36ec9b-bf61-4ba9-
b265-ffb7604cff0a"
Expires: -1
Persistent-Auth: true
X-Frame-Options: SAMEORIGIN
Date: Wed, 31 May 2017 13:49:24 GMT

--2d36ec9b-bf61-4ba9-b265-ffb7604cff0a
Content-Type: application/http; msgtype=response

HTTP/1.1 200 OK
api-version: 2.0
Content-Type: application/json; charset=utf-8

{
"id": 1,
"name": "Administrator"
}
--2d36ec9b-bf61-4ba9-b265-ffb7604cff0a
Content-Type: application/http; msgtype=response

HTTP/1.1 200 OK
api-version: 2.0
Content-Type: application/json; charset=utf-8
```

Response

```
[
{
"code": "1",
"id": 1,
"name": "Ajax"
}
]
--2d36ec9b-bf61-4ba9-b265-ffb7604cff0a--
```

NET Framework and Xamarin (.NET for iOS and Andriod app) already have classes to compose a batch request and parse a batch response. There are several JavaScript libraries out there that can do the same, we need to select one (preferable an jQuery extension:batchjs.zip) to be used for Quick Grid customizations.

Standardized Properties

Name	Type	Description
Id	Long	Resource ID
Name	String	Resource display name
SystemName	String	Resource system name, unique across resource type, case insensitive
Description	String	Resource description
Order	Long	Position of resource in collection
*action*Date	Date	Date of action **Examples:** `CreatedDate ClosedDate` (no `LastModifiedDate` but `UpdatedDate` instead)
*action*ById	Long	ID of user that executed the action **Examples:** `CreatedById, ClosedById`
Is*Flag*	Boolean	A flag **Examples:** `IsClosed, IsActive, IsCalculated, IsShowMessagesEnabled`
Can*Flag*	Boolean	A flag **Examples:**

Name	Type	Description
		`CanCreateStatusReports`
*resource*Id	Long	ID of related resources **Examples:** `TeamLeaderId, ClassId, ProcessModelId`
resource	DTO	Expanded related resource **Examples:** `CreatedBy, ClosedBy, TeamLeader, Class, ProcessModel`
resource + Parent*Resource*	DTO[] + DTO	Hierarchical relationships **Examples:** `/aNodes + ParentNode, Groups + ParentGroup`
Options	Enum	Options and flags
Members	DTO[]	Resource members **Examples:** `Project Team, AccessGroup, SecurityList`
ExtendedFields	DTO[]	Metadata and extended fields **Examples:** `Project Metadata, Deliverable ExtendedFields, User ExtendedFields`
Notes	String	**Examples:** `StatusText, Comments`
*type*DownloadUri	String	**Examples:** `ProjectDocumentVersionDto.DownloadUri, GateDocumentDto.TemplateDownloadUri, ImageDto.DowloadUri`
ModelId	Long	Resource model ID **Examples:** `ProcessModelId, PMPhaseId, PMDeliverableId, PMActivityId, PMGateId`
Links	DTO[]	**Examples:** `Associations, ProjectLinks`
RequestorRights	Enum	Resource rights and permissions for the API caller.

Accolade Web API Navigation

In addition to the overview information provided within the online Help, Accolade provides technical documentation for developers that is available with your Accolade installation. The documentation, which includes the available methods and call examples, can be accessed at *<server name>/help/apihelp*.

> If you are unfamiliar with working in APIs or require more technical support, contact Sopheon's Customer Support group for more information on training opportunities or additional services.

The Accolade Web API documentation is divided into areas representing different areas where information can be accessed within Accolade.

Category	Description
Administration	This category contains the calls that are specific to access and security within Accolade.
Configuration	This category contains the calls that are specific to the configuration components within Accolade.
Core	This category contains the calls that are specific to project data within Accolade.
Data	This category contains the calls that are specific to accessing the data tables for use with your company's BI tool of choice. **Important!** This category requires an Accolade API key in order to access, and should not be used with any advanced configuration within Accolade. See "Creating and Managing Accolade Data API Keys" on page 12 for more information.
Miscellaneous	This category contains the calls that are not applicable to the remaining categories.
Resource Planning	This category contains the calls that are specific to the Resource Planning data within Accolade.

Appendix B

Data API Tables Reference

This section lists the column sets and columns in each subject available for calls by the Accolade Data API.

Note: This section contains calls that correspond to Resource Planning components, and may not return information based on your company's configuration. Resource Planning is an optional Accolade component that you may not have access to. To implement this solution, contact Sopheon Customer Support.

Data API Load Schedule Best Practices

Whenever you refresh data, your BI application must query the Accolade database, possibly load the source data into a dataset, and then update any visualizations in your reports or dashboards that rely on the updated dataset. This process consists of multiple phases, depending on the application.

Keep the following set of best practice recommendations in mind when setting up your Accolade data load schedules for your BI application:

- **Load What You Need** - Since the Data API allows for very fine-grain control of the Accolade data that can be retrieved, you can be very selective when loading data into your BI application. So in general, if you are not using a particular dataset, don't load it. For example, if your organization does not utilize snapshot or scenario functionality, don't tie up resources by calling this information for use in your BI application.
- **Managing Load Volumes** - Querying a large amount of data can potentially impact your Accolade server performance. Scheduling data refreshes during off-hours and avoiding peak use times is recommended in general.
- **Query Timing** - Be mindful when you set up automatic data refreshing, and select schedules that are appropriate for the type of data being retrieved.

  Consider the following scenarios:

  - You may have dashboards that display Accolade project financial data that is only updated at month-end. For optimal results, this data should be scheduled to refresh on a monthly basis that aligns with the Accolade project updates, so that the data presented in your BI application is current and accurate.
  - You many have dashboards that display project deliverable and/or activity status updates that are updated as the project progresses. In this instance, you may want to refresh the data weekly (or more frequently, depending on your projects), in order to report pertinent project information that may impact important strategic meetings or decisions.
  - You may have dashboards that display Accolade project data that is captured in quarterly snapshots. Due to the volume and snapshot schedule, this data should be scheduled to refresh on a quarterly basis that aligns with the snapshot timing, so that you are not using resources reloading the same dataset to your BI application.

User Data Tables

Users

This table contains details on users.

Accessed by: `/api/v2/Data/GetTableData('SGM_CORE_Users')`

Columns	Datatypes	Used for Joins
UserID_SYS	Number	Yes
User Name	String	No
User Login	String	No
Email Address	String	No
Chat Address	String	No
Deleted	String	No
Active	String	No

How to join to other tables:

- **UserID_SYS** can be used to join to other tables that contain the **UserID_SYS** column (or any other user ID column).

Load Order Examples:
`/api/v2/Data/GetTableData('SGM_CORE_Users')` **Important!** This list of examples includes some calls to get you started, but does not include all available options.

User Access Log

This table contains information on user access logs.

Accessed by: `/api/v2/Data/GetTableData('SGM_CORE_User Access Log')`

Columns	Datatypes	Used for Joins
Login Date	Date	No
User Login	String	No
User ID	Number	Yes
User Name	String	No
Login Result	String	No
IP Address	String	No
Language	String	No
User Agent	String	No

How to join to other tables:

- To access more User information, use the **User ID** column to join to the **UserID_SYS** column from the **Users** table.

Load Order Examples:

```
/api/v2/Data/GetTableData('SGM_CORE_Users')
/api/v2/Data/GetTableData('SGM_CORE_User Access Log')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

User Functions

This table links users to their associated functions.

Accessed by: `/api/v2/Data/GetTableData('SGM_CORE_User Functions')`

Columns	Datatypes	Used for Joins
UserID_SYS	Number	Yes
FunctionID_SYS	Number	Yes

How to join to other tables:

- **UserID_SYS** can be used to join to other tables that contain the **UserID_SYS** column (or any other user ID column).
- To access more User information, use the **User ID** column to join to the **UserID_SYS** column from the **Users** table.
- To access more Function information, use the **FunctionID_SYS** column to join to the **FunctionID_SYS** column from the **Functions** table.

Load Order Examples:
`/api/v2/Data/GetTableData('SGM_CORE_Users')`
`/api/v2/Data/GetTableData('SGM_CORE_Functions')`
`/api/v2/Data/GetTableData('SGM_CORE_User Functions')`

Important! This list of examples includes some calls to get you started, but does not include all available options.

Security Data Tables

Security Users

This table contains user information for doing row level security on project data against the accessing user's User Login (aka NTName).

Accessed by: `/api/v2/Data/GetSecurityUsersData`

Columns	Datatypes	Used for Joins
SecurityUserID_SYS	Number	Yes
NTName	String	No

How to join to other tables:

- **SecurityUserID_SYS** can be used to join to other tables that contain the **SecurityUserID_SYS** (or any other user ID column).
- To secure the data in other tables, secure the project data, then use the **NTName** column to map to the login information for the user accessing the BI file.

Load Order Examples:
`/api/v2/Data/GetSecurityUsersData`

Important! This list of examples includes some calls to get you started, but does not include all available options.

Security Lists

This table contains the names of the security lists in Accolade.

Accessed by: `/api/v2/Data/GetSecurityListsData`

Columns	Datatypes	Used for Joins
SecurityListID	Number	Yes
Security List	String	No

How to join to other tables:

- **SecurityListID** can be used to join to other tables that contain the **SecurityListID** column.

Load Order Examples:
`/api/v2/Data/GetSecurityListsData`

Important! This list of examples includes some calls to get you started, but does not include all available options.

Projects Data Tables

User Project Access

This table contains information on a user's project access for use with securing project data on a row level basis.

Accessed by: `/api/v2/Data/GetUserProjectAccessData`

Columns	Datatypes	Used for Joins
SecurityUserID_SYS	Number	Yes
SecurityProjectID_SYS	Number	Yes

How to join to other tables:

- **SecurityUserID_SYS** can be used to join to other tables that contain the **SecurityUserID_SYS** (or any other user ID column).
- **SecurityProjectID_SYS** can be used to join to other tables that contain the **SecurityProjectID_SYS** (or any other project ID column).
- To secure project data, use the **SecurityProjectID_SYS** column to join to the **SecurityProjectID_SYS** column from the **Project Details** table, then use the **SecurityUserID_SYS** column to join that table to the **SecurityUserID_SYS** column from the **Security Users** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserProjectAccessData
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Project Security List Values by List

This table contains security list value information by level for a specific security list for all projects in a system.

Accessed by: `/api/v2/Data/GetProjectSecurityListValuesData ('SecurityListName')`

Columns	Datatypes	Used for Joins
ProjectID_SYS	Number	Yes

Important! The remaining columns are the security list level names returned as strings, which will vary depending on your company's configuration.

Important! For very large data sets using the default format option, (1 - List) may be slower than using other format options such as 2 (Table) or 6 (Table + Headers) or output of CSV. If you encounter timeouts using the List option, try using the other format options.

How to join to other tables:

- **ProjectID_SYS** can be used to join to other project-related tables that contain the **ProjectID_SYS** column.
- To secure the data in this table, secure the project data, then use the **ProjectID_SYS** column to join to the **ProjectID_SYS** column from the **Project Details** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserProjectAccessData
/api/v2/Data/GetTableData('SGM_CORE_Project Details')
/api/v2/Data/GetProjectSecurityListValuesData ('SecurityListName')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Project Details

This table contains details about open and closed projects.

Accessed by: `/api/v2/Data/GetTableData('SGM_CORE_Project Details')`

Columns	Datatypes	Used for Joins
Project Name	String	No
Project Description	String	No
Project ID	String	No
SecurityProjectID_SYS	Number	Yes
Project Creation Date	Date	No
Project Team Leader Name	String	No
Project Last Gate Decision	String	No
Project is in Trouble	String	No
Project Most Recent Status Report Text	String	No
Project Closed	String	No
Project Closed Date	Date	No
Idea Submitter Name	String	No
Migrated-from Project Name	String	No
ProjectID_SYS	Number	Yes
Project Team Leader ID	Number	Yes
Current Phase ID	Number	Yes
Project Model Name	String	No
Project Current Stage Name	String	No
Project Most Recent Status Report Date	Date	No
Project Most Recent Status Report Submitter Name	String	No
Project Created By	String	No
Project Class	String	No
Project Currency Code	String	No
Project Currency Name	String	No
Project Currency Conversion Factor	Number	No
Project Process Type	String	No
Access Group Name	String	No
Parent Access Group Name	String	No

Columns	Datatypes	Used for Joins
Project Start Date	Date	No
Project End Date	Date	No

How to join to other tables:

- **ProjectID_SYS** can be used to join to other project-related tables that contain the **ProjectID_SYS** column.
- To secure project data, use the **SecurityProjectID_SYS** column to join to the **SecurityProjectID_SYS** column from the **User Project Access** table, then use the **SecurityUserID_SYS** column to join that table to the **SecurityUserID_SYS** column from the **Security Users** table.
- To access more Project Team Leader information, use the **Project Team Leader ID** column to join to the **UserID_SYS** column from the **Users** table.
- To get information about current stage of a project, use the **Current Phase ID** column in conjunction with the **ProjectID_SYS** column to join to the **ProjectPhaseID_SYS** and **ProjectID_SYS** columns from the **Project Stages and Gates** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserProjectAccessData
/api/v2/Data/GetTableData('SGM_CORE_Project Details')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Project History

This table contains project events, reasons, and status messages.

Accessed by: `/api/v2/Data/GetTableData('SGM_CORE_Project History')`

Columns	Datatypes	Used for Joins
Event Stage Number	Number	Yes
Event Date	Date	No
Event Type	String	No
Event Reason	String	No
Event Reason Code	String	No
Event Reason Name	String	No
Event User Name	String	No
Event User Login	String	No
Event Stage Name	String	No
Event Gate Number	Number	No
Event Gate Name	String	No
Event Text	String	No
Change-from Date	Date	No
Change-to Date	Date	No
Change-from Decision	String	No
Change-to Decision	String	No
Change-from Name	String	No
Change-to Name	String	No
Change-to in Trouble Flag	Boolean	No
Event User ID	Number	Yes
Change-from Team Leader ID	Number	Yes
Change-to Team Leader ID	Number	Yes
ProjectID_SYS	Number	Yes
Change-from Currency Code	String	No
Change-to Currency Code	String	No
Change-from Project Name	String	No

Columns	Datatypes	Used for Joins
Change-to Project Name	String	No
Event Function Name	String	No

How to join to other tables:

- **ProjectID_SYS** can be used to join to other project-related tables that contain the **ProjectID_SYS** column.
- To secure the data in this table, secure the project data, then use the **ProjectID_SYS** column to join to the **ProjectID_SYS** column from the **Project Details** table.
- To access more Event User information, use the **Event User ID** column to join to the **UserID_SYS** column from the **Users** table.
- To access more Change-from Team Leader information, use the **Change-from Team Leader ID** column to join to the **UserID_SYS** column from the **Users** table.
- To access more Change-to Team Leader information use the **Change-to Team Leader ID** column to join to the **UserID_SYS** column from the **Users** table.
- To get information about which stage the history event happened in, use the **ProjectID_SYS** column in conjunction with the **Event Stage Number** column to join to the **ProjectPhaseID_SYS** and **ProjectID_SYS** columns from the **Project Stages and Gates** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserProjectAccessData
/api/v2/Data/GetTableData('SGM_CORE_Project Details')
/api/v2/Data/GetTableData('SGM_CORE_Project History')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Project Team Members

This table contains details about the members of each project team.

Accessed by: `/api/v2/Data/GetTableData('SGM_CORE_Project Team Members')`

Columns	Datatypes	Used for Joins
Project Team Member Name	String	No
Project Team Member Function	String	No
Project Team Member ID	Number	Yes
ProjectID_SYS	Number	Yes
Project Team Member Is Primary	Boolean	No

How to join to other tables:

- **ProjectID_SYS** can be used to join to other project-related tables that contain the **ProjectID_SYS** column.
- To secure the data in this table, secure the project data, then use the **ProjectID_SYS** column to join to the **ProjectID_SYS** column from the **Project Details** table.
- To access more Project Team Member information, use the **Project Team Member ID** column to join to the **UserID_SYS** column from the **Users** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserProjectAccessData
/api/v2/Data/GetTableData('SGM_CORE_Project Details')
/api/v2/Data/GetTableData('SGM_CORE_Project Team Members')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Project Stages and Gates

This table contains the stage and gate details for projects.

Accessed by: `/api/v2/Data/GetTableData('SGM_CORE_Project $STAGES$ and $GATES$')`

Columns	Datatypes	Used for Joins
Project Stage Exists	String	No
Project Stage Name	String	No
Project Stage Relative Position	String	No
Project Stage Duration	Number	No
Project Gate Exists	String	No
Project Gate Name	String	No
Project Gate Date	Date	No
Project Gate Decision	String	No
Project Gate Meeting Place	String	No
Project Gate Notes	String	No
Project Gate Plan Date	Date	No
Project Gate Relative Position	String	No
ProjectPhaseID_SYS	Number	Yes
Project Gate Conditions	String	No
Project Gate Owner Name	String	No
Project Gate Owner ID	Number	Yes
ProjectID_SYS	Number	Yes
Project Stage Locked	String	No

How to join to other tables:

- **ProjectID_SYS** can be used to join to other project-related tables that contain the **ProjectID_SYS** column.
- **ProjectPhaseID_SYS** can be used in conjunction with the **ProjectID_SYS** column to join to other tables that contain the **ProjectPhaseID_SYS** (or any other phase ID column) and **ProjectID_SYS** columns.
- To secure the data in this table, secure the project data, then use the **ProjectID_SYS** column to join to the **ProjectID_SYS** column from the **Project Details** table.

- To access more Project Gate Owner information, use the **Project Gate Owner ID** column to join to the **UserID_SYS** column from the **Users** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserProjectAccessData
/api/v2/Data/GetTableData('SGM_CORE_Project
Details')
/api/v2/Data/GetTableData('SGM_CORE_Project $STAGES$
and $GATES$')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Project Gatekeepers

This table contains gatekeeper details for projects.

Accessed by: `/api/v2/Data/GetTableData('SGM_CORE_Project $GATEKEEPERS$')`

Columns	Datatypes	Used for Joins
Project Gatekeeper Name	String	No
Project Gatekeeper Function	String	No
Project Gatekeeper ID	Number	Yes
ProjectID_SYS	Number	Yes
ProjectPhaseID_SYS	Number	Yes

How to join to other tables:

- **ProjectID_SYS** can be used to join to other project-related tables that contain the **ProjectID_SYS** column.
- **ProjectPhaseID_SYS** can be used in conjunction with the **ProjectID_SYS** column to join to other tables that contain the **ProjectPhaseID_SYS** (or any other phase ID column) and **ProjectID_SYS** columns.
- To secure the data in this table, secure the project data, then use the **ProjectID_SYS** and **ProjectPhaseID_SYS** columns to join to the **ProjectID_SYS** and **ProjectPhaseID_SYS** columns from the **Project Stages and Gates** table.
- To access more Gatekeeper information, use the **Project Gatekeeper ID** column to join to the **UserID_SYS** column from the **Users** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserProjectAccessData
/api/v2/Data/GetTableData('SGM_CORE_Project Details')
/api/v2/Data/GetTableData('SGM_CORE_Project $STAGES$ and $GATES$')
/api/v2/Data/GetTableData('SGM_CORE_Project $GATEKEEPERS$')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Load Order Examples:

Project Deliverables

This table contains details about deliverables on projects.

Accessed by: `/api/v2/Data/GetTableData('SGM_CORE_Project $DELIVERABLES$')`

Columns	Datatypes	Used for Joins
Project Deliverable Name	String	No
Project Deliverable Owner Name	String	No
Project Deliverable Deadline	Date	No
Project Deliverable Plan Date	Date	No
Project Deliverable Status	String	No
Project Deliverable Status Notes	String	No
Project Deliverable Published	String	No
Project Deliverable Order	Number	No
ProjectDeliverableID_SYS	Number	Yes
Project Deliverable Owner ID	Number	Yes
Project Deliverable Status Changed By	String	No
Project Deliverable Status Change Date	Date	No
ProjectID_SYS	Number	Yes
ProjectPhaseID_SYS	Number	Yes
Project Deliverable Owner Changed By	String	No
Project Deliverable Owner Change Date	Date	No
Project Deliverable is Collaborative	String	No
Project Deliverable Function	String	No
Project Deliverable Functional Area	String	No
Deliverable Last Published Date	Date	No

Columns	Datatypes	Used for Joins
Deliverable Predecessor Name	String	No
Deliverable Predecessor Type	String	No
Deliverable Predecessor Stage Name	String	No

How to join to other tables:

- **ProjectID_SYS** can be used to join to other project-related tables that contain the **ProjectID_SYS** column.
- **ProjectPhaseID_SYS** can be used in conjunction with the **ProjectID_SYS** column to join to other tables that contain the **ProjectPhaseID_SYS** (or any other phase ID column) and **ProjectID_SYS** columns.
- **ProjectDeliverableID_SYS** can be used to join to other project-related tables, such as the **Project Activities** table, that contain the **ProjectDeliverableID_SYS** column.
- To secure the data in this table, secure the project data, then use the **ProjectID_SYS** and **ProjectPhaseID_SYS** columns to join to the **ProjectID_SYS** and **ProjectPhaseID_SYS** columns from the **Project Stages and Gates** table.
- To access more Project Deliverable Owner information, use the **Project Deliverable Owner ID** to join to the **UserID_SYS** column from the **Users** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserProjectAccessData
/api/v2/Data/GetTableData('SGM_CORE_Project Details')
/api/v2/Data/GetTableData('SGM_CORE_Project $STAGES$ and $GATES$')
/api/v2/Data/GetTableData('SGM_CORE_Project $DELIVERABLES$')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Project Activities

This table contains details about activities on projects.

Accessed by: `/api/v2/Data/GetTableData('SGM_CORE_Project $ACTIVITIES$')`

Columns	Datatypes	Used for Joins
Project Activity Name	String	No
Project Activity Owner Name	String	No
Project Activity Deadline	Date	No
Project Activity Plan Date	Date	No
Project Activity Status	String	No
Project Activity Status Notes	String	No
Project Activity Published	String	No
Project Activity Order	Number	No
Project Parent Deliverable Name	String	No
ProjectActivityID_SYS	Number	Yes
Project Activity Owner ID	Number	Yes
Project Activity Status Changed By	String	No
Project Activity Status Change Date	Date	No
ProjectID_SYS	Number	Yes
ProjectPhaseID_SYS	Number	Yes
ProjectDeliverableID_SYS	Number	Yes
Project Activity Owner Changed By	String	No
Project Activity Owner Change Date	Date	No
Project Activity is Collaborative	String	No
Project Activity Function	String	No
Project Activity Functional Area	String	No
Activity Last Published Date	Date	No

Columns	Datatypes	Used for Joins
Activity Predecessor Name	String	No
Activity Predecessor Type	String	No
Activity Predecessor Stage Name	String	No

How to join to other tables:

- **ProjectID_SYS** can be used to join to other project-related tables that contain the **ProjectID_SYS** column.
- **ProjectPhaseID_SYS** can be used in conjunction with the **ProjectID_SYS** column to join to other tables that contain the **ProjectPhaseID_SYS** (or any other phase ID column) and **ProjectID_SYS** columns.
- **ProjectActivityID_SYS** can be used to join to other project-related tables, such as the **Project Deliverables** table, that contain the **ProjectActivityID_SYS** column.
- **ProjectDeliverableID_SYS** can be used to join to the **ProjectDeliverableID_SYS** column from the **Project Deliverables** table.
- To secure the data in this table, secure the project data, then use the **ProjectID_SYS** and **ProjectPhaseID_SYS** columns to join to the **ProjectID_SYS** and **ProjectPhaseID_SYS** columns from the **Project Stages and Gates** table.
- To access more Project Activity Owner information, use the **Project Activity Owner ID** to join to the **UserID_SYS** column from the **Users** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserProjectAccessData
/api/v2/Data/GetTableData('SGM_CORE_Project Details')
/api/v2/Data/GetTableData('SGM_CORE_Project $STAGES$ and $GATES$')
/api/v2/Data/GetTableData('SGM_CORE_Project $DELIVERABLES$')
/api/v2/Data/GetTableData('SGM_CORE_Project $ACTIVITIES$')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Project MS Project Tasks

This table contains details about Microsoft Project tasks for a project.

Accessed by: `/api/v2/Data/GetTableData('SGM_CORE_MS Project Tasks')`

Columns	Datatypes	Used for Joins
MS Project Task Name	String	No
MS Project Task Owner Name	String	No
MS Project Task Description	String	No
MS Project Task Deadline	Date	No
MS Project Task Planned Start Date	Date	No
MS Project Task Planned Finish Date	Date	No
MS Project Task Status	String	No
MS Project Task Status Notes	String	No
MS Project Task Status Changed By	String	No
MS Project Task Changed Date	Date	No
MS Project Task Row Number	Number	No
ProjectMSProjectTaskID_SYS	Number	Yes
MS Project Task Owner ID	Number	Yes
ProjectID_SYS	Number	Yes

How to join to other tables:

- **ProjectID_SYS** can be used to join to other project-related tables that contain the **ProjectID_SYS** column.
- **ProjectMSProjectTaskID_SYS** can be used to join to other project-related tables that contain the **ProjectMSProjectTaskID_SYS** column.
- To secure the data in this table, secure the project data, then use the **ProjectID_SYS** column to join to the **ProjectID_SYS** column from the **Project Details** table.
- To access more MS Project Task Owner information, use the **MS Project Task Owner ID** to join to the **UserID_SYS** column from the **Users** table.

Load Order Examples:
`/api/v2/Data/GetSecurityUsersData`

Load Order Examples:

```
/api/v2/Data/GetUserProjectAccessData
/api/v2/Data/GetTableData('SGM_CORE_Project Details')
/api/v2/Data/GetTableData('SGM_CORE_MS Project Tasks')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Trended Metric History

This table contains information on metric history. Trended metric history tracks when a metric value changed, who made the change, where the change was made, and the value of the metric before and after the change.

Accessed by: `/api/v2/Data/GetTableData('SGM_CORE_Trended Metric History')`

Columns	Datatypes	Used for Joins
ProjectID_SYS	Number	Yes
EventID_SYS	Number	Yes
ChangeSourceID_SYS	Number	No
Stage ID	Number	Yes
Change Source Name	String	No
Change Event Date	Date	No

How to join to other tables:

- **ProjectID_SYS** can be used to join to other project-related tables that contain the **ProjectID_SYS** column.
- **Stage ID** can be used in conjunction with the **ProjectID_SYS** column to join to other tables that contain the **ProjectPhaseID_SYS** (or any other phase ID column) and **ProjectID_SYS** columns.
- To access more information about the metrics and what their values were when the trending event happened, use the **EventID_SYS** column to join to the **EventID_SYS** column from the **Trended Project Metrics** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserProjectAccessData
/api/v2/Data/GetTableData('SGM_CORE_Project Details')
/api/v2/Data/GetTableData('SGM_CORE_Trended Metric History')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Trended Project Metrics

This table contains trended project metrics data for a specific category.

It is recommended to name these "Trended Project Metrics - <*category*>" so that you can keep each category separate.

Accessed by: `/api/v2/Data/GetTrendedProjectMetricsData('category')`

Columns	Datatypes	Used for Joins
ProjectID_SYS	Number	Yes
EventID_SYS	Number	Yes

Important! The remaining columns are the trended metrics in this category, which will vary depending on your company's configuration.

How to join to other tables:

- **ProjectID_SYS** can be used to join to other project-related tables that contain the **ProjectID_SYS** column.
- To secure the data in this table, secure the project data, then use the **ProjectID_SYS** column to join to the **ProjectID_SYS** column from the **Project Details** table.
- To access more information about the event that caused the metrics to be trended, use the **EventID_SYS** column to join to the **EventID_SYS** column from the **Trended Metric History** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserProjectAccessData
/api/v2/Data/GetTableData('SGM_CORE_Project Details')
/api/v2/Data/GetTableData('SGM_CORE_Trended Metric History')
/api/v2/Data/GetTrendedProjectMetricsData ('category')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Project Matrix Metrics

This table contains project matrix metric data for a specific matrix.

> It is recommended to name these "Project Matrix Metric - *<matrix name>*" so that you can keep each matrix separate.

Accessed by: `/api/v2/Data/GetProjectMatrixMetricsData ('matrixSystemName')`

Columns	Datatypes	Used for Joins
ProjectID_SYS	Number	Yes

Important! The remaining columns are the metrics in the matrix, which will vary depending on your company's configuration.

How to join to other tables:

- **ProjectID_SYS** can be used to join to other project-related tables that contain the **ProjectID_SYS** column.
- To secure the data in this table, secure the project data, then use the **ProjectID_SYS** column to join to the **ProjectID_SYS** column from the **Project Details** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserProjectAccessData
/api/v2/Data/GetTableData('SGM_CORE_Project Details')
/api/v2/Data/GetProjectMatrixMetricsData ('matrixSystemName')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Project Metrics

This table contains data on metrics in a project for a specific category.

It is recommended to name these "Project Metrics - <*category*>" so that you can keep each category separate.

Accessed by: `/api/v2/Data/GetProjectMetricsData('category')`

Columns	Datatypes	Used for Joins
ProjectID_SYS	Number	Yes

Important! The remaining columns are the metrics in this category, which will vary depending on your company's configuration.

How to join to other tables:

- **ProjectID_SYS** can be used to join to other project-related tables that contain the **ProjectID_SYS** column.
- To secure the data in this table, secure the project data, then use the **ProjectID_SYS** column to join to the **ProjectID_SYS** column from the **Project Details** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserProjectAccessData
/api/v2/Data/GetTableData('SGM_CORE_Project Details')
/api/v2/Data/GetProjectMetricsData('category')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Related Projects Data Table

This table contains related project data, and can be used to link the project data of one project to the project data of another project.

Accessed by: `/api/v2/Data/GetRelatedProjectsData`

Columns	Datatypes	Used for Joins
ProjectID_SYS	Number	Yes
ToProjectID_SYS	Number	Yes
Level	Number	No
Link Type	String	No
Dependency Type	String	No
In Conflict	Boolean	No
Conflict Created On	Date	No
Conflict Created By	String	No

How to join to other tables:

- **ProjectID_SYS** can be used to join to other project-related tables that contain the **ProjectID_SYS** column.
- **ToProjectID_SYS** can be used to join to other project-related tables that contain the **ToProjectID_SYS** column.
- To secure the data in this table, secure the project data, then use the **ProjectID_SYS** column to join to the **ProjectID_SYS** column from the **Project Details** table.
- To prevent circular references to project data, create a copy of the project data already loaded into your BI application and change the **ProjectID_SYS** in those tables to **ToProjectID_SYS**.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserProjectAccessData
/api/v2/Data/GetTableData('SGM_CORE_Project Details')
/api/v2/Data/GetRelatedProjectsData()
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Reference Tables Data Table

This table contains reference table data.

It is recommended to name these "Reference Table - *<reference table name>*" so that you can keep each reference table separate.

Accessed by: `/api/v2/Data/GetReferenceTableData('systemName')`

Columns	Datatypes	Used for Joins
ReferenceTableID_SYS	Number	No
RowID_SYS	Number	No

Important! The remaining columns are the columns in the reference table, which will vary depending on your company's configuration.

How to join to other tables:

- Joining reference tables to other Accolade data will vary depending on your company's configuration.

Load Order Examples:
`/api/v2/Data/GetReferenceTableData('systemName')`

Important! This list of examples includes some calls to get you started, but does not include all available options.

Resources Data Tables

Resource Pools

This table contains information on resource pools used in Resource Planning.

Accessed by: `/api/v2/Data/GetTableData('SGM_CORE_Resource Pools')`

Columns	Datatypes	Used for Joins
Resource Pool Name	String	No
Resource Pool Owner Name	String	No
Resource Pool is Active	Boolean	No
ResourcePoolID_SYS	Number	Yes
Resource Pool Owner ID	Number	Yes
Resource Pool Unit of Measure	String	No

How to join to other tables:

- **ResourcePoolID_SYS** can be used to join to other resource-related tables that contain the **ResourcePoolID_SYS** column.
- To access more Resource Pool Owner information, use the **Resource Pool Owner User ID** column to join to the **UserID_SYS** column from the **Users** table.

Load Order Examples:
`/api/v2/Data/GetTableData('SGM_CORE_Resource Pools')`

Important! This list of examples includes some calls to get you started, but does not include all available options.

Resources

This table contains information on resources used in Resource Planning.

Accessed by: `/api/v2/Data/GetTableData('SGM_CORE_Resources')`

Columns	Datatypes	Used for Joins
Resource Name	String	No
ResourceID_SYS	Number	Yes
Resource Pool Member User ID	Number	Yes
Resource Account Status	String	No
ResourcePoolID_SYS	Number	Yes
Resource System Name	String	No

How to join to other tables:

- **ResourceID_SYS** can be used to join to other resource-related tables that contain the **ResourceID_SYS** column.
- **ResourcePoolID_SYS** can be used to join to other resource-related tables that contain the **ResourcePoolID_SYS** column.
- To access more Resource Pool Member information, use the **Resource Pool Member User ID** column to join to the **UserID_SYS** column from the **Users** table.

Load Order Examples:

```
/api/v2/Data/GetTableData('SGM_CORE_Resource Pools')
/api/v2/Data/GetTableData('SGM_CORE_Resources')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Resource Capacities

This table contains information on resource capacities used in Resource Planning.

Accessed by: `/api/v2/Data/GetTableData('SGM_CORE_Resource Capacities')`

Columns	Datatypes	Used for Joins
Resource Capacity Start Date	Date	No
Resource Capacity End Date	Date	No
Resource Capacity	Number	No
Resource Capacity Unavailable	Number	No
ResourcePeriodID_SYS	Number	Yes
ResourceID_SYS	Number	Yes

How to join to other tables:

- **ResourceID_SYS** can be used to join to other resource-related tables that contain the **ResourceID_SYS** column.
- **ResourcePeriodID_SYS** can be used to join to other resource planning-related tables that contain the **ResourcePeriodID_SYS** column.
- To access more information about the time period, use the **ResourcePeriodID_SYS** column to join to the **TimePeriodID_SYS** column from the **Time Period** table.
- To access more information about the resource pool, use the **ResourceID_SYS** column to join to the **ResourceID_SYS** column from the **Resource** table, then use the **ResourcePoolID_SYS** column from the **Resources** table to join to the **ResourcePoolID_SYS** column from the **Resource Pools** table.

Load Order Examples:

```
/api/v2/Data/GetTimePeriodsData
/api/v2/Data/GetTableData('SGM_CORE_Resource Pools')
/api/v2/Data/GetTableData('SGM_CORE_Resources')
/api/v2/Data/GetTableData('SGM_CORE_Resource
Capacities')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Resource Demands

This table contains information on resource demands used in Resource Planning.

Accessed by: `/api/v2/Data/GetTableData('SGM_CORE_Resource Demands')`

Columns	Datatypes	Used for Joins
Resource Demand Project Name	String	No
Resource Demand Project Stage Name	String	No
Resource Demand Start Date	Date	No
Resource Demand End Date	Date	No
Resource Demand Type	String	No
Resource Demand Value	Number	No
Resource Demand is Active	String	No
ProjectID_SYS	Number	Yes
Resource Demand Project Phase ID	Number	Yes
Resource Demand Type ID	Number	No
ResourceID_SYS	Number	Yes
ResourcePeriodID_SYS	Number	Yes
Resource Demand Curve	String	No
Resource Demand Multiplier	Number	No
Resource Demand Effective Time Period	String	No
Resource Demand System Name	String	No

How to join to other tables:

- **ProjectID_SYS** can be used to join to other project-related tables that contain the **ProjectID_SYS** column.
- **ResourceID_SYS** can be used to join to other resource-related tables that contain the **ResourceID_SYS** column.
- **ResourcePeriodID_SYS** can be used to join to other resource planning-related tables that contain the **ResourcePeriodID_SYS** column.

- **Resource Demand Project Phase ID** can be used in conjunction with the **ProjectID_SYS** column to join to other tables that contain the **ProjectPhaseID_SYS** (or any other phase ID column).
- To secure the data in this table, secure the project data, then use the **ProjectID_SYS** column to join to the **ProjectID_SYS** column from the **Project Details** table.
- To access more information about the time period, use the **ResourcePeriodID_SYS** column to join to the **TimePeriodID_SYS** column from the **Time Period** table.
- To access more information about the resource pool, use the **ResourceID_SYS** column to join to the **ResourceID_SYS** column from the **Resource** table, then use the **ResourcePoolID_SYS** column from the **Resources** table to join to the **ResourcePoolID_SYS** column from the **Resource Pools** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserProjectAccessData
/api/v2/Data/GetTableData('SGM_CORE_Project Details')
/api/v2/Data/GetTableData('SGM_CORE_Project $STAGES$ and $GATES$')
/api/v2/Data/GetTimePeriodsData
/api/v2/Data/GetTableData('SGM_CORE_Resource Pools')
/api/v2/Data/GetTableData('SGM_CORE_Resources')
/api/v2/Data/GetTableData('SGM_CORE_Resource Capacities')
/api/v2/Data/GetTableData('SGM_CORE_Resource Demands')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Time Periods

This table contains the time period information used in Resource Planning.

Accessed by: `/api/v2/Data/GetTimePeriodsData`

Columns	Datatypes	Used for Joins
TimePeriodID_SYS	Number	Yes
Caption	String	No
Period Start Date	Date	No
Period End Date	Date	No

How to join to other tables:

- **TimePeriodID_SYS** can be used to join to the **ResourcePeriodID_SYS** column in resource planning-related tables like **Resource Demands** and **Resource Capacities**.

Load Order Examples:
`/api/v2/Data/GetTimePeriodsData`

Important! This list of examples includes some calls to get you started, but does not include all available options.

Functions Data Table

This table contains information on functions.

Accessed by: `/api/v2/Data/GetTableData('SGM_CORE_Functions')`

Columns	Datatypes	Used for Joins
FunctionID_SYS	Number	Yes
Function Name	String	No
Functional Area Name	String	No

How to join to other tables:

- To access more user information, use the **FunctionID_SYS** column to join to the **FunctionID_SYS** column from the **Users** table.

Load Order Examples:
`/api/v2/Data/GetTableData('SGM_CORE_Functions')`

Important! This list of examples includes some calls to get you started, but does not include all available options.

Time Tracking Data Tables

Timesheets

This table contains details on timesheets.

Accessed by: `/api/v2/Data/GetTableData('SGM_CORE_Timesheets')`

Columns	Datatypes	Used for Joins
TimesheetID_SYS	Number	Yes
Timesheet Start Date	Date	No
Timesheet End Date	Date	No
TimesheetOwnerID_SYS	Number	Yes
Timesheet Owner Name	String	No
TimesheetStatusID_SYS	Number	No
Timesheet Status	String	No
Timesheet Status Comment	String	No
TimesheetLastUpdatedByID_SYS	Number	Yes
Timesheet Last Updated By Name	String	No
Timesheet Last Updated Date	Date	No
TimesheetApproverID_SYS	Number	Yes
Timesheet Approver Name	String	No
Timesheet Approver Date	Date	No
TimesheetOriginalApproverID_SYS	Number	Yes
Timesheet Original Approver Name	String	No

How to join to other tables:

- To secure the data in this table, secure the project data, then use the **TimesheetID_SYS** and **TimesheetRowID_SYS** columns to join to the **TimesheetID_SYS** and **TimesheetRowID_SYS** columns from the **Timesheet Rows** table.
- To access row information for a timesheet, use the **TimesheetID_SYS** column to join to the **TimesheetID_SYS** column from the **Timesheet Rows** table.
- To access more Timesheet Owner information, use the **TimesheetOwnerID_SYS** column to join to the **UserID_SYS** column from the **Users** table.

- To access more Timesheet Last Updated By information, use the **TimesheetLastUpdatedByID_SYS** column to join to the **UserID_SYS** column from the **Users** table.
- To access more Timesheet Approver information, use the **TimesheetApproverID_SYS** column to join to the **UserID_SYS** column from the **Users** table.
- To access more Timesheet Original Approver information, use the **TimesheetOriginalApproverID_SYS** column to join to the **UserID_SYS** column from the **Users** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserProjectAccessData
/api/v2/Data/GetTableData('SGM_CORE_Project Details')
/api/v2/Data/GetTableData('SGM_CORE_Timesheets')
/api/v2/Data/GetTableData('SGM_CORE_Timesheet Rows')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Timesheet Rows

This table contains rows of timesheet data.

Accessed by: `/api/v2/Data/GetTableData('SGM_CORE_Timesheet Rows')`

Columns	Datatypes	Used for Joins
TimesheetID_SYS	Number	Yes
TimesheetRowID_SYS	Number	Yes
ProjectID_SYS	Number	Yes
Timesheet Row Project Name	String	No
TimesheetRowApproverID_SYS	Number	Yes
Timesheet Row Approver Name	String	No
Timesheet Row Approval Date	Date	No
TimesheetOriginalApproverID_SYS	Number	Yes
Timesheet Original Approver Name	String	No

How to join to other tables:

- **ProjectID_SYS** can be used to join to other project-related tables that contain the **ProjectID_SYS** column.
- To secure the data in this table, secure the project data, then use the **ProjectID_SYS** column to join to the **ProjectID_SYS** column from the **Project Details** table.
- To access more information about the timesheet, use the **TimesheetID_SYS** column to join to the **TimesheetID_SYS** column from the **Timesheets** table.
- To access more Timesheet Row Approver information, use the **TimesheetRowApproverID_SYS** column to join to the **UserID_SYS** column from the **Users** table.
- To access more Timesheet Original Approver information, use the **TimesheetOriginalApproverID_SYS** column to join to the **UserID_SYS** column from the **Users** table.
- To get the corresponding timesheet entries for a timesheet row, use the **TimesheetRowID_SYS** column in conjunction with **TimesheetID_SYS** column to join to the **TimesheetRowID_SYS** and **TimesheetID_SYS** columns from the **Timesheet Entries** table.

Load Order Examples:
`/api/v2/Data/GetSecurityUsersData`

Load Order Examples:

```
/api/v2/Data/GetUserProjectAccessData
/api/v2/Data/GetTableData('SGM_CORE_Project Details')
/api/v2/Data/GetTableData('SGM_CORE_Timesheets')
/api/v2/Data/GetTableData('SGM_CORE_Timesheet Rows')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Timesheet Entries

This table contains the entries for a row of timesheet data.

Accessed by: `/api/v2/Data/GetTableData('SGM_CORE_Timesheet Entries')`

Columns	Datatypes	Used for Joins
TimesheetID_SYS	Number	Yes
TimesheetRowID_SYS	Number	Yes
Timesheet Entry Date	Date	No
Timesheet Entry Value	Number	No

How to join to other tables:

- To secure the data in this table, secure the project data, then use the **TimesheetID_SYS** and **TimesheetRowID_SYS** columns to join to the **TimesheetID_SYS** and **TimesheetRowID_SYS** columns from the **Timesheet Rows** table.
- To access more information about the timesheet, use the **TimesheetID_SYS** column to join to the **TimesheetID_SYS** column from the **Timesheets** table.
- To get the corresponding timesheet row for timesheet entries, use the **TimesheetRowID_SYS** column in conjunction with **TimesheetID_SYS** column to join to the **TimesheetRowID_SYS** and **TimesheetID_SYS** columns from the **Timesheet Rows** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserProjectAccessData
/api/v2/Data/GetTableData('SGM_CORE_Project Details')
/api/v2/Data/GetTableData('SGM_CORE_Timesheets')
/api/v2/Data/GetTableData('SGM_CORE_Timesheet Rows')
/api/v2/Data/GetTableData('SGM_CORE_Timesheet Entries')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

User Scenario Access and Favorite Information Data Table

This table contains information on a user's scenario access and scenario favorites for use with securing scenario data on a row level basis.

Accessed by: `/api/v2/Data/GetUserScenarioAccessData`

Columns	Datatypes	Used for Joins
SecurityUserID_SYS	Number	Yes
SecurityScenarioID_SYS	Number	Yes
ScenarioIsFavorite	Number	No

How to join to other tables:

- **SecurityUserID_SYS** can be used to join to other tables that contain the **SecurityUserID_SYS** (or any other user ID column).
- **SecurityScenarioID_SYS** can be used to join to other tables that contain the **SecurityScenarioID_SYS** (or any other project ID column).
- To secure scenario project data, secure the scenario data, then use the **SecurityUserID_SYS** column to join to the **SecurityUserID_SYS** column from the **Security Users** table to get the **NTName** for the user.

Load Order Examples:

`/api/v2/Data/GetSecurityUsersData`
`/api/v2/Data/GetUserScenarioAccessData`
 	
Important! This list of examples includes some calls to get you started, but does not include all available options.

Scenarios Data Table

This table contains details on scenarios.

Accessed by: `/api/v2/Data/GetTableData('APE_CORE_Scenarios')`

Columns	Datatypes	Used for Joins
ScenarioID_SYS	Number	Yes
Scenario Name	String	No
Scenario Created By ID	Number	Yes
Scenario Created By Name	String	No
Scenario Creation Date	Date	No
Scenario Updated By ID	Number	Yes
Scenario Updated By Name	String	No
Scenario Updated Date	Date	No
Scenario is Public	String	No
Scenario Last Commit Date	Date	No
Scenario Last Commit By ID	Number	Yes
Scenario Last Commit By Name	String	No

How to join to other tables:

- **ScenarioID_SYS** can be used to join to other scenario-related tables that contain the **ScenarioID_SYS** column.
- To secure scenario data, use the **ScenarioID_SYS** column to join to the **SecurityScenarioID_SYS** column from the **User Scenario Access** table, then use the **SecurityUserID_SYS** column to join that table to the **SecurityUserID_SYS** column from the **SecurityUsers** table.
- To access more Scenario Created By information, use the **Scenario Created By ID** column to join to the **UserID_SYS** column from the **Users** table.
- To access more Scenario Updated By information, use the **Scenario Updated By ID** column to join to the **UserID_SYS** column from the **Users** table.
- To access more Scenario Last Commit By information, use the **Scenario Last Commit By ID** column to join to the **UserID_SYS** column from the **Users** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserScenarioAccessData
/api/v2/Data/GetTableData('APE_CORE_Scenarios')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Scenario Editors Data Table

This table contains data about scenario editors.

Accessed by: `/api/v2/Data/GetTableData('APE_CORE_Scenario Editors')`

Columns	Datatypes	Used for Joins
ScenarioID_SYS	Number	Yes
ScenarioUserID_SYS	Number	Yes
Scenario Editor Name	String	No
Scenario Checked Out	String	No

How to join to other tables:

- **ScenarioID_SYS** can be used to join to other scenario-related tables that contain the **ScenarioID_SYS** column.
- To secure the data in this table, secure the scenario data, then use the **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns to join to the **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns from the **Scenario Project Details** table.
- To access more Scenario Editor information, use the **ScenarioUserID_SYS** column to join to the **UserID_SYS** column from the **Users** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserScenarioAccessData
/api/v2/Data/GetTableData('APE_CORE_Scenarios')
/api/v2/Data/GetTableData('APE_CORE_Scenario
Editors')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Scenario Projects Data Tables

Scenario User Project Access

This table contains information on a user's scenario project access for use with securing scenario project data on a row level basis.

Accessed by: `/api/v2/Data/GetScenarioUserProjectAccessData`

Columns	Datatypes	Used for Joins
SecurityUserID_SYS	Number	Yes
SecurityScenarioID_SYS	Number	Yes
SecurityProjectID_SYS	Number	Yes

How to join to other tables:

- **SecurityUserID_SYS** can be used to join to other tables that contain the **SecurityUserID_SYS** (or any other user ID column).
- **SecurityScenarioID_SYS** can be used to join to other tables that contain the **SecurityScenarioID_SYS** (or any other project ID column).
- **SecurityProjectID_SYS** can be used to join to other tables that contain the **SecurityProjectID_SYS** (or any other project ID column).
- To secure scenario project data, secure the scenario data, then use the **SecurityUserID_SYS** column to join to the **SecurityUserID_SYS** column from the **Security Users** table to get the **NTName** for the user.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserScenarioAccessData
/api/v2/Data/GetSnapshotUserProjectAccessData
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Scenario Project Security List Values by List

This table contains security list value information by level for a specific security list for all scenario projects in a system.

Accessed by: `/api/v2/Data/GetScenarioProjectSecurityListData ('SecurityListName')`

Columns	Datatypes	Used for Joins
ScenarioID_SYS	Number	Yes
ScenarioProjectID_SYS	Number	Yes

Important! The remaining columns are the security list level names returned as strings, which will vary depending on your company's configuration.

How to join to other tables:

- **ScenarioID_SYS** can be used to join to other scenario-related tables that contain the **ScenarioID_SYS** column.
- **ScenarioProjectID_SYS** can be used in conjunction with the **ScenarioID_SYS** column to join to other scenario-related tables that contain the **ScenarioProjectID_SYS** and **ScenarioID_SYS** columns.
- To secure the data in this table, secure the scenario data, then use the **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns to join to the **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns from the **Scenario Project Details** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserScenarioAccessData
/api/v2/Data/GetScenarioUserProjectAccessData
/api/v2/Data/GetTableData('APE_CORE_Scenarios')
/api/v2/Data/GetTableData('APE_CORE_Scenario Project Details')
/api/v2/Data/GetScenarioProjectSecurityListData ('SecurityListName')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Scenario Project Details

This table contains scenario project details.

Accessed by: `/api/v2/Data/GetTableData('APE_CORE_Scenario Project Details')`

Columns	Datatypes	Used for Joins
ScenarioID_SYS	Number	Yes
ScenarioProjectID_SYS	Number	Yes
Project ID	String	No
Project Name	String	No
Access Group ID	Number	No
Access Group	String	No
Parent Access Group ID	Number	No
Parent Access Group	String	No
Project Description	String	No
Project Process Type	String	No
Project Team Leader ID	Number	Yes
Project Team Leader Name	String	No
Project is In Trouble	String	No
Project Creation Date	Date	No
Project Closed	String	No
Current Phase ID	Number	Yes
Project Last Gate Decision	String	No
Project Most Recent Status Report Text	String	No
Project Last Gate Decision Code	Number	No
Project Model ID	Number	No
Project Model Name	String	No
Project Current Stage Name	String	No
Project Class ID	Number	No
Project Class	String	No
Project Created By ID	Number	Yes
Project Created By	String	No
Project Rank	Number	No
Project is Active	String	No

Columns	Datatypes	Used for Joins
Project Scenario Rank	Number	No
Project Start Date	Date	No
Project End Date	Date	No

How to join to other tables:

- **ScenarioID_SYS** can be used to join to other scenario-related tables that contain the **ScenarioID_SYS** column.
- **ScenarioProjectID_SYS** can be used in conjunction with the **ScenarioID_SYS** column to join to other scenario-related tables that contain the **ScenarioProjectID_SYS** and **ScenarioID_SYS** columns.
- To secure snapshot project data, use the **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns to join to the **SecurityScenarioID_SYS** and **SecurityProjectID_SYS** columns from the **Scenario User Project Access** table, then use the **SecurityUserID_SYS** column to join to the **SecurityUserID_SYS** column from the **Security Users** table.
- To access more Project Team Leader information, use the **Project Team Leader ID** column to join to the **UserID_SYS** column from the **Users** table.
- To access more Project Created By information, use the **Project Created By ID** column to join to the **UserID_SYS** column from the **Users** table.
- To get information about current stage of a project, use the **Current Phase ID** column in conjunction with the **ScenarioProjectID_SYS** column to join to the **ScenarioProjectPhaseID_SYS** and **ScenarioProjectID_SYS** columns from the **Scenario Project Stages and Gates** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserScenarioAccessData
/api/v2/Data/GetScenarioUserProjectAccessData
/api/v2/Data/GetTableData('APE_CORE_Scenarios')
/api/v2/Data/GetTableData('APE_CORE_Scenario Project Details')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Scenario Project Stages and Gates

This table contains information on scenario project stages and gates.

Accessed by: `/api/v2/Data/GetTableData('APE_CORE_Scenario Project $STAGES$ and $GATES$')`

Columns	Datatypes	Used for Joins
ScenarioID_SYS	Number	Yes
ScenarioProjectID_SYS	Number	Yes
ScenarioProjectPhaseID_SYS	Number	Yes
Project Stage Exists	String	No
Project Gate Name	String	No
Project Gate Exists	String	No
Project Gate Date	Date	No

How to join to other tables:

- **ScenarioID_SYS** can be used to join to other scenario-related tables that contain the **ScenarioID_SYS** column.
- **ScenarioProjectID_SYS** can be used in conjunction with the **ScenarioID_SYS** column to join to other scenario-related tables that contain the **ScenarioProjectID_SYS** and **ScenarioID_SYS** columns.
- **ScenarioProjectPhaseID_SYS** can be used in conjunction with the **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns to join to other scenario-related tables that contain the **ScenarioProjectPhaseID_SYS** (or any other phase ID column), **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns.
- To secure the data in this table, secure the scenario data, then use the **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns to join to the **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns from the **Scenario Project Details** table.

Load Order Examples:
`/api/v2/Data/GetSecurityUsersData`
`/api/v2/Data/GetUserScenarioAccessData`
`/api/v2/Data/GetScenarioUserProjectAccessData`
`/api/v2/Data/GetTableData('APE_CORE_Scenarios')`
`/api/v2/Data/GetTableData('APE_CORE_Scenario Project Details')`

Load Order Examples:

```
/api/v2/Data/GetTableData('APE_CORE_Scenario Project
$STAGES$ and $GATES$')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Scenario Project Matrix Metrics

This table contains scenario project matrix metric data for a specific matrix.

It is recommended to name these "Scenario Project Matrix Metric - *<matrix name>*" so that you can keep each matrix separate.

Accessed by: `/api/v2/Data/GetScenarioProjectMatrixMetricsData ('matrixSystemName')`

Columns	Datatypes	Used for Joins
ScenarioID_SYS	Number	Yes
ScenarioProjectID_SYS	Number	Yes
RowID	Number	No

Important! The remaining columns are the metrics in the matrix, which will vary depending on your company's configuration.

How to join to other tables:

- **ScenarioID_SYS** can be used to join to other scenario-related tables that contain the **ScenarioID_SYS** column.
- **ScenarioProjectID_SYS** can be used in conjunction with the **ScenarioID_SYS** column to join to other scenario-related tables that contain the **ScenarioProjectID_SYS** and **ScenarioID_SYS** columns.
- To secure the data in this table, secure the scenario data, then use the **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns to join to the **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns from the **Scenario Project Details** table.

Load Order Examples:

`/api/v2/Data/GetSecurityUsersData`
`/api/v2/Data/GetUserScenarioAccessData`
`/api/v2/Data/GetScenarioUserProjectAccessData`
`/api/v2/Data/GetTableData('APE_CORE_Scenarios')`
`/api/v2/Data/GetTableData('APE_CORE_Scenario Project Details')`
`/api/v2/Data/GetScenarioProjectMatrixMetricsData ('matrixSystemName')`
Important! This list of examples includes some calls to get

Load Order Examples:
you started, but does not include all available options.

Scenario Project Metrics

This table contains scenario data on metrics in a project for a specific category.

It is recommended to name these "Scenario Project Metrics - *<category>*" so that you can keep each category separate.

Accessed by: `/api/v2/Data/GetScenarioProjectMetricsData('category')`

Columns	Datatypes	Used for Joins
ScenarioID_SYS	Number	Yes
ScenarioProjectID_SYS	Number	Yes

Important! The remaining columns are the metrics in this category, which will vary depending on your company's configuration.

How to join to other tables:

- **ScenarioID_SYS** can be used to join to other scenario-related tables that contain the **ScenarioID_SYS** column.
- **ScenarioProjectID_SYS** can be used in conjunction with the **ScenarioID_SYS** column to join to other scenario-related tables that contain the **ScenarioProjectID_SYS** and **ScenarioID_SYS** columns.
- To secure the data in this table, secure the scenario data, then use the **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns to join to the **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns from the **Scenario Project Details** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserScenarioAccessData
/api/v2/Data/GetScenarioUserProjectAccessData
/api/v2/Data/GetTableData('APE_CORE_Scenarios')
/api/v2/Data/GetTableData('APE_CORE_Scenario Project Details')
/api/v2/Data/GetScenarioProjectMetricsData ('category')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Scenario Resources Data Tables

Scenario Resource Pools

This table contains information on scenario resource pools used in Resource Planning.

Accessed by: `/api/v2/Data/GetTableData('APE_CORE_Scenario Resource Pools')`

Columns	Datatypes	Used for Joins
ScenarioID_SYS	Number	Yes
Resource Pool Name	String	No
ScenarioResourcePoolID_SYS	Number	Yes
Resource Pool Unit of Measure	String	No

How to join to other tables:

- **ScenarioID_SYS** can be used to join to other scenario-related tables that contain the **ScenarioID_SYS** column.
- **ScenarioResourcePoolID_SYS** can be used in conjunction with the **ScenarioID_SYS** column to join to other scenario-related tables that contain the **ScenarioResourcePoolID_SYS** and **ScenarioID_SYS** columns.
- To secure the data in this table, secure the scenario data, then use the **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns to join to the **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns from the **Scenario Project Details** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserScenarioAccessData
/api/v2/Data/GetTableData('APE_CORE_Scenarios')
/api/v2/Data/GetTableData('APE_CORE_Scenario Resource Pools')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Scenario Resources

This table contains information on scenario resources used in Resource Planning.

Accessed by: `/api/v2/Data/GetTableData('APE_CORE_Scenario Resources')`

Columns	Datatypes	Used for Joins
ScenarioID_SYS	Number	Yes
Resource Name	String	No
ScenarioResourceID_SYS	Number	Yes
ScenarioResourcePoolID_SYS	Number	Yes

How to join to other tables:

- **ScenarioID_SYS** can be used to join to other scenario-related tables that contain the **ScenarioID_SYS** column.
- **ScenarioResourceID_SYS** can be used in conjunction with the **ScenarioID_SYS** column to join to other resource planning-related tables that contain the **ScenarioResourceID_SYS** and **ScenarioID_SYS** columns.
- **ScenarioResourcePoolID_SYS** can be used in conjunction with the **ScenarioID_SYS** column to join to other scenario-related tables that contain the **ScenarioResourcePoolID_SYS** and **ScenarioID_SYS** columns.
- To secure the data in this table, secure the scenario data, then use the **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns to join to the **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns from the **Scenario Project Details** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserScenarioAccessData
/api/v2/Data/GetTableData('APE_CORE_Scenarios')
/api/v2/Data/GetTableData('APE_CORE_Scenario
Resources')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Scenario Resource Capacities

This table contains information on scenario resource capacities used in Resource Planning.

Accessed by: `/api/v2/Data/GetTableData('APE_CORE_Scenario Resource Capacities')`

Columns	Datatypes	Used for Joins
ScenarioID_SYS	Number	Yes
Resource Capacity Start Date	Date	No
Resource Capacity End Date	Date	No
Resource Capacity	Number	No
Resource Capacity Unavailable	Number	No
ScenarioResourcePeriodID_SYS	Number	Yes
ScenarioResourceID_SYS	Number	Yes

How to join to other tables:

- **ScenarioID_SYS** can be used to join to other scenario-related tables that contain the **ScenarioID_SYS** column.
- **ScenarioResourceID_SYS** can be used in conjunction with the **ScenarioID_SYS** column to join to other resource planning-related tables that contain the **ScenarioResourceID_SYS** and **ScenarioID_SYS** columns.
- **ScenarioResourcePeriodID_SYS** can be used to in conjunction with the **ScenarioID_SYS** column to join to other resource planning-related tables that contain the **ScenarioResourcePeriodID_SYS** and **ScenarioID_SYS** columns.
- To secure the data in this table, secure the scenario data, then use the **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns to join to the **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns from the **Scenario Project Details** table.
- To access more information about the time period, use the **ScenarioResourcePeriodID_SYS** column to join to the **TimePeriodID_SYS** column from the **Time Period** table.

Load Order Examples:
`/api/v2/Data/GetSecurityUsersData`
`/api/v2/Data/GetUserScenarioAccessData`
`/api/v2/Data/GetTimePeriodsData`

Load Order Examples:

```
/api/v2/Data/GetTableData('APE_CORE_Scenarios')
/api/v2/Data/GetTableData('APE_CORE_Scenario Resource Pools')
/api/v2/Data/GetTableData('APE_CORE_Scenario Resources')
/api/v2/Data/GetTableData('APE_CORE_Scenario Resource Capacities')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Scenario Resource Demands

This table contains information on scenario resource demands used in Resource Planning.

Accessed by: `/api/v2/Data/GetTableData('APE_CORE_Scenario Resource Demands')`

Columns	Datatypes	Used for Joins
ScenarioID_SYS	Number	Yes
Resource Demand Project Name	String	No
Resource Demand Start Date	Date	No
Resource Demand End Date	Date	No
Resource Demand Value	Number	No
Resource Demand is Active	String	No
ScenarioProjectID_SYS	Number	Yes
ScenarioResourceID_SYS	Number	Yes
ScenarioResourcePeriodID_SYS	Number	Yes
Resource Demand Curve	String	No
Resource Demand Multiplier	Number	No
Resource Demand Effective Time Period	String	No

How to join to other tables:

- **ScenarioID_SYS** can be used to join to other scenario-related tables that contain the **ScenarioID_SYS** column.
- **ScenarioProjectID_SYS** can be used in conjunction with the **ScenarioID_SYS** column to join to other scenario-related tables that contain the **ScenarioProjectID_SYS** and **ScenarioID_SYS** columns.
- **ScenarioResourceID_SYS** can be used in conjunction with the **ScenarioID_SYS** column to join to other resource planning-related tables that contain the **ScenarioResourceID_SYS** and **ScenarioID_SYS** columns.
- **ScenarioResourcePeriodID_SYS** can be used to in conjunction with the **ScenarioID_SYS** column to join to other resource planning-related tables that contain the **ScenarioResourcePeriodID_SYS** and **ScenarioID_SYS** columns.
- To secure the data in this table, secure the scenario data, then use the **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns to join to the **ScenarioID_SYS** and **ScenarioProjectID_SYS** columns from the **Scenario Project Details** table.

- To access more information about the time period, use the **ScenarioResourcePeriodID_SYS** column to join to the **TimePeriodID_SYS** column from the **Time Period** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetUserScenarioAccessData
/api/v2/Data/GetScenarioUserProjectAccessData
/api/v2/Data/GetTimePeriodsData
/api/v2/Data/GetTableData('APE_CORE_Scenarios')
/api/v2/Data/GetTableData('APE_CORE_Scenario Project Details')
/api/v2/Data/GetTableData('APE_CORE_Scenario Resource Pools')
/api/v2/Data/GetTableData('APE_CORE_Scenario Resources')
/api/v2/Data/GetTableData('APE_CORE_Scenario Resource Capacities')
/api/v2/Data/GetTableData('APE_CORE_Scenario Resource Demands')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Snapshots Data Table

This table contains details on snapshots.

Accessed by: `/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Details')`

Columns	Datatypes	Used for Joins
SnapshotID_SYS	Number	Yes
Snapshot Name	String	No
Snapshot Date	Date	No
Snapshot Description	String	No
Snapshot Type	String	No
Snapshot Position - Type	Number	No
Snapshot Position - Global	Number	No

How to join to other tables:

- **SnapshotID_SYS** can be used to join to other snapshot-related tables that contain the **SnapshotID_SYS** column.

Load Order Examples:
`/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Details')`

Important! This list of examples includes some calls to get you started, but does not include all available options.

Snapshot Projects Data Tables

Snapshot User Project Access

This table contains information on a user's snapshot project access for use with securing snapshot project data on a row level basis.

Accessed by: `GetSnapshotUserProjectAccessData`

Columns	Datatypes	Used for Joins
SecuritySnapshotID_SYS	Number	Yes
SecurityUserID_SYS	Number	Yes
SecurityProjectID_SYS	Number	Yes

How to join to other tables:

- **SecuritySnapshotID_SYS** can be used to join to other tables that contain the **SecuritySnapshotID_SYS** (or any other snapshot ID column).
- **SecurityUserID_SYS** can be used to join to other tables that contain the **SecurityUserID_SYS** (or any other user ID column).
- **SecurityProjectID_SYS** can be used to join to other tables that contain the **SecurityProjectID_SYS** (or any other project ID column).
- To secure snapshot project data, secure the snapshot data, then use the **SecurityUserID_SYS** column to join to the **SecurityUserID_SYS** column from the **Security Users** table to get the **NTName** for the user.

Load Order Examples:
`/api/v2/Data/GetSecurityUsersData` `GetSnapshotUserProjectAccessData`

Important! This list of examples includes some calls to get you started, but does not include all available options.

Snapshot Project Security List Values by List

This table contains security list value information by level for a specific security list for all snapshot projects in a system.

Accessed by: `/api/v2/Data/GetSnapshotProjectSecurityListValuesData('SecurityListName')`

Columns	Datatypes	Used for Joins
SnapshotID_SYS	Number	Yes
SnapshotProjectID_SYS	Number	Yes

Important! The remaining columns are the security list level names returned as strings, which will vary depending on your company's configuration.

How to join to other tables:

- **SnapshotID_SYS** can be used to join to other snapshot-related tables that contain the **SnapshotID_SYS** column.
- **SnapshotID_SYS** can be used to join to other snapshot-related tables that contain the **SnapshotID_SYS** column.
- To secure the data in this table, secure the snapshot data, then use the **SnapshotID_SYS** and **SnapshotProjectID_SYS** columns to join to the **SnapshotID_SYS** and **SnapshotProjectID_SYS** columns from the **Snapshot Project Details** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetSnapshotUserProjectAccessData
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Details')
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Project Details')
/api/v2/Data/GetSnapshotProjectMetricData('category')
/api/v2/Data/GetSnapshotProjectSecurityListValuesData('Security List 1')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Snapshot Project Details

This table contains information on snapshot project details.

Accessed by: `/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Project Details')`

Columns	Datatypes	Used for Joins
SnapshotID_SYS	Number	Yes
SnapshotProjectID_SYS	Number	Yes
Project Name	String	No
Project Description	String	No
Project ID	String	No
Project Creation Date	Date	No
Project Team Leader Name	String	No
Project is In Trouble	String	No
Project Most Recent Status Report Text	String	No
Project Closed	String	No
Project Closed Date	Date	No
Idea Submitter Name	String	No
Project Team Leader ID	Number	Yes
Current Phase ID	Number	Yes
Last Gate Decision Code	Number	No
Migrated-from Project ID	Number	Yes
Project Model Name	String	No
Project Model ID	Number	No
Project Current Stage Name	String	No
Project Most Recent Status Report Date	Date	No
Project Most Recent Status Report Submitter Name	String	No
Project Created By	String	No
Project Class	String	No
Project Currency Code	String	No
Project Currency Name	String	No
Project Currency Conversion Factor	Number	No

Columns	Datatypes	Used for Joins
Project Process Type	String	No
Project Snapshot Date	Date	No
Project Start Date	Date	No
Project End Date	Date	No

How to join to other tables:

- **SnapshotID_SYS** can be used to join to other snapshot-related tables that contain the **SnapshotID_SYS** column.
- **SnapshotProjectID_SYS** can be used in conjunction with the **SnapshotID_SYS** column to join to other snapshot-related tables that contain the **SnapshotProjectID_SYS** and **SnapshotID_SYS** columns.
- **Migrated-from Project ID** can be used to join to other tables that contain the **ProjectID_SYS** column. **Migrated-from Project ID** can also be used in conjunction with the **SnapshotID_SYS** and **SnapshotProjectID_SYS** columns to join to other snapshot-related tables that contain the **SnapshotProjectID_SYS** and **SnapshotID_SYS** columns.
- To secure snapshot project data, use the **SnapshotID_SYS** and **SnapshotProjectID_SYS** columns to join to the **SecuritySnapshotID_SYS** and **SecurityProjectID_SYS** columns from the **Snapshot User Project Access** table, then use the **SecurityUserID_SYS** column to join to the **SecurityUserID_SYS** column from the **Security Users** table.
- To access more Project Team Leader information, use the **Project Team Leader ID** column to join to the **UserID_SYS** column from the **Users** table.
- To get information about current stage of a project, use the **Current Phase ID** column in conjunction with the **SnapshotProjectID_SYS** and **SnapshotID_SYS** columns to join to the **ProjectPhaseID_SYS**, **SnapshotProjectID_SYS**, and **SnapshotID_SYS** columns from the **Snapshot Project Stages and Gates** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetSnapshotUserProjectAccessData
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Details')
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Project
Details')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Snapshot Project Stages and Gates

This table contains information on snapshot project stages and gates.

Accessed by: `/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Project $STAGES$ and $GATES$')`

Columns	Datatypes	Used for Joins
Project Stage Exists	String	No
Project Stage Name	String	No
Project Stage Relative Position	String	No
Project Stage Duration	Number	No
Project Gate Exists	String	No
Project Gate Name	String	No
Project Gate Date	Date	No
Project Gate Decision	String	No
Project Gate Meeting Place	String	No
Project Gate Notes	String	No
Project Gate Plan Date	Date	No
Project Gate Relative Position	String	No
ProjectPhaseID_SYS	Number	Yes
Project Gate Conditions	String	No
Project Gate Owner Name	String	No
Project Gate Owner ID	Number	Yes
SnapshotProjectID_SYS	Number	Yes
Project Stage Locked	String	No
SnapshotID_SYS	Number	Yes

How to join to other tables:

- **SnapshotID_SYS** can be used to join to other snapshot-related tables that contain the **SnapshotID_SYS** column.
- **SnapshotProjectID_SYS** can be used in conjunction with the **SnapshotID_SYS** column to join to other snapshot-related tables that contain the **SnapshotProjectID_SYS** and **SnapshotID_SYS** columns.
- **ProjectPhaseID_SYS** can be used in conjunction with the **SnapshotProjectID_SYS** and **SnapshotID_SYS** columns to join to other tables that contain the

ProjectPhaseID_SYS (or any other phase ID column), **SnapshotProjectID_SYS** and **SnapshotID_SYS** columns.

- To secure the data in this table, secure the snapshot data, then use the **SnapshotID_SYS** and **SnapshotProjectID_SYS** columns to join to the **SnapshotID_SYS** and **SnapshotProjectID_SYS** columns from the **Snapshot Project Details** table.
- To access more Project Gate Owner information, use the **Project Gate Owner ID** column to join to the **UserID_SYS** column from the **Users** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetSnapshotUserProjectAccessData
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Details')
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Project Details')
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Project $STAGES$ and $GATES$')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Snapshot Project Matrix Metrics

This table contains snapshots of project matrix metric data for a specific matrix.

It is recommended to name these "Snapshot Project Matrix Metric - *<matrix name>*" so that you can keep each matrix separate.

Accessed by: `/api/v2/Data/GetSnapshotProjectMatrixMetricsData ('matrixSystemName')`

Columns	Datatypes	Used for Joins
SnapshotID_SYS	Number	Yes
SnapshotProjectID_SYS	Number	Yes
RowID	Number	No

Important! The remaining columns are the metrics in the matrix, which will vary depending on your company's configuration.

How to join to other tables:

- **SnapshotID_SYS** can be used to join to other snapshot-related tables that contain the **SnapshotID_SYS** column.
- **SnapshotProjectID_SYS** can be used in conjunction with the **SnapshotID_SYS** column to join to other snapshot-related tables that contain the **SnapshotProjectID_SYS** and **SnapshotID_SYS** columns.
- To secure the data in this table, secure the snapshot data, then use the **SnapshotID_SYS** and **SnapshotProjectID_SYS** columns to join to the **SnapshotID_SYS** and **SnapshotProjectID_SYS** columns from the **Snapshot Project Details** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetSnapshotUserProjectAccessData
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Details')
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Project Details')
/api/v2/Data/GetSnapshotProjectMatrixMetricsData ('matrixSystemName')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Snapshot Project Metrics

This table contains snapshot data on metrics in a project for a specific category.

It is recommended to name these "Snapshot Project Metrics - <*category*>" so that you can keep each category separate.

Accessed by: `/api/v2/Data/GetSnapshotProjectMetricData('category')`

Columns	Datatypes	Used for Joins
SnapshotID_SYS	Number	Yes
SnapshotProjectID_SYS	Number	Yes

Important! The remaining columns are the metrics in this category, which will vary depending on your company's configuration.

How to join to other tables:

- **SnapshotID_SYS** can be used to join to other snapshot-related tables that contain the **SnapshotID_SYS** column.
- **SnapshotProjectID_SYS** can be used in conjunction with the **SnapshotID_SYS** column to join to other snapshot-related tables that contain the **SnapshotProjectID_SYS** and **SnapshotID_SYS** columns.
- To secure the data in this table, secure the snapshot data, then use the **SnapshotID_SYS** and **SnapshotProjectID_SYS** columns to join to the **SnapshotID_SYS** and **SnapshotProjectID_SYS** columns from the **Snapshot Project Details** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetSnapshotUserProjectAccessData
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Details')
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Project Details')
/api/v2/Data/GetSnapshotProjectMetricData ('category')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Snapshot Resources Data Tables

Snapshot Resource Pools

This table contains information on snapshot resource pools used in Resource Planning.

Accessed by: `/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Resource Pools')`

Columns	Datatypes	Used for Joins
SnapshotID_SYS	Number	Yes
Resource Pool Name	String	No
Resource Pool Owner Name	String	No
Resource Pool is Active	Boolean	No
ResourcePoolID_SYS	Number	Yes
Resource Pool Owner ID	Number	Yes
Resource Pool Unit of Measure	String	No

How to join to other tables:

- **SnapshotID_SYS** can be used to join to other snapshot-related tables that contain the **SnapshotID_SYS** column.
- **ResourcePoolID_SYS** can be used in conjunction with the **SnapshotID_SYS** column to join to other resource planning-related tables that contain the **ResourcePoolID_SYS** and **SnapshotID_SYS** columns.
- To access more Resource Pool Owner information, use the **Resource Pool Owner User ID** column to join to the **UserID_SYS** column from the **Users** table.

Load Order Examples:

```
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Details')
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Resource
Pools')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Snapshot Resources

This table contains information on snapshot resources used in Resource Planning.

Accessed by: `/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Resources')`

Columns	Datatypes	Used for Joins
SnapshotID_SYS	Number	Yes
Resource Name	String	No
ResourceID_SYS	Number	Yes
Resource Pool Member User ID	Number	Yes
Resource Account Status	String	No
ResourcePoolID_SYS	Number	Yes

How to join to other tables:

- **SnapshotID_SYS** can be used to join to other snapshot-related tables that contain the **SnapshotID_SYS** column.
- **ResourceID_SYS** can be used in conjunction with the **SnapshotID_SYS** column to join to other resource planning-related tables that contain the **ResourceID_SYS** and **SnapshotID_SYS** columns.
- **ResourcePoolID_SYS** can be used in conjunction with the **SnapshotID_SYS** column to join to other resource planning-related tables that contain the **ResourcePoolID_SYS** and **SnapshotID_SYS** columns.
- To access more Resource Pool Member information, use the **Resource Pool Member User ID** column to join to the **UserID_SYS** column from the **Users** table.

Load Order Examples:

```
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Details')
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Resources')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Snapshot Resource Capacities

This table contains information on snapshot resource capacities used in Resource Planning.

Accessed by: `/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Resource Capacities')`

Columns	Datatypes	Used for Joins
SnapshotID_SYS	Number	Yes
Resource Capacity Start Date	Date	No
Resource Capacity End Date	Date	No
Resource Capacity	Number	No
Resource Capacity Unavailable	Number	No
ResourcePeriodID_SYS	Number	Yes
ResourceID_SYS	Number	Yes

How to join to other tables:

- **SnapshotID_SYS** can be used to join to other snapshot-related tables that contain the **SnapshotID_SYS** column.
- **ResourceID_SYS** can be used in conjunction with the **SnapshotID_SYS** column to join to other resource planning-related tables that contain the **ResourceID_SYS** and **SnapshotID_SYS** columns.
- **ResourcePeriodID_SYS** can be used to in conjunction with the **SnapshotID_SYS** column to join to other resource planning-related tables that contain the **ResourcePeriodID_SYS** and **SnapshotID_SYS** columns.
- To access more information about the time period, use the **ResourcePeriodID_SYS** column to join to the **TimePeriodID_SYS** column from the **Time Period** table.

Load Order Examples:
`/api/v2/Data/GetTimePeriodsData`
`/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Details')`
`/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Resource Pools')`
`/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Resources')`
`/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Resource Capacities')`

Load Order Examples:

Important! This list of examples includes some calls to get you started, but does not include all available options.

Snapshot Resource Demands

This table contains information on snapshot resource demands used in Resource Planning.

Accessed by: `/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Resource Demands')`

Columns	Datatypes	Used for Joins
SnapshotID_SYS	Number	Yes
Resource Demand Project Name	String	No
Resource Demand Project Stage Name	String	No
Resource Demand Start Date	Date	No
Resource Demand End Date	Date	No
Resource Demand Type	String	No
Resource Demand Value	Number	No
Resource Demand is Active	String	No
ProjectID_SYS	Number	Yes
Resource Demand Project Phase ID	Number	Yes
Resource Demand Type ID	Number	No
ResourceID_SYS	Number	Yes
ResourcePeriodID_SYS	Number	Yes
Resource Demand Curve	String	No
Resource Demand Multiplier	Number	No

How to join to other tables:

- **ProjectID_SYS** can be used in conjunction with the **SnapshotID_SYS** column to join to other project-related tables that contain the **ProjectID_SYS** and **SnapshotID_SYS** columns.
- **SnapshotID_SYS** can be used to join to other snapshot-related tables that contain the **SnapshotID_SYS** column.
- **ResourceID_SYS** can be used in conjunction with the **SnapshotID_SYS** column to join to other resource planning-related tables that contain the **ResourceID_SYS** and **SnapshotID_SYS** columns.

- **ResourcePeriodID_SYS** can be used to in conjunction with the **SnapshotID_SYS** column to join to other resource planning-related tables that contain the **ResourcePeriodID_SYS** and **SnapshotID_SYS** columns.
- **Resource Demand Project Phase ID** can be used in conjunction with the **SnapshotID_SYS** and **ProjectID_SYS** columns to join to other tables that contain the **ProjectPhaseID_SYS** (or any other phase ID column), **SnapshotID_SYS** and **ProjectID_SYS** columns.
- To secure the data in this table, secure the snapshot data, then use the **SnapshotID_SYS** and **SnapshotProjectID_SYS** columns to join to the **SnapshotID_SYS** and **SnapshotProjectID_SYS** columns from the **Snapshot Project Details** table.
- To access more information about the time period, use the **ResourcePeriodID_SYS** column to join to the **TimePeriodID_SYS** column from the **Time Period** table.

Load Order Examples:

```
/api/v2/Data/GetSecurityUsersData
/api/v2/Data/GetSnapshotUserProjectAccessData
/api/v2/Data/GetTimePeriodsData
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Details')
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Project Details')
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Resource Pools')
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Resources')
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Resource Capacities')
/api/v2/Data/GetTableData('SGM_SNAPSHOTS_Resource Demands')
```

Important! This list of examples includes some calls to get you started, but does not include all available options.

Custom Data Tables

This is a general topic for all custom reporting tables that have been added to Accolade.

It is recommended to name these "Custom Table - <*custom table name*>" so that you can keep each reporting table separate.

Accessed by: `/api/v2/Data/GetCustomTableData('systemName')`

Columns	Datatypes	Used for Joins
Important! The columns and datatypes available in the reporting table will vary depending on your company's configuration.		

How to join to other tables:

- Joining custom tables to other Accolade data will vary depending on your company's configuration.

Load Order Examples:
`/api/v2/Data/GetCustomTableData('systemName')`
Important! Like join options, the load order examples will vary depending on your company's configuration.

Sopheon Corporation

6870 West 52nd Avenue, Suite 215

Arvada, CO 80002