



# Getting Started with the SDK

Version 2019.8.0



This edition of the *Getting Started with the SDK* refers to version 2019.8.0 of Black Duck.

This document created or updated on Tuesday, July 23, 2019.

**Please send your comments and suggestions to:**

Synopsys  
800 District Avenue, Suite 201  
Burlington, MA 01803-5061 USA

Copyright © 2019 by Synopsys.

All rights reserved. All use of this documentation is subject to the license agreement between Black Duck Software, Inc. and the licensee. No part of the contents of this document may be reproduced or transmitted in any form or by any means without the prior written permission of Black Duck Software, Inc.

Black Duck, Know Your Code, and the Black Duck logo are registered trademarks of Black Duck Software, Inc. in the United States and other jurisdictions. Black Duck Code Center, Black Duck Code Sight, Black Duck Hub, Black Duck Protex, and Black Duck Suite are trademarks of Black Duck Software, Inc. All other trademarks or registered trademarks are the sole property of their respective owners.

<b>Chapter 1: About the Black Duck SDK</b>	<b>1</b>
Viewing and Testing the REST APIs	1
Authentication	1
Generating an API token	2
Using API token	2
Using the bearer token in API calls	3
API requests using authorization-based authentication	3
API versions	6
<b>Chapter 2: Example of using the Black Duck SDK</b>	<b>7</b>
Step 1: Authentication	7
Step 2: Get the Application1 project	7
Step 3: Get the Application1 project versions	8
Step 4: Get the Risk Profile for “1.0” version	8
<b>Chapter 3: CSRF security</b>	<b>10</b>
Other names	10
CSRF security in Black Duck	10
<b>Chapter 4: API list</b>	<b>13</b>

## Black Duck documentation

The documentation for Black Duck consists of online help and these documents:

Title	File	Description
Release Notes	release_notes.pdf	Contains information about the new and improved features, resolved issues, and known issues in the current and previous releases.
Installing Black Duck using Docker Compose	install_compose.pdf	Contains information about installing and upgrading Black Duck using Docker Compose.
Installing Black Duck using Docker Swarm	install_swarm.pdf	Contains information about installing and upgrading Black Duck using Docker Swarm.
Installing Black Duck using Kubernetes	install_kubernetes.pdf	Contains information about installing and upgrading Black Duck using Kubernetes.
Installing Black Duck using OpenShift	install_openshift.pdf	Contains information about installing and upgrading Black Duck using OpenShift.
Getting Started	getting_started.pdf	Provides first-time users with information on using Black Duck.
Scanning Best Practices	scanning_best_practices.pdf	Provides best practices for scanning.
Getting Started with the SDK	getting_started_sdk.pdf	Contains overview information and a sample use case.

Title	File	Description
Report Database	report_db.pdf	Contains information on using the report database.
User Guide	user_guide.pdf	Contains information on using Black Duck's UI.

Black Duck integration documentation can be found on [Confluence](#).

## Customer support

If you have any problems with the software or the documentation, please contact Synopsys Customer Support.

You can contact Synopsys Support in several ways:

- Online: <https://www.synopsys.com/software-integrity/support.html>
- Email: [software-integrity-support@synopsys.com](mailto:software-integrity-support@synopsys.com)
- Phone: See the Contact Us section at the bottom of our [support page](#) to find your local phone number.

Another convenient resource available at all times is the [online customer portal](#).

## Synopsys Software Integrity Community

The Synopsys Software Integrity Community is our primary online resource for customer support, solutions, and information. The Community allows users to quickly and easily open support cases and monitor progress, learn important product information, search a knowledgebase, and gain insights from other Software Integrity Group (SIG) customers. The many features included in the Community center around the following collaborative actions:

- Connect - Open support cases and monitor their progress, as well as, monitor issues that require Engineering or Product Management assistance
- Learn - Insights and best practices from other SIG product users to allow you to learn valuable lessons from a diverse group of industry leading companies. In addition, the Customer Hub puts all the latest product news and updates from Synopsys at your fingertips, helping you to better utilize our products and services to maximize the value of open source within your organization.
- Solve - Quickly and easily get the answers you're seeking with the access to rich content and product knowledge from SIG experts and our Knowledgebase.
- Share - Collaborate and connect with Software Integrity Group staff and other customers to crowdsource solutions and share your thoughts on product direction.

[Access the Customer Success Community](#). If you do not have an account or have trouble accessing the system, click [here](#) to get started, or send an email to [community.manager@synopsys.com](mailto:community.manager@synopsys.com).

## Training

Synopsys Software Integrity, Customer Education (SIG Edu) is a one-stop resource for all your Black Duck education needs. It provides you with 24x7 access to online training courses and how-to videos.

New videos and courses are added monthly.

At Synopsys Software Integrity, Customer Education (SIG Edu), you can:

- Learn at your own pace.
- Review courses as often as you wish.
- Take assessments to test your skills.
- Print certificates of completion to showcase your accomplishments.

Learn more at <https://community.synopsys.com/s/education>.

# Chapter 1: About the Black Duck SDK

Black Duck REST APIs provide a standard interface for interacting with Black Duck.

REST APIs provide access to resources (data entities) via URI paths. To use a REST API, your application makes a HTTP request and parses the response. The methods used by your application are the standard HTTP methods of GET, POST, PUT and DELETE. REST APIs operate over HTTP(S) making it easy to use with any programming language or framework. The input and output format for the majority of Black Duck REST APIs is JSON.

**Important:** Black Duck does not support REST APIs that begin with "v1" or "internal".

## Viewing and Testing the REST APIs

Black Duck provides online access and documentation to the REST servers and individual REST APIs. After logging in to Black Duck, you can enter the following url to access this information:

`http://<your Black Duck server url>/api.html`

Each supported REST API is documented with the required input parameters and expected response.

A **Try it out** button enables you to do interactive testing of the API.

**Note:** The API page uses Swagger to document Black Duck APIs. Swagger documentation should not be considered part of Black Duck REST APIs and can change without warning or backward compatibility.

## Authentication

Black Duck enables you to generate one or more tokens for accessing Black Duck APIs.

These tokens are intended to replace the use of username/password credentials when accessing Black Duck APIs. With access tokens, if a security breach occurs, the user's credentials (which might be their SSO or LDAP credentials) are not directly compromised.

**To access Black Duck API, you must authenticate by doing the following steps:**

1. Generate an API token in Black Duck by using the Black Duck UI to navigate to your Profile page to generate API token.
2. Pass the API token in an HTTP POST to `/api/tokens/authenticate` to generate a Bearer token,

which you use for authorization.

3. Pass the bearer token in the authorization header of you API requests to get data from your Black Duck instance.

## Generating an API token

Log in to your Black Duck instance and generate an API token.

To generate an API token

1. On the top navigation bar, click **System > My Profile**.
2. In the **User Access Token** section, type a name in the **Name** field.
3. **Optional:** In the **Description** field, type a description or definition.
4. Select **Read Access** and/or **Write Access**.
5. Click **Generate**.  
The API token displays in a pop-up window. For security reasons, this is the only time your user API token displays. Please save this token. If the token is lost, you must regenerate it.

## Using API token

To use an API token, make an HTTP POST to `/api/tokens/authenticate` with the following header:  
Authorization: token <API token>

This produces a response with a bearer token.

The screenshot shows a REST client interface with the following details:

- URL:** `https://eng-...lan/api/tokens/authenticate`
- Method:** POST
- Headers:**
  - Authorization:** token YyydpitokenapitokenY1YWMtM2VjMi00YzUzLWE3ODUtMDYyMDc2MD...
  - Accept:** application/vnd.blackducksoftware.user-4+json
- Body:** (Empty)
- Response:**

```

1 {
2   "bearerToken": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9
   .eyJodwJfawQioiIiwMDAwMDAwMC0wMDAwLTAwMDAwMDAwMS0wMDAwMDAwMDAwMDEiLCJ1c2VyX25hbWUiOiJzeXNhZG1pb2IiInNjb381IjpbInJlYWQiLCJ3cm10Z
   SRUxZQVNFx0xJlU1Q1LCJQUk9KRUNUX1JFQUQiLCJQUdFUKVBRClSik5PVE1GSUN8VE1PT19SRUFEIiw1VMTkVSQUJ3TE1UwVJFTUVESUFUSU90X1JFQUQiLCJ0
   DQ5FREVMRVRFIiw1VMTkVSQUJ3TE1UwV9SRUFEIiw009ORk1HX1JFQUQiLCJQUk9KRUNUX1NDSEVEVUxFIiwu0NBt19VUER8VEU1LCJQUdFQ1JFQVRFIiw1T
   FTEVURSIkNPTkZ3R19DUkV8VEU1LCJQUk9KRUNUX0xJU1Q1LCJKT0JfUkVBRClSikNPREVMT0NBVE1PT19SRUFEIiw1VTV9NX0ZJRUXEX1JFQUQiLCJ0VNVUT
   EQVRFIiwu0NBt19DUKVBVEU1LCJTRUFSQ0hfUkV3JTkRFWCIk98VVRX0NHSUUV0VF9DUKVBVEU1LCJ0VNVUT01fRk1FTERfVVB8EQVRFIiw1RVhURVJ0QUxfRVhUR
  
```

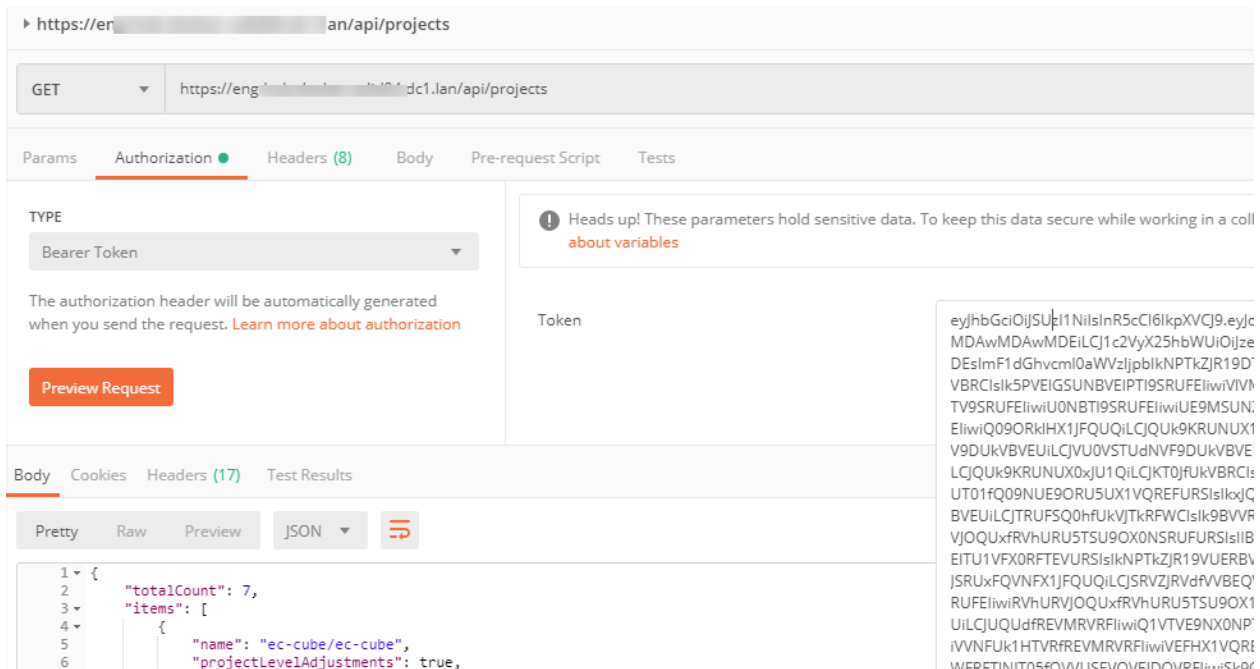


The following example shows a cURL POST request in Postman with the API token (Authorization token) from Black Duck to generate the Bearer token.

```
curl -X POST \
https://eng-hub-docker-vkkrkkk.ran/api/tokens/authenticate \
-H 'Accept: application/vnd.blackducksoftware.user-4+json' \
-H 'Authorization: token My-api-tokenGMwOC00MGQ2LTkMy_api_tokenZDZkOWI0ZDQwZjJiOjVjOTc3NmM2LTczZWEtNGZhNS04YWVhLTU0MTM2Y2VhZmEwMw==' \
```

## Using the bearer token in API calls

The following example in the Postman client shows a GET request using the bearer token that was generated from the API token.



## API requests using authorization-based authentication

Note how the token provided is sent back in the subsequent request within the 'Authorization' header with 'Bearer' authentication scheme

Additionally, no CSRF, (as described [here](#)) is included since it is not required when using the 'Authorization' header to pass the token.

```
curl -v -H "Authorization: Bearer <token>" -X GET
"https://<hostname>/api/components/b2d3d36c-15d3-4e1a-bef5-
8e907f107048/versions/35f6d76e-e4dd-4b87-bae5-133da093dcd2"
```

The following example shows the component-version REST API call, which returns component information:

```
GET /api/components/{componentId}/versions/{versionId}
```

where {componentID} and {versionID} are the internal Black Duck IDs for the OSS component in the Black Duck KB.

```
curl -X --header "Cookie: AUTHORIZATION: Bearer <token>"  
"Accept:application/json" "https://<Black Duck  
SERVER>/api/components/b2d3d36c-15d3-4e1a-bef5-8e907f107048/versions/35f6d76e-  
e4dd-4b87-bae5-133da093dcd2"
```

#### Request URL

```
https://<HUB_SERVER>/api/components/b2d3d36c-15d3-4e1a-bef5-8e907f107048/versions/35f6d76e-e4dd-4b87-bae5-133da093dcd2
```

#### Response Body

```
{  
  "versionName": "7.0.22",  
  "releasedOn": "2011-09-27T04:00:00.000Z",  
  "source": "KB",  
  "license": {  
    "type": "DISJUNCTIVE",  
    "licenses": [  
      {  
        "name": "Apache License 2.0",  
        "ownership": "OPEN_SOURCE",  
        "codeSharing": "PERMISSIVE",  
        "licenses": [],  
        "license": "https://saleshub.blackducksoftware.com/api/licenses/7cae335f-1193-421e-92f1-8802b4243e93"  
      }  
    ]  
  },  
}
```

The equivalent API call can also be made using the API GUI, as shown in the following image:

GET /api/components/{componentId}/versions/{versionId} findVersion

Response Class (Status 200)  
OK

Model Example Value

```
{
  "license": {
    "present": true
  },
  "releasedOn": "2018-04-11T14:48:46.631Z",
  "source": "CUSTOM",
  "versionName": "string"
}
```

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
componentId	b2d3d36c-15d3-4e1a-bef5-8e907f107048	componentId	path	string
versionId	35f6d76e-e4dd-4b87-bae5-133da090dcd2	versionId	path	string

Try it out! [Hide Response](#)

Curl

```
curl -X GET --header 'Accept: application/json' --header 'X-CSRF-TOKEN: 02d565bb-ea6b-4d29-ae96-7f1500fd6ceb' 'https://eng-hub-docker-va1id03.dcl.lan:443/api/components/b2d3d36c-15d3-4e1a-bef5-8e907f107048/versions/35f6d76e-e4dd-4b87-bae5-133da090dcd2'
```

Request URL

```
https://eng-hub-docker-va1id03.dcl.lan:443/api/components/b2d3d36c-15d3-4e1a-bef5-8e907f107048/versions/35f6d76e-e4dd-4b87-bae5-133da090dcd2
```

Response Body

```
{
  "versionName": "7.0.22",
  "releasedOn": "2011-09-27T04:00:00.000Z",
  "source": "KB",
  "license": {
    "name": "Apache License 2.0",
    "ownership": "OPEN_SOURCE",
    "codeSharing": "PERMISSIVE",
    "licenseDisplay": "Apache License 2.0",
    "licenses": [],
    "license": "https://eng-hub-docker-va1id03.dcl.lan/api/components/ae4667bd-0bf2-4004-9edd-133a6c3550e0/versions/b0fff07c-9ce1-4b87-bae5-133da090dcd2"
  },
  "_meta": {
    "allow": {
      "GET": true
    }
  },
  "href": "https://eng-hub-docker-va1id03.dcl.lan/api/components/ae4667bd-0bf2-4004-9edd-133a6c3550e0/versions/b0fff07c-9ce1-4b87-bae5-133da090dcd2",
  "links": [
    {
      "rel": "self",
      "href": "https://eng-hub-docker-va1id03.dcl.lan/api/components/ae4667bd-0bf2-4004-9edd-133a6c3550e0/versions/b0fff07c-9ce1-4b87-bae5-133da090dcd2"
    }
  ]
}
```

Response Code

200

Response Headers

```
{
  "x-csrf-param": "csrf",
  "date": "Wed, 11 Apr 2018 14:52:14 GMT",
  "content-encoding": "gzip",
  "x-content-type-options": "nosniff",
  "x-csrf-token": "02d565bb-ea6b-4d29-ae96-7f1500fd6ceb",
  "transfer-encoding": "chunked",
  "connection": "keep-alive",
  "vary": "Accept-Encoding",
  "x-xss-protection": "1; mode=block",
  "x-csrf-header": "X-CSRF-TOKEN",
  "pragma": "no-cache",
  "server": "nginx/1.13.6",
  "x-frame-options": "SAMEORIGIN",
  "allow": "GET",
  "strict-transport-security": "max-age=31536000; includeSubDomains",
  "content-type": "application/json; charset=UTF-8",
  "cache-control": "no-cache, no-store, max-age=0, must-revalidate",
  "expires": "0"
}
```

## API versions

Black Duck APIs are backward compatible within a Black Duck media type (excluding application/json).

To avoid issues, when sending or receiving data, specify the Black Duck media type in the Accept header.

## Chapter 2: Example of using the Black Duck SDK

Use Case: I'm interested in using the REST APIs to find risk counts for my project 'Application1' version '1.0'.

### Step 1: Authentication

You must first authenticate with the Black Duck application.

For example, the following curl commands shows how to authenticate with the Black Duck application and get a bearer token prior to making the REST API calls:

```
curl -X POST \
https://Black Duck/api/tokens/authenticate \
-H 'Accept: application/vnd.blackducksoftware.user-4+json' \
-H 'Authorization: token <API token from Black Duck>' \
```

Using the bearer token that is output when you authenticate, Get a list of projects:

```
curl -X GET \
https://<Black Duck server url>/api/projects \
-H 'Accept: */*' \
-H 'Authorization: Bearer <bearer token>' \
```

### Step 2: Get the Application1 project

```
curl -X GET \
https://<Black Duck server url>:443/api/projects?q=name%3AApplication1 \
-H 'Accept: */*' \
-H 'Authorization: Bearer <bearer token>' \
```

```
{
  totalCount: 1
  -items: [1]
    -0: {
      name: "Application1"
      description: "Application 1"
      projectTier: 1
      source: "CUSTOM"
      -_meta: {
        -allow: [3]
          0: "GET"
          1: "PUT"
          2: "DELETE"
        href: "https://[redacted]/api/projects/4bfe0d5a-6b3d-4d11-8212-6b3b2f7b61b5"
        -links: [2]
          -0: {
            rel: "versions"
            href: "https://[redacted]/api/projects/4bfe0d5a-6b3d-4d11-8212-6b3b2f7b61b5/versions"
          }
          -1: {
            rel: "canonicalVersion"
            href: "https://[redacted]/api/projects/4bfe0d5a-6b3d-4d11-8212-6b3b2f7b61b5/versions/f7838043-f144-4853-9c62-d1c4c49b909f"
          }
        }
      }
    }
  }
}
```

Included in the JSON response are all your available http calls. Notice you can call “versions” or “canonicalVersion”.

## Step 3: Get the Application1 project versions

Get the versions of Application1 project.

Using the links provided in the JSON response in step 1 I can use the following to get my project versions:

<https://<Black Duck server url>/api/projects/4bfe0d5a-6b3d-4d11-8212-6b3b2f7b61b5/versions>

```
{
  totalCount: 1
  -items: [1]
    -0: {
      versionName: "1.0"
      phase: "DEVELOPMENT"
      distribution: "EXTERNAL"
      source: "CUSTOM"
      -_meta: {
        -allow: [3]
          0: "GET"
          1: "PUT"
          2: "DELETE"
        href: "https://[redacted]/api/projects/4bfe0d5a-6b3d-4d11-8212-6b3b2f7b61b5/versions/f7838043-f144-4853-9c62-d1c4c49b909f"
        -links: [2]
          -0: {
            rel: "versionReport"
            href: "https://[redacted]/api/versions/f7838043-f144-4853-9c62-d1c4c49b909f/reports"
          }
          -1: {
            rel: "riskProfile"
            href: "https://[redacted]/api/projects/4bfe0d5a-6b3d-4d11-8212-6b3b2f7b61b5/versions/f7838043-f144-4853-9c62-d1c4c49b909f/risk-profile"
          }
        }
      }
    }
  }
}
```

Included in the JSON response are all your available http calls. Notice you can call “riskProfile”.

## Step 4: Get the Risk Profile for “1.0” version

Using the links in the JSON response from above I can call the following to get the risk profile:

<https://<Black Duck server url>/api/projects/4bfe0d5a-6b3d-4d11-8212-6b3b2f7b61b5/versions/f7838043-f144-4853-9c62-d1c4c49b909f/risk-profile>

```
{
  -categories: {
    -VERSION: { ... }
    -ACTIVITY: { ... }
    -VULNERABILITY: {
      HIGH: 0
      MEDIUM: 1
      LOW: 0
      OK: 8
      UNKNOWN: 0
    }
    -LICENSE: {
      HIGH: 0
      MEDIUM: 1
      LOW: 0
      OK: 8
      UNKNOWN: 0
    }
    -OPERATIONAL: { ... }
  }
  -_meta: {
    -allow: [1]
    0: "GET"
    href: "https://[redacted]/api/projects/4bfe0d5a-6b3d-4d11-8212-6b3b2f7b61b5/versions/f7838043-f144-4853-9c62-d1c4c49b909f/risk-profile"
    -links: [1]
    -0: {
      rel: "version"
      href: "https://[redacted]/api/projects/4bfe0d5a-6b3d-4d11-8212-6b3b2f7b61b5/versions/f7838043-f144-4853-9c62-d1c4c49b909f"
    }
  }
}
```

The JSON response includes all risk counts for Version, Activity, Vulnerability, License, and Operational risk. The user can now use those values for whatever external activity they'd like to do (fail a build, create bug tracking tickets, and so on)

Cross-site request forgeries (CSRF) are types of attacks which perform unwanted actions on a web application without the user's intent. This type of vulnerability happens on web applications which use cookies to identify the user. For example, the attacker uses the same cookies to make a fraudulent request, and attempts to lure the user to click a fraudulent link. If the user clicks the link after being authenticated, the fraudulent request performs any actions available to the logged-in user, but without the knowledge of the user.

The forged request works because browser requests automatically include all credentials associated with the web application, such as the logged-in user's session information in cookies. The server cannot distinguish between the forged request from the user and the legitimate request from the user.

### Other names

CSRF attacks are also known by other names; the main difference is the technique used to perform the attack. The names include XSRF, Sea Surf, Session Riding, and Hostile Liking.

### CSRF security in Black Duck

Black Duck now features improved security for attempted cross-site request forgeries (CSRF).

**Note:** CSRF is only needed if the request uses cookies in the request.

Black Duck uses a Synchronized Token Pattern (STP) to address CSRF. This is a technique where a secret and unique token value is passed along with every request through form data or the header. This token is then verified on the server side. If the tokens do not match, then the request is ignored, and a *403 forbidden* error displays. The unique token is generated for each session, not for each request. The same unique token is used for the duration of the session, and a new token is generated for each new session. The token is destroyed when its session is destroyed. The following is an example of the CSRF format:

```
headerName = X-CSRF-TOKEN
parameterName = _csrf
token = 484dcea0-82a7-4f3e-9158-f80513ed86f9
```

When making a REST call to authenticate the user, a POST request (login request) is made to the URL `/j_spring_security_check`. The CSRF information is returned in the response header. The following is an example of the *login request header*:

```
POST /j_spring_security_check HTTP/1.1
Host: qa-hub21.dc1.1an
Connection: keep-alive
```



```
Content-Length: 40
Origin: https://qa-hub21.dc1.lan
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_5)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110
Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Accept: */*
X-Requested-With: XMLHttpRequest
X-FirePHP-Version: 0.0.6
Referer: https://qa-hub21.dc1.lan/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.8,ja;q=0.6,cs;q=0.4
```

The following is an example of the returned *login response header*. The CSRF token is highlighted.

```
HTTP/1.1 204
Server: nginx/1.10.3
Date: Wed, 14 Jun 2017 16:27:50 GMT
Connection: keep-alive
X-CSRF-HEADER: X-CSRF-TOKEN
X-CSRF-PARAM: _csrf
X-CSRF-TOKEN: 484dcea0-82a7-4f3e-9158-f80513ed86f9
Set-Cookie: AUTHORIZATION_BEARER=<token>
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: SAMEORIGIN
```

**To add the CSRF security header to the request:**

1. *Only if cookies are used*, copy the CSRF token from the login response header, and include the CSRF token in each subsequent POST, PUT, PATCH and DELETE request headers.

The following is a sample request with the CSRF token passed in using the request header:

```
POST /api/v1/projects HTTP/1.1
Host: qa-hub21
```

```
Connection: keep-alive
Content-Length: 180
Origin: https://qa-hub21
X-CSRF-TOKEN: 484dcea0-82a7-4f3e-9158-f80513ed86f9
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_5)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110
Safari/537.36
Content-Type: application/json
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
X-FirePHP-Version: 0.0.6
Referer: https://qa-hub21/ui/projects/view:create/action:modal
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.8,ja;q=0.6,cs;q=0.4
Cookie: AUTHORIZATION_BEARER=<token>
```

**Note:** You must use the CSRF token for POST, PUT, or DELETE requests when using cookies. Any integration with a Black Duck 4.0.0 (or higher) server must include the token.

Each supported REST API is documented with the required input parameters. Below is the API list for Black Duck 2019.8.0:

**aggregate-bom-rest-server : Aggregate Bom Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/components/{componentId}/versions/{componentVersionId}/references	findProjectVersionByComponentVersionId
GET	/api/licenses/{licenseId}/bom-counts	getLicenseBomCounts
GET	/api/projects/{projectId}/versions/{projectVersionId}/components	findBomComponentsV1
POST	/api/projects/{projectId}/versions/{projectVersionId}/components	addBomComponent
GET	/api/projects/{projectId}/versions/{projectVersionId}/components-filters	getBomComponentFilter
GET	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}	findBomComponentV1
PUT	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}	updateBomComponentV1
GET	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}/custom-fields	findProjectVersionComponentCustomFields
GET	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}/custom-fields/{customFieldId}	findProjectVersionComponentCustomField
PUT	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}/custom-fields/{customFieldId}	storeProjectVersionComponentCustomFieldValue
GET	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}/licenses/{licenseId}	findBomComponentLicenseV6
GET	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}/licenses/{licenseId}/text	findBomComponentLicenseText
GET	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}/versions/{componentVersionId}	findBomComponentVersionV1
PUT	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}/versions/{componentVersionId}	updateBomComponentVersionV2
GET	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}/versions/{componentVersionId}/custom-fields	findProjectVersionComponentVersionCustomFields
GET	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}/versions/{componentVersionId}/custom-fields/{customFieldId}	findProjectVersionComponentVersionCustomField
PUT	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}/versions/{componentVersionId}/custom-fields/{customFieldId}	storeProjectVersionComponentVersionCustomFieldValue
GET	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}/versions/{componentVersionId}/licenses/{licenseId}	findBomComponentVersionLicense
GET	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}/versions/{componentVersionId}/licenses/{licenseId}/text	findBomComponentVersionLicenseText

**api-token-rest-server : Api Token Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/current-user/tokens	getApiTokens
POST	/api/current-user/tokens	createApiToken
DELETE	/api/current-user/tokens/{tokenId}	deleteApiToken
GET	/api/current-user/tokens/{tokenId}	getApiToken
POST	/api/current-user/tokens/{tokenId}	regenerateApiToken
PUT	/api/current-user/tokens/{tokenId}	updateApiToken
POST	/api/tokens/authenticate	authenticateApiToken

**bom-component-comment-rest-server : Bom Component Comment Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/projects/{projectId}/versions/{versionId}/components/{componentId}/comments	findBomComponentComments
POST	/api/projects/{projectId}/versions/{versionId}/components/{componentId}/comments	createBomComponentComment
DELETE	/api/projects/{projectId}/versions/{versionId}/components/{componentId}/comments/{commentId}	deleteBomComponentComment
GET	/api/projects/{projectId}/versions/{versionId}/components/{componentId}/comments/{commentId}	findBomComponentComment
PUT	/api/projects/{projectId}/versions/{versionId}/components/{componentId}/comments/{commentId}	updateBomComponentComment
GET	/api/projects/{projectId}/versions/{versionId}/components/{componentId}/component-versions/{componentVersionId}/comments	findBomComponentVersionComments
POST	/api/projects/{projectId}/versions/{versionId}/components/{componentId}/component-versions/{componentVersionId}/comments	createBomComponentVersionComment
DELETE	/api/projects/{projectId}/versions/{versionId}/components/{componentId}/component-versions/{componentVersionId}/comments/{commentId}	deleteBomComponentVersionComment
GET	/api/projects/{projectId}/versions/{versionId}/components/{componentId}/component-versions/{componentVersionId}/comments/{commentId}	findBomComponentVersionComment
PUT	/api/projects/{projectId}/versions/{versionId}/components/{componentId}/component-versions/{componentVersionId}/comments/{commentId}	updateBomComponentVersionComment

**bom-component-issue-rest-server : Bom Component Issue Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

POST	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}/component-versions/{componentVersionId}/issues	Creates a new issue for component version. Created issue is mapped to issue tracker issue
DELETE	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}/component-versions/{componentVersionId}/issues/{issueId}	deleteIssue
GET	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}/component-versions/{componentVersionId}/issues/{issueId}	Gets an issue for a component version
PUT	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}/component-versions/{componentVersionId}/issues/{issueId}	Updates issue for component version. Updated issue is mapped to issue tracker issue
POST	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}/issues	Creates a new issue for component. Created issue is mapped to issue tracker issue
DELETE	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}/issues/{issueId}	deleteIssue
GET	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}/issues/{issueId}	Gets an issue for a component
PUT	/api/projects/{projectId}/versions/{projectVersionId}/components/{componentId}/issues/{issueId}	Updates issue for component. Updated issue is mapped to issue tracker issue
GET	/api/projects/{projectId}/versions/{projectVersionId}/issues	getProjectVersionIssues
GET	/api/releases/{releaseId}/issues	findIssuesByProjectVersionId

**code-location-rest-server : Code Location Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/codelocations	findCodeLocations
DELETE	/api/codelocations/{codeLocationId}	deleteCodeLocation
GET	/api/codelocations/{codeLocationId}	findCodeLocation
PUT	/api/codelocations/{codeLocationId}	updateCodeLocation
GET	/api/internal/codelocations/{codeLocationId}	findInternalCodeLocation
GET	/api/projects/{projectId}/versions/{versionId}/codelocations	findCodeLocationForProjectVersions

**component-filter-rest-server : Component Filter Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/component-approval-status-filters	getComponentApprovalStatusFilters
GET	/api/component-source-filters	getComponentSourceFilters

**component-license-rest-server : Component License Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/components/{componentId}/versions/{versionId}/origin/{originId}/file-licenses	getOriginFileLevelData
GET	/api/components/{componentId}/versions/{versionId}/origin/{originId}/file-licenses-fuzzy	getOriginFuzzyFileLevelData
GET	/api/components/{componentId}/versions/{versionId}/origin/{originId}/file-licenses/{licenseId}	getOriginLicenseFileLevelData

**component-origin-rest-server : Component Origin Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/components/{componentId}/versions/{versionId}/origin	findOriginsV5
GET	/api/components/{componentId}/versions/{versionId}/origin/{originId}	findOriginV5
GET	/api/components/{componentId}/versions/{versionId}/origin/{originId}/direct-dependencies	directDependencies

**component-version-filter-rest-server : Component Version Filter Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/component-version-approval-status-filters	getComponentVersionApprovalStatusFilters
-----	--	--

**component-rest-server : Component Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/components	getComponents
POST	/api/components	createCustomComponent
GET	/api/components-filters	geComponentFilter
DELETE	/api/components/{componentId}	deleteComponent
GET	/api/components/{componentId}	findComponent
PUT	/api/components/{componentId}	updateComponent
GET	/api/components/{componentId}/custom-fields	findComponentCustomFields
GET	/api/components/{componentId}/custom-fields/{customFieldId}	findComponentCustomField
PUT	/api/components/{componentId}/custom-fields/{customFieldId}	storeComponentCustomFieldValue
GET	/api/components/{componentId}/legacy-suite-ids	getLegacySuiteIds

**component-version-rest-server : Component Version Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/components/{componentId}/versions	findVersionsByProjectIdV5
POST	/api/components/{componentId}/versions	createCustomComponentVersion
GET	/api/components/{componentId}/versions-filters	getComponentVersionFilterV4
DELETE	/api/components/{componentId}/versions/{componentVersionId}	deleteCustomComponentVersion
GET	/api/components/{componentId}/versions/{componentVersionId}	findVersionV4
PUT	/api/components/{componentId}/versions/{componentVersionId}	updateCustomComponentVersionV4
GET	/api/components/{componentId}/versions/{componentVersionId}/legacy-suite-ids	getLegacySuiteIds
GET	/api/components/{componentId}/versions/{versionId}/crypto-algorithms	getVersionCryptoAlgorithms
GET	/api/components/{componentId}/versions/{versionId}/licenses/{licenseId}	findLicenseForComponentVersionV5
GET	/api/components/{componentId}/versions/{versionId}/licenses/{licenseId}/text	getLicenseText
GET	/api/components/{componentId}/versions/{versionId}/remediating	getRemediationOptions
GET	/api/releases/{releaseId}/redirect	getRedirect

**component-version-risk-profile-rest-server : Component Version Risk Profile Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/components/{componentId}/versions/{versionId}/risk-profile	findComponentVersionRiskProfile
-----	---	---------------------------------

**composite-code-location-rest-server : Composite Code Location Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

DELETE	/api/codelocations/{codeLocationId}	deleteCodeLocation
--------	-------------------------------------	--------------------



**custom-field-rest-server : Custom Field Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/custom-fields/objects	getCustomFieldObjects
GET	/api/custom-fields/objects/{customFieldObject}	getCustomFieldObject
GET	/api/custom-fields/objects/{customFieldObject}/field-filters	getCustomFieldFilter
GET	/api/custom-fields/objects/{customFieldObject}/fields	findCustomFieldByObject
POST	/api/custom-fields/objects/{customFieldObject}/fields	createCustomField
DELETE	/api/custom-fields/objects/{customFieldObject}/fields/{customFieldId}	deleteCustomField
GET	/api/custom-fields/objects/{customFieldObject}/fields/{customFieldId}	findCustomField
PUT	/api/custom-fields/objects/{customFieldObject}/fields/{customFieldId}	updateCustomField
GET	/api/custom-fields/objects/{customFieldObject}/fields/{customFieldId}/options	findCustomFieldOptions
POST	/api/custom-fields/objects/{customFieldObject}/fields/{customFieldId}/options	createCustomFieldOption
DELETE	/api/custom-fields/objects/{customFieldObject}/fields/{customFieldId}/options/{customFieldOptionId}	deleteOption
PUT	/api/custom-fields/objects/{customFieldObject}/fields/{customFieldId}/options/{customFieldOptionId}	updateCustomFieldOption
GET	/api/custom-fields/types	getCustomFieldTypes

**dashboard-rest-server : Dashboard Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/dashboard-summary	Get the aggregated links needed to construct the summary dashboard
-----	------------------------	--

**end-user-license-agreement-rest-server : End User License Agreement Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/enduserlicenseagreement	getEndUserLicenseAgreementAcceptance
POST	/api/enduserlicenseagreement	createOrUpdateEndUserLicenseAgreementAcceptance

**file-level-data-rest-server : File Level Data Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/components/{componentId}/versions/{componentVersionId}/origin/{componentVersionOriginId}/file-copyrights	getOriginFileLevelCopyrightData
GET	/api/components/{componentId}/versions/{versionId}/origin/{originId}/file-licenses	getOriginFileLevelData
GET	/api/components/{componentId}/versions/{versionId}/origin/{originId}/file-licenses-fuzzy	getOriginFuzzyFileLevelData
GET	/api/components/{componentId}/versions/{versionId}/origin/{originId}/file-licenses/{licenseId}	getOriginLicenseFileLevelData

**file-source-rest-server : File Source Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/file-source-contents/{sha1Id}	getFileContentExternal
-----	------------------------------------	------------------------

**health-check-rest-server : Health Check Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/health-checks/liveness	Determine the liveness status to determine if the system is in a normal, healthy state.
GET	/api/health-checks/readiness	Determine the readiness status to determine if the system is ready to receive requests.

**job-rest-server : Job Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/job-statistics	findStatistics
GET	/api/jobs-filters	getJobFilters

**journal-rest-server : Journal Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/journal/projects/{projectId}	findJournalForProject
GET	/api/journal/projects/{projectId}/journal-project-filters	getJournalProjectFilter
GET	/api/journal/projects/{projectId}/versions/{projectVersionId}	findJournalForProjectVersion
GET	/api/journal/projects/{projectId}/versions/{projectVersionId}/journal-version-filters	getJournalVersionFilter

**license-rest-server : License Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/licenses	findLicensesV5
POST	/api/licenses	createLicense
GET	/api/licenses-filters	getLicenseFilter
DELETE	/api/licenses/{licenseId}	deleteLicense
GET	/api/licenses/{licenseId}	findLicenseV5
PUT	/api/licenses/{licenseId}	updateLicenseV5
GET	/api/licenses/{licenseId}/license-terms	findLicenseTerms
POST	/api/licenses/{licenseId}/license-terms	createLicenseTermsToLicensesAssociations
GET	/api/licenses/{licenseId}/license-terms-filters	getLicenseTermFilter
DELETE	/api/licenses/{licenseId}/license-terms/{licenseTermId}	deleteLicenseLicenseTermAssociation
GET	/api/licenses/{licenseId}/license-terms/{licenseTermId}	getLicenseTerm
PUT	/api/licenses/{licenseId}/license-terms/{licenseTermId}	updateLicenseLicenseTermAssociation
GET	/api/licenses/{licenseId}/obligations	findLicenseObligations
GET	/api/licenses/{licenseId}/obligations-filters	getLicenseObligationFilter
GET	/api/licenses/{licenseId}/text	getLicenseTextView
PUT	/api/licenses/{licenseId}/text	updateLicenseTextView

**license-comment-rest-server : License Comment Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/licenses/{licenseId}/comments	findLicenseComments
POST	/api/licenses/{licenseId}/comments	createLicenseComment
DELETE	/api/licenses/{licenseId}/comments/{commentId}	deleteLicenseComment
GET	/api/licenses/{licenseId}/comments/{commentId}	findLicenseComment
PUT	/api/licenses/{licenseId}/comments/{commentId}	updateLicenseComment

**license-family-rest-server : License Family Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/license-families	findLicenseFamilies
POST	/api/license-families	createLicenseFamily
GET	/api/license-families-filters	getLicenseFamilyFilters
DELETE	/api/license-families/{licenseFamilyId}	deleteLicenseFamily
GET	/api/license-families/{licenseFamilyId}	findLicenseFamilyById
PUT	/api/license-families/{licenseFamilyId}	updateLicenseFamily

**license-term-rest-server : License Term Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/license-terms	findLicenseTerms
POST	/api/license-terms	createLicenseTerm
GET	/api/license-terms-filters	getLicenseTermFilters
DELETE	/api/license-terms/{licenseTermId}	deleteLicenseTerm
GET	/api/license-terms/{licenseTermId}	findLicenseTermById
PUT	/api/license-terms/{licenseTermId}	updateLicenseTerm
GET	/api/license-terms/{licenseTermId}/licenses	findLicenses
POST	/api/license-terms/{licenseTermId}/licenses	createLicensesToLicenseTermAssociations
DELETE	/api/license-terms/{licenseTermId}/licenses/{licenseId}	deleteLicenseTermLicenseAssociation
GET	/api/license-terms/{licenseTermId}/licenses/{licenseId}	getLicenseTermLicenseAssociation
PUT	/api/license-terms/{licenseTermId}/licenses/{licenseId}	updateLicenseTermLicenseAssociation

**license-term-category-rest-server : License Term Category Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/license-term-categories	findLicenseTermCategories
POST	/api/license-term-categories	createLicenseTermCategory
GET	/api/license-term-categories-filters	getLicenseFilter
DELETE	/api/license-term-categories/{licenseTermCategoryId}	deleteLicenseTermCategory
GET	/api/license-term-categories/{licenseTermCategoryId}	getLicenseTermCategory

**linked-data-rest-server : Linked Data Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)**HEAD** /api/bom-import head**POST** /api/bom-import importBom**matched-file-rest-server : Matched File Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)**GET** /api/projects/{projectId}/versions/{versionId}/components/{componentId}/matched-files findMatchedFiles**GET** /api/projects/{projectId}/versions/{versionId}/components/{componentId}/versions/{componentVersionId}/matched-files  
findMatchedFiles**GET** /api/projects/{projectId}/versions/{versionId}/components/{componentId}/versions/{componentVersionId}/origins/{componentOriginId}/matched-files  
findMatchedFiles**meta-rest-server : Meta Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)**GET** /api/current-version findCurrentHubVersion**POST** /api/reload-sso-config reloadSsoConfig**notification-rest-server : Notification Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)**GET** /api/notification-state-filters getNotificationStateFilters**GET** /api/notification-type-filters getNotificationTypeFilters**GET** /api/notifications findNotifications**GET** /api/notifications-filters getNotificationFilter**GET** /api/notifications/{notificationId} findNotification**GET** /api/users/{userId}/notification-counts findUserNotificationsCount**PUT** /api/users/{userId}/notification-states updateAllNotificationState**GET** /api/users/{userId}/notifications findUserNotifications**GET** /api/users/{userId}/notifications-filters getUserNotificationFilter**GET** /api/users/{userId}/notifications/{notificationId} findUserNotification**PUT** /api/users/{userId}/notifications/{notificationId} updateUserNotification**notification-subscription-rest-server : Notification Subscription Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)**GET** /api/users/{userId}/notification-subscriptions findNotificationSubscriptions**POST** /api/users/{userId}/notification-subscriptions createNotificationSubscription**DELETE** /api/users/{userId}/notification-subscriptions/{subscriptionId} deleteNotificationSubscription**GET** /api/users/{userId}/notification-subscriptions/{subscriptionId} findNotificationSubscription

**policy-rule-rest-server : Policy Rule Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/policy-rule-enabled-filters	getPolicyRuleEnabledFilters
GET	/api/policy-rules	findPolicyRulesV2
POST	/api/policy-rules	createPolicyRuleV5
GET	/api/policy-rules-filters	getPolicyRuleFilters
DELETE	/api/policy-rules/{policyRuleId}	deletePolicyRule
GET	/api/policy-rules/{policyRuleId}	findPolicyRuleV5
PUT	/api/policy-rules/{policyRuleId}	updatePolicyRuleV5

**project-assignment-rest-server : Project Assignment Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/projects/{projectId}/assignable-usergroups	getAssignableUserGroups
GET	/api/projects/{projectId}/assignable-users	getAssignableUsers
GET	/api/projects/{projectId}/usergroups	getAssignedUserGroups
POST	/api/projects/{projectId}/usergroups	addAssignedUser
DELETE	/api/projects/{projectId}/usergroups/{userGroupId}	removeAssignedUserGroup
GET	/api/projects/{projectId}/usergroups/{userGroupId}	getAssignedUserGroup
GET	/api/projects/{projectId}/users	getAssignedUsers
POST	/api/projects/{projectId}/users	addAssignedUser
DELETE	/api/projects/{projectId}/users/{userId}	removeAssignedUser
GET	/api/projects/{projectId}/users/{userId}	getAssignedUser
GET	/api/usergroups/{userGroupId}/assignable-projects	getAssignableProjectsForUserGroup
GET	/api/usergroups/{userGroupId}/projects	getProjectsAssignedToUserGroup
GET	/api/users/{userId}/assignable-projects	getAssignableProjectsForUser
GET	/api/users/{userId}/projects	getProjectsAssignedToUserOrUsersGroups

**project-mapping-rest-server : Project Mapping Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/projects/{projectId}/project-mappings	getProjectMappings
POST	/api/projects/{projectId}/project-mappings	createProjectMapping
DELETE	/api/projects/{projectId}/project-mappings/{projectMappingId}	deleteProjectMapping
GET	/api/projects/{projectId}/project-mappings/{projectMappingId}	findProjectMapping
PUT	/api/projects/{projectId}/project-mappings/{projectMappingId}	updateProjectMapping

**project-version-bom-comparison-rest-server : Project Version Bom Comparison Rest Server**

	Show/Hide	List Operations	Expand Operations
GET	/api/projects/{projectId}/versions/{projectVersionId}/comparison		diffBomComponents

**project-rest-server : Project Rest Server**

	Show/Hide	List Operations	Expand Operations
GET	/api/projects		findProjects
POST	/api/projects		createProject
DELETE	/api/projects/{projectId}		deleteProject
GET	/api/projects/{projectId}		findProject
PUT	/api/projects/{projectId}		updateProject
GET	/api/projects/{projectId}/custom-fields		findProjectCustomFields
GET	/api/projects/{projectId}/custom-fields/{customFieldId}		findProjectCustomField
PUT	/api/projects/{projectId}/custom-fields/{customFieldId}		storeProjectCustomFieldValue

**project-version-rest-server : Project Version Rest Server**

	Show/Hide	List Operations	Expand Operations
GET	/api/projects/{projectId}/versions		findVersionsByProjectIdV5
POST	/api/projects/{projectId}/versions		createVersionV5
GET	/api/projects/{projectId}/versions-filters		getVersionsFilter
DELETE	/api/projects/{projectId}/versions/{projectVersionId}		deleteVersion
GET	/api/projects/{projectId}/versions/{projectVersionId}		findVersionV4
PUT	/api/projects/{projectId}/versions/{projectVersionId}		updateVersionV4
GET	/api/projects/{projectId}/versions/{projectVersionId}/custom-fields		findProjectCustomFields
GET	/api/projects/{projectId}/versions/{projectVersionId}/custom-fields/{customFieldId}		findProjectCustomField
PUT	/api/projects/{projectId}/versions/{projectVersionId}/custom-fields/{customFieldId}		storeProjectCustomFieldValue

**registration-rest-server : Registration Rest Server**

	Show/Hide	List Operations	Expand Operations
GET	/api/registration		getRegistration
POST	/api/registration		createRegistration
PUT	/api/registration		updateRegistration
GET	/api/registration-attributes		getInternalAttributes
GET	/api/registration-summary		getInternalSummary

**report-rest-server : Report Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

DELETE	/api/reports/{reportId}	delete
GET	/api/reports/{reportId}	download
GET	/api/reports/{reportId}/contents	findContents
GET	/api/versions/{versionId}/license-reports	findVersionLicenseReports
POST	/api/versions/{versionId}/license-reports	createVersionLicenseReport
DELETE	/api/versions/{versionId}/license-reports/{reportId}	deleteVersionLicenseReport
GET	/api/versions/{versionId}/license-reports/{reportId}	findVersionLicenseReport
GET	/api/versions/{versionId}/reports	findVersionReports
POST	/api/versions/{versionId}/reports	createVersionReport
DELETE	/api/versions/{versionId}/reports/{reportId}	deleteVersionReport
GET	/api/versions/{versionId}/reports/{reportId}	findVersionReport
GET	/api/vulnerability-remediation-reports	findVulnerabilityRemediationReports
POST	/api/vulnerability-remediation-reports	createVulnerabilityRemediationReport
DELETE	/api/vulnerability-reports/{reportId}	deleteVulnerabilityReport
GET	/api/vulnerability-reports/{reportId}	findVulnerabilityReport
GET	/api/vulnerability-status-reports	findVulnerabilityStatusReports
POST	/api/vulnerability-status-reports	createVulnerabilityStatusReport
GET	/api/vulnerability-update-reports	findVulnerabilityUpdateReports
POST	/api/vulnerability-update-reports	createVulnerabilityUpdateReport

**risk-profile-rest-server : Risk Profile Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/catalog-risk-profile-dashboard	getCombinedProjectUsageAndComponentRiskProfiles
GET	/api/risk-profile-dashboard	getProjectDashboardRiskAmalgamation
GET	/api/risk-profile-filters	getLicenseFilter

**role-rest-server : Role Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/roles	getRolesPublic
GET	/api/roles/{roleId}	getRolePublic



**scan-service-proxy-rest-server : Scan Service Proxy Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/codelocations/{codeLocationId}/scan-summaries	findScanSummaries
POST	/api/fp-import/scans/{scanId}	fpImportPost
GET	/api/scan-summaries/{scanId}	findScanSummary
GET	/api/scan/data/	listBdio2
POST	/api/scan/data/	uploadBdio2ViaForm
GET	/api/scan/data/{id}	summarizeBdio2
GET	/api/scan/data/{id}.bdio	download

**search-rest-server : Search Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/search/components	searchByComponentName
-----	------------------------	-----------------------

**source-upload-rest-server : Source Upload Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/projects/{projectId}/versions/{versionId}/scans/{scanId}/source-upload-file-contents/{sha1Id}	findSourceFileContent
-----	--	-----------------------

**source-view-rest-server : Source View Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/projects/{projectId}/versions/{projectVersionId}/snippet-filters	getSnippetFilter
GET	/api/projects/{projectId}/versions/{projectVersionId}/source-view-filters	getSourceViewFilter
GET	/api/projects/{projectId}/versions/{versionId}/scans/{scanId}/nodes/{nodeId}/declared-component-trees	findDeclaredNodeChildren
GET	/api/projects/{projectId}/versions/{versionId}/scans/{scanId}/nodes/{nodeId}/file-trees	findFileNodeChildren
GET	/api/projects/{projectId}/versions/{versionId}/scans/{scanId}/nodes/{nodeId}/snippet-bom-entries	Returns snippets for given node satisfying provided filter conditions
GET	/api/projects/{projectId}/versions/{versionId}/source-trees	Returns source tree nodes for given project version

**tag-rest-server : Tag Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/components/{componentId}/tags	getTagsByComponentId
POST	/api/components/{componentId}/tags	createTagsByCustomComponentId
DELETE	/api/components/{componentId}/tags/{tagId}	deleteCustomComponentTag
GET	/api/components/{componentId}/tags/{tagId}	getCustomComponentTag
GET	/api/projects/{projectId}/tags	getTagsByProjectId
POST	/api/projects/{projectId}/tags	createTagsByProjectId
DELETE	/api/projects/{projectId}/tags/{tagId}	deleteProjectTag
GET	/api/projects/{projectId}/tags/{tagId}	getTag



**url-directory-rest-server : Url Directory Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/	getUrlDirectory
-----	-------	-----------------

**user-filters-rest-server : User Filters Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/user-status-filters	findUserStatusFilterView
-----	--------------------------	--------------------------

**user-group-rest-server : User Group Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/usergroups	page
POST	/api/usergroups	create
DELETE	/api/usergroups/{userGroupId}	delete
GET	/api/usergroups/{userGroupId}	get
PUT	/api/usergroups/{userGroupId}	update
GET	/api/usergroups/{userGroupId}/roles	getRolesByUserGroup
POST	/api/usergroups/{userGroupId}/roles	addRoleForUserGroup
DELETE	/api/usergroups/{userGroupId}/roles/{roleAssignmentId}	deleteRoleAssignmentForUserGroup
GET	/api/usergroups/{userGroupId}/roles/{roleAssignmentId}	getRoleAssignmentForUserGroup
GET	/api/usergroups/{userGroupId}/users	getAssignedUsers
POST	/api/usergroups/{userGroupId}/users	addUsersToUserGroup
DELETE	/api/usergroups/{userGroupId}/users/{userId}	removeUserFromUserGroup
GET	/api/usergroups/{userGroupId}/users/{userId}	getAssignedUser
GET	/api/users/{userId}/usergroups	getUserAssignedGroups
POST	/api/users/{userId}/usergroups	addUserToGroups
DELETE	/api/users/{userId}/usergroups/{userGroupId}	deleteUserFromGroup
GET	/api/users/{userId}/usergroups/{userGroupId}	getSpecificUserAssignedToGroup

**user-notification-filter-rest-server : User Notification Filter Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/notification-state-filters	findNotificationStateFilterView
GET	/api/notification-type-filters	getNotificationTypeFilters

**user-rest-server : User Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/current-user	getCurrentUser
GET	/api/user-status-filters	findUserStatusFilterView
GET	/api/users	findUsers
POST	/api/users	createUser
GET	/api/users-filters	getUserFilter
GET	/api/users/{userId}	findUser
PUT	/api/users/{userId}	updateUser
PUT	/api/users/{userId}/changepassword	changePasswordCurrentUser
PUT	/api/users/{userId}/resetpassword	resetPassword

**user-role-rest-server : User Role Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/roles/{roleId}/inheriting-users	getUsersByInheritedRolePublic
GET	/api/roles/{roleId}/users	getUsersByRolePublic
GET	/api/users/{userId}/inherited-roles	getInheritedRolesByUserPublic
GET	/api/users/{userId}/roles	getRolesByUserPublic
POST	/api/users/{userId}/roles	addRoleForUserPublic
DELETE	/api/users/{userId}/roles/{roleAssignmentId}	deleteRoleAssignmentForUserPublic
GET	/api/users/{userId}/roles/{roleAssignmentId}	getRoleAssignmentForUserPublic

**version-bom-policy-rest-server : Version Bom Policy Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/projects/{projectId}/versions/{versionId}/components/{componentId}/policy-rules	getPolicyRulesForComponentV6
GET	/api/projects/{projectId}/versions/{versionId}/components/{componentId}/policy-status	getPolicyStatusForComponent
GET	/api/projects/{projectId}/versions/{versionId}/components/{componentId}/versions/{componentVersionId}/policy-rules	getPolicyRulesForComponentVersionV6
GET	/api/projects/{projectId}/versions/{versionId}/components/{componentId}/versions/{componentVersionId}/policy-status	getPolicyStatusForComponentVersion
GET	/api/projects/{projectId}/versions/{versionId}/policy-rules	getPolicyRulesForRelease
GET	/api/projects/{projectId}/versions/{versionId}/policy-status	findScanSummary

**version-bom-status-rest-server : Version Bom Status Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/projects/{projectId}/versions/{projectVersionId}/bom-status	findVersionBomStatus
-----	--	----------------------

**version-risk-profile : Version Risk Profile**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/projects/{projectId}/versions/{versionId}/risk-profile	find version risk-profile
-----	---	---------------------------

**vuln-db-report-db-copy-rest-server : Vuln Db Report Db Copy Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)**POST** /api/vulnerabilities/vulndb-copy[createCopy](#)**vulnerability-rest-server : Vulnerability Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)**GET** /api/components/{componentId}/versions/{versionId}/origin/{originId}/vulnerabilities[Finds vulnerabilities by component version and origin](#)**GET** /api/components/{componentId}/versions/{versionId}/vulnerabilities[Finds vulnerabilities by component version](#)**GET** /api/components/{componentId}/vulnerabilities[Finds vulnerabilities by component](#)**GET** /api/cwes/{cwelId}[Finds a specific CWE \(common weakness enumeration\) record](#)**GET** /api/vulnerabilities/{vulnerabilityId}[Finds a specific vulnerability record](#)**vulnerable-component-rest-server : Vulnerable Component Rest Server**[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)**GET** /api/projects/{projectId}/versions/{versionId}/vulnerable-bom-components[findVulnerableComponents](#)