

Getting Started with the Black Duck API

Black Duck SCA 2026.7.2

Copyright ©2026 by Black Duck.

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.

03-11-2026

Contents

Preface....4
Black Duck documentation....4
Customer support....4
Black Duck Community....5
Training....5
Black Duck Statement on Inclusivity and Diversity....5
Black Duck Security Commitments....6

1. Getting Started with the Black Duck API....7

2. Authenticating with the API....8

3. Using the right media types....10

4. Using the Black Duck API with Postman....12

5. Using the Black Duck API with OpenAPI....14

6. API list....15

Preface

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 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.
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.

The installation methods for installing Black Duck software in a Kubernetes or OpenShift environment are Helm. Click the following links to view the documentation.

- Helm is a package manager for Kubernetes that you can use to install Black Duck. Black Duck supports Helm3 and the minimum version of Kubernetes is 1.13.

Black Duck integration documentation is available on:

- https://sig-product-docs.blackduck.com/bundle/detect/page/integrations/integrations.html
- https://docs.blackduck.com/p/bridgeintegrations

Customer support

If you have any problems with the software or the documentation, please contact Black Duck Customer Support:

- Online: https://community.blackduck.com/s/contactsupport
- To open a support case, please log in to the Black Duck Community site at https://community.blackduck.com/s/contactsupport.
- Another convenient resource available at all times is the online Community portal.

Black Duck Community

The Black Duck 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 Black Duck 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 Black Duck 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 Black Duck 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 Black Duck experts and our Knowledgebase.
- Share – Collaborate and connect with Black Duck 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@blackduck.com.

Training

Black Duck Customer Education 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.

In Black Duck Education, 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://blackduck.skilljar.com/page/black-duck or for help with Black Duck, select **Black Duck Tutorials** from the Help menu () in the Black Duck UI.

Black Duck Statement on Inclusivity and Diversity

Black Duck is committed to creating an inclusive environment where every employee, customer, and partner feels welcomed. We are reviewing and removing exclusionary language from our products and supporting customer-facing collateral. Our effort also includes internal initiatives to remove biased language from our

engineering and working environment, including terms that are embedded in our software and IPs. At the same time, we are working to ensure that our web content and software applications are usable to people of varying abilities. You may still find examples of non-inclusive language in our software or documentation as our IPs implement industry-standard specifications that are currently under review to remove exclusionary language.

Black Duck Security Commitments

As an organization dedicated to protecting and securing our customers' applications, Black Duck is equally committed to our customers' data security and privacy. This statement is meant to provide Black Duck customers and prospects with the latest information about our systems, compliance certifications, processes, and other security-related activities.

This statement is available at: Security Commitments | Black Duck

1. Getting Started with the Black Duck API

The Black Duck REST API allows you to programmatically interact with your Black Duck instance—enabling automation, custom integrations, and enhanced visibility into your security, license, and component data.

Whether you're writing scripts, building tools, or connecting Black Duck with other systems, this guide helps you get started with the core API functionality.

Accessing the API documentation

You can explore the available endpoints and try them out in your browser using the built-in API documentation.

To access it:

- Go to `https://<your-black-duck-server>/api-doc/public.html`
- Or, in the top navigation bar, click the Help icon  and select **REST API Developers Guide**.

This interactive documentation shows each supported endpoint, the required parameters, response structure, and supported media types.

Important: Only APIs documented in this guide are supported. Endpoints that begin with `/v1` or `internal` are not supported for external use.

2. Authenticating with the API

Black Duck uses API tokens for authentication. This approach improves security and makes it easier to integrate with external tools.

Step 1: Generate an API token

1. Log into the Black Duck UI.
2. From the top-right menu, select **System** → **Access Tokens**.
3. Click **Create New Token** and follow the prompts.
   - Name the token
   - (Optional) Add a description
   - Choose **Read Access Only** or **Read and Write Access**
   - Click **Create**.
4. Save the token somewhere safe. For security reasons, it will only be shown once.

Step 2: Exchange the token for a bearer token

Make an HTTP `POST` request to:

```
/api/tokens/authenticate
```

Include your API token in the Authorization header:

```
curl -X POST \
    https://<your-black-duck-server>/api/tokens/authenticate \
    -H "Accept: application/vnd.blackducksoftware.user-4+json" \
    -H "Authorization: token <your-api-token>"
```

This returns a **Bearer token**, which you use to authorize all subsequent API requests.

Step 3: Use the Bearer token in your requests

Example:

```
curl -X GET \
    https://<your-black-duck-server>/api/projects/ \
    -H "Authorization: Bearer <your-bearer-token> \
    -H "Accept: application/vnd.blackducksoftware.project-detail-4+json" \
```

Changing the expiration time for a bearer token

To extend the expiration time of a bearer token used in REST API, use the `docker-compose.local-overrides.yml` file to override the default setting by configuring the `HUB_AUTHENTICATION_ACCESS_TOKEN_EXPIRE` environment variable with the new expiration value in seconds.

The `HUB_AUTHENTICATION_ACCESS_TOKEN_EXPIRE` property is the number of seconds that the access tokens take to expire.

Note: The expiration configuration change only works for API tokens that are created after you change the setting in the `docker-compose.local-overrides.yml` file. The expiration time that you configure isn't updated for existing database records/API tokens when the setting is changed and the service is restarted.

3. Using the right media types

The Black Duck API uses **custom media types** to ensure stable and versioned responses.

Specify the media type in the `Accept` and `Content-Type` headers of your request. This ensures backward compatibility and consisten reponses.

Example:

```
Content-Type: application/vnd.blackducksoftware.user-4+json
Accept: application/vnd.blackducksoftware.user-4+json
```

Note: You can find the recommended media types listed next to each endpoint in the REST API documentation.

Why media types matter

Media types (also known as MIME types) are used in API requests and responses to specify the format of the data being exchanged. In Black Duck, media types serve an additional, important purpose: they ensure backward compatibility and versioned stability of the API.

Each API endpoint may support one or more custom Black Duck media types. By explicitly specifying a media type in your request headers, you ensure that:

- You receive a predictable and structured response.
- Your integration continues to work even if the underlying API evolves.
- The correct version of a resource is returned, especially if multiple versions exist.

Failing to set the correct media type (e.g., using `application/json` as a generic fallback) may result in unsupported behavior, incomplete data, or even failed requests—especially for endpoints that return complex or versioned resources.

Tip: You can find the correct media type for each endpoint in the REST API documentation.

Examples of using media types

Here are some practical examples of how to use media types when working with the Black Duck API.

Example 1: Setting Accept and Content-Type headers

When making a `PUT` or `POST` request with a request body, you typically include both `Content-Type` and `Accept` headers.

```
PUT /api/users/{userId}
Content-Type: application/vnd.blackducksoftware.user-4+json
Accept: application/vnd.blackducksoftware.user-4+json
```

This tells the server:

- The request body is formatted according to `user-4+json`
- You expect the response to also be in that format

Example 2: GET request for a project

```
curl -X GET \
    https://<your-black-duck-server>/api/projects
    -H "Accept: application/vnd.blackducksoftware.project-detail-7+json
```

```
    -H "Authorization: Bearer <your-bearer-token>
```

This returns a list of projects in a versioned, structured JSO format that the client knows how to handle.

Example 3: Receiving a BDIO ZIP

Some endpoints, like those related to data import/export, return or accept binary data (e.g., BDIO):

```
application/vnd.blackducksoftware.bdio+zip
```

You would use this media type when downloading or uploading BDIO scan archives.

4. Using the Black Duck API with Postman

Postman is a popular tool for exploring and testing APIs without writing any code. Black Duck provides an official Postman collection that you can import and use to try out API endpoints with your own instance.

Why use Postman?

Postman makes it easy to:

- Explore and test API endpoints in a visual interface
- Authenticate using API tokens
- Send requests and view responses without writing scripts
- Save common requests as part of a reusable collection

It's especially helpful if you're new to the API or just want to experiment with functionality before automating anything.

Importing the Black Duck Postman collection

1. Open a browser and log in to your Black Duck instance.
2. Navigate to the following URL, replacing the server name as needed:

   ```
   https://<your-black-duck-server>/api-doc/postman-collection-public.json
   ```

3. Right-click anywhere on the page and Save As `postman-collection-public.json`.
4. Open Postman and import the saved file:
   - Clicking Import int he top-left corner of Postman.
   - Choose the file you just saved.
5. The collection will now appear in your Postman sidebar, organized into folders by endpoint category.

Setting up authentication in Postman

To use the API, you'll need to authenticate using an API token. This process involves generating a bearer token and using it in your requests.

In Postman, create a new request:

- Method: `POST`
- URL: `https://<your-black-duck-server>/api/tokens/authenticate`
- Headers:
  - `Accept: application/vnd.blackducksoftware.user-4+json`
  - `Authorization: token <your-api-token>`

Copy the `bearerToken` from the response. You can now use the token for all future API calls:

- In Postman, go to your collection or request.
- Under the Authorization tab, set:
  - Type: `Bearer Token`

- Token: Paste your bearer token

Alternatively, use the `Authorization: Bearer <token>` header manually in each request.

Making your first API request

With authentication configured, try a simple request like retrieving all projects:

- Method: `GET`
- URL: `https://<your-black-duck-server>/api/projects`
- Header: `Accept: application/vnd.blackducksoftware.project-detail-7+json`

Hit **Send**, and you should see the response from your Black Duck instance in the lower pane.

Fastpath:

For more information on how to authenticate, see Authenticating with the API.

To dive deeper into request structure, see Using the right media types.

5. Using the Black Duck API with OpenAPI

For users using OpenAPI Specification (OAS), you can generate customer-facing endpoints through `/api-doc/openapi3-public.json`.

1. Log into Black Duck.
2. Open a browser tab and paste the following URL using you Black Duck server address.

   `https://<your_black_duck_server>/api-doc/openapi3-public.json`
3. On the page that's generated, right-click and save as `openapi3-public.json`
4. Import the saved `openapi3-public.json` into your application.

6. API list

For more information on API requests, please refer to the REST API Developers Guide available in Black Duck.