TECHNICAL DOCUMENTATION

Dashboard

Complete documentation.
One clear reference.

Version 1.3.1

Contents

Dashboard **5**
Documentation 7
Licensing **7**
License Comparison 7
Community License 8
Commercial License 12
Deployment **12**
Running Dashboard 12
Running a Specific Dashboard Version 12
Update Dashboard 13
Authentication 13
Projects, Users and Roles 13
Systemd 13
Workers 14
Proxy 14
Backup 14
Restore 15
Security **15**
TLS 15
Password Storage 15
Reset Root Password 15
Login **16**
Overview Page **16**
Project Page **17**
Datepicker 19
Statistics 19
Test Results 19
Requirements Coverage 21
Badges 23
Requirements Coverage **24**
Adding Requirements using HILSTER Testing Framework 24
Adding Requirements with other Tools 25
Submitting Requirements 25
Command Line Interface **26**
Installation 26
Run CLI in Docker Compose 27
Authentication Options 27
Server Options 28
Verbosity 28

Session Storage ... 29
TLS/SSL Certificate Verification ... 29
User and Role Overview ... 29
User Management ... 30
Projects ... 33
Reports ... 35
Tokens ... 36
Requirements ... 38
Cache ... 40
Bash-Completion ... 40
Enable Bash-Completion ... 40

QABench htf Integration ... 41

API Endpoints ... 41
Authentication ... 42
Projects ... 42
Project Information ... 43
Reports ... 44
Requirements ... 45
Tokens ... 47
Cache ... 48
User Management ... 48

Cyber Resilience Act Conformity ... 50
Statement of Conformity ... 51
Security Measures ... 51
Reporting Vulnerabilities ... 51

Changelog ... 51
dashboard-1.3.1: ... 51
dashboard-1.3.0 ... 52
dashboard-1.2.5 ... 52
dashboard-1.2.4 ... 52
dashboard-1.2.3 ... 52
dashboard-1.2.2 ... 52
dashboard-1.2.1 ... 52
dashboard-1.2.0 ... 52
dashboard-1.1.0 ... 53
dashboard-1.0.1 ... 53
dashboard-1.0.0 ... 53

Open-Source Licenses ... 54
Bootstrap ... 54
Chart.js ... 54
Datatables ... 55
Daterangepicker ... 55
Fontawesome-free ... 55
jQuery ... 56
jQuery Easing ... 57
Moment.js ... 58

Patternomaly 58
SB-Admin-2 59

Dashboard

The QABench Dashboard (formerly known as HILSTER Dashboard) is an interactive analytics and test-management tool for test results during development. It measures test results and requirements coverage and generates informative and meaningful charts.

Fully integrated into QABench htf, test reports can be sent automatically to the Dashboard upon conclusion of a test run.

Additionally, JUnit-XML compatible reports generated by other testing suites can also be uploaded to allow for a comprehensive overview of the project's current test status.

It is also possible to measure requirements coverage over varying test levels and different tools with the Dashboard.

It serves users in both managing and developing roles, showing the current project status as well as trends over time.

Project requirements can be supplied to the Dashboard, enabling users to see whether requirements are already covered by tests.



Dashboard browser overview

Download recording · Download transcript

```
Dashboard browser overview
==========================

Visual transcript and walkthrough of the silent recording.
Based on reviewed screen captions/interface actions and the associated written example.

Original recording: dashboard_overview.mp4

1. The recording starts with the Dashboard project list and its report counts, test-
   result summaries, requirements coverage, and last-modified column.

2. Opening a project displays test-result and requirements-coverage timelines.

3. Recent-report doughnut charts summarize test outcomes and requirements coverage.
   Pointing at chart segments exposes their values.

4. The project page also lists individual requirements and their coverage status. The
   associated Dashboard documentation explains these views in detail.
```

The QABench Dashboard can be used for free with the community license or you can get a commercial license.

Documentation

Licensing

QABench Dashboard is offered as part of a community license and a commercial license.

The community licenses allows private persons to use QABench Dashboard for free for up to 10 projects and allows up to 3 users in a commercial context.

If you need more than three users please get a commercial license.

License Comparison

	Community License	Commercial License
Use for free in private context	Yes	No
Use for free in commercial context	Yes, but only for up to 3 users and 10 projects	No
Number of users	Up to 3 users and 10 projects	Depends on the license

Community License

Please also note our End-User License Agreement (EULA) and our Privacy Policy.

Note

End-User License Agreement (EULA) of QABench Dashboard

This End-User License Agreement (“EULA”) is a legal agreement between you and HILSTER

This EULA agreement governs your acquisition and use of our QABench Dashboard software (“Software”) directly from HILSTER or indirectly through a HILSTER authorized reseller or distributor (a “Reseller”).

Please read this EULA agreement carefully before completing the installation process and using the QABench Dashboard software. It provides a license to use the QABench Dashboard software and contains warranty information and liability disclaimers.

If you register for a free trial of the QABench Dashboard software, this EULA agreement will also govern that trial. By clicking “accept” or installing and/or using the QABench Dashboard software, you are confirming your acceptance of the Software and agreeing to become bound by the terms of this EULA agreement.

If you are entering into this EULA agreement on behalf of a company or other legal entity, you represent that you have the authority to bind such entity and its affiliates to these terms and conditions. If you do not have such authority or if you do not agree with the terms and conditions of this EULA agreement, do not install or use the Software, and you must not accept this EULA agreement.

This EULA agreement shall apply only to the Software supplied by HILSTER herewith regardless of whether other software is referred to or described herein. The terms also apply to any HILSTER updates, supplements, Internet-based services, and support services for the Software, unless other terms accompany those items on delivery. If so, those terms apply. This EULA was created by EULA Template for QABench Dashboard.

License Grant

HILSTER hereby grants you a personal, non-transferable, non-exclusive licence to use the QABench Dashboard software on your devices in accordance with the terms of this EULA agreement.

You are permitted to load the QABench Dashboard software (for example a PC, laptop, mobile or tablet) under your control. You are responsible for ensuring your device meets the minimum requirements of the QABench Dashboard software.

You are not permitted to:

- Edit, alter, modify, adapt, translate or otherwise change the whole or any part of the Software nor permit the whole or any part of the Software to be combined with or become incorporated in any other software, nor decompile, disassemble or reverse engineer the Software or attempt to do any such things
- Reproduce, copy, distribute, resell or otherwise use the Software for any commercial purpose for more than 3 users
- Allow any third party to use the Software on behalf of or for the benefit of any third party
- Use the Software in any way which breaches any applicable local, national or international law

- use the Software for any purpose that HILSTER considers is a breach of this EULA agreement

Intellectual Property and Ownership

HILSTER shall at all times retain ownership of the Software as originally downloaded by you and all subsequent downloads of the Software by you. The Software (and the copyright, and other intellectual property rights of whatever nature in the Software, including any modifications made thereto) are and shall remain the property of HILSTER.

HILSTER reserves the right to grant licences to use the Software to third parties.

Termination

This EULA agreement is effective from the date you first use the Software and shall continue until terminated. You may terminate it at any time upon written notice to HILSTER.

It will also terminate immediately if you fail to comply with any term of this EULA agreement. Upon such termination, the licenses granted by this EULA agreement will immediately terminate and you agree to stop all access and use of the Software. The provisions that by their nature continue and survive will survive any termination of this EULA agreement.

Governing Law

This EULA agreement, and any dispute arising out of or in connection with this EULA agreement, shall be governed by and construed in accordance with the laws of de.

Note

Privacy Policy

At HILSTER License Server, accessible from https://licenses.hilster.io, one of our main priorities is the privacy of our visitors. This Privacy Policy document contains types of information that is collected and recorded by HILSTER License Server and how we use it.

If you have additional questions or require more information about our Privacy Policy, do not hesitate to contact us.

General Data Protection Regulation (GDPR)

We are a Data Controller of your information.

HILSTER legal basis for collecting and using the personal information described in this Privacy Policy depends on the Personal Information we collect and the specific context in which we collect the information:

HILSTER needs to perform a contract with you You have given HILSTER permission to do so Processing your personal information is in HILSTER legitimate interests HILSTER needs to comply with the law HILSTER will retain your personal information only for as long as is necessary for the purposes set out in this Privacy Policy. We will retain and use your information to the extent necessary to comply with our legal obligations, resolve disputes, and enforce our policies. Our Privacy Policy was generated with the help of GDPR Privacy Policy Generator and the GDPR Privacy Policy Generator.

If you are a resident of the European Economic Area (EEA), you have certain data protection rights. If you wish to be informed what Personal Information we hold about you and if you want it to be removed from our systems, please contact us.

In certain circumstances, you have the following data protection rights:

- The right to access, update or to delete the information we have on you.
- The right of rectification.
- The right to object.
- The right of restriction.
- The right to data portability
- The right to withdraw consent

Log Files

HILSTER License Server follows a standard procedure of using log files. These files log visitors when they visit websites. All hosting companies do this and a part of hosting services' analytics. The information collected by log files include internet protocol (IP) addresses, browser type, Internet Service Provider (ISP), date and time stamp, referring/exit pages, and possibly the number of clicks. These are not linked to any information that is personally identifiable. The purpose of the information is for analyzing trends, administering the site, tracking users' movement on the website, and gathering demographic information.

Consent

By using the Community License, you hereby consent to our Privacy Policy and agree to its terms.

Commercial License

To get a commercial license, please contact us via e-mail.

Deployment

The Dashboard server is designed to be run as a number of Docker containers using Docker Compose.

All files needed for deployment are located on our GitHub.

Clone the repository with:

```
git clone https://github.com/hilstertestingsolutions/dashboard-deployment.git
```

Navigate to the folder containing the `docker-compose.yml`. Edit `docker-compose.yml` according to your needs.

Warning

By default the activation key for the community license is used. If you need a commercial license for your Dashboard, please contact us to get one.

Running Dashboard

Start the Dashboard server using:

```
docker compose up
```

This will collect all the necessary containers from Dockerhub and start the Dashboard server.

Running a Specific Dashboard Version

To run a specific Dashboard version, modify the following line in `docker-compose.yml` by replacing `latest` with your desired version.

```
image: hilster/dashboard:latest
```

For example, to run `Dashboard-1.2.5` use:

```
image: hilster/dashboard:1.2.5
```

Update Dashboard

To use the to the newest version of Dashboard, run

```
docker compose pull
```

Authentication

On the initial run with an empty database, a user `root` with password `root` is created automatically.

You should immediately change the password using the command line interface by running

```
dashboard -u root -p root https://dashboard.server passwd
```

Read the command line interface documentation to learn how to add other users, projects and roles.

Learn how to reset root's password.

Projects, Users and Roles

You can create projects and users in the Dashboard, and assign roles for users in those projects. The available roles are `user` and `admin`.

Users with the role `user` can view all data of projects the role is applied for, but they're not allowed to upload requirements, delete reports or manage tokens.

Users with the `admin` role have full access to a project.

Systemd

On Linux `systemd` is a popular daemon to start and stop services.

To run Dashboard as a systemd service, create and edit `dashboard.service` to fit your needs.

```
[Unit]
Description=Dashboard
Requires=docker.service
After=docker.service

[Service]
WorkingDirectory=/services/dashboard
ExecStart=/usr/bin/docker compose up
ExecStop=/usr/bin/docker compose down
TimeoutStartSec=0
```

```
Restart=always
RestartSec=20s

[Install]
WantedBy=multi-user.target
```

To install the service, run

```
cp dashboard.service /etc/systemd/system/dashboard.service
systemctl daemon-reload
systemctl enable dashboard.service
service dashboard start
```

To view logs of Dashboard running as a service, use

```
journalctl -u dashboard.service -f
```

Workers

The number of workers for the front-end and back-end can be altered in `dashboard/dashboard.yml` by changing `workers` to an appropriate value.

In the back-end, a value of `workers: 0` means that one worker is started per cpu. All other positive integer numbers state the actual number of processes to be started.

Proxy

`nginx` is used as a proxy server. You can change `nginx/default.conf` to meet your needs, e.g. add SSL certificates, etc.

Backup

To create a backup of the Dashboard database, run the following command while the server is running:

```
./backup_database.sh
```

or manually

```
docker compose exec mongodb mongodump
```

The database contents will be dumped to the `data/dump` folder.

Restore

To restore the Dashboard database from a backup, run the following command while the server is running:

```
./restore_database.sh
```

or manually

```
docker compose exec mongodb mongorestore
```

The database contents will be restored from the `data/dump` folder.

Security

TLS

You should consider running the dashboard using TLS (SSL) security layer.

Password Storage

All passwords are stored as `bcrypt hashes`. At this point in time, no rainbow-table and brute-force attacks are available against `bcrypt`.

Reset Root Password

To reset the `root` password you need to have access to the machine running the dashboard. Run

```
dashboard-admin set-root-password [--password <password>]
```

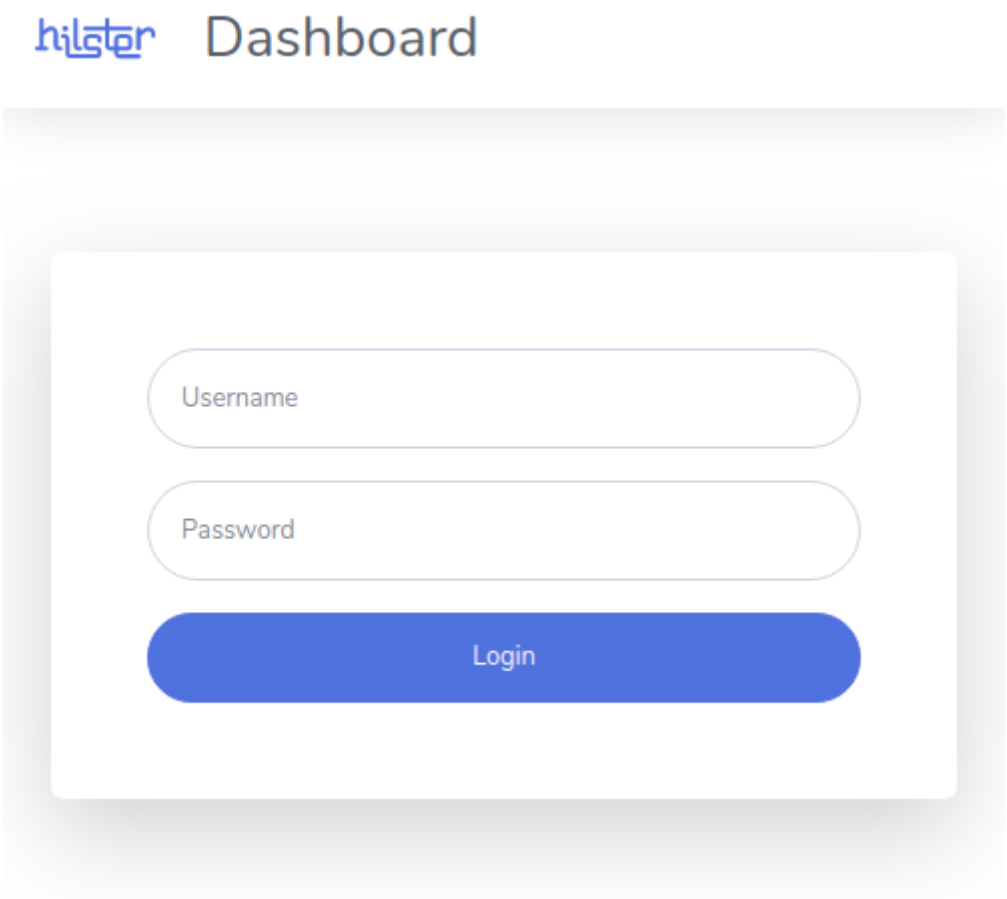
in Docker for example:

```
sudo docker compose exec dashboard dashboard-admin set-root-password --password root
```

If no `<password>` is supplied, a password is generated and printed to the terminal.

Login

To access overview and project, pages you have to login first.



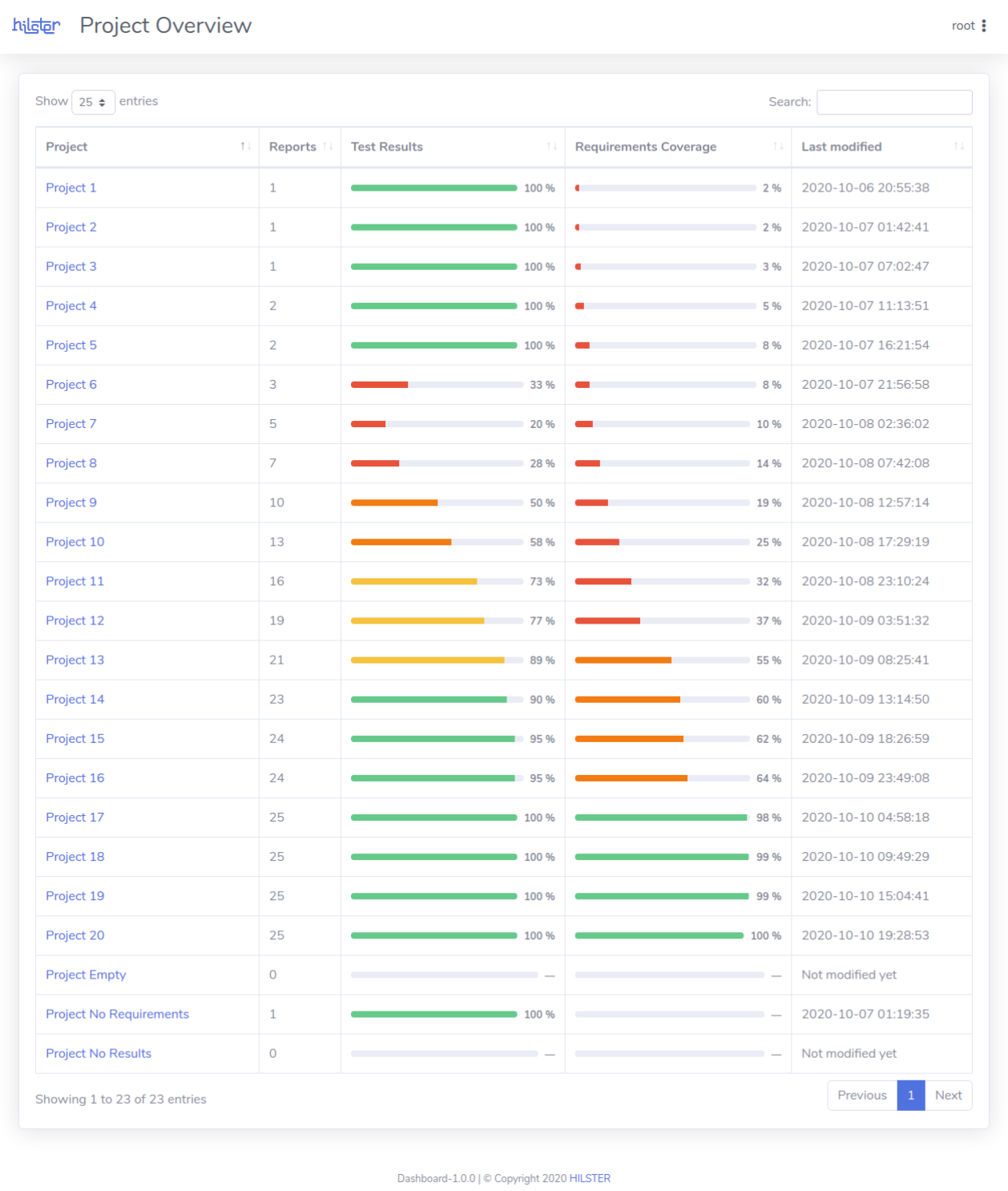
Login page.

Overview Page

The overview page shows the user's projects in a table with the following contents:

- Name of the project (The alias given during project creation)
- Number of submitted test reports
- Percentage of successful tests in the most recent report
- Percentage of successfully covered requirements in the most recent report
- Last modified date

They serve as to give a fast outline for each individual project.



Sample overview page of the Dashboard.

Project Page

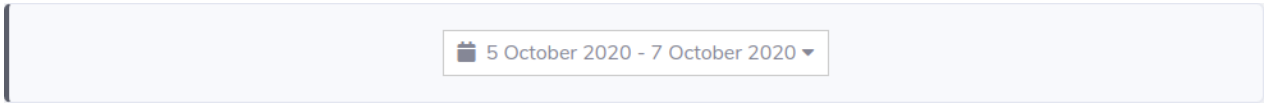
The project view page shows all information for a specific project.



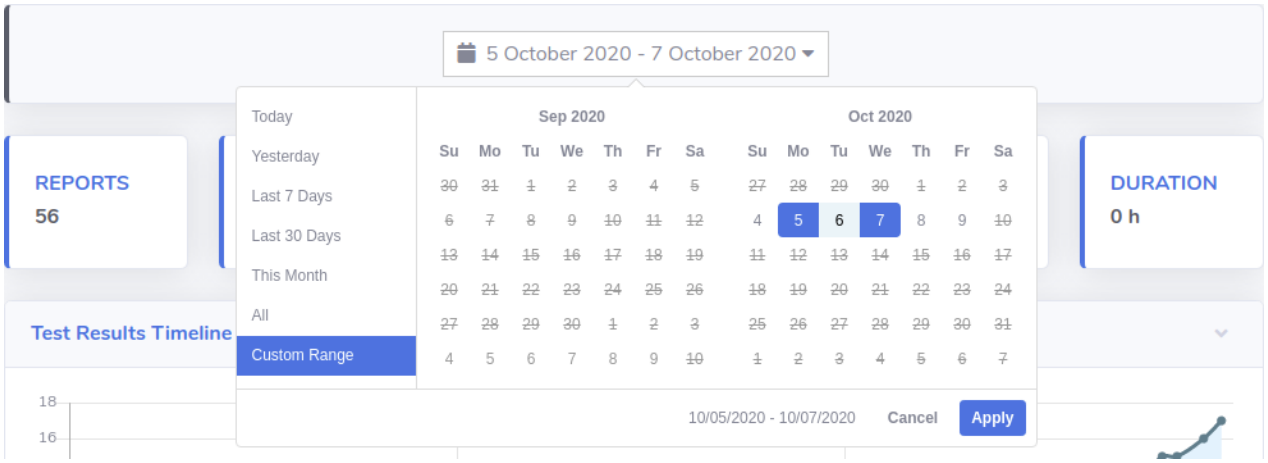
Project view

Datepicker

You can select the visible date range using the datepicker by clicking on it. The project page reloads automatically and show the desired results within the selected date range.



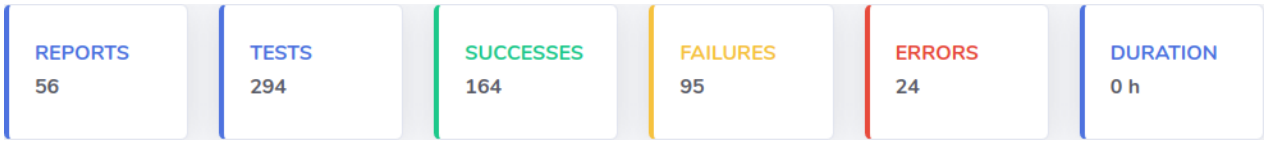
Datepicker



Expanded Datepicker

Statistics

The statistics callouts contain important statistics about the project.



Statistic Callouts

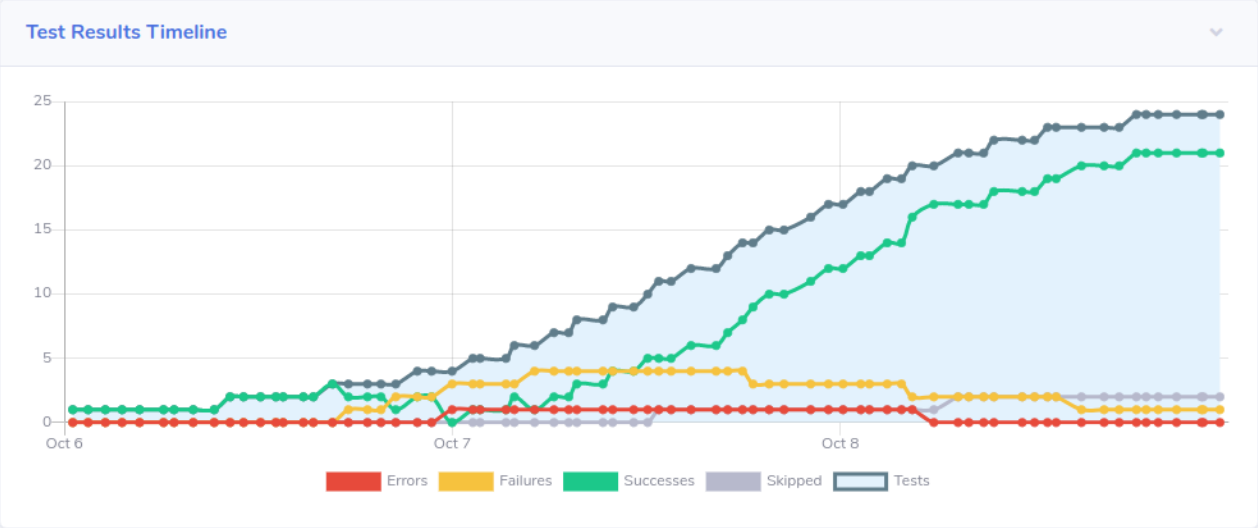
Test Results

The Dashboard tracks test results and generates meaningful charts.

Test Results Timeline

The test results timeline shows the submitted test results over the time.

The chart contains successful, failed, defective and skipped tests as well as the total number of tests.

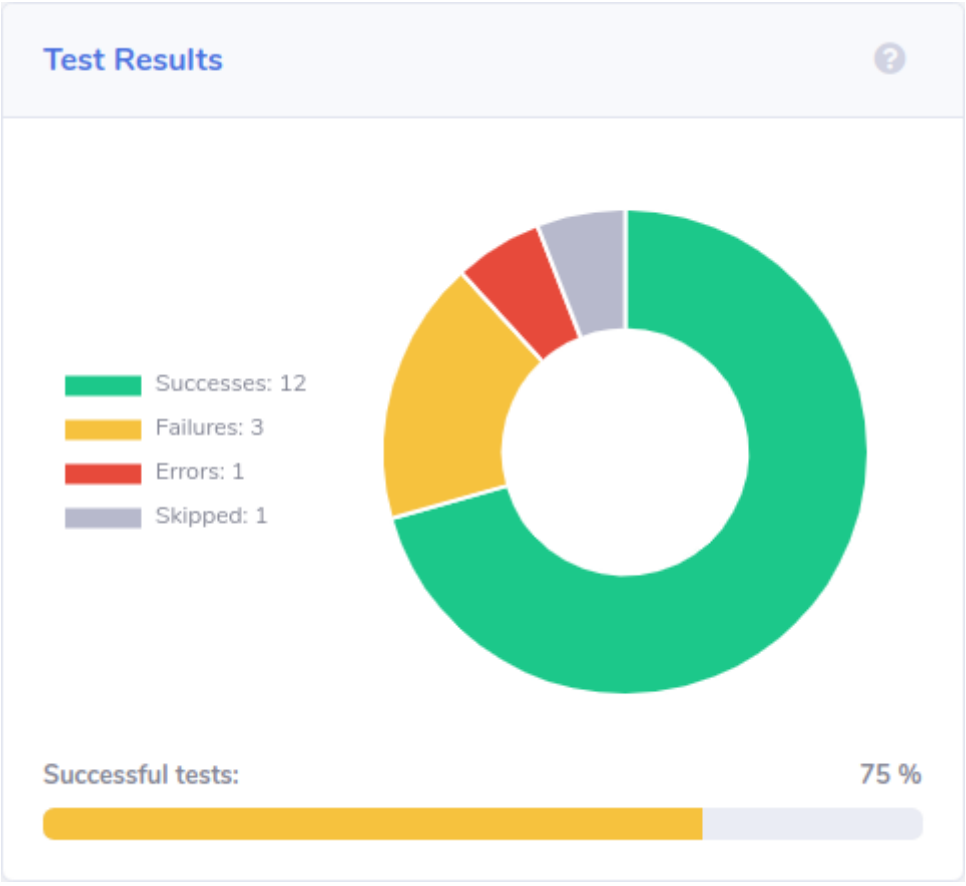


Test Results Timeline

Test Results Doughnut

A test results doughnut chart is generated for the most recent test report. It gives an overview about the latest test result as well as the percentage of successful tests.

The percentage is the ratio of the successful tests to the number of total run tests, excluding skipped tests.



Test Results Doughnut

Requirements Coverage

The Dashboard also generates meaningful charts and tables for each project that tracks requirements.

Requirements Coverage Timeline

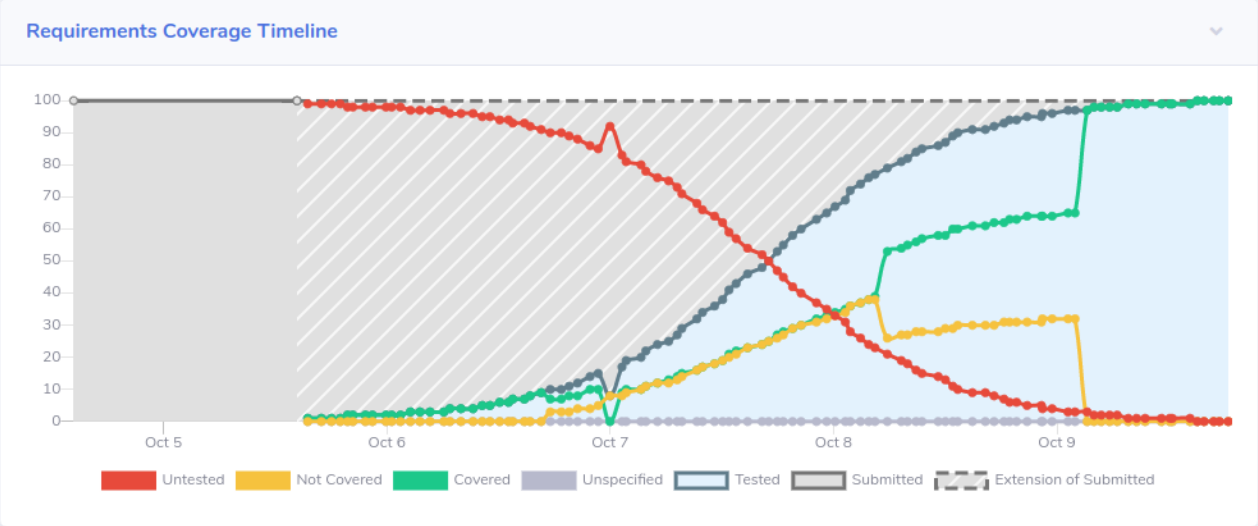
The Requirements Coverage Timeline shows the submitted requirements over the time (gray) and the requirements coverage over the time. The latest submitted requirements are extended for a better overview (gray hatched).

Covered requirements are requirements that are submitted and tested successfully.

Not covered requirements are requirements that are submitted and have at least one failing test.

Untested requirements are submitted requirements that are not tested yet.

Unspecified requirements are requirements that are tested, but not yet submitted.

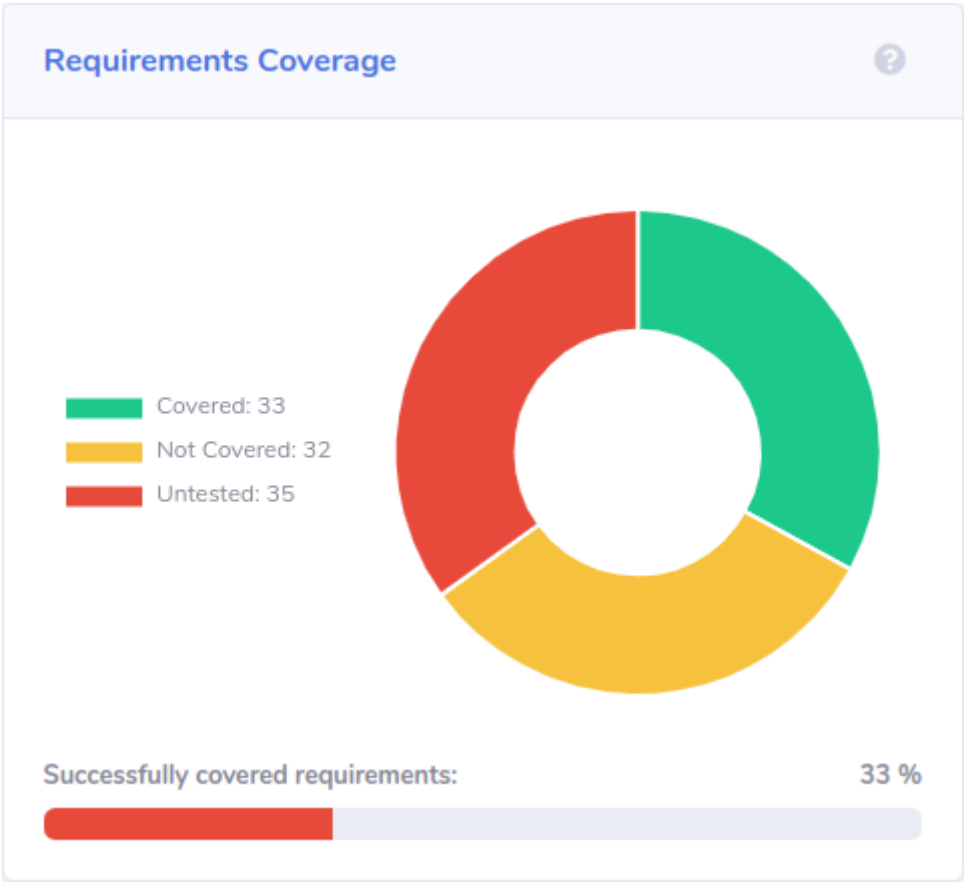


Requirements Coverage Timeline

Requirements Coverage Doughnut

A requirements coverage doughnut chart is generated for the most recent test report. It shows an overview about the latest requirement coverage state, as well as the percentage of the requirements coverage.

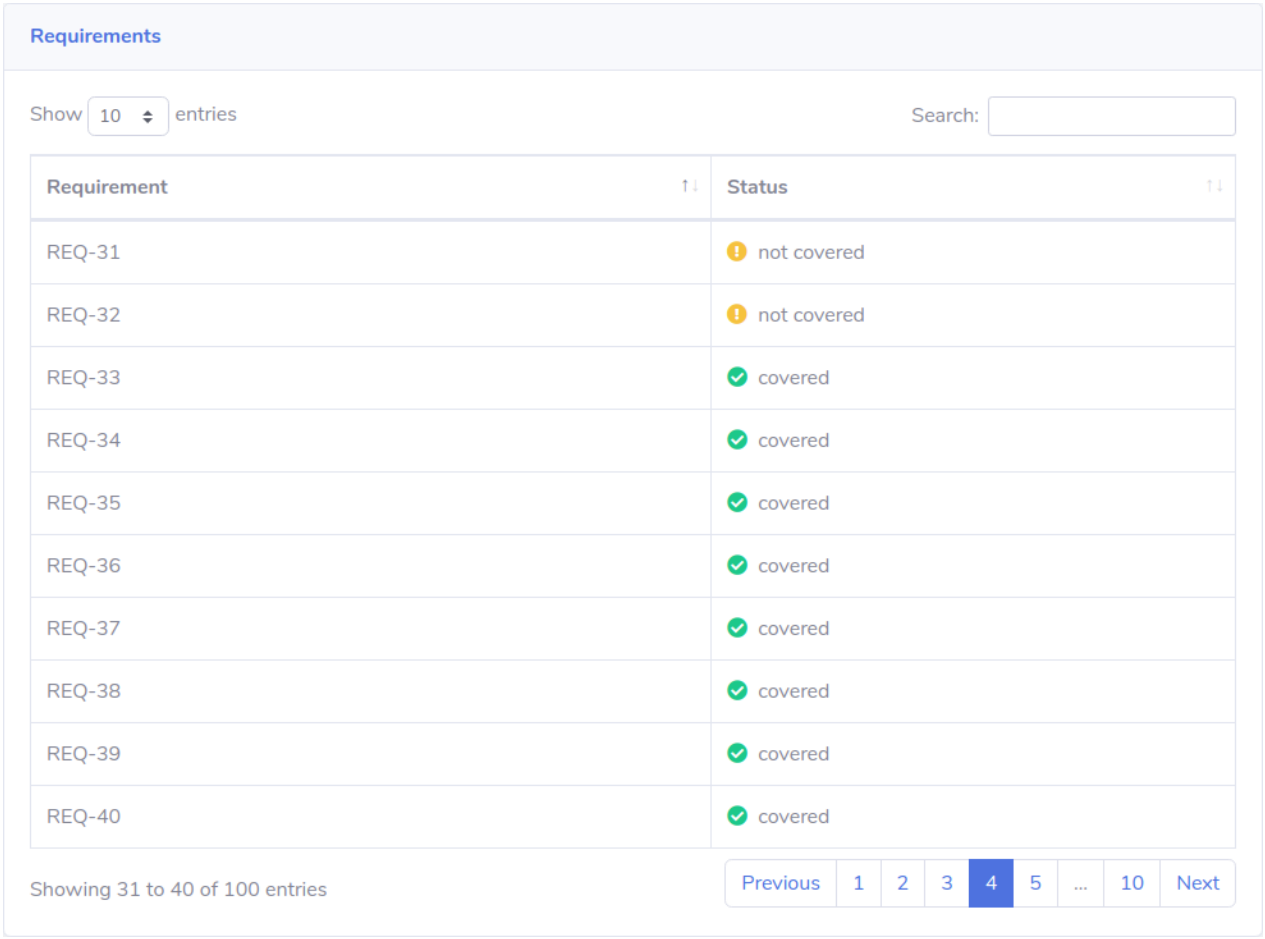
The percentage of the requirements coverage is the ratio of covered requirements to submitted requirements.



Requirements Coverage Doughnut

Requirements Coverage Table

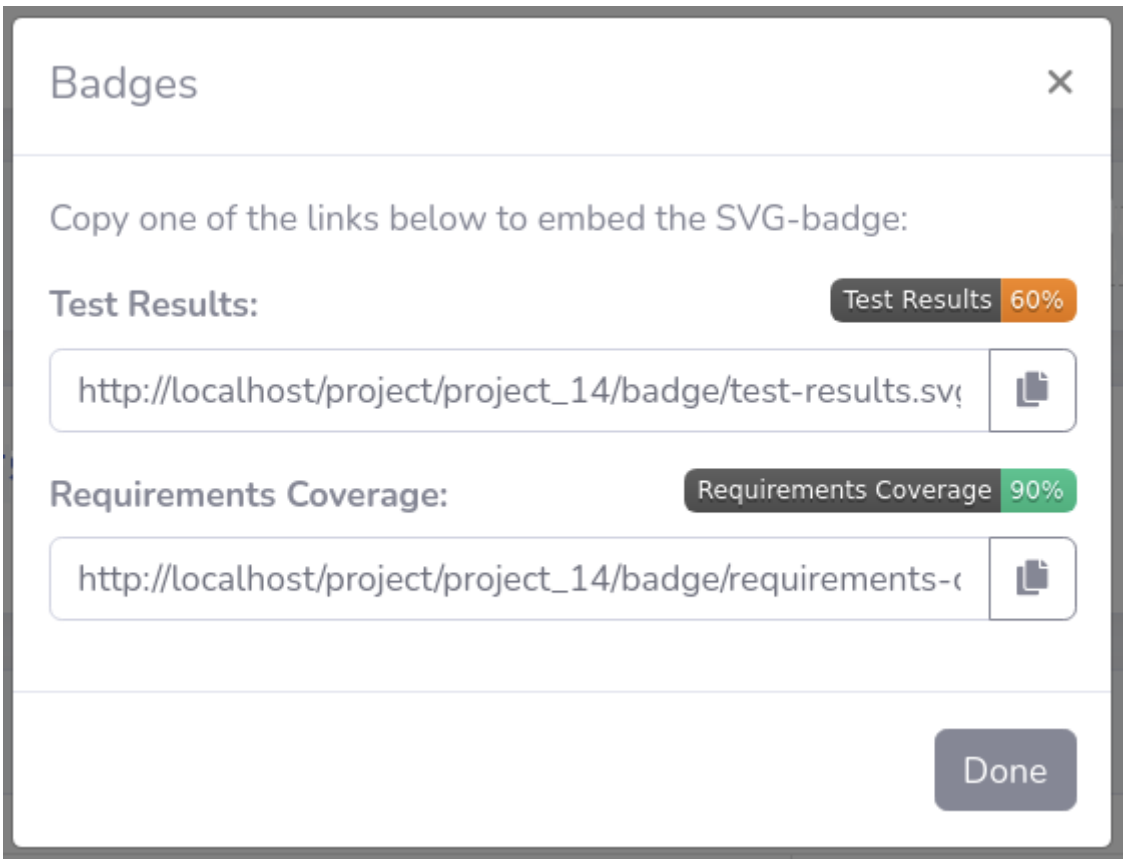
The requirements coverage table shows the coverage state of all submitted and tested requirements.



Requirements Coverage Table

Badges

The Dashboard offers an easy way to visualize the status of your most recent test report as an embeddable SVG. Available badges are listed in the Badges-Submenu located in Three-Dot-Menu in the top right.



Embeddable Badges

Requirements Coverage

The Dashboard is able to track requirements and create requirements coverage graphs. You can test requirements in different ways.

Adding Requirements using HILSTER Testing Framework

Requirements can be added to tests using HILSTER Testing Framework's `htf.requirements()` decorator.

```
@htf.requirements('REQ-1', 'REQ-2', 'REQ-3')
def test_with_requirements(step):
    with step("This is the first test step"):
        pass
```

Adding Requirements with other Tools

Alternatively requirements can be printed to stdout and stderr so that requirements can be extracted from all types of test reports, i.e. JUnit-XML, etc. Thus the Dashboard allows requirements coverage with any kind of testing framework that supports JUnit-XML reports.

Printed requirements must fulfill the following rules:

- Each requirement is printed on a single line.
- Each line containing a requirement contains

  `REQUIREMENT`,

  `REQ` or

  `FULFILLS`, followed by a colon (`:`) and

  - at least one space followed by the requirement name itself.
- Each requirement name consists of at least one characters out of `A-Z`, `0-9`, `-`, `_` and `.`.
- Requirements are case-insensitive.

```
def test_with_printed_requirementsref():
    print("REQ: REQ-1")
    print("REQUIREMENT: REQ-2")
    print("FULLFILLS: REQ-3")
```

Submitting Requirements

Requirements can be submitted to the dashboard using the command line interface.

Requirement files need to be `csv`-files with an arbitrary header line and the requirement IDs located in the leftmost column. The first line, the header line, is ignored. E.g.:

```
ID, ...
REQ-0, ...
REQ-1, ...
.
.
```

The included requirements in this case would be `REQ-0`, `REQ-1`, `...`.

Multi-line values are supported using `"` as quotechar:

```
ID,Content,Last-Modified
REQ-0,Content for REQ-0,2100-01-01
```

```
REQ-1,Content for REQ-1,2100-01-01
REQ-2,"Content for
REQ-2",2100-01-01
REQ-3,Content for REQ-2,2100-01-01
.
.
```

In the user-interface the column with the longest entry will be used as the content.

Learn how to upload requirements via the command line interface.

Command Line Interface

The Dashboard command line interface allows the user to interact with and control the Dashboard server.

This way the usage of the Dashboard is completely scriptable.

Installation

The Dashboard command line interface is shipped in the Docker image. If you wish to run it elsewhere, you have to install it first.

The Dashboard command line interface is installed with

```
pip install -i https://pypi.hilster.io dashboard-cli
```

To check the correct installation and see all possible commands, type:

```
dashboard -h
```

```
dashboard [-h] [-s SERVER] [-u SERVER_USERNAME] [-p SERVER_PASSWORD] [-t TIMEOUT] [-v]
          {login,logout,add-user,delete-user,passwd,user-info,whoami,set-roles,list-users,create-
project,delete-project,list-projects,list-reports,delete-report,list-requirements,delete-
requirements,create-token,delete-token,list-tokens,delete-tokens,export-project,import-
project,upload-requirements,upload-reports,rebuild-cache}
          ...

QABench Dashboard CLI - https://www.hilster.io

positional arguments:
  {login,logout,add-user,delete-user,passwd,user-info,whoami,set-roles,list-users,create-
project,delete-project,list-projects,list-reports,delete-report,list-requirements,delete-
requirements,create-token,delete-token,list-tokens,delete-tokens,export-project,import-
project,upload-requirements,upload-reports,rebuild-cache}
    login               Login
    logout              Logout
    add-user            Add a user
    delete-user         Add a user
```

```
    passwd              Change password
    user-info           Get user information
    whoami              Get current username
    set-roles           Set user's roles for a project
    list-users          List users
    create-project      Create a project
    delete-project      Delete a project
    list-projects       Show a list of projects
    list-reports        Show a list of reports
    delete-report       Delete a report
    list-requirements   Show a list of requirements
    delete-requirements
                        Delete requirements
    create-token        Create a token
    delete-token        Delete a token
    list-tokens         List project tokens
    delete-tokens       Delete all project tokens
    export-project      Export a project to an archive
    import-project      Import a project from an archive
    upload-requirements
                        Upload requirements
    upload-reports      Upload reports
    rebuild-cache       Rebuild cache

optional arguments:
  -h, --help            show this help message and exit
  -s SERVER, --server SERVER
                        Server with HTTP URI scheme (default: http://localhost)
  -u SERVER_USERNAME, --username SERVER_USERNAME
                        Username
  -p SERVER_PASSWORD, --password SERVER_PASSWORD
                        Password
  -t TIMEOUT, --timeout TIMEOUT
                        Timeout in seconds (default: 30 s)
  -v, --verbose         Verbose output
```

Run CLI in Docker Compose

To use the Dashboard command line interface in Docker Compose, run

```
docker compose exec dashboard-server dashboard -h
```

Authentication Options

The Dashboard and the command line interface support authentication.

You can either login and logout persistently with distinct commands, or add the username and/or password to all commands.

If only the username is supplied, the command line interface interactively asks for the password.

In both cases the login is persistently stored to `~/.dashboard_session.bin`.

The stored login credentials are used automatically until you log out.

For example, to login to a server without running a specific command, use

```
dashboard login
```

This will save an authentication cookie alongside the session locally.

To remove the session data and logout, use

```
dashboard logout
```

To list all available projects with credentials, run

```
dashboard -u <username> -p <password> list-projects
```

Alternatively the environment variables `$DASHBOARD_USERNAME` and `$DASHBOARD_PASSWORD` can be used to set the credentials.

```
DASHBOARD_USERNAME=root DASHBOARD_PASSWORD=root dashboard login
```

Server Options

To set the server via command line, use the `-s` or `--server` options.

```
dashboard --server https://dashboard.server list-projects
```

You can alternatively set an environment variable to set the server url.

```
DASHBOARD_SERVER=https://dashboard.server dashboard list-projects
```

Verbosity

To enable verbose output, use the `-v` or `--verbose` options.

```
dashboard --verbose whoami
```

You can alternatively set an environment variable to enable verbosity.

```
DASHBOARD_VERBOSE=true dashboard whoami
```

If `$DASHBOARD_VERBOSE` is set to any value, verbosity is enabled.

Session Storage

To set the filename to store session information, use the `-S` or `--session-file` options.

```
dashboard --session-file session.txt whoami
```

You can alternatively set an environment variable to set the session file.

```
DASHBOARD_SESSION_FILE=session.txt dashboard whoami
```

TLS/SSL Certificate Verification

To disable TLS/SSL certificate verification, use the `--insecure` option.

```
dashboard --server https://dashboard.server --insecure login
```

You can alternatively set an environment variable to disable TLS/SSL certificate verification

```
DASHBOARD_INSECURE=true dashboard --server https://dashboard.server login
```

This can be helpful in case of self signed certificates or SSL interception proxy servers.

User and Role Overview

		user “root”	role “admin”	role “user”
User Management				
	List users	x		
	Add user	x		
	Delete user	x		
	Change own password	x	x	x
	Change other user’s passwords	x		
	Get own user info	x	x	x
	Get other user’s info	x		
	Get own username	x	x	x
	Set other user’s roles	x		
Projects				

		user “root”	role “admin”	role “user”
	Create project	x		
	List projects the user has a role in	x	x	x
	List all projects	x		
	Delete project	x		
	Export project	x	x	x
	Import project	x	x	
Reports				
	Upload report	x	x	x
	List reports	x	x	x
	Delete report	x	x	
Tokens				
	Create token	x	x	
	List tokens	x	x	
	Delete token	x	x	
	Delete all tokens	x	x	
Requirements				
	Upload requirements	x	x	
	List requirements	x	x	x
	Delete requirements	x	x	
Cache				
	Rebuild cache	x	x	

User Management

List Users

To list all users, run

```
dashboard list-users
```

You need to be `root` to list users.

Add User

To add a user, run

```
dashboard add-user <username> [--password <password>]
```

The password can optionally supplied via command line. If this is not the case, it is asked interactively twice.

Example:

```
dashboard add-user foo
```

You need to be `root` to add users.

Delete User

To delete a user, run

```
dashboard delete-user <username>
```

Example:

```
dashboard delete-user foo
```

You need to be `root` to delete users.

Change Password

To change your own password, run

```
dashboard passwd [--password <password>]
```

The password can optionally supplied via command line. If not, it is asked interactively twice.

Example:

```
dashboard passwd
```

To change another user's password, run

```
dashboard passwd --username <username> [--password <password>]
```

You need to be `root` to change other user's passwords.

Example:

```
dashboard passwd --username foo
```

User Information

To get user information for the current user, run

```
dashboard user-info
```

It will list the current username and project roles.

To get user information for other users, run

```
dashboard user-info --username <username>
```

as `root`.

Get Username

To get the currently logged in user name, run

```
dashboard whoami
```

Set User's Project Roles

To set user's roles for a project, run

```
dashboard set-roles <project> <username> [<list of roles>]
```

as `root`.

Available roles are `admin` and `user`.

Example:

```
dashboard set-roles project_1 user_name admin
```

To remove user's roles just run the command without the optional roles.

Example:

```
dashboard set-roles project_1 user_name
```

Projects

Create Project

You can create a new project by running

```
dashboard create-project <project-name> <project-alias>
```

Example:

```
dashboard create-project project "My first project"
```

Only `root` users can create projects.

List Projects

To list all currently available projects in the Dashboard you can, run

```
dashboard list-projects
```

```
Project             Alias                Number of Reports
------------------  -----              -------------------
unit_tests          Unit Tests                         200
integration_tests   Integration Tests                  300
system_tests        System Tests                       400
```

Only projects available for the user are listed. A project is available if the user has at least the `user` role for the specified project.

Running this command as `root` user lists all projects.

Project Status

To show the latest project status in the command line, run

```
dashboard status <project>
```

```
dashboard status unit-tests
Project: Unit Tests
Last modified: Sat, 11 Dec 2021 16:38:31 GMT
Successful tests: 92 % (23/25)
Requirements coverage: 98 % (98/100)
```

> **Warning**
> After uploading a new report it might take a few seconds until the cached contents are rebuilt.

Delete Project

To delete a project and the corresponding reports, requirements and tokens, run

```
dashboard delete-project <project-name>
```

Example:

```
dashboard delete-project project
```

Only `root` users can delete projects.

Export Project

To export a project to an archive, run

```
dashboard export-project <project> <archive-filename>
```

The exported archive is a `zip` file containing all supplied reports, requirements and tokens to be stored as a backup or to be imported elsewhere.

Example:

```
dashboard export-project project archive.zip
```

Projects can be exported by users with at least the `user` role.

Import Project

To import an exported project from an archive, run

```
dashboard import-project <project> <archive-filename> [--wait <timeout>]
```

Note that a project needs to exist before you can import it. Projects can be imported to existing projects by users with the `admin` role.

The `root` user can create a project on import by running

```
dashboard import-project -c <project> <archive-filename>
```

Example:

```
dashboard import-project -c <project> archive.zip
```

If `--wait` is supplied the command will wait until the cache was updated after importing all data or a timeout occurs.

To import a project, update the cache and wait up to one minute run

```
dashboard import-project project archive.zip --wait 60
```

Reports

Upload Reports

Archived reports can be uploaded using the command line interface.

To upload multiple files in a directory, run

```
dashboard upload-reports <project> <list-of-filenames>
```

This interface supports the filetypes: `YAML`- and `JSON`-Reports generated by the QABench htf (formerly known as HILSTER Testing Framework) and `JunitXML`-Reports generated by any testing tool.

Example:

```
dashboard upload-reports <project> *.yaml *.yml *.json *.xml
```

> **Note**
>
> When using tokens for authentication, you have to supply a token to the project url since the upload uses the same API endpoint as `htf` does. `HTTP 403 Forbidden` is returned if a token is missing.

Reports can be uploaded by users with at least the `user` role.

List Reports

To print the list of reports uploaded for a project, run

```
dashboard list-reports <project>
```

Example:

```
dashboard list-reports <project>
```

```
ID                                   Date                   Number of Tests  Results
-----------------------------------  -------------------  -----------------
---------------------------------------
5173e06a-d705-4560-be2a-a6f9e96f3025  2020-01-01 00:00:00                 2  Success 2 Failure 0
Error 0 Skipped 0
41bf9c00-216f-4afd-b838-12d495e41802  2020-01-02 00:00:00                 2  Success 2 Failure 0
Error 0 Skipped 0
1036f49d-d460-406e-887c-e9919cca3a7f  2020-01-03 00:00:00                 2  Success 2 Failure 0
Error 0 Skipped 0
```

Reports can be listed by users with at least the `user` role.

Delete Report

To delete a report, run

```
dashboard delete-report <project> <report-id>
```

Example:

```
dashboard delete-report <project> 5173e06a-d705-4560-be2a-a6f9e96f3025
```

Reports can be deleted by users with the `admin` role.

Tokens

By default all endpoints of the Dashboard need authentication, except the endpoint used by QABench htf (formerly known as HILSTER Testing Framework) to upload reports.

Instead of authenticating QABench htf (formerly known as HILSTER Testing Framework), you can add tokens to your project that are used in the server url. Thus tokens can be used to restrict the upload of test reports.

If no tokens exist for a project, the endpoint for QABench htf (formerly known as HILSTER Testing Framework) does not need a token to be supplied in the server url. If a token is supplied in this case, an exception is raised.

If one or many tokens exist for a project, one of these tokens must be supplied in the server url. If no token is supplied in this case, an exception is raised.

Create Token

To create a token, run

```
dashboard create-token <project> <token-name>
```

Example:

```
dashboard create-token <project> my_token
```

The newly created token is printed on the command line and can be used after that. The token is simply added to the project url.

To use QABench htf (formerly known as HILSTER Testing Framework) with a token, run

```
htf -r https://dashboard.server/project/<project>/token/<token> .
```

Example:

```
htf -r https://dashboard.server/project/my_project/token/5f214bca1ddb227668c086a9 .
```

Tokens can be created by users with the `admin` role.

List Tokens

To print the list of tokens for a project, run

```
dashboard list-tokens <project>
```

Example:

```
dashboard list-tokens <project>
```

```
Name   Token
-----  ------------------------------------------------------------
token  4479723524bcb8a562e861cd8ef1998b431114ef4213e38b16dba026
```

Tokens can be created by users with the `admin` role.

Delete Token

To delete a token, run

```
dashboard delete-token <project> <token>
```

Example:

```
dashboard delete-token <project> 4479723524bcb8a562e861cd8ef1998b431114ef4213e38b16dba026
```

Tokens can be deleted by users with the `admin` role.

Delete all Tokens

To delete all tokens, run

```
dashboard delete-tokens <project>
```

Tokens can be deleted by users with the `admin` role.

Requirements

Upload Requirements

To upload requirements to a project, run

```
dashboard upload-requirements <project> <list-of-filenames>
```

Requirement files need to be `csv`-files.

More information regarding the requirement format can be found here

Example:

```
dashboard upload-requirements <project> requirements.csv
```

By default the current date will be used when uploading the requirements. To force a specific date, use

```
dashboard upload-requirements <project> <list-of-filenames> --date "%Y-%m-%d %H:%M:%S"
```

Example:

```
dashboard upload-requirements <project> requirements.csv --date "2020-01-01 00:00:00"
```

Requirements can be uploaded by users with the `admin` role.

List Requirements

To print a list of requirements uploaded to a project, run

```
dashboard list-requirements <project>
```

Example:

```
dashboard list-requirements <project>
```

```
ID                        Date                   Number of Requirements  Requirements
------------------------  -------------------  ------------------------
----------------------------
5f214bca1ddb227668c086a9  2020-01-01 00:00:00                       0  ['REQ_0']
5f214bca1ddb227668c086ad  2020-01-02 00:00:00                       4  ['REQ_0', 'REQ_1']
5f214bca1ddb227668c086b1  2020-01-03 00:00:00                       6  ['REQ_0', 'REQ_1', 'REQ_2']
```

Requirements can be listed by users with at least the `user` role.

Delete Requirements

To delete a set of requirements, run

```
dashboard delete-requirements <project> <requirements-id>
```

Example:

```
dashboard delete-requirements <project> 5f214bca1ddb227668c086a9
```

Requirements can be deleted by users with the `admin` role.

Cache

To rebuild the project cache, run

```
dashboard rebuild-cache <project> [--wait <timeout>]
```

The cache can be rebuilt by users with the `admin` role.

If `--wait` is supplied the command will wait until the cache was updated or a timeout occurs.

To update the cache and to wait up to one minute run

```
dashboard rebuild-cache <project> --wait 60
```

Bash-Completion

`dashboard` supports Bash-Completion which provides automatic command completion for the `bash` shell.

To easly run `dashboard create-project` type

```
dashboard cr<TAB>
    create-project  create-token
dashboard create-project
```

Enable Bash-Completion

To enable Bash-Completion

```
eval $(dashboard complete)
eval $(dashboard.admin complete)
```

To persistently enable bash completion, add the command to your `~.bashrc`. alternatively put it into a new file called `/etc/bash_completion.d/dashboard`, eg.

```
echo "eval $(dashboard complete)" >> ~/.bashrc
echo "eval $(dashboard.admin complete)" >> ~/.bashrc
. ~/.bashrc
```

or

```
echo "eval $(dashboard complete)" > /etc/bash_completion.d/dashboard
. /etc/bash_completion.d/dashboard
```

```
echo "eval $(dashboard.admin complete)" > /etc/bash_completion.d/dashboard.admin
. /etc/bash_completion.d/dashboard.admin
```

To check if Bash-Completion is enabled type

```
complete -p | grep dashboard
complete -p | grep dashboard.admin
```

QABench htf Integration

The Dashboard can be used with QABench htf (formerly known as HILSTER Testing Framework).

> **Warning**
>
> To use this version of dashboard, you need at least `htf-1.3.10` or `htf-2.0.6`.

To upload a report directly from a test run, append the `-r` parameter to a htf command, followed by the server url:

```
htf <test-specifiers> -r https://report.server/project/example  # the url in your browser
htf <test-specifiers> -r https://report.server/project/example/token/
4479723524bcb8a562e861cd8ef1998b431114ef4213e38b16dba026
```

a shorter form is also possible

```
htf <test-specifiers> -r https://report.server/example
htf <test-specifiers> -r https://report.server/example/
4479723524bcb8a562e861cd8ef1998b431114ef4213e38b16dba026
```

It's also possible to upload a report from the `report-tool`, as shown in the following example:

```
report-tool unittests.xml integrationtests.html -r https://report.server/project/example
```

API Endpoints

The dashboard exposes a RESTful API.

Authentication

All endpoints support signed cookie based authentication.

Login

To login, send a POST request to

```
POST /api/v1/login
```

with the following data

```
{'username': <username>, 'password': <password>}
```

Returns `204 No Content` on success or `403 Forbidden` if the username or password was not found. On success a signed cookie is set.

Logout

Send a GET or POST request without parameters to

```
POST /api/v1/logout
```

Returns `204 No Content` on success or `403 Forbidden` is the user was not logged in. On success the signed cookie is cleared.

Projects

Create a Project

To create a new project send a POST request to

```
POST /api/v1/project/<project>
```

`<project>` must be a valid project name consisting of `A-Z`, `a-z`, `0-9`, `-` and `_`.

POST data:

```
{'alias': <optional human readable name>}
```

Returns `204 No Content` on success.

`<alias>` is an optional human-readable name for the project.

List projects

To list all projects, send a GET request to

```
GET /api/v1/projects
```

Returns `200 OK` on success.

Response:

```
{"my_project": {"name": "my_project", "alias": "My Project", ...}, ...}
```

Delete a Project

To delete a project and all associated reports, requirements and tokens, send a DELETE request to

```
DELETE /api/v1/project/<project>
```

Returns `204 No Content` on success.

Project Information

To get project information about a visible project, send a GET request to

```
GET /api/v1/project/<project>
```

Response:

```
{
    "name": <project-name>,
    "alias": <project-alias>,
    "last-modified": <utc-timestamp>,
    "number_of_reports": <int or null>,
    "current_number_of_successes": <int or null>,
    "current_number_of_tests": <int or null>,
    "current_number_of_successfully_covered_requirements": <int or null>,
    "current_number_of_requirements": <int or null>
}
```

On success `200 OK` is returned. If the current user is not allowed to access the project, `404 Not Found` is returned.

Reports

Upload a Report

To upload a test report to a project, send a POST request to

```
POST /api/v1/project/<project>/report[/token/<token>]
```

POST data:

```
{'filename': <report filename>,
 'type': <report type>,
 'contents': <file contents>}
```

The `/token/<token>` is optional. `type` can be `null` or `'junit'`.

Returns `204 No Content` on success. `409 Conflict` is returned if the report already exists.

List Reports

To list all test reports of a project, send a GET request to

```
GET /api/v1/project/<project>/reports
```

Returns `200 OK` on success.

Response:

```
{"reports":
    [{"id": "0b2c2e38-3b57-47c7-a868-4e4c209a029a",
      "title": "Test Report",
      [...]
     },
     [...]
    ]
}
```

Export a Report

To export a test report, send a GET request to

```
GET /api/v1/project/<project>/report/<report-id>
```

Returns `200 OK` on success and `400 Bad Request` if `report-id` is not supplied.

Response:


```
{"id": "0b2c2e38-3b57-47c7-a868-4e4c209a029a",
 "title": "Test Report",
 "date": "2020-01-01 00:00:00",
 "start_time": "2020-01-01 00:00:00",
 "start_timestamp": 1577833200.0,
 "stop_time": "2020-01-01 00:00:00",
 "stop_timestamp": 1577833200.0,
 "duration": "",
 "seconds": 4503.94160399791,
 "test_amount": 1,
 "successes": 1,
 "errors": 0,
 "failures": 0,
 "skipped": 0,
 "unexpected_successes": 0,
 "expected_failures": 0,
 "metadata": "",
 "htf_version": "1.3.0",
 "python_version": "3.7",
 "license_information": "",
 "platform": "Linux-4.15.0-55-generic-x86_64-with-Ubuntu-18.04-bionic",
 "hostname": "test host",
 "machine": "x86_64",
 "tests": [{"result": "success",
            "data": {"links": []}
           }],
 "number_of_tested_requirements": 0,
 "number_of_successfully_tested_requirements": 0,
 "number_of_unsuccessfully_tested_requirements": 0,
 "tested_requirements": ["REQ-1", "REQ-2", "REQ-3"],
 "successfully_tested_requirements": ["REQ-1", "REQ-2"],
 "unsuccessfully_tested_requirements": ["REQ-3"]}
```


Delete a Report

To delete a test report from a project, send a DELETE request to

```
DELETE /api/v1/project/<project>/report/<report-id>
```

Returns `204 No Content` on success.

Requirements

Upload Requirements

To upload requirements to a project, send a POST request to

```
POST /api/v1/project/<project>/requirements
```

POST data

```
{'filename': <requirements filename>,
 'type': <requirements type>,
 'contents': <file contents>,
 'date': <date>}
```

Date must be in strptime-format `%Y-%m-%d %H:%M:%S`. `type` can be `null` or `'csv'`.

Returns `204 No Content` on success.

List Uploaded Requirements

To list uploaded requirements, send a GET request to

```
GET /api/v1/project/<project>/requirements
```

Returns `200 OK` on success.

Response:

```
[{"date": "2020-01-01 00:00:00",
  "requirements": ["REQ-1", "REQ-2", "REQ-3"],
  "number_of_requirements": 0,
  "id": "5f2a6479b5503fc71875a778"},
 [...]
]
```

Export Requirements

To export requirements, send a GET request to

```
GET /api/v1/project/<project>/requirements/<requirements-id>
```

Returns `200 OK` on success.

Response:

```
{"date": "2020-01-01 00:00:00",
 "requirements": ["REQ-1", "REQ-2", "REQ-3"],
 "number_of_requirements": 0,
 "id": "5f2a6479b5503fc71875a778"}
```

Delete Requirements

To delete requirements from a project, send a DELETE request to

```
DELETE /api/v1/project/<project>/requirements/<requirements-id>
```

Returns `204 No Content` on success.

Tokens

Create Token

To create a new token for a project, send a POST request to

```
POST /api/v1/project/<project>/token
```

POST data:

```
{'name': <token identifier>}
```

Returns `201 Created` on success and response.

```
{'token': <token identifier>}
```

List Tokens

To list all tokens for a project, send a GET request to

```
GET /api/v1/project/<project>/tokens
```

Returns `200 OK` on success.

Response:

```
{
    "707bcd17ef364263848bf5e3ecfeb755f01e11fee492d9f6361d3c0d": "token_0",
    "4479723524bcb8a562e861cd8ef1998b431114ef4213e38b16dba026": "token_1",
    # ...
}
```

Delete Token

To delete a token from a project, send a DELETE request to

```
DELETE /api/v1/project/<project>/token/<token>
```

Returns `204 No Content` on success.

Cache

Results that are shown in the user interface are cached. A cache rebuild can be forced by sending a POST request to

```
POST /api/v1/project/<project>/cache
```

Returns `204 No Content` on success or `404 Not Found` if the project was not found.

User Management

Users and roles can be managed via the API.

List Users

Send a GET request to

```
GET /api/v1/users
```

Returns `200 OK` on success with the payload

```
[
    {
        'username': <username>,
        'projects': {
            '<project-1>': {
                'roles': ['manager', 'user'],
            },
            # ...
    },
    # ...
]
```

Add User

To add a user, send a POST request to

```
POST /api/v1/user/<username>
```

with the payload

```
{'password': <password>}
```

Returns `204 No Content` on success or `409 Conflict` if the user already exists.

Requires user `root`.

Delete User

To add a user, send a DELETE request to

```
DELETE /api/v1/user/<username>
```

Returns `204 No Content` on success, `404 Not Found` if the user does not exist and `403 Forbidden` if the current user is not root or the supplied username is `root`. The user `root` cannot be removed.

Requires user `root`.

Get User Information

To get user information, send a GET request to

```
GET /api/v1/user
GET /api/v1/user/<username>
```

```
{
    'username': <username>,
    'projects': {
        '<project-1>': {
            'roles': ['manager', 'user'],
        },
        # ...
}
```

If no `<username>` is supplied, the current username is used.

Returns `200 OK` on success, `404 Not Found` if the user was not found and `403 Forbidden` if a user is not root or does not read its own information.

Requires user `root` to read user information for different users.

Change User's Password

To change the user's password, send a PUT request to

```
PUT  /api/v1/user
PUT  /api/v1/user/<username>
```

with payload

```
{'password': <password>}
```

If no `<username>` is supplied the current username is used.

Returns `204 No Content` on success, `404 Not Found` if the user does not exist and `403 Forbidden` if a user is not root and does not change its own password.

Requires user `root` to change password for other users.

Set User's Project Roles

To set roles for projects, send a PUT request as root to

```
PUT /api/v1/user/<username>/roles
```

with payload

```
{
    <project-0>: [role-1, role-2, role-N],
    # ...
}
```

Each role must be one out of `"admin", "user"`. If you want to remove a user's role, leave the list of roles empty.

Returns `204 No Content` on success, `404 Not Found` if the user was not found and `403 Forbidden` if the current user is not root.

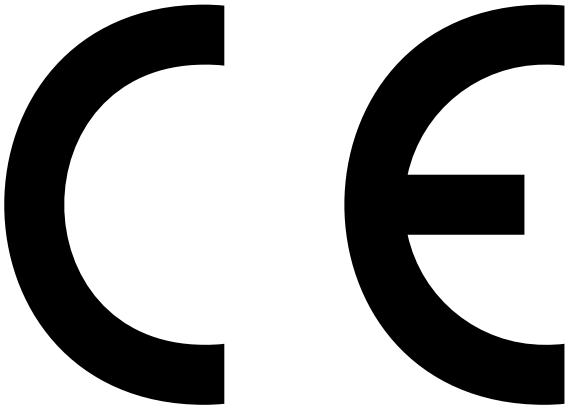
Projects don't have to exist. Thus roles can be created before a project is created.

Cyber Resilience Act Conformity

This software is distributed by

> HILSTER Testing Solutions GmbH
> Amtsstraße 10
> 22143 Hamburg
> Germany

Statement of Conformity



This software is placed on the market as a product with digital elements in the sense of Regulation (EU) 2024/2847 (Cyber Resilience Act, “CRA”). It is therefore subject to the conformity assessment obligations of the CRA.

This software is classified as “standard” risk class under the CRA.

Security Measures

As required by the CRA, HILSTER Testing Solutions GmbH takes the following measures to ensure a high level of security:

- Regular review of the source code and the dependencies used
- Prompt provision of updates for known security vulnerabilities
- Minimised collection and storage of user data
- No telemetry or automatic transmission of data to third parties

Reporting Vulnerabilities

Security vulnerabilities can be reported at any time by sending an e-mail to security@hilster.io.

> **Note**
> This statement is for information purposes only and does not constitute a declaration of conformity within the meaning of the Cyber Resilience Act.

Changelog

dashboard-1.3.1:

- update dependencies

dashboard-1.3.0

- implement Cyber Resilience Act Conformity (CRA)
- use `Python 3.14`
- use `MongoDB 8` (see update instruction in `docker-compose.yml`)
- use `nginx 1.31`
- update dependencies
- fix database initialization with multiple workers

dashboard-1.2.5

- add SBOM

dashboard-1.2.4

- update reports
- update dependencies
- update underlying Docker image
- add build information and requirements for validation purposes
- use `docker compose` instead of `docker-compose` in documentation

dashboard-1.2.3

- fix exception handling with incorrectly formatted docstrings in HTML test report

dashboard-1.2.2

- add skip reasons to HTML Test Report Overview chart

dashboard-1.2.1

- Draft reports are not stored anymore

dashboard-1.2.0

- update look and feel of QABench
- update and modernize HTML Test Report
- projects can now be imported and exported in the browser
- reports can now be uploaded and managed in the browser
- tokens can now be created and managed in the browser
- requirements can now be uploaded and managed in the browser
- requirements may contain contents that are shown in the browser

- user management is now possible in the browser
- changed commands to import and export projects for command-line utility
- cache is updated once after project import is finished
- project names are checked in ui and cli
- after password reset the user is redirected to the login page
- simplify root password reset
- update dependencies
- use Python 3.9

Warning

After an update from `dashboard-1.1` or earlier you need to update the project caches by running

```
dashboard login
dashboard update-cache <project>
```

for every project to let the requirement contents be displayed correctly.

dashboard-1.1.0

- test reports can now be reviewed in the browser
- test reports can now be downloaded in different formats (`HTML`, `JSON`, `YAML` and `JUnit-XML`)
- streamline user interface
- add badges

dashboard-1.0.1

- fix support for multiline requirements supplied by `csv` files

dashboard-1.0.0

- initial release
- upload reports from HILSTER Testing Framework and JUnit-XML
- upload requirements via CSV
- import and export
- test status overview
- requirements coverage overview
- user management and user-roles per project
- usage via command line interface
- deployment via Docker Compose

Open-Source Licenses

Bootstrap


```
The MIT License (MIT)

Copyright (c) 2011-2020 Twitter, Inc.
Copyright (c) 2011-2020 The Bootstrap Authors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
```


Chart.js


```
The MIT License (MIT)

Copyright (c) 2018 Chart.js Contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and
associated documentation files (the "Software"), to deal in the Software without restriction,
including without limitation the rights to use, copy, modify, merge, publish, distribute,
sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial
portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT
NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES
OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

Datatables

```
Copyright SpryMedia Limited and other contributors
http://datatables.net

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
```

Daterangepicker

```
The MIT License (MIT)

Copyright (c) 2012-2020 Dan Grossman

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and
associated documentation files (the "Software"), to deal in the Software without restriction,
including without limitation the rights to use, copy, modify, merge, publish, distribute,
sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial
portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT
NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES
OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

Fontawesome-free

```
Font Awesome Free License
-------------------------

Font Awesome Free is free, open source, and GPL friendly. You can use it for
commercial projects, open source projects, or really almost whatever you want.
```

```
Full Font Awesome Free license: https://fontawesome.com/license/free.

# Icons: CC BY 4.0 License (https://creativecommons.org/licenses/by/4.0/)
In the Font Awesome Free download, the CC BY 4.0 license applies to all icons
packaged as SVG and JS file types.

# Fonts: SIL OFL 1.1 License (https://scripts.sil.org/OFL)
In the Font Awesome Free download, the SIL OFL license applies to all icons
packaged as web and desktop font files.

# Code: MIT License (https://opensource.org/licenses/MIT)
In the Font Awesome Free download, the MIT license applies to all non-font and
non-icon files.

# Attribution
Attribution is required by MIT, SIL OFL, and CC BY licenses. Downloaded Font
Awesome Free files already contain embedded comments with sufficient
attribution, so you shouldn't need to do anything additional when using these
files normally.

We've kept attribution comments terse, so we ask that you do not actively work
to remove them from files, especially code. They're a great way for folks to
learn about Font Awesome.

# Brand Icons
All brand icons are trademarks of their respective owners. The use of these
trademarks does not indicate endorsement of the trademark holder by Font
Awesome, nor vice versa. **Please do not use brand logos for any purpose except
to represent the company, product, or service to which they refer.**
```

jQuery

```
Copyright JS Foundation and other contributors, https://js.foundation/

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

jQuery Easing

```
jQuery Easing v1.3.1 - http://gsgd.co.uk/sandbox/jquery/easing/

Uses the built in easing capabilities added In jQuery 1.1
to offer multiple easing options

TERMS OF USE - jQuery Easing

Open source under the BSD License.

Copyright © 2008 George McGinley Smith
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer.
Redistributions in binary form must reproduce the above copyright notice, this list
of conditions and the following disclaimer in the documentation and/or other materials
provided with the distribution.

Neither the name of the author nor the names of contributors may be used to endorse
or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
OF THE POSSIBILITY OF SUCH DAMAGE.



TERMS OF USE - EASING EQUATIONS
Open source under the BSD License.
Copyright © 2001 Robert Penner
All rights reserved.
Redistribution and use in source and binary forms, with or without modification
are permitted provided that the following conditions are met:
Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer.
Redistributions in binary form must reproduce the above copyright notice, this list
of conditions and the following disclaimer in the documentation and/or other materials
provided with the distribution.
Neither the name of the author nor the names of contributors may be used to endorse
or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
```

```
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
OF THE POSSIBILITY OF SUCH DAMAGE.
```

Moment.js

```
Copyright (c) JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.
```

Patternomaly

```
The MIT License (MIT)

Copyright (c) 2016 Tom Loudon

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

SB-Admin-2

```
The MIT License (MIT)

Copyright (c) 2013-2020 Start Bootstrap LLC

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
```