VILT Minium Manager

Reference documentation

Version 2.0

Contents

Summary

This document provides a quick start guide as well as detailed instructions on how to setup and use a VILT Minium Manager 2.0 system. Usage instructions for Minium Recorder, which is exclusive to Minium Manager customers, are also included.

Quick start

This chapter describes the quickest way to get a Minium Manager up and running. This is useful to try out Minium Manager and do not want to configure a production ready system.

1. Install and Run Minium Manager

To install and run Minium Manager:

1. Contact VILT and request the Minium Manager installation bundle.
2. Extract `minium-manager.zip` to a directory on a local machine.
3. Execute the script `./bin/start-minium-manager-all.(sh|bat)`
4. Minium Manager is now available at port `8080`. You can access it by requesting the following URL: .

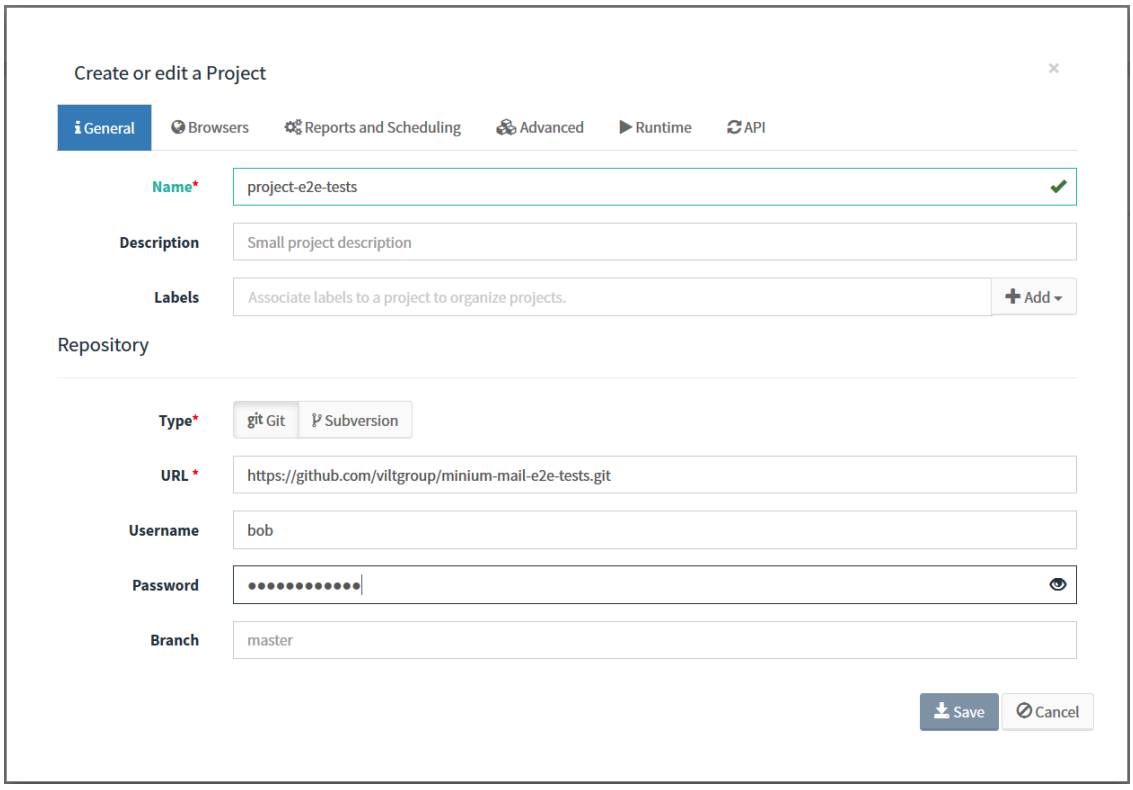
```
http://localhost:8080/
```

2. Give it a try

To try it, access Minium Manager:

```
http://localhost:8080/
```

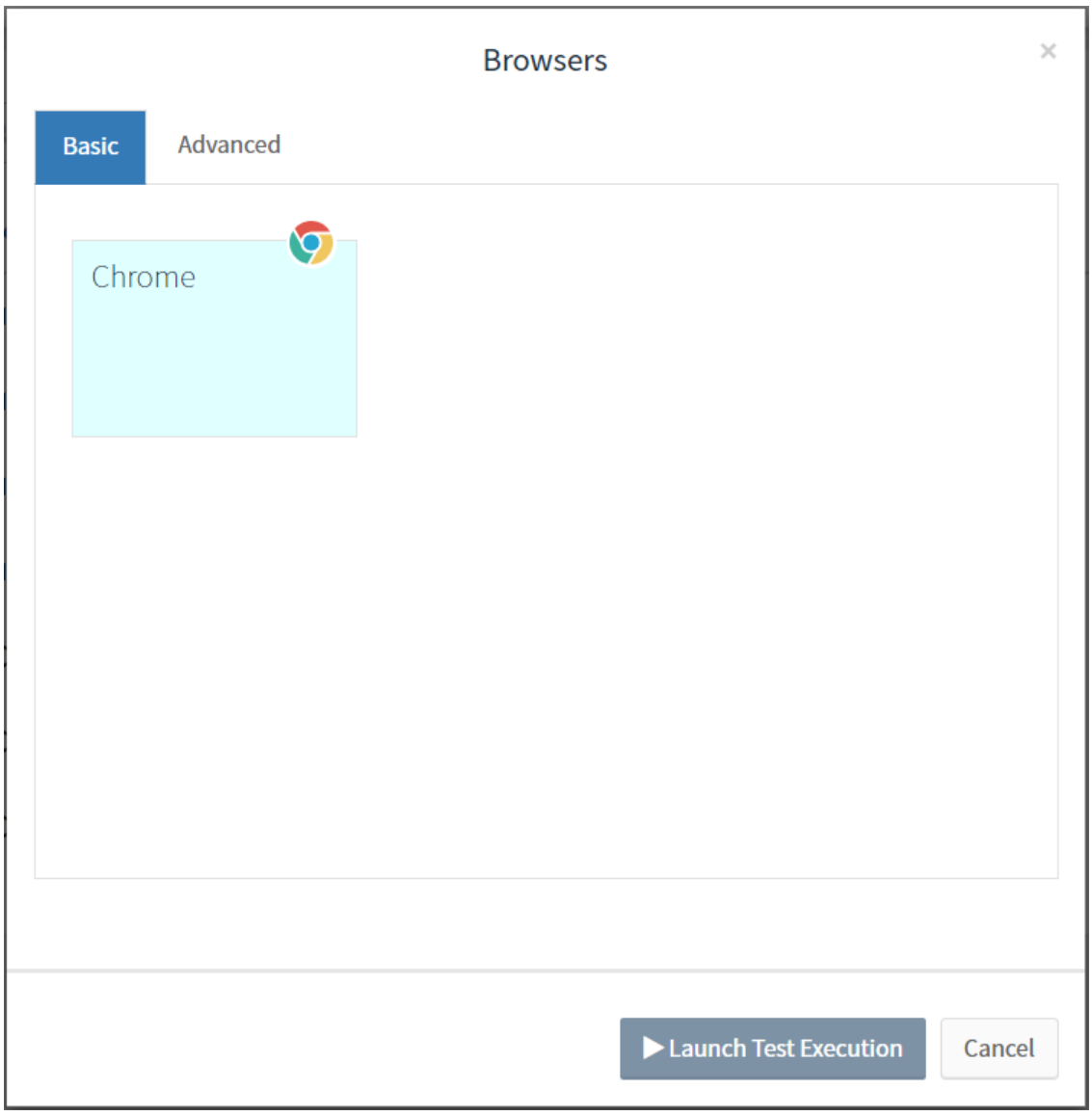
- Log in the Minium Manager and click `Create New Project`.



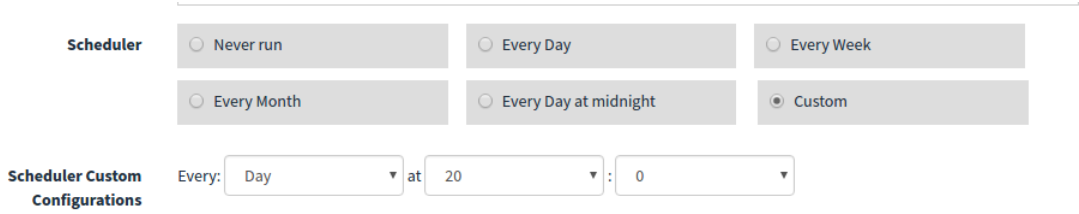
- Fill the form:

Name	Project name
Description	Description of the project
Type SCM	Type of SCM repository
Repository Url	URL of the repository where the project with the features is stored. The following are examples of valid git URLs (or a local file path): https://github.com/github/git.git //dolos/repos/gmail-e2e-tests

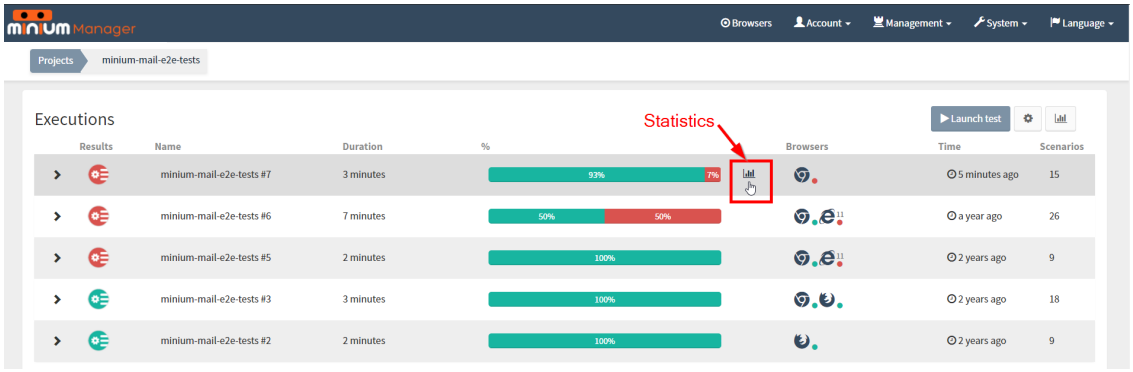
- Click `Save` to create the New Project.
- In the projects list, click on `Launch test` to launch test executions of the features and choose the browsers where you want to run the features:



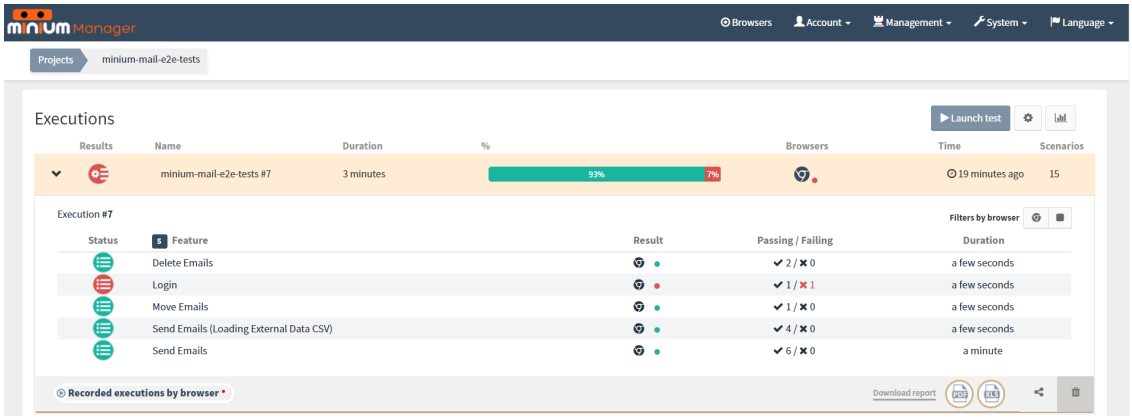
- Choose an installed browser on the machine and click `Launch Test Execution` to Start the test.
- Schedule the test execution (it the project configuration):



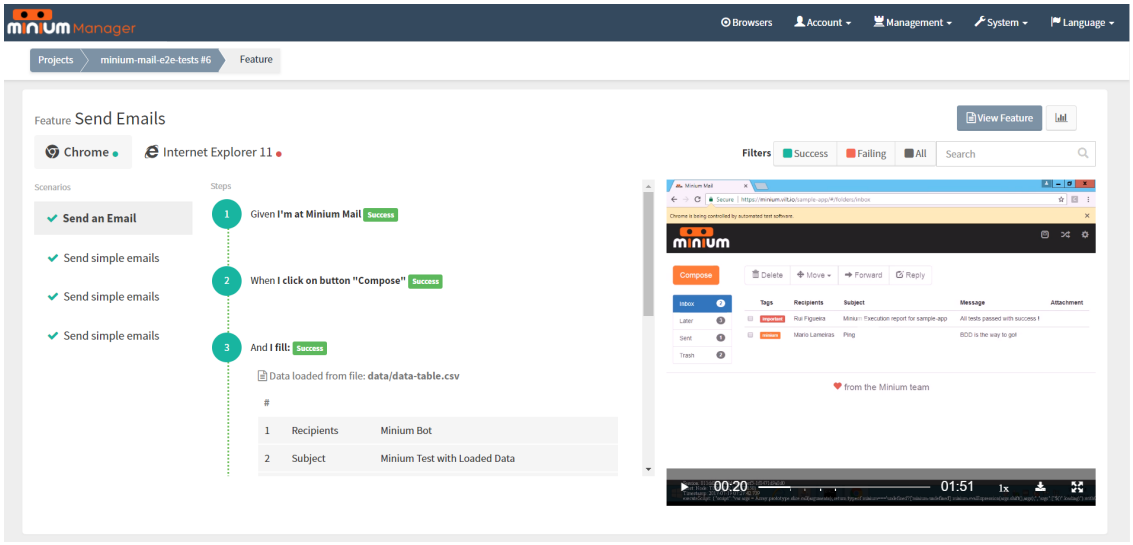
- Click on the project name (from the projects list) to see the status of the all launched executions:



- Click on the test execution to see the list of features executed and their results



- At the feature's page, see the scenario of this particular feature. Collapse the scenario to see all the steps and their results.



Installation Guide

This chapter describes the installation and configuration of the Minium Manager.

3. Supported Environments and Compatibility

This chapter provides a reference to matrices of the Minium Manager components and configurations tested and validated.

3.1. OS

The following table shows Minium Manager OS support:

OS	Versions	Minium Manager
Linux	Ubuntu 16.04 LTS	✔
MS Windows	Windows 7+	✔

3.2. Databases

The following table shows Minium Manager database support:

Database	Versions	Minium Manager
H2	—	—
MySQL	The minimum supported MySQL version is 5.6.33 Community Server 64 bits	✔

3.3. Java

Minium Manager supports Java 1.8.

3.4. Selenium Grid

Minium Manager uses Selenium Grid Extras to run tests in parallel on multiple machines, and to manage different browser versions and browser configurations centrally.

The compatibility of Selenium Grid with OS are:

OS	Versions	Selenium Grid
Linux	Ubuntu 16.04 LTS	✔
MS Windows	Windows 7+	✔

About the web browsers (based on the last versions of the webdrivers), the Selenium Grid supports:

Browser	Versions	Selenium Grid
Firefox	The latest release, the previous release, the latest ESR release and the previous ESR release.	✔
MS Internet Explorer	11	✔
Chrome	Check the release notes to see which version of the webdriver is required to the chrome version installed.	✔

You can check (in more details) the supported platforms at https://selenium.dev/downloads/

3.4.1. Selenium Grid - WebDrivers

If you are running the Node in a 32-bit Linux distros machine, you cannot update your Google Chrome web browser to the latest versions (Google drop support for all 32-bit Linux distros in March, 2016). The last version of the Google Chrome for the 32-bit Linux distros is 48. So, to continue to perform tests with Google Chrome (version 48) in 32-bit Linux distros, you should use the version 2.21 of the chrome webdriver. For the 64-bit Linux distros, you can update the Google Chrome (64-bit) to the latest version and use the latest chrome webdriver.

About the Firefox, with the release of Firefox 47, the extension based version FirefoxDriver no longer works. If you are using Firefox 46 (or prior versions), you can continue to use the FirefoxDriver. For the Firefox 47, you need to update the Firefox to the latest version and start using use the MarionetteDriver. The other option is to downgrade your version of Firefox (preferably to the Firefox 45 ERS) and continue using the FirefoxDriver. But keep in mind that the FirefoxDriver will be discontinued.

For the Internet Explorer, it is required a MS Windows OS machine.

3.5. Application Servers

The following table shows which Application Servers you can use with Minium Manager system:

Application Server	Versions	Minium Manager
Apache Tomcat	8.x	✔

3.6. Supported Browsers

Minium Manager web application is supported by the following browsers:

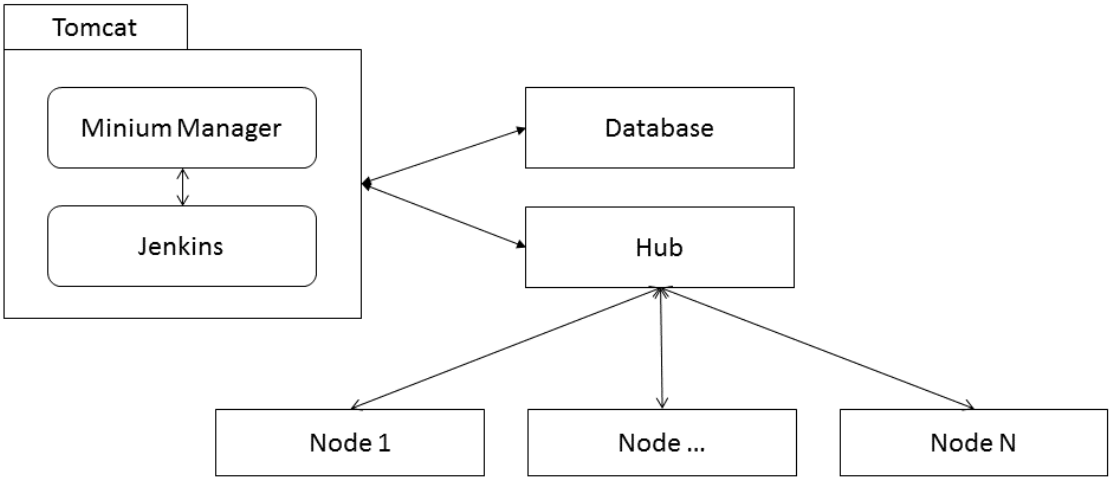
1. **Chrome**
2. **Firefox**
3. **IE 9+**

4. Minium Manager

Minium Manager is a powerful console that provides useful reports of the ongoing projects and has a strong component in continuous integration.

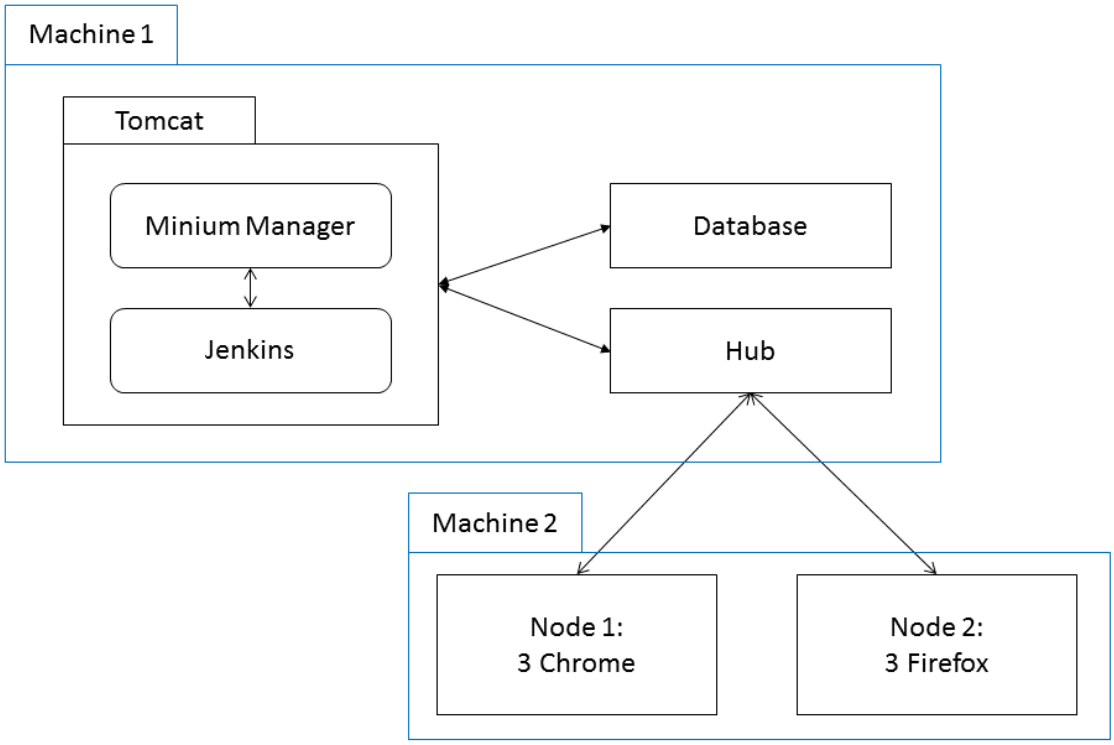
4.1. Minium Manager Architecture

The following diagram shows the generic architecture of Minium Manager:



The main components of Minium Manager are Jenkins to continuous testing and Selenium Grid: one **Hub** (to monitor and manage the tests) and several **Nodes** (runs a browser to perform the tests).

One of the standard installation for Minium Manager is:



This configuration have two machines:

- **Machine 1** with the Minium Manager, Jenkins and the Hub installed. This is the machine that will manage the tests.
- **Machine 2** with two Nodes. This is the machine that will perform the tests.

4.1.1. Machine 1

This machine runs Minium Manager with Jenkins that allows the user to create and launch the tests. The Hub will connect one or more nodes that tests will be delegated to.

To be able to create reports (result of the tests), connect to the SCM repositories and build the projects, it is required that this machine has installed the following software:

- `Java 8`
- `Git` and/or `SVN`

Note:

It is required to set the `JAVA_HOME` environment variable.

4.1.2. Machine 2

A node machine requires the installation of the following software:

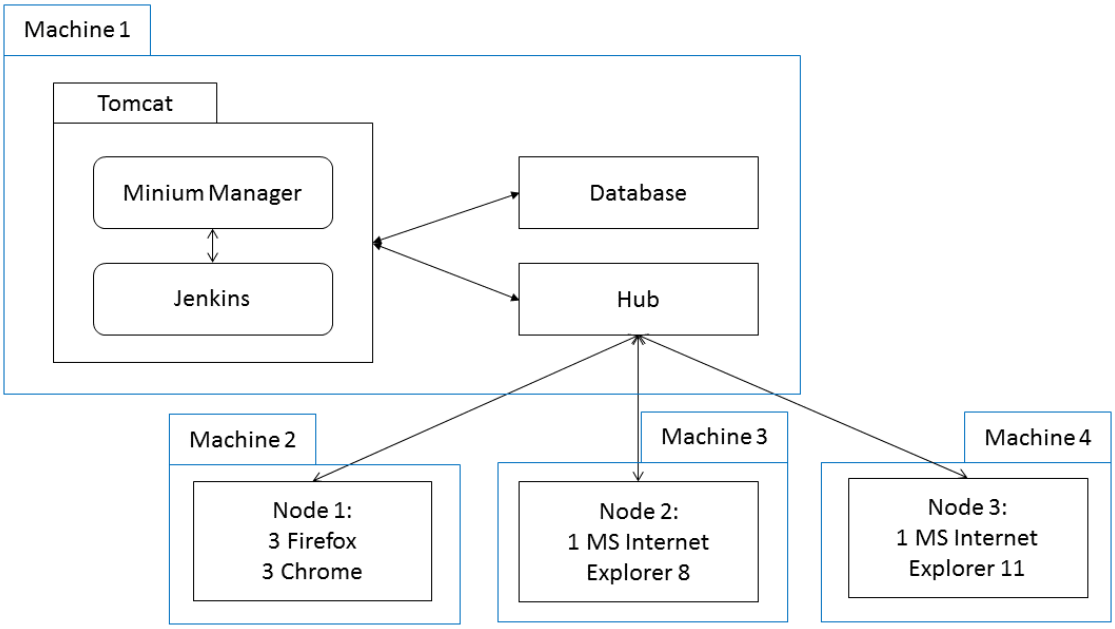
- `Java 8`

- The browser(s) and its web drivers that the node will use to execute the tests (provided by the installation package)

In this configuration, it is only necessary to run the tests in Chrome and Firefox browsers.

The node machines are independent from the hub or the other nodes (concerning to OS or browser selection). A single machine can have multiple nodes running, but is recommended a node (running only single type of browser) per machine. A node on MS Windows OS might have the capability of offering Internet Explorer as a browser option, whereas this wouldn't be possible on Linux or Mac.

An alternative standard installation for Minium Manager is:



The main difference of this configuration to the first is how the Nodes are arranged. This configuration run a Node per machine (**Machine 2**, **Machine 3** and **Machine 4**). The **Machine 3** and **Machine 4** was MS Windows OS. This installation will run the tests in parallel.

Alternatively, run the Hub in a different machine.

NOTE:

It is required a machine for each Internet Explorer version.

4.1.3. Node Configuration - Internet Explorer

To configure a node with MS Internet Explorer the following configuration is required:

- On IE 7 or higher on Windows Vista or Windows 7, set the Protected Mode settings for each zone to be the same value. The value can be on or off, as long as it is the same for every zone. To set the Protected Mode settings, choose "Internet Options…" from the Tools menu, and click on the Security tab. For each zone, there will be a checkbox at the

bottom of the tab labeled "Enable Protected Mode".

- Additionally, "Enhanced Protected Mode" must be disabled for IE 10 and higher. This option is found in the Advanced tab of the Internet Options dialog.
- The browser zoom level must be set to 100% so that the native mouse events can be set to the correct coordinates.
- For IE 11 only, it will need to set a registry entry on the target computer so that the driver can maintain a connection to the instance of Internet Explorer it creates. For 32-bit Windows installations, the key to examine in the registry editor is `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_BFCACHE`. For 64-bit Windows installations, the key is `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_BFCACHE`. Please note that the `FEATURE_BFCACHE` subkey may or may not be present, and should be created if it is not present.

Important:

Inside this key, create a DWORD value named `iexplore.exe` with the value of 0.

Check the rest of the configurations at the following website: https://github.com/SeleniumHQ/selenium/wiki/InternetExplorerDriver#required-configuration

4.1.4. Jenkins Agent

This machine runs a Jenkins Agent that executes Minium Projects.

To be able to connect to the SCM repositories, build the projects and launch the Minium Projects, it is required that this machine has installed the following software:

- `Java 8`
- `Git` and/or `SVN`
- `Maven`

Also, it is required to download the `agent.jar` to be able to connect to the Jenkins. To download the archive jar, access the link `<jenkins_ulr>/jnlpJars/agent.jar`.

4.2. System Requirements

Minium Manager minimum requirements are:

Hard-Drive:

- **Minium Manager + Database + Selenium Grid Hub**:

  To be able to run the Minium Manager the system needs to have at least 700 Mb free. Basically enough space to extract the ZIP file, run the tomcat and to write log files. Relatively to the data produced by Minium Manager, is needed to take into account how many projects will have and how many executions are scheduled (and that will be

stored). Excluding the size of the project source code, execute 100000 scenarios (number of scenarios available for one month), it is required of 20 GB (worst case) to store the history of the executions performed by jenkins. This value may vary depending on the success of the test, the number of projects and the number of executions that will be stored (old executions will be deleted). If the scenario fails, it is required a greater amount of space to store the screenshots where the error occurred during the test and the logs. For each scenario, it is needed, at least, 210 KB to save the data produced by the test. This amount does not take into consideration the space needed for the database.

Software:

- JAVA 8

If the server machine is a LINUX system with no graphical interface. You will need to install the following packages:

- xvfb
- xfonts-75dpi
- fontconfig
- libjpeg62-turbo
- libxrender1
    - **Selenium Grid Node**:

      To be able to install and run the node the system needs to have at least 100 Mb free. Not taking into consideration the Web Browsers.

Memory:

- **Minium Manager + Database + Selenium Grid Hub**: is required 8 GB.
- **Selenium Grid Node**: is required 1 GB.
- **Jenkins Node**: depends on the load but at the very least it would be 8 GB.

Offline Server:

- **RPM packages for LINUX systems**:

If the server machine does not have Internet access the RPM packages will be provided, then follow the steps below:

- Extract the packages into the server machine
- Go to the folder where the packages were extracted
- rpm -Uvh *.rpm
    - **Repository Maven**:

Local repository maven to execute the project will also be provided, then follow the steps below:

- Extract the folder into the default location of repository maven (${user.home}/.m2/repository)

5. Installing Minium Manager

This chapter describes how to install Minium Manager in a standalone application or with docker.

To install (as standalone):

- Contact VILT and request the Minium Manager installation bundle.
- Extract `minium-manager.zip` to a directory of your choice.

5.1. Database Configuration

By default, Minium Manager uses a **H2** database but an **Oracle**, a **SQL Server** or a **MySQL** database is recommended.

Before proceeding, it is necessary to create the database named `minium`.

Note:

Be free to give another name to the database. After, it is necessary to set the database name in the endpoint of the database connection at the application server.

5.1.1. MySQL

The Minium Manager supported MySQL 5.6 Community Edition.

To define the new configuration of the **MySQL** edit the file `$MINIUM_MANAGER_HOME/config/application-prod.yml`, in order to replace the `<dbhost>`, `<username>` and `<password>`:

```
spring:
  datasource:
    driver-class-name: com.mysql.jdbc.jdbc2.optional.MysqlDataSource
    url: jdbc:mysql://<dbhost>/minium
    username: <username>
    password: <password>
  jpa:
    database-platform: org.hibernate.dialect.MySQLInnoDBDialect
    database: MYSQL
```

It is also recommend to increase the buffer pool size of MySQL to avoid future problems when the amount of data stored on the database increases (see **Minium Manager is slow** on section Troubleshooting Minium Manager).

5.2. Selenium Grid Configuration

Minium Manager uses **Selenium Grid** to provide an easy way to run tests in parallel on multiple machines and manage different browser versions and browser configurations centrally.

The default configuration run the **Hub** and the **Node** in the same machine as the Minium Manager, but it can choose to run the **Hub** and the **Node** on another machine or use the browserstack as provider to run the tests.

5.2.1. Move the Hub and the Node to another machine

If choose to run the **Hub** on another machine, copy the `./selenium` folder to the new machine and configure the new endpoint.

Inside this folder (`./selenium`), will find two folders: `hub` and `node`.

The `hub` folder contains a configuration to run the **Hub** (see below).

The `node` folder contains two folders with configurations to run the **Node** in a MS Windows and in a Linux OS machine:

- The `win` folder contains a configuration to run the **Node** in a MS Windows OS machine. In this configuration (`./selenium/node/win/node_5555.json` file), it is configured 3 instances of firefox, 3 instances of chrome and 1 instance of internet explorer (version 11).
- The `linux` folder contains a configuration to run the **Node** in a Linux OS machine (default configuration). In this configuration (`./selenium/node/linux/node_5555.json` file), it is configured 3 instances of firefox and 3 instances of chrome.

It is necessary to add/update the Minium Manager configuration (replace `<selenium_host>` at the `./config/application-prod.yml` file) in order to set the new provider:

```
minium:
  manager:
    providers:
      - type: seleniumgridextras
        selenium-grid-extras-url: <selenium_host>/grid
```

After this launch the **Hub** from another machine with `./bin/start-selenium-hub.(sh|bat)` script (from the `./selenium/hub` copied folder):

Note:

The new machine must have the java installed and configured (i.e. the `JAVA_HOME` environment variable).

It is possible to choose a different port (default: 4444), to do this edit the `hub_4444.json` (at the `./selenium/hub` copied folder) before launching the **Hub**, to replace the port property.

At the `./selenium/hub` folder find the `hub_4444.json` file with the following properties:

Property	Type (Default Value)	Description
port	Integer (4444)	The port that **Hub** will use (explained above).
newSessionWaitTimeout	Integer (-1)	The time (in milliseconds) after which a new test waiting for a node to become available will time out. When that happens, the test will throw an exception before starting a browser.
throwOnCapabilityNotPresent	Boolean (true)	If true, the hub will reject test requests right away if no proxy is currently registered that can host that capability. Set it to false to have the request queued until a node supporting the capability is added to the grid.
nodePolling	Integer	Interval (in milliseconds) between alive checks of node how often the hub. Checks if the node is still alive.
cleanUpCycle	Integer	How often (in milliseconds) a proxy will check for timed out thread.
browserTimeout	Integer	The timeout (in milliseconds) a browser can hang.
timeout	Integer	The timeout (in milliseconds) before the hub automatically ends a test that hasn't had any activity in the last X seconds. The browser will be released for another test to use. This typically takes care of the client crashes.
maxSession	Integer (5)	The max number of tests that can run at the same time on the nodes, independently of the browser used.

Node Configuration

The default configuration run one **Node** in the same machine as the Minium Manager.

In order to run instances of the **Node** on another machine, copy the `./selenium` folder to the new machine and configure the endpoint to **Hub**. It is required that the new machine have java installed.

To define the endpoint of the **Hub**, edit the **Node** configuration (on the new machine) at `./selenium/node/(linux|win)/node_5555.json` file, to replace `<machine_ip>` (the host/"ip address" of the new machine), `<hub_host>` (the host/"ip address" of the Hub) and `<hub_port>` (the port used by the Hub):

```
{
  ... ,
  "configuration": {
    ... ,
    "host": "<machine_ip>",
    ...
    "hubHost": <hub_host>,
    "hubPort": "<hub_port>",
    ...
  }
}
```

Launch the **Node** from another machine with `./bin/start-selenium-node.(sh|bat)` script (from the `./selenium/node/(linux|win)` copied folder):

Note:

Before start any **Node** instance, make sure that the **Hub** instance is started.

Note:

Be free to change this configuration, but before apply the changes, stop the **Node**, update the `node_5555.json` file, and launch the **Node** again.

At the capabilities section (in the `node_5555.json` file) we can set the browser(s) for the tests:

Property	Type	Description
browserName	String	The name of the browser being used; this value should be one of `{android|chrome|firefox|htmlunit|internet explorer|iPhone|iPad|opera|safari}`.
version	String	The browser version, or the empty string if unknown.

Property	Type	Description
platform	String	A key specifying which platform the browser should be running on. This value should be one of `{WINDOWS|XP|VISTA|MAC|LINUX|UNIX|ANDROID}`. When requesting a new session, the client may specify `ANY` to indicate any available platform may be used.
maxInstances	Integer	Maximum number of instances to allow to connect to grid.

Check the rest of the properties at the following website: https://raw.githubusercontent.com/wiki/SeleniumHQ/selenium/DesiredCapabilities.md

WebDrivers Configuration

In order to update the webdrivers at the **Node**, due the incompatibility with the web browser version (check Selenium Grid - WebDrivers to see more about the compatibility), edit the `selenium_grid_extras_config.json` file. In this file will find the several webdrivers supported by Selenium:

- **webdriver** - The old webdriver that supports Firefox (Supports Firefox 46 or prior versions). It will be discontinued.
- **iedriver** - The webdriver that supports Internet Explorer
- **chromedriver** - The webdriver that supports Google Chrome.
- **marionettedriver** - The new webdriver that supports Firefox (Supports Firefox 47+).

So, in order to define the version of the webdriver to use, open and edit the `selenium_grid_extras_config.json` file:

```
{
  ... ,
  "theConfigMap": {
    ... ,
    "{webdriver|iedriver|chromedriver|marionettedriver}": [
      "version": "<nr_version>"
    ],
    ...
  }
}
```

Will find a block for each webdriver. It is only necessary replace `<nr_version>`.

Selenium Grid Extras also automatically updates the webdrivers binaries. In order to automatically update the webdrivers binaries, open and edit the

`selenium_grid_extras_config.json` file to set the property `auto_update_drivers` with the value '1':

```
{
  ... ,
  "auto_update_drivers": "1",
  ... ,
}
```

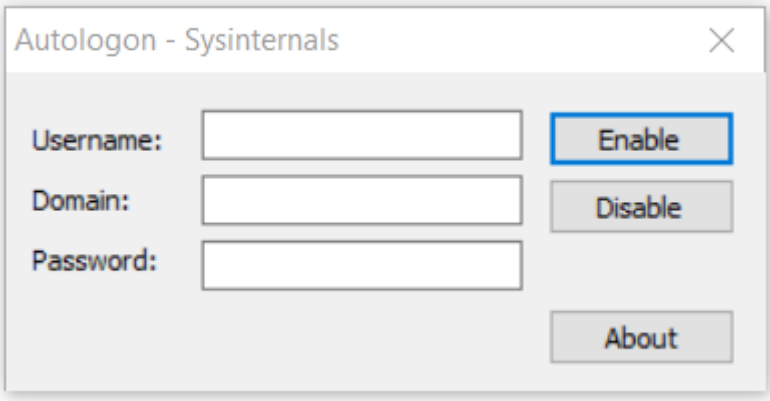
Auto Restart Nodes

Selenium Grid Extras can restart the nodes automatically after a certain amount of builds have been executed and the node is currently not busy. This helps to keep the nodes in pristine state for longer periods of time, and clears up and browser crashes, which may have occurred. To restart node machine automatically, it is required the following setup:

1. Setup default login user
2. Setup default start up task
3. Give permission to access OS

Windows Node

To setup the default login user, you can use the tool Autologon from Sysinternals. After you download the file `AutoLogon.zip`, extract the zip file at the node machine and run the `Autologon.exe`:



Enter the username, domain and password or the user you want to autologin after the restart an click `Enable`. If the Autologon was successfully configured, the machine will autologon automatically with the user configured.

The user, configured at the autologin, must be able to run the command `shutdown -r -t 1 -f` at the terminal.

To configure the startup of the Selenium Node at windows, you need to add the batch script to the StartUp directory:

1. Create a shortcut to the batch file:
    a. The shortcut that we should create is `./selenium/node/win/start-selenium-node.bat`.
2. Once the shortcut has been created, right-click the file and select Cut.
3. Press the Start button and type Run and press enter (or use the keyboard shortcut Win+R).
4. In the Run window, type `shell:startup` to open the Startup folder.
5. Once the Startup folder has been opened, click the Home tab at the top of the folder and select Paste to paste the shortcut into the folder.

For windows, we don't need to setup permissions as long as the user (configured at autologon) can run the following command in the Terminal: `shutdown -r -t 1 -f`

Check the rest of the Selenium Grid Extras configurations at the following website: https://github.com/groupon/Selenium-Grid-Extras

Additional browser capabilities

By default, the selection of the browsers on which the tests will run is as simple as possible, by only taking into account their names. In case more specificity is needed, additional browser capabilities can be configured. If any of those capabilities is not `browserName`, `version`, `platform` or `applicationName`, start by making sure that:

- The `minium-grid.extensions.jar` file is on the `selenium` folder.
- The property `hub_additional_classpath` on the `selenium/hub/selenium-grid-extras_config.json` file includes the value `minium-grid-extensions.jar`.
- The property `capabilityMatcher` on the `selenium/hub/hub_4444.json` file is set to `minium.manager.grid.MiniumCapabilityMatcher`.

After setting up the hub, add the capabilities to the JSON configuration files of the nodes (e.g., `node_5555.json`). Example containing the version, platform and environment of each browser:

```
{
  "capabilities": [
    {
      "browserName": "chrome",
      "maxInstances": 1,
      "version": "62",
      "platform": "Windows",
      "environment": "Development"
    },
    {
      "browserName": "firefox",
      "maxInstances": 1,
      "version": "56",
      "platform": "Windows",
      "environment": "Development"
    }
  ]
}
```

Note that the names of the capabilities are in camel case (`browserName`, `applicationName`,

etc.).

Lastly, add the capability names to the `config/application-prod.yml` file, so that they will be considered and displayed on the user interface:

```
minium:
  manager:
    webdrivers:
      capabilities-to-consider:
        - version
        - platform
        - environment
```

5.2.2. BrowserStack

To run the tests through browserstack as provider is necessary to update the default configuration.

First disable the default configuration in order to set the new provider. So, edit the `./config/application-prod.yml` file to uncomment the `minium.manager.providers.browserstack-username` and the `minium.manager.providers.browserstack-access-key` (remove the '# '), set the `minium.manager.providers.type` with the `browserstack` value and replace `<browserstack_username>` and `<browserstack_access_key>`:

```
minium:
  manager:
    providers:
      - type: browserstack
        browserstack-username: <browserstack_username>
        browserstack-access-key: <browserstack_access_key>
```

The `minium.manager.providers.type` property value should be one of:

- *seleniumgridextras*: for a local Selenium Grid configuration (it also requires the configuration of the `minium.manager.providers.selenium-grid-extras-url` property).
- *browserstack*: to use the browserstack as provider (it also requires the configuration of `minium.manager.providers.browserstack-username` and `minium.manager.providers.browserstack-access-key` properties).

5.2.3. Multiple providers

Minium Manager also supports the configuration of multiple providers.

To add multiple providers, edit the Minium Manager configuration (replace the placeholders, explained above, and the `<provider_label>` at the `./config/application-prod.yml` file) in order to set the list of providers:

```
minium:
  manager:
   providers:
     - label: <provider_label> (e.g: Selenium Grid Extras)
       type: seleniumgridextras
       selenium-grid-extras-url: <selenium_host>/grid
     - label: <provider_label> (e.g: BrowserStack)
       type: browserstack
       browserstack-username: <browserstack_username>
       browserstack-access-key: <browserstack_access_key>
     - label: ...
       ...
```

For each new provider, we must define the index of the provider (the previous provider index plus 1).

The `provider_label` configuration must be an unique value and will define the name of the provider that will be displayed at the Minium Manager interface.

After the configuration of multiple providers, restart Minium Manager, to be able to launch the tests in different providers.

5.3. Standalone Installation

The quickest way to start using Minium Manager is to launch it as a standalone application.

5.3.1. Start Minium Manager

Minium Manager is launched by running the script `./bin/start-minium-manager-all.(sh|bat)`.

Note:

Minium Manager is launched with 1024M of heap size. To increase the heap size, you must edit the `JAVA_OPTS` environment variable at `./tomcat/bin/setenv.(sh|bat)`.

Change the default configuration (e.g. run **Selenium Grid** and/or **Selenium Node** in another host), then launch the **Selenium Grid** and/or **Selenium Node** from the different hosts (see Selenium Grid Configuration). After this run the script `./bin/start-minium-manager.(sh|bat)` to start Minium Manager.

To start the **Selenium Grid** and **Selenium Node** from another host, check the Selenium Grid Configuration.

After Minium Manager is started, and if the default configuration is used, it is ready to use and available at `http://<hostname>:8080/`

5.4. Docker Installation

An Alternative way to start using Minium Manager is through a docker container.

To run Minium Manager with docker, it is required have docker installed, docker compose too and a machine with Linux.

To install:

1. Contact VILT and request the Minium Manager installation bundle.
2. Extract `minium-manager.zip` to a directory on the local machine.

5.4.1. Start Minium Manager

Minium Manager can be started by executing the following commands:

1. Navigate to the directory created by the extraction of the installation bundle `minium-manager.zip`.
2. Run the Minium Manager image, with the following command:

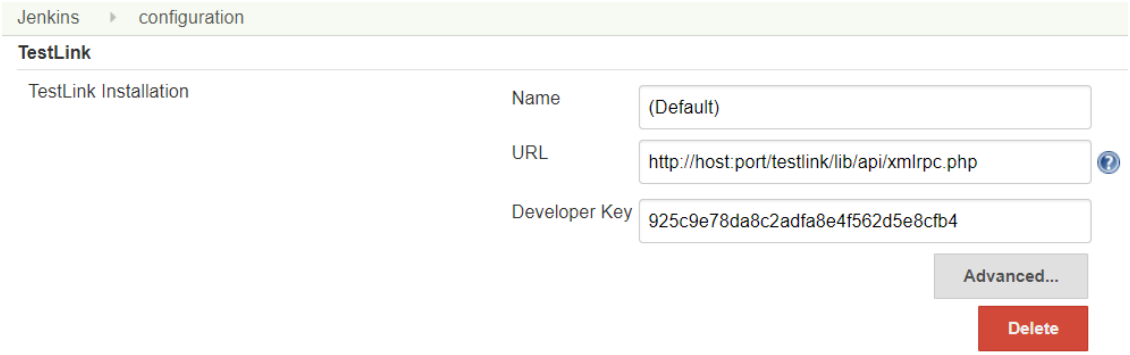
```
docker-compose up -d &
```

Note:

To configure Minium Manager (i.e. Database connection, Selenium, …), check the Installing Minium Manager section.

5.5. TestLink

Minium Manager can be configured to export test results to TestLink. To configure the integration with TestLink, start by making sure that the TestLink Plugin is installed on Jenkins. After that, go to `Manage Jenkins > Configure System` and configure the URL of TestLink and a developer key with permissions to create builds on all the desired projects:



Then, open the `config/application.yml` and add the `TestLink` value to the `minium.manager.modules` property:

```
minium:
  manager:
    modules:
      - TestLink
```

To log the batch update requests, configure the following logging levels:

```
logging:
  level:
    # logs the user who performed the request
    minium.manager.web.rest.TestLinkResource: INFO
    # logs the number of projects being updated
    minium.manager.service.TestLinkService: INFO
```

5.6. Add Jenkins Node

Minium Manager supports a new type of project: the monitoring project that are executed with high frequency (every 5, 10, 20 minutes), and can give us a report about the availability of a certain page and performance metrics, with the objective to check that the page are working as expected.

Before configuring a this new project type, first configure jenkins to add a new agent:

To add a new agent:

1. Go to `Manage Jenkins` > `Manage Nodes` > `New Node`.
2. Set the Node name field (store the node name), select `Permanent Agent` and click `Ok`.
3. Configure the `# of executors` (e.g. `5`), the `Remote root directory` (at the Jenkins Agent machine, e.g. `/tmp`) and click `Save`.
4. Open the new node menu (`<jenkins_ulr>/computer/<node_name>/`) and copy command to start the new Agent.
5. Lanch the Jenkins Agent (by executing the command previously retrieved) and check if the Agent is connected.

> **Important:**
>
> Check the Jenkins Agent to see the requirements of a new node.

6. After Installing Minium Manager

Minium Manager allows tweaking configurations such as follows:

6.1. License

This chapter describes the procedure to gather all the information required to produce a valid license and how to setup the installation with a license. Minium Manager allows to configure Projects without a license but do not allow to run tests without a valid license in place.

The license has a limit of executions per month. Each month the executions are reseted. An alert will be shown in Minium Manager when the number of executions gets close to the limit.

As an example, if exists a project with 10 features and each features have 5 scenarios, when an execution is launched in one browser it will consider 50 scenarios executed at the end of the execution. If launch the execution in 2 browsers, it will count 100 scenarios (2 x 50 scenarios) executed.

6.1.1. Obtaining a Valid License

Request a license form VILT and attach the company name.

6.1.2. Configuring Minium Manager License

With a valid license it only needs to add its contents to the file `./config/license/key.license` and restart Minium Manager. After this should be able to run tests normally.

To configure the Minium Manager license, edit the `./config/application-prod.yml` file in order to set the `minium.manager.license.client-name` and `minium.manager.license.scenarios-alert-notification` in order the replace the `<client_name>` (the name of the client) and the `<number_of_scenarios>` (the number of scenarios executed before launch a notification) values.

```
minium:
  manager:
    license:
      client-name: <client_name>
      scenarios-alert-notification: <number_of_scenarions>
```

6.2. Logs

This chapter describes how to manage the logging of the several components of Minium Manager.

6.2.1. Minium Manager

By default, Minium Manager writes logs to `./logs/minium.log`. Change easily the following properties in `application-prod.yml` to tweak the logs:

Property	Description	Default Value
`logging.level.root`	Default logging level for all classes	`INFO`
`logging.level.`*`fullyqualifiedname`*	Specify log level for specific packages by replacing *fullyqualifiedname* with the full class name. For example: org.springframework	□–□

Next an example of a possible logging configuration:

```
logging:
  file: ${minium.manager.home}/logs/minium.log ①
  level:
    org.springframework: WARN ②
    minium.manager: DEBUG ②
```

① Path where minium manager will write the log file

② Add or remove keys to suit the needs

Note:

The properties `logging.level.root` and `logging.level.`*`fullyqualifiedname`* can be set with one of the following values: `ERROR`, `WARN`, `INFO`, `DEBUG` or `TRACE`.

To stop the minium manager logging for the `./logs/minium.log` file, please comment the line `file: ${minium.manager.home}/logs/minium.log` the logging configuration (Add the hash sign "#" at the beginning of the line) at the `application-prod.yml` file and the restart the tomcat. The minium manager logging will start to be written at the `./logs/catalina.<date>.log`

Jenkins

If you intend to separate the jenkins file in a new file, you can create the file `./tomcat/webapps/jenkins/WEB-INF/classes/logging.properties` with the content:

```
handlers = org.apache.juli.AsyncFileHandler

org.apache.juli.AsyncFileHandler.level = FINE
org.apache.juli.AsyncFileHandler.directory = ${minium.manager.home}/logs
org.apache.juli.AsyncFileHandler.prefix = ${classloader.webappName}.
org.apache.juli.AsyncFileHandler.maxDays = 3
```

Note:

The path `./tomcat/webapps/jenkins/WEB-INF/classes/` will be generated after the tomcat starts for the first time and the file `./tomcat/webapps/jenkins.war` be deployed.

After tomcat is restarted, the jenkins logging will start to be written at the `./logs/jenkins.<date>.log`.

Find more information at https://tomcat.apache.org/tomcat-8.5-doc/logging.html#Using_java.util.logging_(default)

6.2.2. Selenium Grid

Selenium will write the logs at the `./selenium/{hub|linux|win}/log` folder at INFO (logging level for all classes).

To update the default configuration, edit the `selenium_grid_extras_config.json` file, in order to add the following block:

```
{
  ... ,
  "theConfigMap": {
    ... ,
    "grid_jvm_options": {
      "selenium.LOGGER.level": "<level>"
    },
    ...
  }
}
```

The `<level>` can be replaced with:

- **OFF:** Turns off logging.
- **SEVERE:** Messages about things that went wrong. For instance, an unknown command.
- **WARNING:** Messages about things that may be wrong but was handled. For instance, a handled exception.
- **INFO:** Messages of an informative nature. For instance, information about received commands.
- **DEBUG:** Messages for debugging. For instance, information about the state of the driver.

- **ALL:** All log messages. A way to collect all information regardless of which log levels that are supported.

Find more information at https://github.com/SeleniumHQ/selenium/wiki/Logging.

6.3. E-mail

Configure e-mail reports for Minium Manager executions. This way, the executions status can be sent by e-mail, which is specially handy for large tests.

Before defining the e-mail recipients on the project configuration, configure the e-mail server.

To configure the e-mail gather the following properties:

- **Protocol**: The protocol used by the email server
- **Hostname**: The email server hostname
- **Port**: The email server port number
- **Username**: The login user of the server
- **Password**: The login password of the server
- **From**: The e-mail field `from`

This configuration is done on `./config/application-prod.yml` as follows:

```
spring:
  mail:
    protocol: smtp
    host: mail.example.com
    port: 25
    user: SomeUser
    password: SomePassword
    tls: false
    auth: false
    from: minium@example.com
```

6.4. LDAP

To use LDAP authentication, set the following properties in the `config/application-prod.yml` file:

- **URLs**: The URLs of the LDAP servers.
- **Username**: DN of a binding user.
- **Password**: Password of the binding user.
- **Base**: Base suffix from which all operations should originate.
- **User search base**: The search base for user searches.
- **User search filter**: The filter used to search for users, where `{0}` will get replaced by the user's login name.

- **Group search base**: The search base for group membership searches.
- **Group search filter**: The filter used to search for group membership, where `{0}` will get replaced by the DN of the user.
- **Group name attribute**: The attribute that contains the name of the group.
- **Admin**: The login name of the initial administrator.

Here is an example:

```
spring:
  ldap:
    urls:
      - ldap://myserver1:389
      - ldap://myserver2:389
    username: uid=admin,dc=example,dc=com
    password: secret
    base: dc=example,dc=com
    base-environment:
      user-search-base: ou=users
      user-search-filter: uid={0}
      group-search-base: ou=groups
      group-search-filter: uniqueMember={0}
      group-name-attribute: cn
      admin: an.admin.user
```

6.5. Videos

To record the tests and present the videos at executions and features pages, set the following properties in the `config/application-prod.yml` file:

- **location-path**: The path where the videos are stored.
- **max-disk-space**: Maximum disk space available for the video files before the system warn the user at the health check page (in megabytes).

Here is an example:

```
minium:
  manager:
    video:
      location-path: /opt/minium/videos
      # size in MB
      max-disk-space: 10240
```

6.6. Minium version auto-update

For the version of `minium-cucumber-parent` to be automatically updated for the projects have that auto-update enabled:

- Set the `minium.manager.jenkins.miniumCucumberParentVersion` property to the target version. To use the latest version instead of a specific version, set the value of the property to the minimum version and add a `,` after it (for example `1.9.5,`). If this property is not defined, the tests will run with the version defined on the POM even if

auto-update is enabled.

- The Maven installation used by Jenkins (`Manage Jenkins > Global Tool Configuration`) must have name `maven_jenkins` and the user who runs Jenkins must have permission to execute the binary.

6.7. Error Message (Jenkins)

To print the errors at the console output, you need to add the error pattern (regular expression based in the `java.util.regex` package) that occurs in the console output of the child job (e.g. `/ERROR: Subversion checkout has been canceled/`). You can configure the patterns at the `minium.manager.jenkins.error-patterns` property list. When the pattern matchs with a line at the console output of the child job, the console output will be added to the pipeline job until find an empty line or the line "Build was aborted". The error will be propagated to the pipeline job and will be printed at the front-end of Minium Manager.

At front-end, Minium Manager displays an user-friendly message for the common cases: `/org\.mozilla\.javascript\.JavaScriptException: Error: Module \".+\" not found\..*/` (Module missing at the project) and `/org\.tmatesoft\.svn\.core\.SVNAuthenticationCancelledException: svn: E200015: Authentication cancelled/`, `/ERROR: Subversion checkout has been canceled/` (Wrong credentials for the svn repository).

6.8. Monitoring

To configure the Minium Manager to execute monitoring projects, set the following properties in the config/application-prod.yml file:

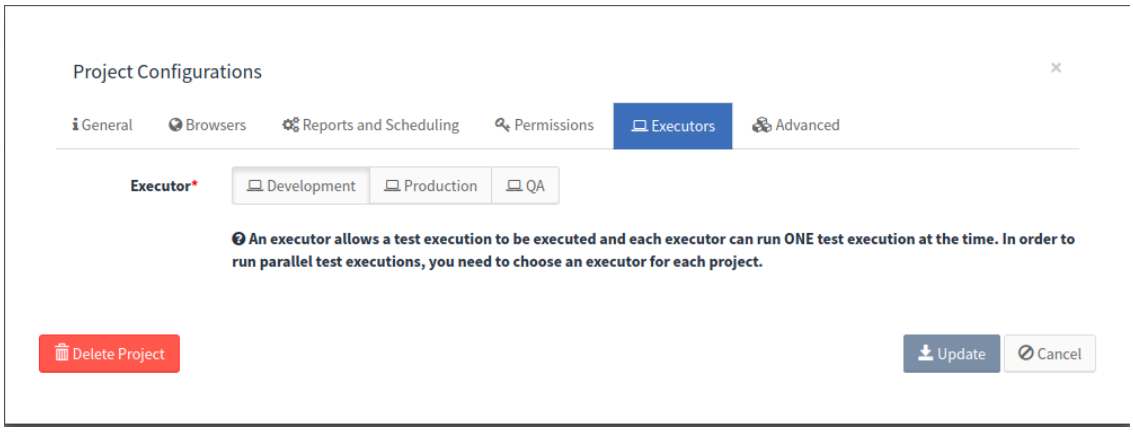
- **agents-label:** The given name of Jenkins Node.
- **max-projects-per-agent:** Maximum number of projects that executes per node.

Here is an example:

```
minium:
  manager:
    jenkins:
      agents-label:
        - monitoring
      max-projects-per-agent: 10
```

6.9. Executors

An executor allows a test execution to be executed and each executor can run ONE test execution at the time. In order to run parallel test executions, you need to choose an executor for each project.



- **home-dir**: Jenkins home directory
- **executors**: List of executors. Each executor has a name and label (this label will be shown in the UI).

Here is an example:

```
minium:
     manager:
        jenkins:
          home-dir: /home/minium-manager/data/jenkins
        executors:
          - name: dev
            label: Development
          - name: prod
            label: Production
          - name: qa
            label: QA
```

6.10. Additional WebDriver configurations

Additional WebDriver configurations can be set by adding profiles to the `config/application-webdrivers.yml` corresponding to the browser names and/or versions. Examples:

```
# adds the marionette capability when running tests on Firefox 47
spring:
  profiles: firefox47

minium:
  webdriver:
    desiredCapabilities:
      marionette: false
---
# starts Chrome, regardless of the version, with the no-sandbox argument
spring:
  profiles: chrome

minium:
  webdriver:
    chromeOptions:
      args:
        - no-sandbox
```

6.11. Integration with HP ALM

Minium Manager allow integration with HP Application Lifecycle Management (HP ALM). After Minium Manager execute the tests, the results are imported into HP ALM. Afterwards, see the results published in ALM.

Before configuring a project that imports the tests results to HP ALM, first configure the HP ALM Server at jenkins.

To configure e-the HP ALM Server at jenkins, edit the `./tools/jenkins/com.hp.application.automation.tools.settings.AlmServerSettingsBuilder.xml` file to replace `${ALM_NAME}` and `${ALM_URL}` properties with the name of the ALM Server and the endpoint to the ALM Server, respectively.

If already launched Minium Manager (i.e. the folder `./data/jenkins/` already exists), edit the `./data/jenkins/com.hp.application.automation.tools.settings.AlmServerSettingsBuilder.xml` file as well.

It is also required edit the ./config/application-prod.yml` as follows:

```
minium:
  manager:
    alm:
      url: <alm_url> - also defined at the
com.hp.application.automation.tools.settings.AlmServerSettingsBuilder.xml
file
      server-name: <alm_server_name> - also defined at the
com.hp.application.automation.tools.settings.AlmServerSettingsBuilder.xml
file
      username: <alm_username> - the username of the user configured at HP
ALM
      password: <alm_password> - the password of the username of the user
configured at HP ALM encrypted (see the Encrypt password section below how
to encrypt the password)
      baseURL: <baseURL_Minium_Manager> - the base URL of Minium Manager
to create the link at HP ALM
```

Launch Minium Manager and create/configure a project imports the tests results to HP ALM. The ALM properties can be found at the project properties by clicking at `Advanced configuration` button and selecting the `ALM` at Modules.

6.11.1. Encrypt password

Access the jenkins console through:

```
http://localhost:8080/jenkins/script
```

And run the following script with the `<password_here>` replaced with the password desired to encrypt.

```
println(hudson.util.Secret.fromString('<password_here>').getEncryptedValue
())
```

Press `Run` and retrieve the password at the result.

6.12. Update Minium Manager

1. Download the `WAR` file provided by VILT
2. Stop the Minium Manager by running `minium-manager\tomcat\bin\shutdown.(bat|sh)` script
3. Delete `ROOT.war` (if exists) and the folder ROOT in `minium-manager\tomcat\webapps`
4. Copy and paste the new .war in `minium-manager\tomcat\webapps`
5. Restart tomcat by running `minium-manager\tomcat\bin\start.(bat|sh)` script

7. Troubleshooting Minium Manager

Most common problems with the Minium Manager are reviewed in this chapter.

Cannot start Minium Manager:

- Check if already have another service listening on the same port of the Minium Manager (default port: 8080). To configure a new port (to a standalone installation), change the `port` attribute value of the connector with the protocol="HTTP/1.1", at the `./tomcat/conf/server.xml` file. For the docker installation, check this website in order to update the `docker-compose.yml` file.
- If configured Minium Manager with **MySQL** database, check if Minium Manager database is running and is reachable from the Minium Manager host.
- Check if the **CATALINA_HOME** environment variable is defined with the path to a different tomcat.

Cannot execute tests with Selenium Node:

- The error `org.openqa.selenium.remote.UnreachableBrowserException: Could not start a new session. Possible causes are an invalid address of the remote server or browser start-up failure.`
  - This error can occur when the Node configuration (`capabilities` section at the `node_5555.json` file) is wrong or the webdriver version is no compatible with the browser version. Check the [Support Matrix] chapter to see if the webdriver version is compatible with the browser version. Also, check the Node Configuration chapter to see if the configuration is correct.
- The errors `org.openqa.selenium.WebDriverException: Session * was terminated due to SO_TIMEOUT` and `org.openqa.selenium.WebDriverException: Error forwarding the new session Error forwarding the request Read timed out Command duration or timeout: * seconds`
  - Raise the browserTimeout value at the `./selenium/hub/hub_4444.json` and restart the Selenium Grid (Hub and Nodes).
- The error `org.openqa.selenium.WebDriverException: JavaScript error (WARNING: The server did not provide any stacktrace information)`
  - Check if the browser has the JavaScript disable. If disabled, so enable it.: http://www.enable-javascript.com/
- The error `org.mozilla.javascript.WrappedException: Wrapped org.openqa.selenium.WebDriverException: Failed to navigate to https://<username>:<password>@<host>/. This usually means that a call to the COM method IWebBrowser2::Navigate2() failed.`
  - This error can occur when we are using the basic authentication at internet explorer. To fix this you need to register a key:
    - **The 64-bit variant of IE with 64-bit variant of IEDriverServer.exe**: HKEY_LOCAL_MACHINE/SOFTWARE/Wow6432Node/Microsoft/Internet Explorer/MAIN/FeatureControl/FEATURE_HTTP_USERNAME_PASSWORD_DISABLE
    - **The 32-bit variant of IE with 32-bit variant of IEDriverServer.exe**: HKEY_LOCAL_MACHINE/SOFTWARE/Microsoft/Internet Explorer/MAIN/FeatureControl/FEATURE_HTTP_USERNAME_PASSWO

RD_DISABLE

- After create a DWORD `iexplore.exe` with value `0` in this registry key that you just created. More info at https://stackoverflow.com/a/23519791

Cannot connect to the internet due to proxies issues:

- **Maven:** edit the file `settings.xml` of the maven used by Jenkins (at `./tools/maven/conf`) in order to put the proxies detail inside (uncomment the proxy options and fill in the proxy server detail) and save the file.
- **VN at Jenkins (java based):** check the documentation to pass the settings to the tomcat (e.g. -Dhttps.proxyHost=host … -Dhttp.nonProxyHosts=localhost|…)
- **Git:** check the documentation to configure the proxy.

Minium Manager is slow

If some pages are taking too long to load, it might be because the buffer pool size of MySQL is too small for the amount of data stored on the database. To increase it, change the value of the `innodb_buffer_size` configuration option of MySQL. A value of `1G` should be enough for most cases.

PDF reports do not look as expected

If the appearance of PDF reports do not look as expected, it usually can be fixed by adding some options to the properties `minium.manager.pdf-generator.additional-settings` and `minium.manager.pdf-generator.additional-settings-object`. If the elements on PDF reports look too big or too small, set the `load.zoomFactor` parameter accordingly. Or, if there are icons missing on some reports, add the `load.jsdelay` parameter with a value of around 500 (milliseconds). Example:

```
minium:
  manager:
    pdf-generator:
      additional-settings-object:
        load.zoomFactor: 0.75
        load.jsdelay: 400
```

You can check the properties for additional-settings here and for additional-settings-object here.

Black videos (and black screenshots at progress) for Selenium Nodes (Windows)

When using Remote Desktop to connect to a remote computer, closing Remote Desktop locks out the computer, displaying the logon screen. If you use RDP to connect to the machine, everytime you exit the machine the GUI is diabled and the video will be recorded as black. Also, the progress screenshots will be black.

To avoid problems with GUI, use the tscon utility to disconnect from Remote Desktop. *tscon* returns the control to the original local session on the remote computer, bypassing the logon screen. All programs on the remote computer continue running normally, including GUI tests.

To solve this issue:

1. create a batch file with this code:

```
for /f "skip=1 tokens=3" %%s in ('query user %USERNAME%') do (
  %windir%\System32\tscon.exe %%s /dest:console
  timeout 5
)
```

2. Create a desktop shortcut to this file. To do this, right-click the batch file and select `Send to | Desktop (create shortcut)`.
3. In the shortcut properties, click Advanced and select Run as administrator.

Now, when you need to disconnect from Remote Desktop, double-click this shortcut on the remote computer (in the Remote Desktop window).

It is recommended to configure the Autologon. With the Autologon, when the Selenium Node machine boots, it will sign in with an user and will have the GUI available.

An alternative is to use the VNC server to access the Selenium Nodes machine instead of RDP.

8. Uninstalling Minium Manager

Before actually removing Minium Manager please read all of the chapters in this section.

8.1. Remove Databases

If using the embedded **H2** database the database directory is removed by deleting the directory created by the extraction of the installation bundle `minium-manager.zip`.

If **MySQL** it is be used it is necessary to manually drop the database.

8.2. Remove Standalone Installation Files

Note

Before proceeding shutdown the Minium Manager.

Finally, after everything else is removed, remove the Minium Manager install files by deleting the directory created by the extraction of the installation bundle `minium-manager.zip`.

8.3. Remove Docker Installation Files

Through the docker commands, remove the installed docker with the following command:

```
docker rmi minium-manager:production
```

Finally, after everything else is removed, remove the Minium Manager configuration files by deleting the directory created by the extraction of the installation bundle `minium-manager.zip`.

User Guide

VILT Minium Manager: User Guide

9. Introduction

This document describes the user interface of the VILT - Minium Manager, version 2.0.

Minium Manager is a platform to manage, run and analyze end to end tests, designed to run tests across all possible combination of browsers and OS's on a continuous integration fashion. It allows managers to control the quality of a project, providing detailed reports with useful information for failures with errors messages and screenshots.

10. Projects

Configure projects in Minium Manager, in order to run and analyze end to end tests. All projects need to have a repository (`Git` or `SVN`) associated, where the code is stored.

There are two type of projects: Web application testing and Monitoring.

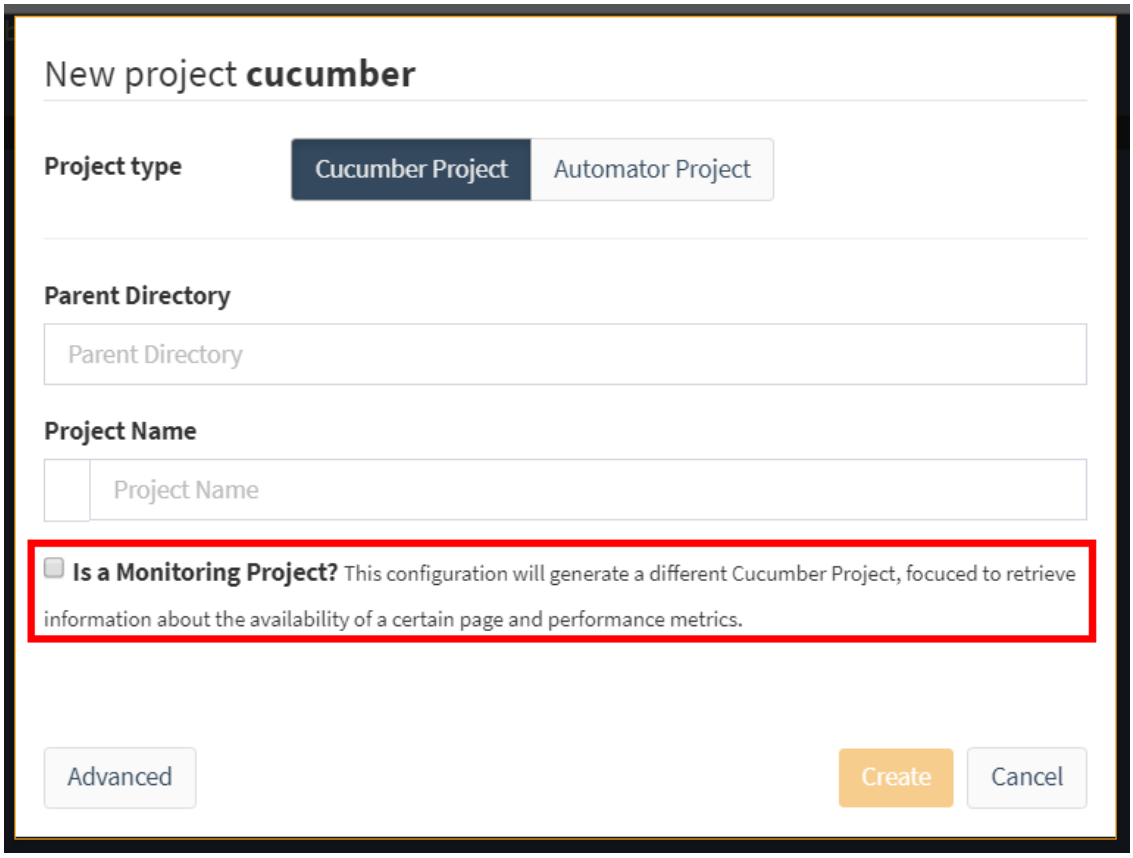
The Web application testing project is a Minium project that test a website in the same way a human would.

The Monitoring project is a Minium project that are executed with high frequency (every 5, 10, 20 minutes), and can give us a report about the availability of a certain page and performance metrics, with the objective to check that the page are working as expected.

The development workflow for a Monitoring project in Minium Developer will be similar to the a web application testing project, providing an easy and quick way to write test. The configuration in Minium Manager is similar to other projects. The monitoring project does not have the browser configuration.

10.1. Monitoring

To start the development of a monitoring project, you need to check the checkbox `Is a Monitoring Project?` to generate the monitoring project:



Here are some examples of the development workflow in Minium Developer of the monitoring

project:

10.1.1. Check if the website is ok

Use case: You want to test if the URL is ok and get the performance in the reports

Scenario:

```
Scenario: Check Blog La caixa
  When Check if website responds: "https://blog.caixabank.es/"
```

Step:

```
When(/^Check if website responds: "([^"]*)"$/, function(url) {
  browser.get(url);
});
```

For a monitoring project, the expression `browser.get(url);` retrieves the data related to page load performance, the URL status, the number of requests, the page size and javascript errors. This information will be presented at the Minium Manager report.

10.1.2. Do a search on the website with refresh and check how long it took to load.

Use case: You want to perform a search in the website (where's there's a refresh after the search) and check if the results appear and get the performance of the page.

Scenario:

```
Scenario: Search
    Given I'm at
"https://www.caixabank.es/particular/home/particulares_es.html"
    When Search for "CaixaBank" and check the results
```

Steps:

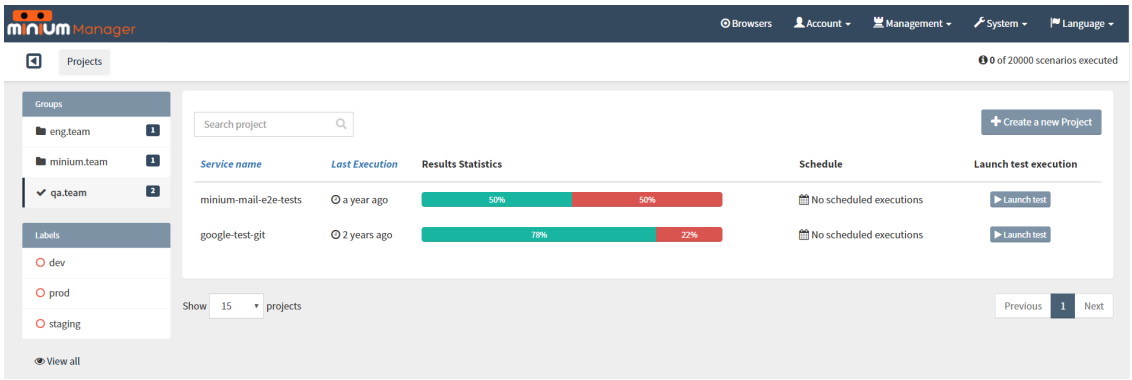
```
When(/^I'm at "([^"]*)"$/, function(url) {
  browser.get(url);
});

When(/^Search for "([^"]*)" and check the results$/, function(search) {
  $("#cookies-accept-full a").click();
  $("#search-field").fill(search);
  $("#prebuscadorCabecera input[type='submit']").click();
  expect($(".search-result-block")).to.exist();
  scenario.write(browser.getPerformance());
});
```

The expression `scenario.write(browser.getPerformance());` retrieves and store the data

related to the last page load performance, the URL status, the number of requests, the page size and javascript errors. This information will be presented at the Minium Manager report.

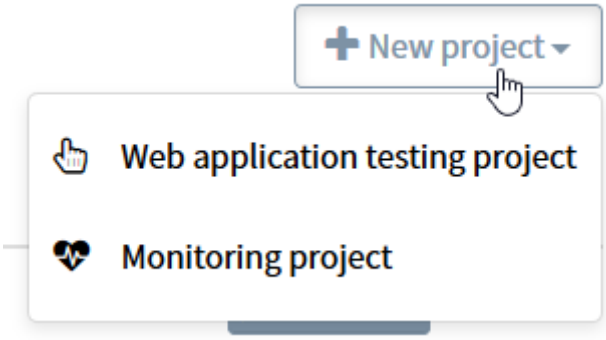
10.1.3. Click on a link and get the performance of the page

```
When(/^I click on link with text "([^"]*)"$/, function(text) {
 $("a").withText(text).click();
 expect($(".page-title")).to.exist();
 scenario.write(browser.getPerformance());
});
```

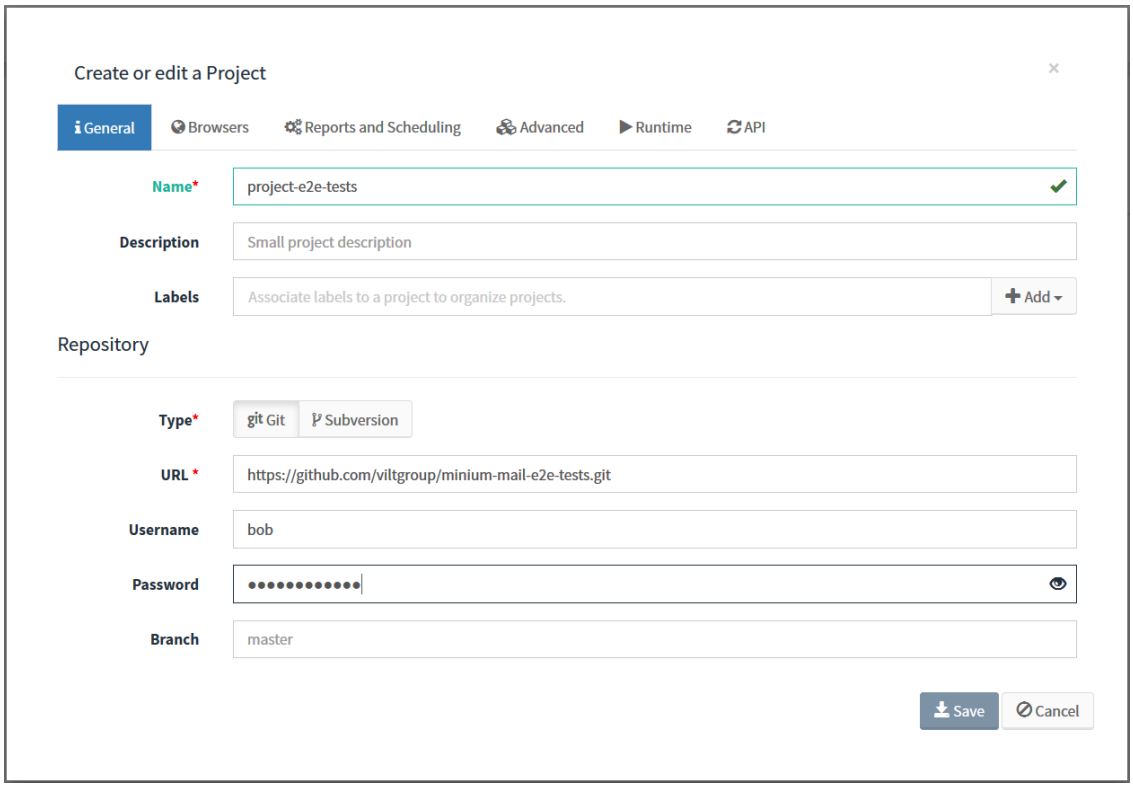


10.2. Setup a project

To setup a project, click on the button `New Project`:

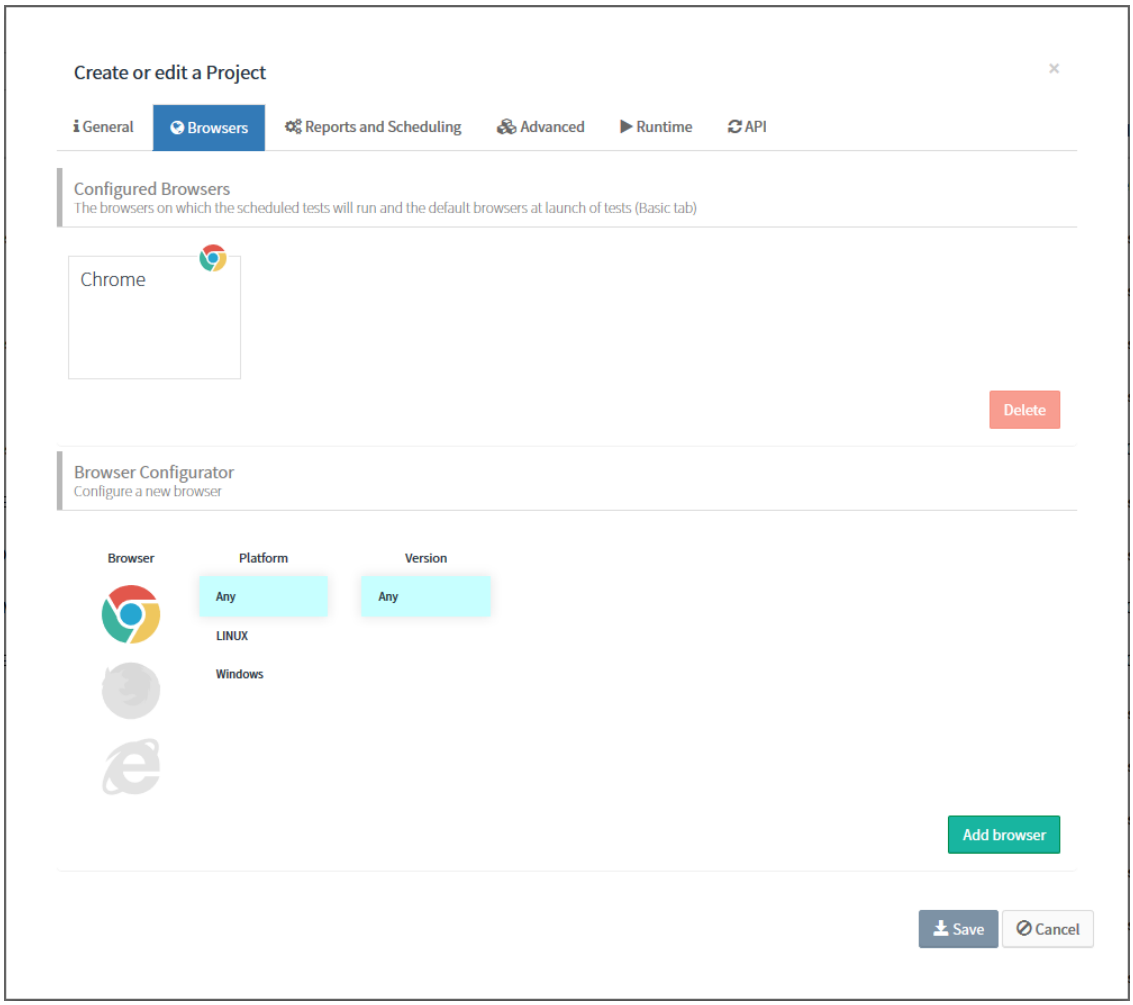


Then, fill the form:



On the `General` tab, fill the following fields:

Name	Display name of the project (e.g gmail-e2e-tests)
Description	Optionally provide a project description
Labels	Associated labels to a project in order to organize and filter projects.
Type	Type of SCM repository
Url	URL of the repository where the project with the features is stored. The following are examples of valid git URL's (or a local file path): https://github.com/github/git.git //dolos/repos/gmail-e2e-tests
Username/Password	Credentials for authentication on the repository.
Branch	The name of the branch you want to execute (GIT only). The default value is `master`



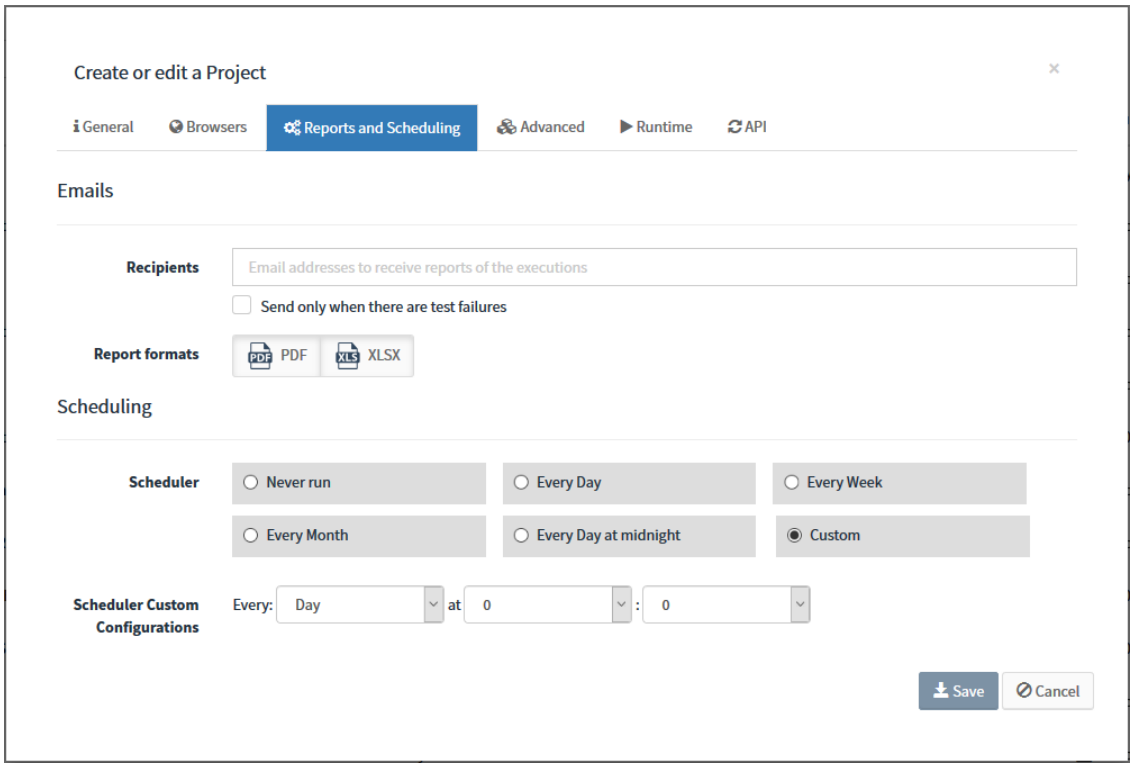
On the `Browsers` tab, choose the browsers on which the scheduled tests will run and the default browsers at launch of tests (Basic tab)

To configure the browsers, click `Browser Configurator` to expand the available browsers and select a browser (and the properties). After, click `Add browser`.

If you want to remove a configured browser, select the browser configuration (below the configured browsers) and click "Delete".

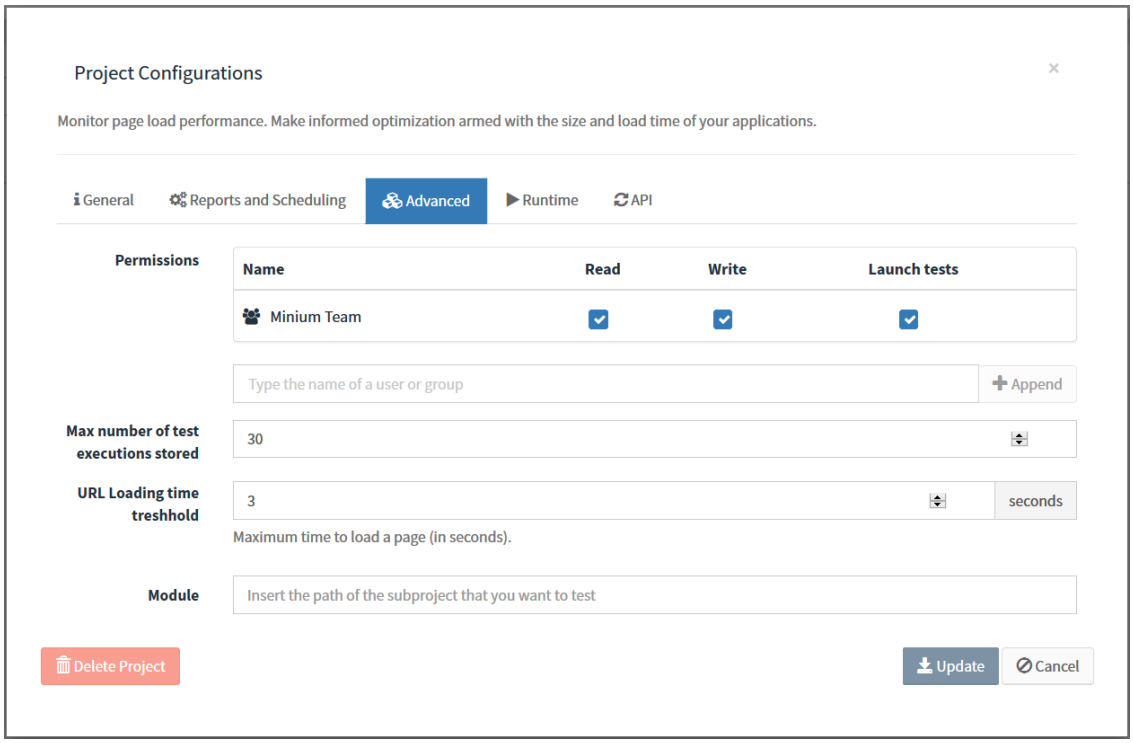
Note:

The `Browsers` tab is not available for the monitoring projects.



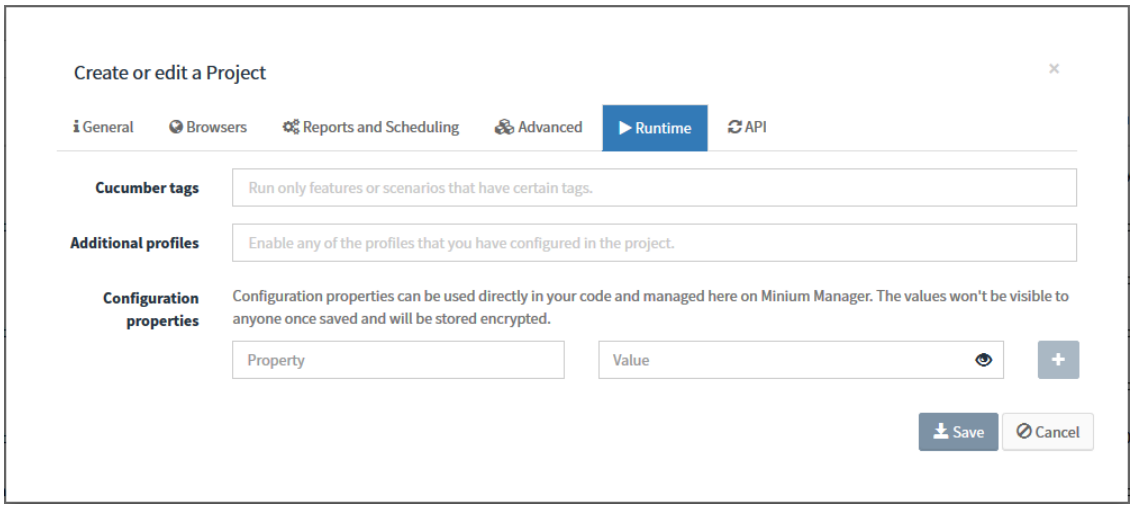
On the `Reports and Scheduling` tab, fill the following fields:

Recipients	Set email addresses to receive a report of execution for the project. If you only want to receive the report of the executions that contains failed tests, check the option "Send only when there are test failures".
Report formats	Set the format(s) that the report of execution will be sent to the email.
Scheduler	Set the schedule you want to execute the project and send the report of the execution. If you choose the option `Custom` you can set a custom schedule.
Scheduler Custom Configurations	Set the schedule you want to execute the project and send the report of the execution with different levels of granularity.



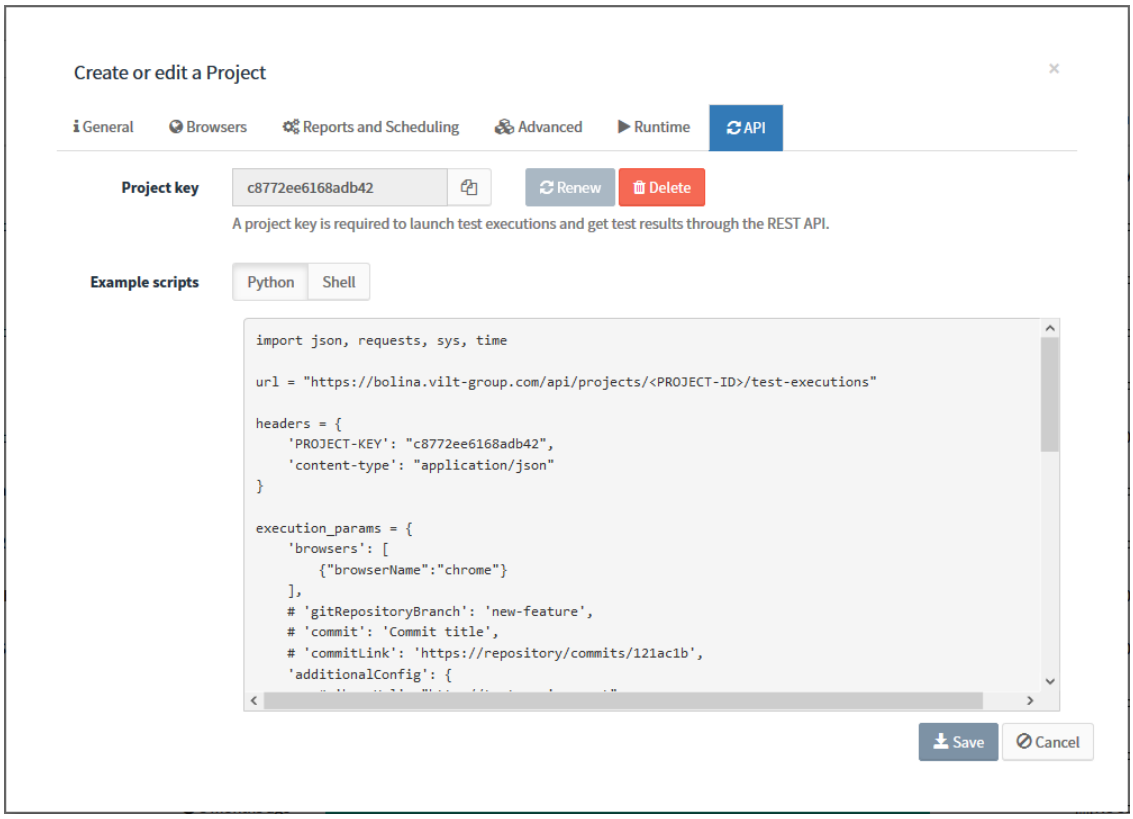
On the `Advanced` tab, fill the following fields:

Permissions	Add and personalize the level of access that assigns to each user or group.
Max number of test executions stored	Set the max number of executions to be stored at Minium Manager.
URL Loading time treshhold	Maximum time to load a page (in seconds) to warning the user at the executions page. **Available only to Monitoring projects**
Module	The path to the folder where the features are in the repository (e.g minium-developer-e2e-tests).



On the `Runtime` tab, fill the following fields:

Cucumber tags	Use this option to tell Minium Manager that only run features or scenarios that have certain tags.
Additional Profiles	Enable any of the profiles that are configured in the `./config/application.yml` project file. See Configuration profiles for more details.
Configuration properties	This configuration properties can be used directly in your code and managed here on Minium Manager. The values won't be visible to anyone once saved and will be stored encrypted.



On the `API` tab, fill the following fields:

Project key	Generate a project key (required to launch test executions and get test results through the REST API).
Example scripts	Once the project has an API key, the scripts will be fulfilled with the project-specific data and ready to use. The first part of the script shows how to launch a test execution.

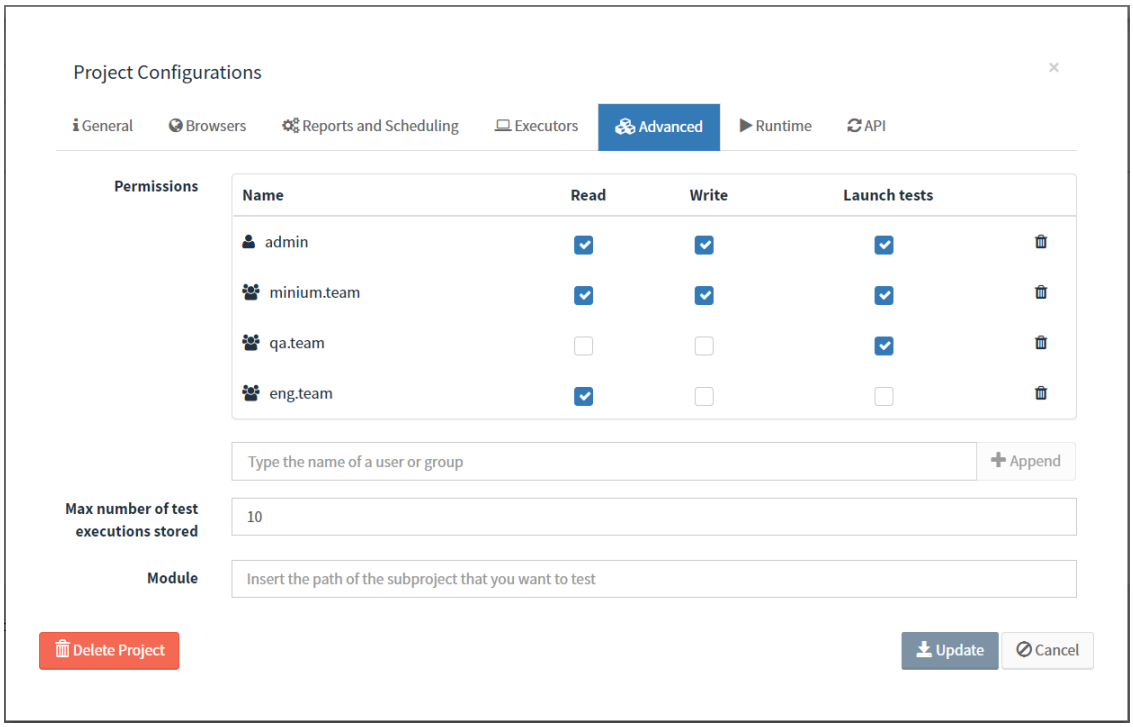
10.3. Project permissions

Manage the access levels of the users/groups in the projects. It is also possible to add and personalize the level of access that assigns to each user or group, per-project.

Note:

Write permissions is required to update the permissions of a project.

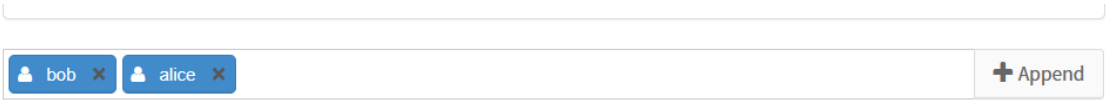
In order to change the project permission the first step is go to the project configuration and click on tab `Permissions`.



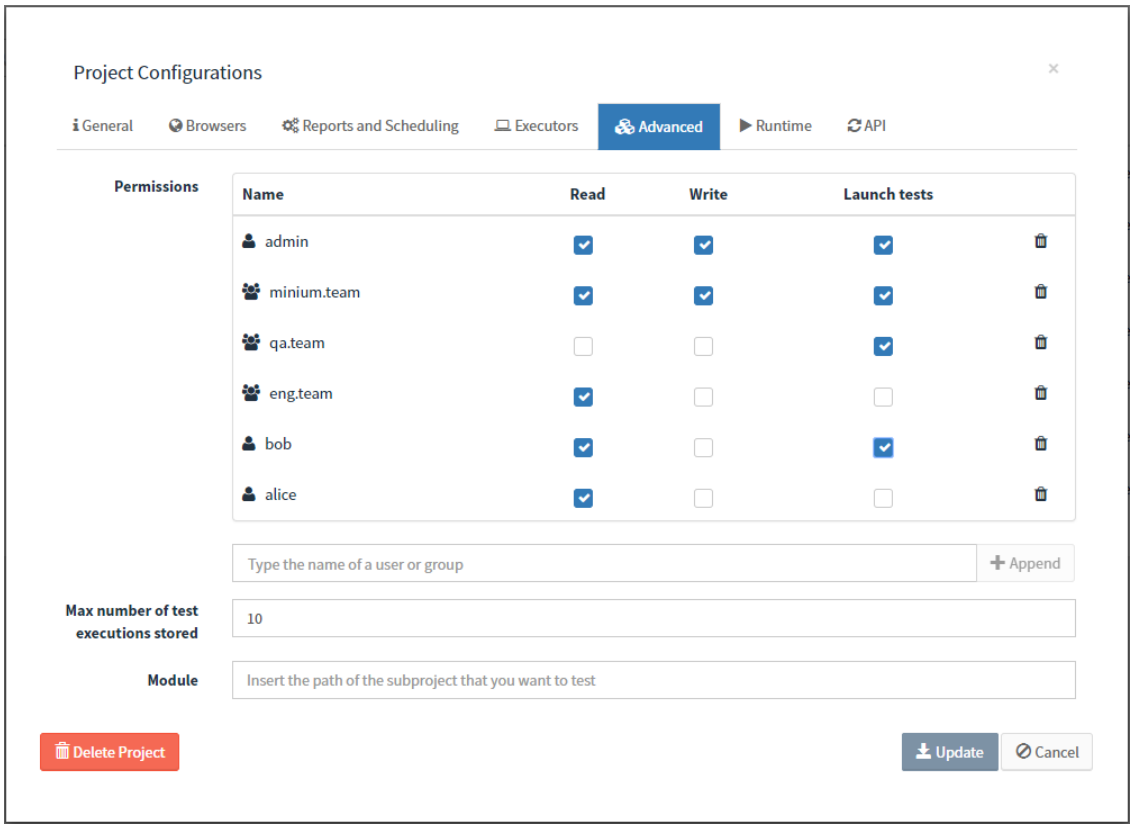
Add user or groups to the permission table. Below the permission table, start typing the name of the user or group desired. And then click on button `Append`.

Note:

It is possible to select more than one user or group.

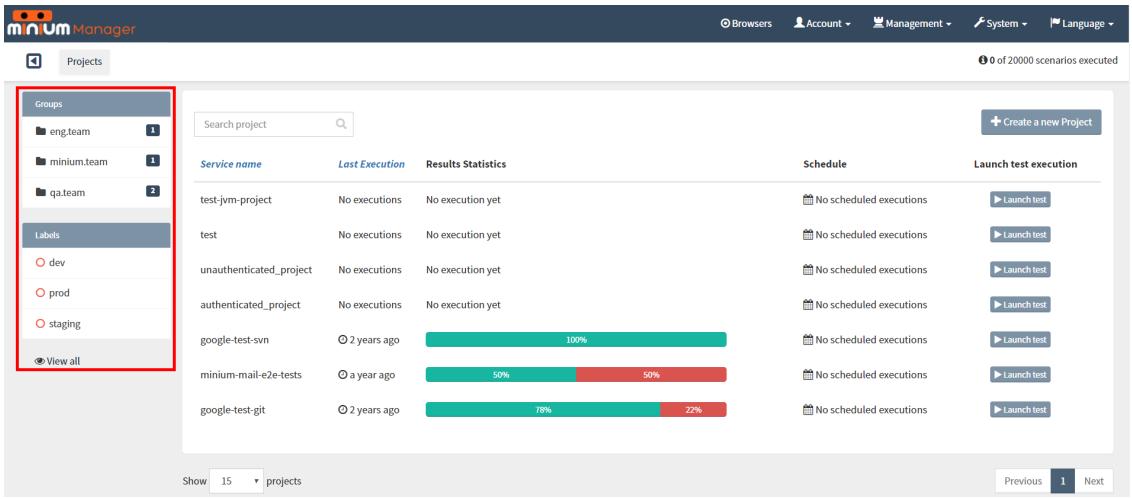


After the users or groups was appended to the permissions table, define the permission for each entry appended and click `Update` to save the project configurations.



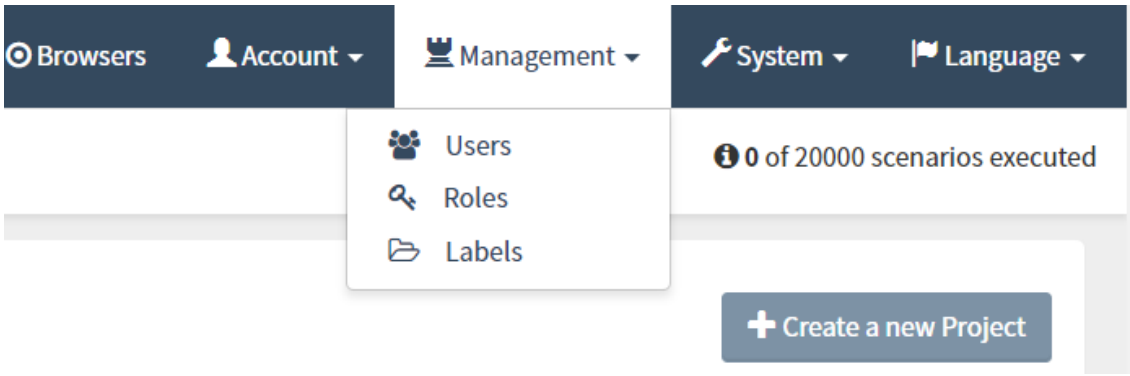
10.4. Organise projects by Labels or Groups

Labels were meant to be used only to filter projects in the Projects page providing a useful way to organize the projects.

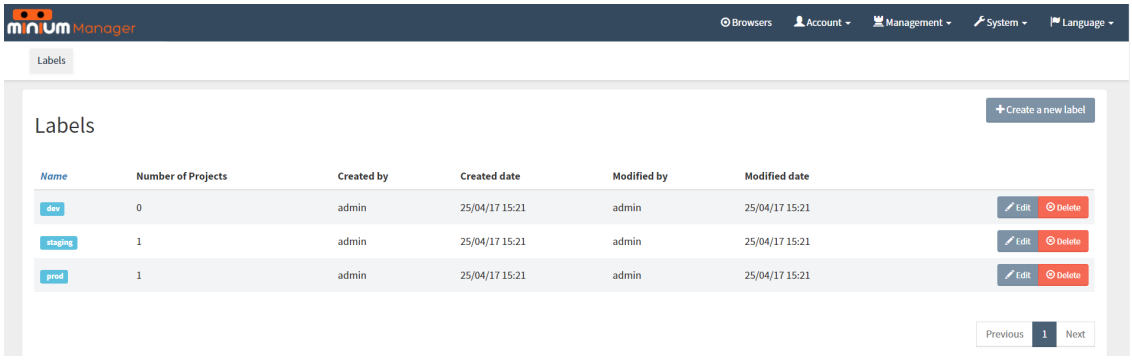


10.4.1. Manage Labels

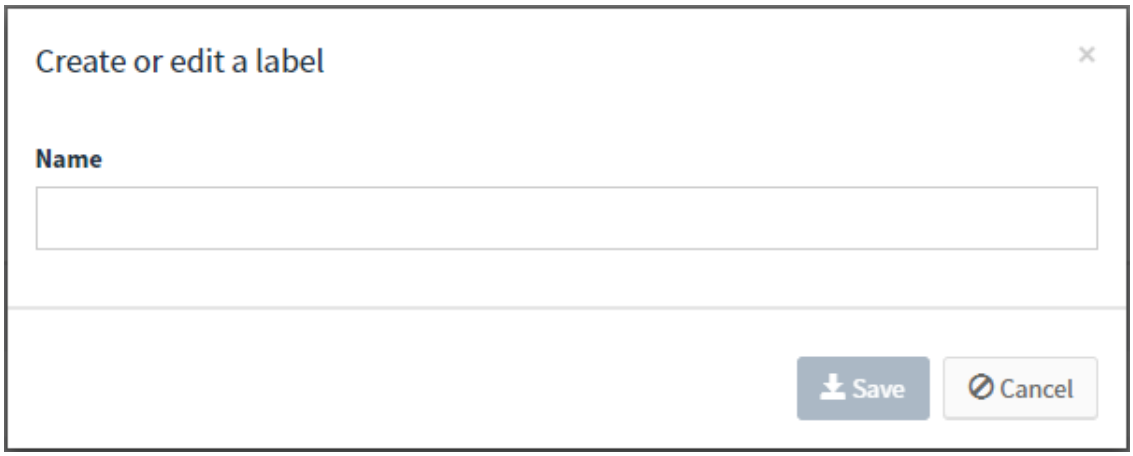
To manage the labels, open the `Management` menu and click on `Labels`:



The Labels page is shown with all current labels. Edit, delete and create new ones if desired.

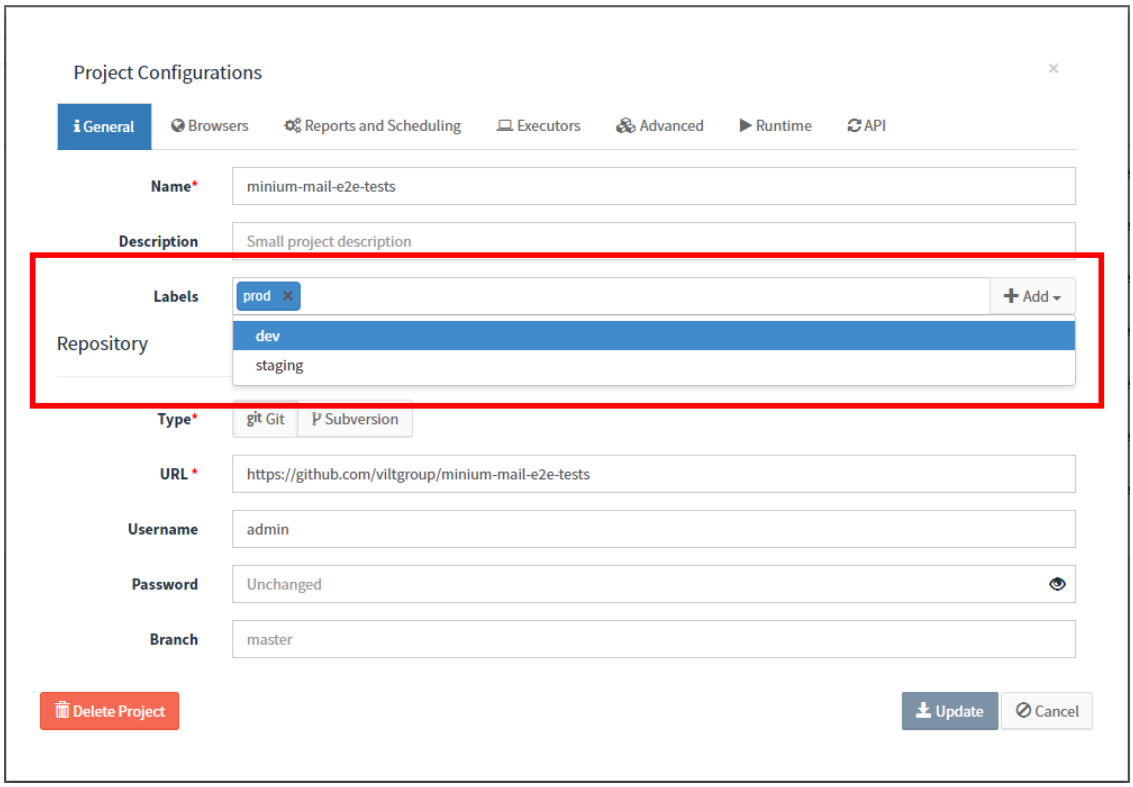


To create a new label, click on `Create a new label`. To delete or edit an existing label, click on `Edit` or `Delete` respectively.

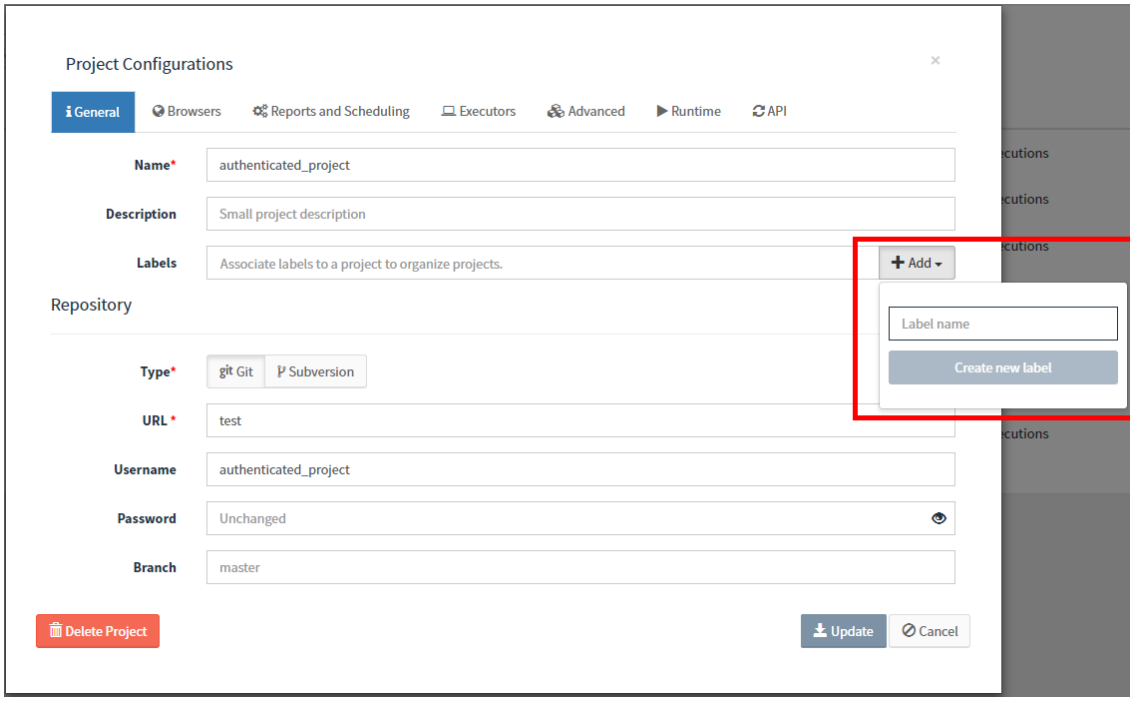


10.4.2. Assign labels to the projects

To assign some labels to a project, go to the project configurations:



The `Add` button can be used to create new labels if needed.



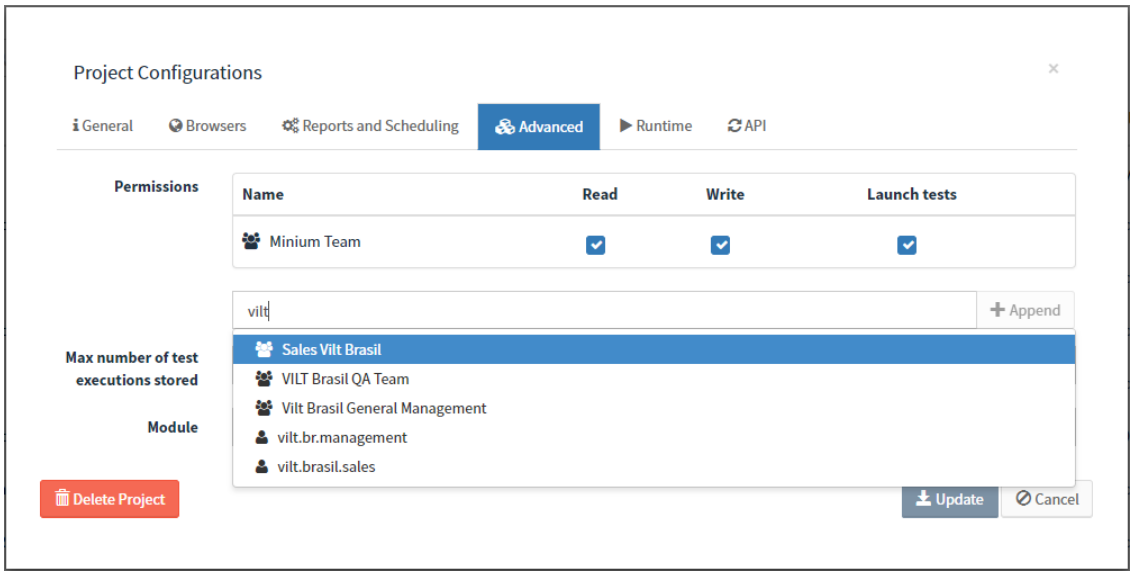
10.4.3. Manage groups

Groups are intended to organize projects at a higher level and to perform changes in several projects at a time.

Manage groups using LDAP authentication

If Minium Manager is using a LDAP to manage the users, the groups are configured through the LDAP.

In the project configuration, search for groups. Below the permission table, start typing the name of the user or group. This will perform a search into the LDAP directory.

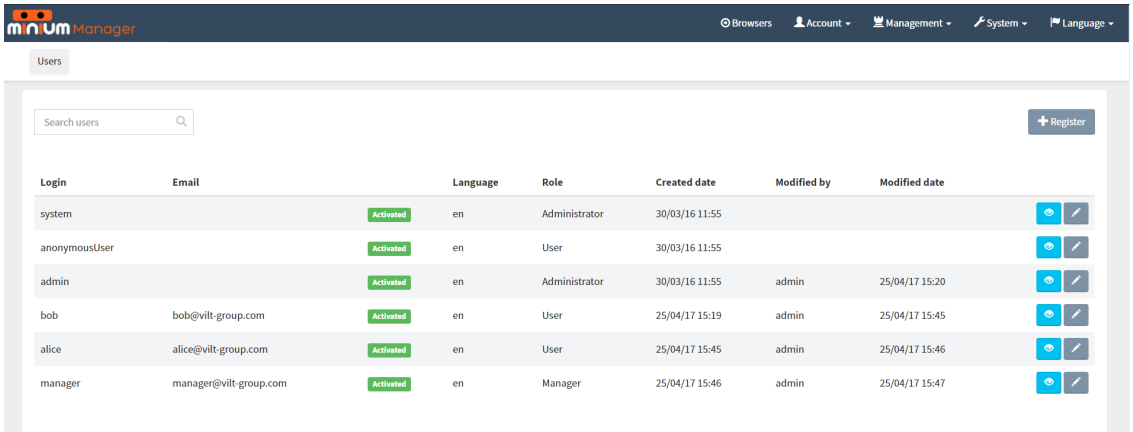


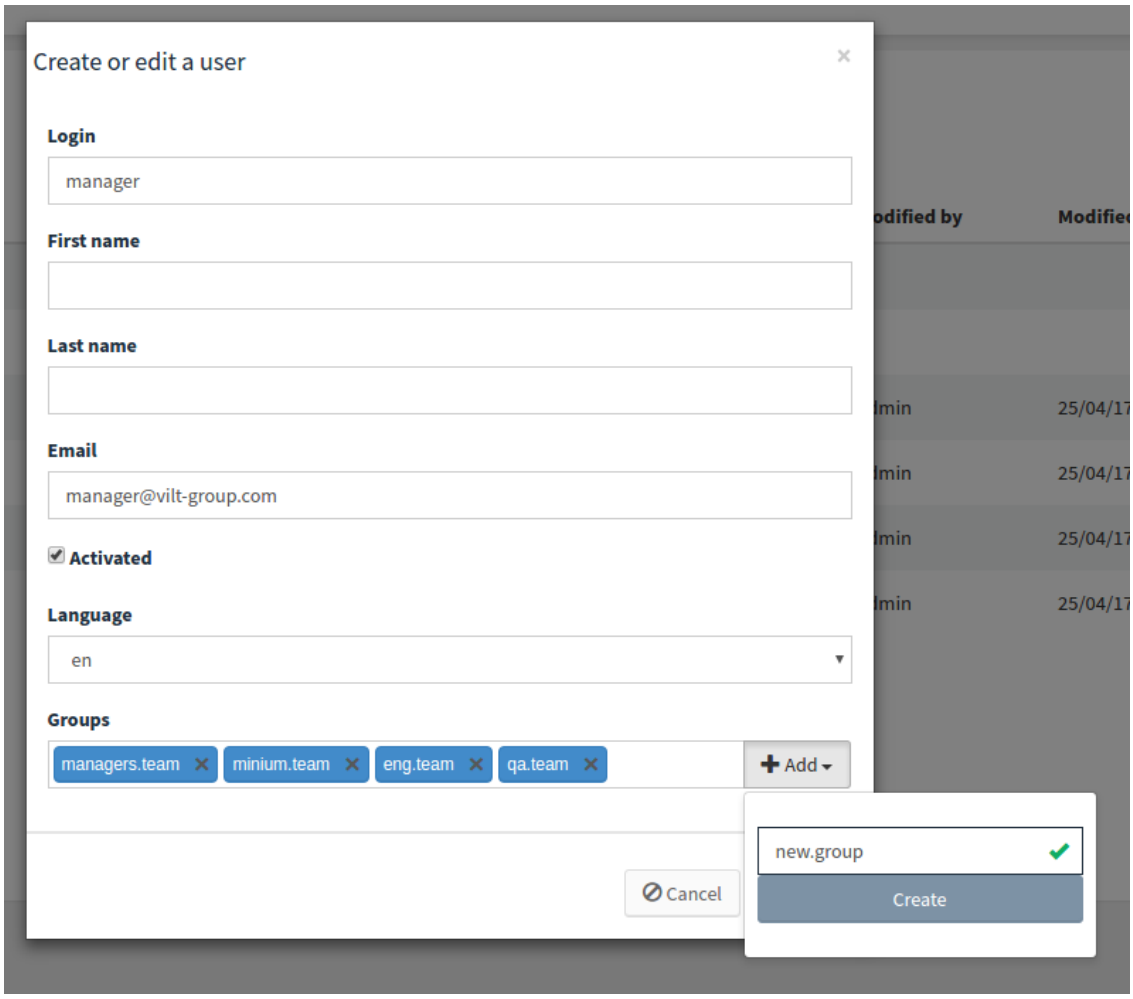
Manage groups without LDAP authentication

If Minium Manager is **not** using LDAP authentication, create new groups through the user management UI.

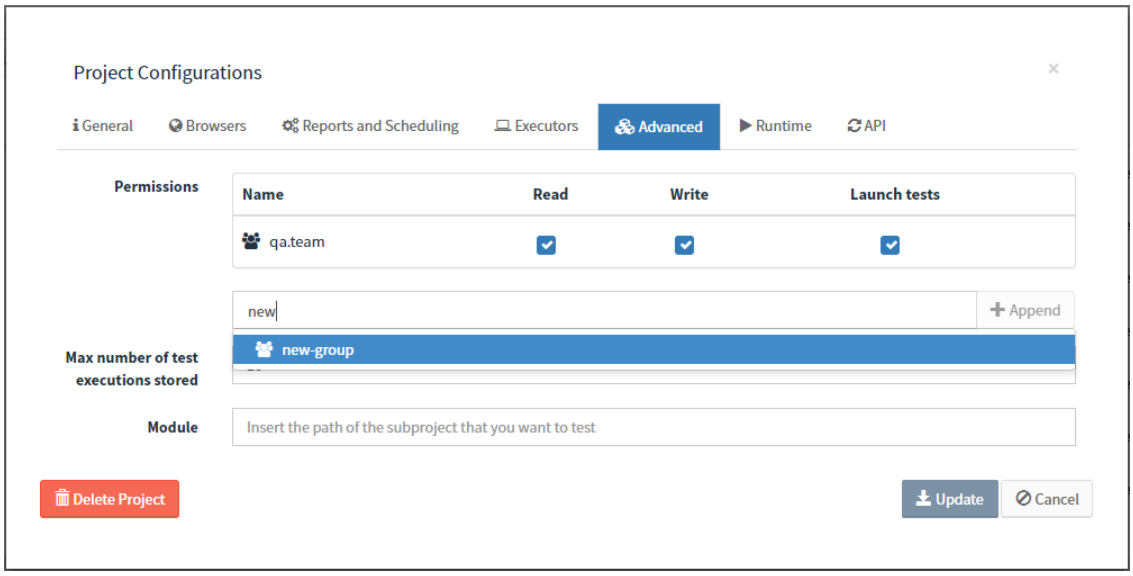
Note:

Only the users with `Admin` privileges are allowed to create groups.



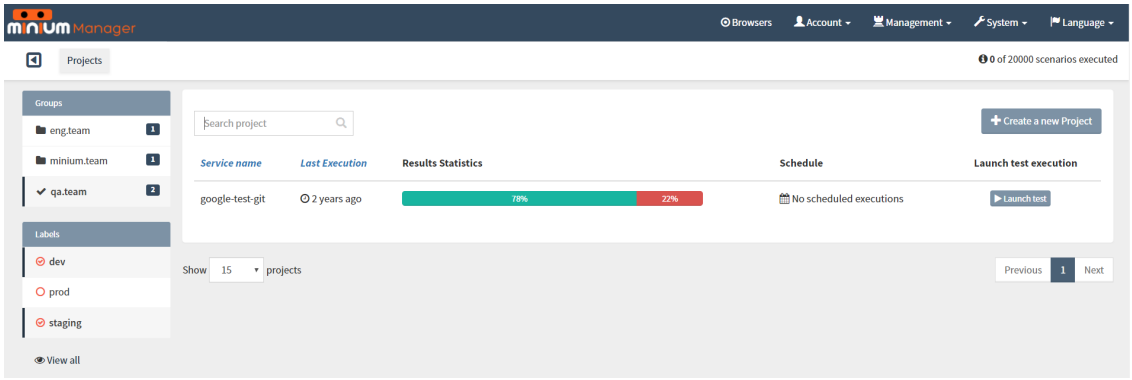


If desired add permissions to the group recently created in the project configuration. Below the permission table, start typing the name of the group created.



10.4.4. Organise Projects

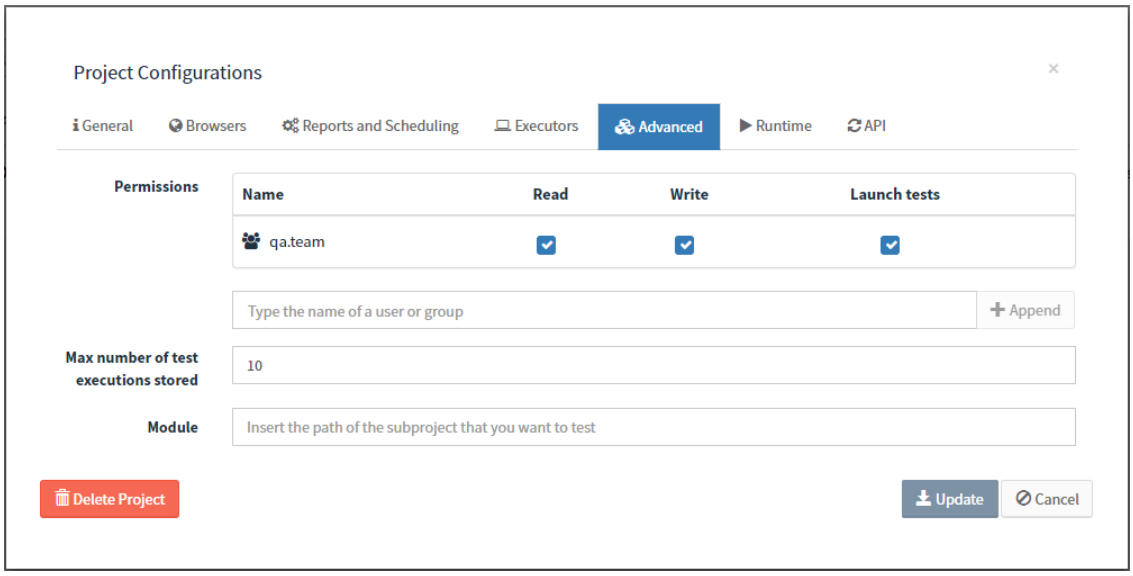
Using **groups** and **labels** to organise the projects selecting a group and labels from the sidebar.



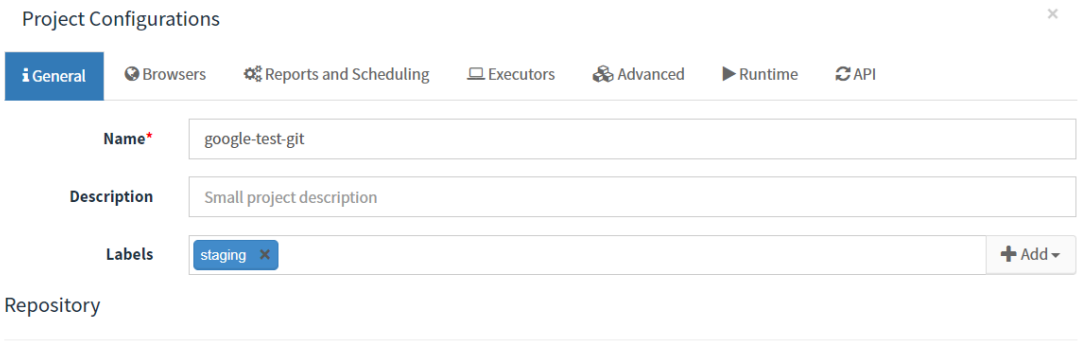
Select one **group** at a time, but select multiple **labels** and combine both.

In the case of the figure above, notice that the **group qa.team** is selected and two **labels** are also selected (**dev** and **staging**). With this filters applied, exists one project shown in the project list.

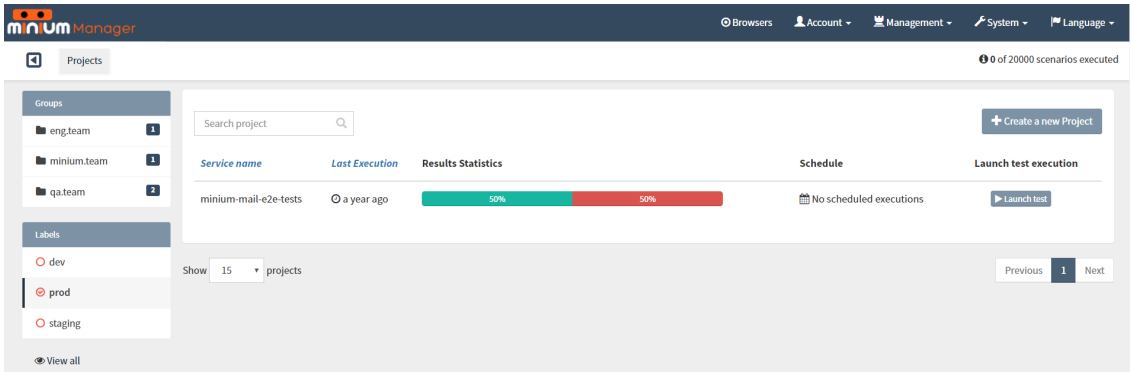
This means that the one project belong to the **group qa.team** (see it in the figure below).



And the project have either the **label dev** or **staging** associated with it in the project configuration (as shown in the figure below).

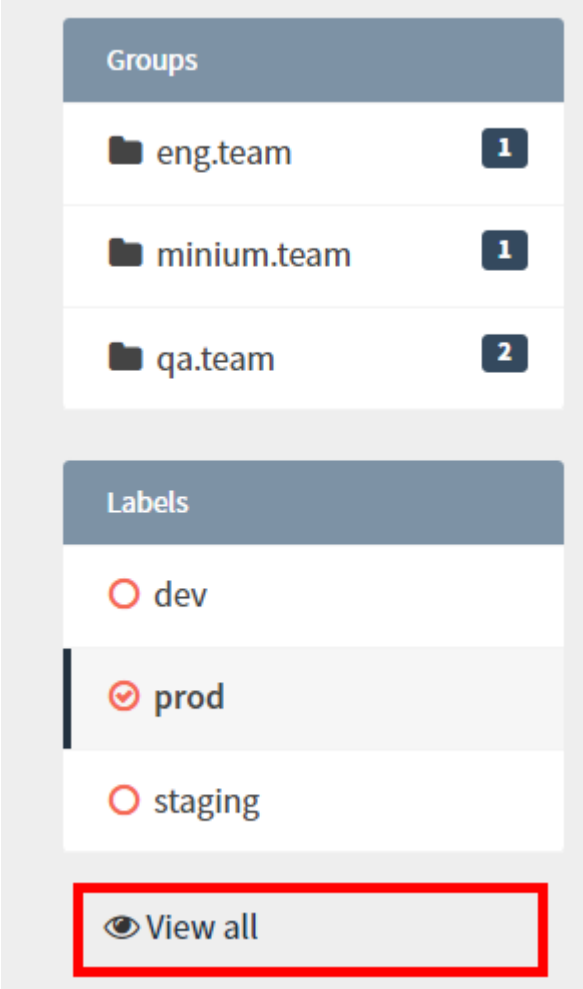


Select **a group** or **one or more label**, like shown in the figure below. In this specific case, are selected only the **label prod**, without selecting any group. It means that all the projects in the list have the **label prod** associated.



Clean project filters

By clicking on the button **View all**, it will remove all the selected filter.



10.5. Configuration profiles

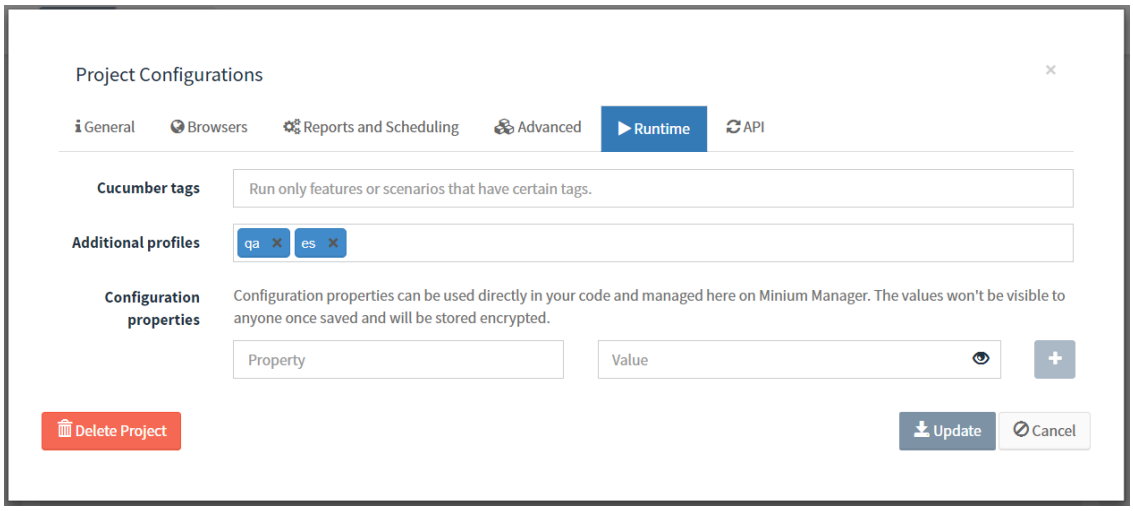
Configuration profiles can be defined on the `config/application.yml` of a project and used to load configuration properties that vary based on the environment being tested. Such configuration could be for example the URL of the application or the language:

```yaml
# default values
minium:
  config:
    baseUrl: http://localhost
    language: English
---
spring.profiles: qa
minium:
  config:
    baseUrl: http://staging
---
spring.profiles: es
minium:
  config:
    language: Spanish
```

These configuration properties can then be used on the step definitions like this:

```
When(/^I go to the homepage"$/, function() {
  browser.get(config.baseUrl);
  $(".dropdown").withText("Language").click();
  $(".dropdown a").withText(config.language).click();
});
```

Then to run test executions with the configuration properties of a profile, go to `Project Configurations` and add it to `Additional profiles`:



10.6. Secret configuration properties

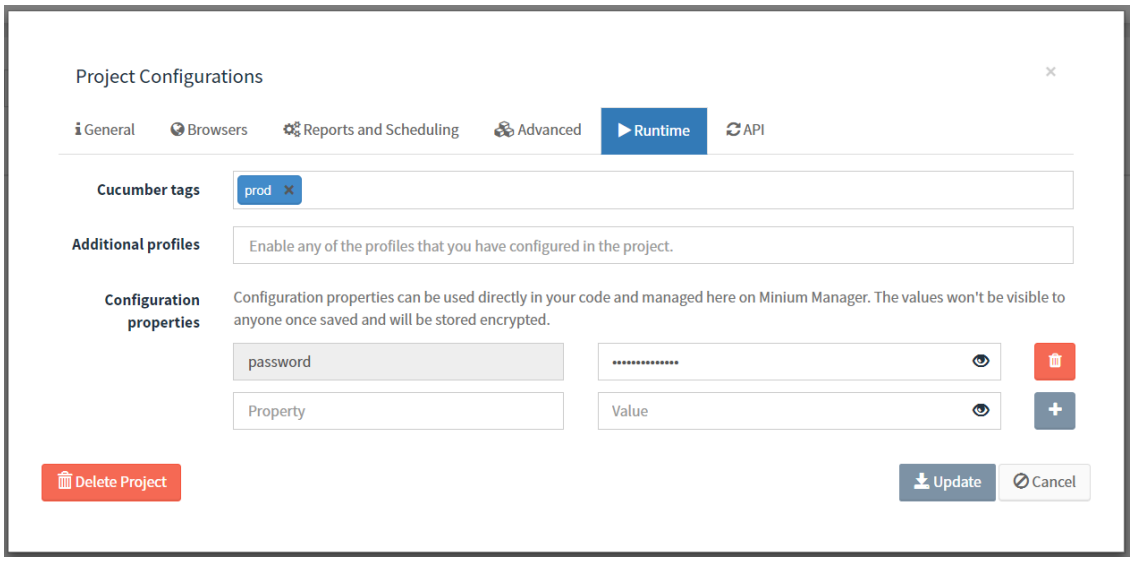
Sensitive configuration properties that cannot be included on the repository of the tests can be safely stored on Minium Manager. They can be used just like the configuration properties defined on the `config/application.yml` file of the project. As an example, consider that some tests need to be executed against a production environment which requires a password that cannot be exposed on the code. In that case, only the password to be used on the test environment would be defined on the `config/application.yml` file:

```
minium:
  config:
    baseUrl: http://staging
    username: test
    password: minium

---

spring:
  profiles: prod

minium:
  config:
    baseUrl: https://production
    username: admin
```

Which would be accessible through the `config.password` during the tests:

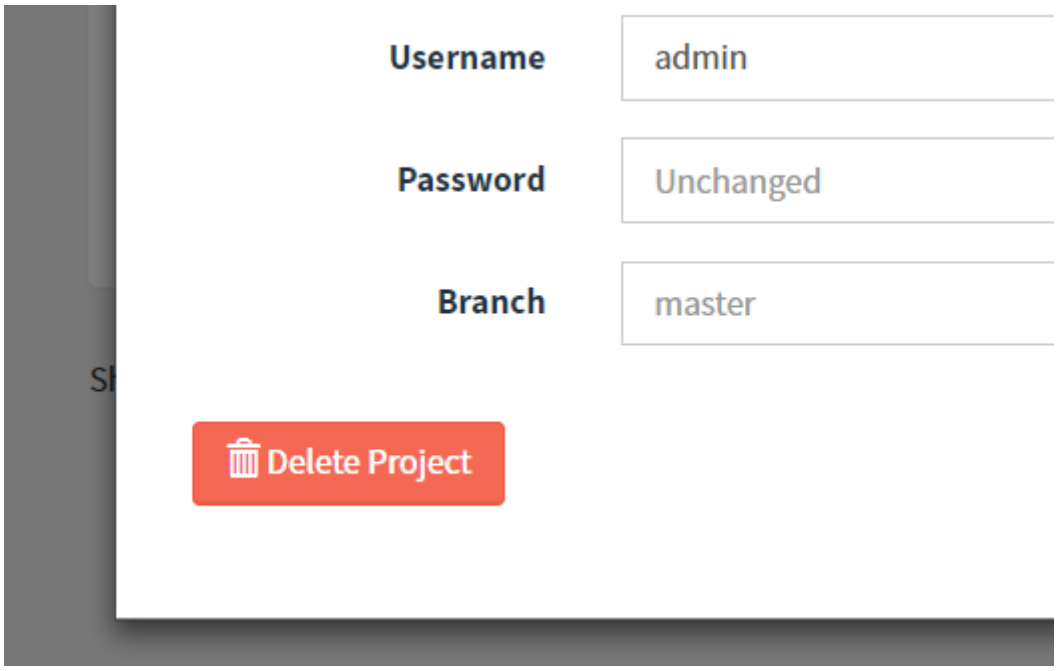
```
When(/^I login"$/, function() {
  browser.get(config.baseUrl);
  $(":text").fill(config.username);
  $(":password").fill(config.password);
  $(":submit").click();
});
```

The password to be used on the production environment would then be defined on the Project Configurations:



10.7. Delete a project

To delete a project, go to the project **configurations** and click on the button `Delete Project`, at the bottom-left corner:

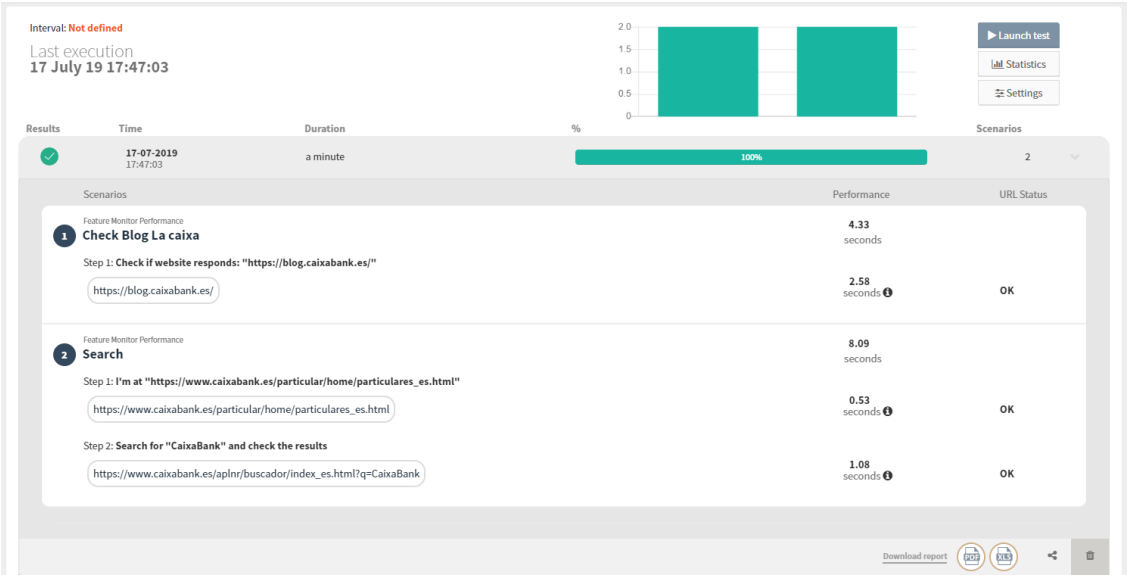


10.7.1. Click on a link and get the performance of the page

```
When(/^I click on link with text "([^"]*)"$/, function(text) {
 $("a").withText(text).click();
 expect($(".page-title")).to.exist();
 scenario.write(browser.getPerformance());
});
```

10.7.2. Monitoring Report

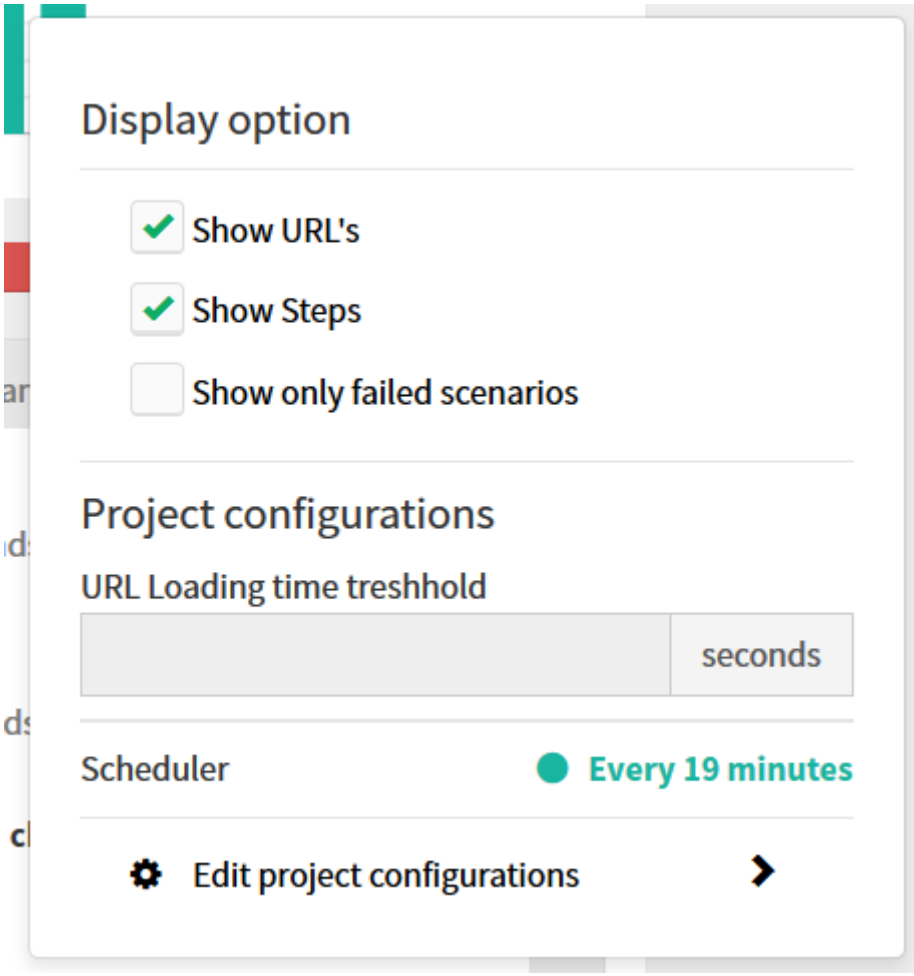
The result of an execution of a Monitoring project is presented at the execution page:



At execution page, we can see the time of each URL took to load. Also we can see the URL status and the time that took to execute the scenario.

10.7.3. Display options

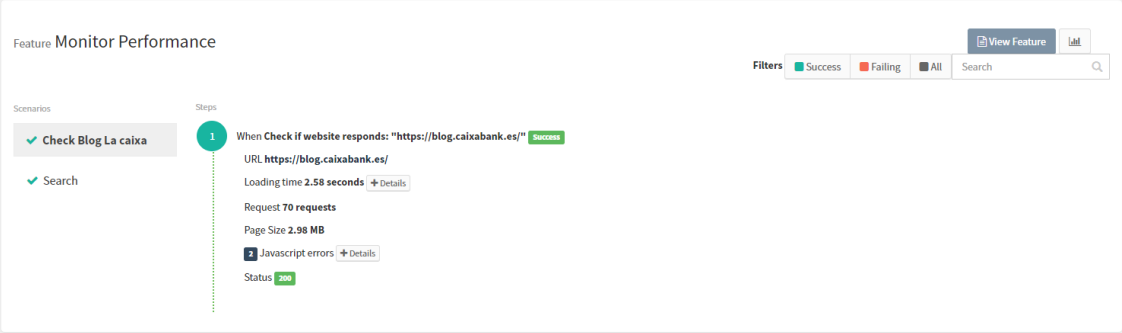
The information at the execution page can be filtered to show/hide some details.



The display options allow us to:

- Show/Hide the URL's
- Show/Hide the steps
- Show only the sailed scenarios
- Consult the URL loading time threshold (This property is configured at the project configurations at the advanced tab and warns the user (at the execution page) when the loading time of an URL is higher than the loading time threshold)
- Consult the scheduler (The scheduled is configured at the project configurations at the reports and scheduling tab)

Feature page of a Monitoring project scenario:

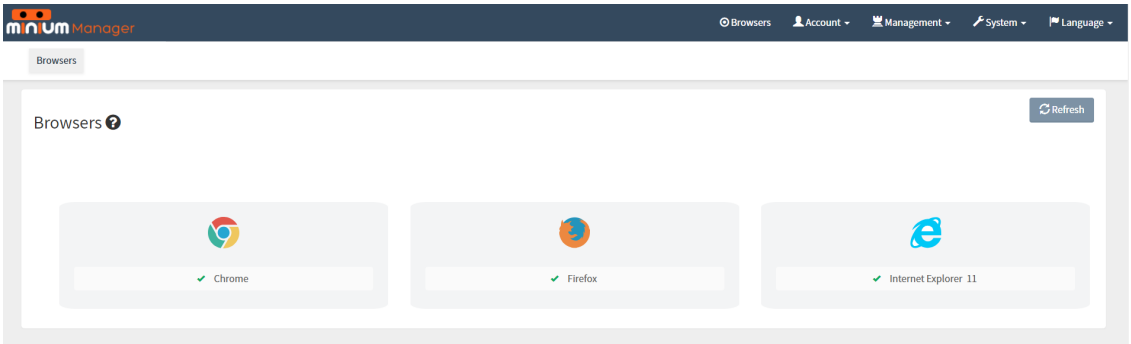


At the feature page, we can see more details about the page load performance:

- The url loaded
- The loading time with the times of the Backend Performance, Frontend Performance and DOM Content Loading
- The number of requests
- The Javascript Errors (if any)
- The URL status code

11. Check the available browsers

To check the available browsers, click on `Browsers`, at the navigation bar.

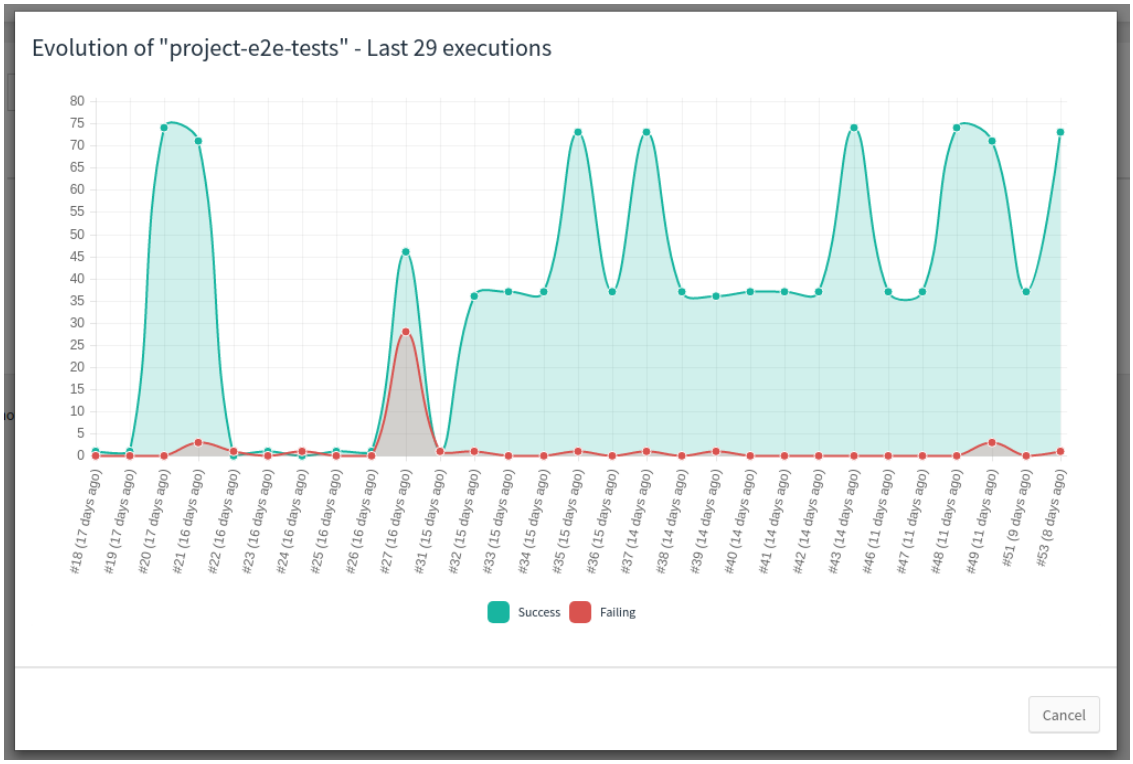


In the figure above, we only have one provider available, but we can have two or more providers configured (e.g. "BrowserStack" and/or "Selenium Grid Extras"). The provider configured has the browsers Chrome, Firefox and Internet Explorer available.

This means that with this environment is able to run the tests in Chrome, Firefox and Internet Explorer at the provider configured.

12. Project Statistics

Evolution chart to the test executions of a project.



Statistics for a single execution.


Execution: "project-e2e-tests #53" (8 days ago)
Scenarios
Failing: 1
Success: 73
Success
Failing
Browsers
Chrome
Profiles : chrome
Success: 37
Success
Failing
Firefox
Profiles : firefox
Failing: 1
Success: 36
Success
Failing
Close

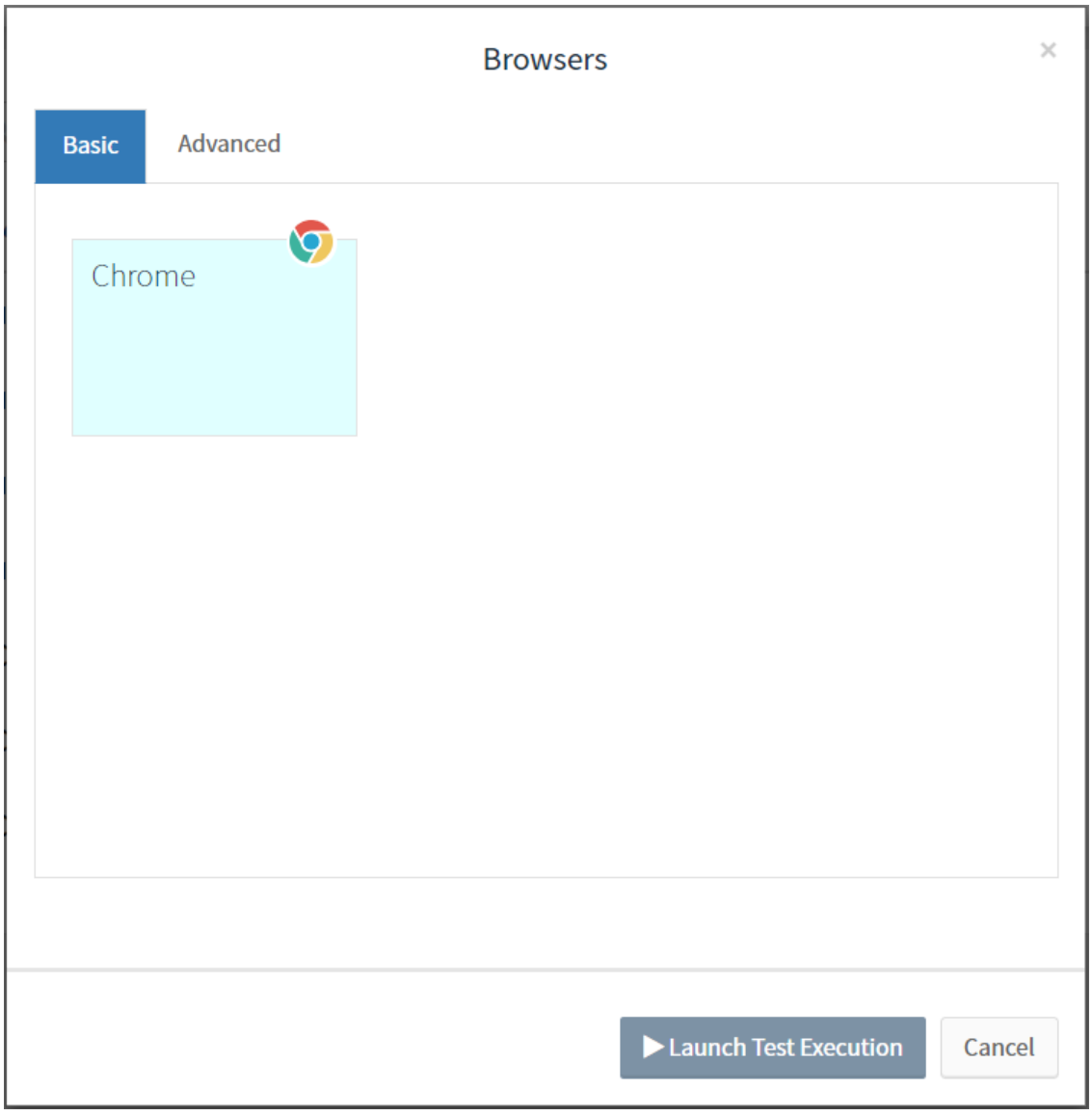
13. Test executions

13.1. Launch test executions

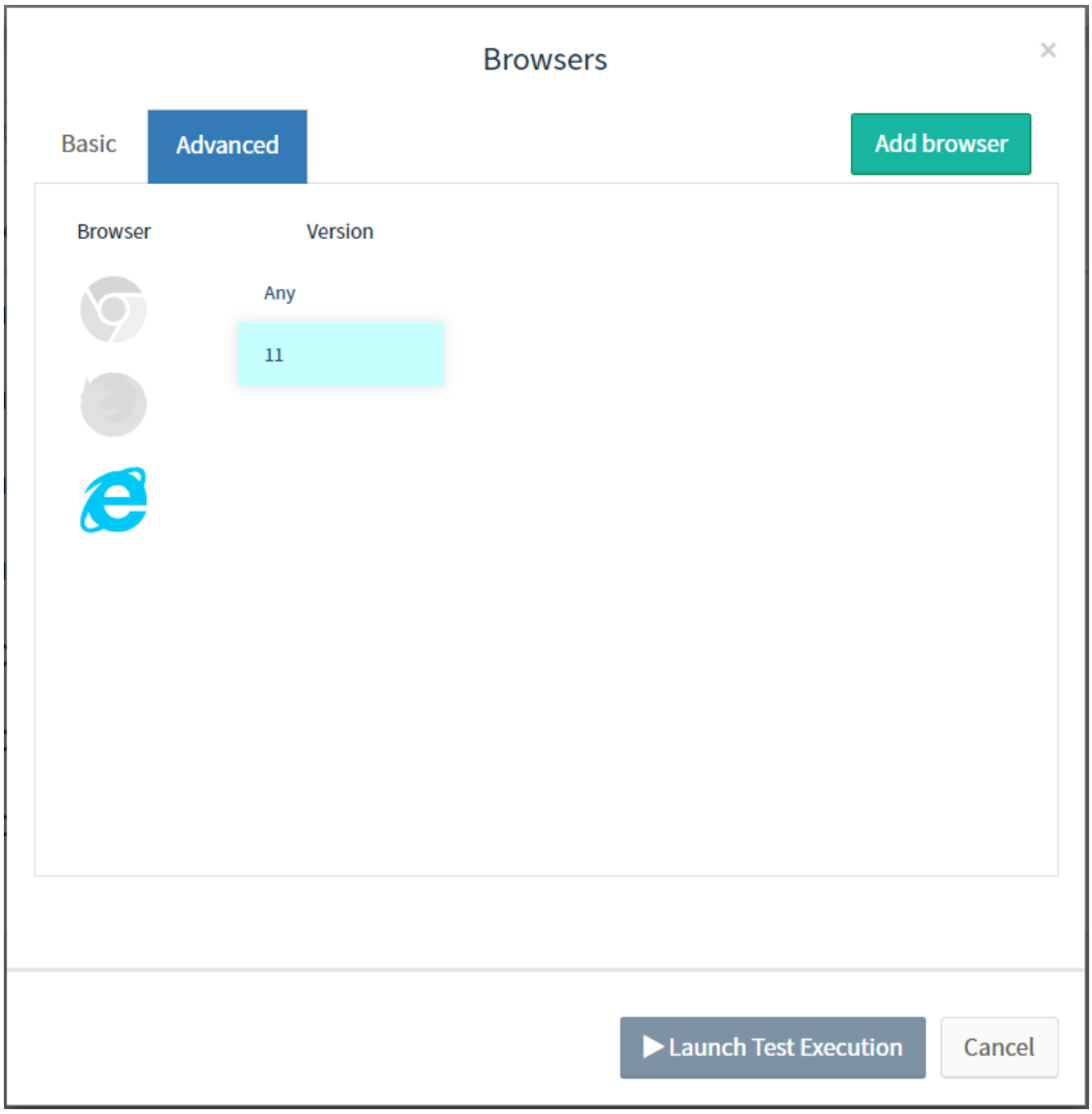
A test execution will execute the tests present in a project on one or more different browsers. There are alternative ways for launching test executions.

13.1.1. Launch a test execution manually

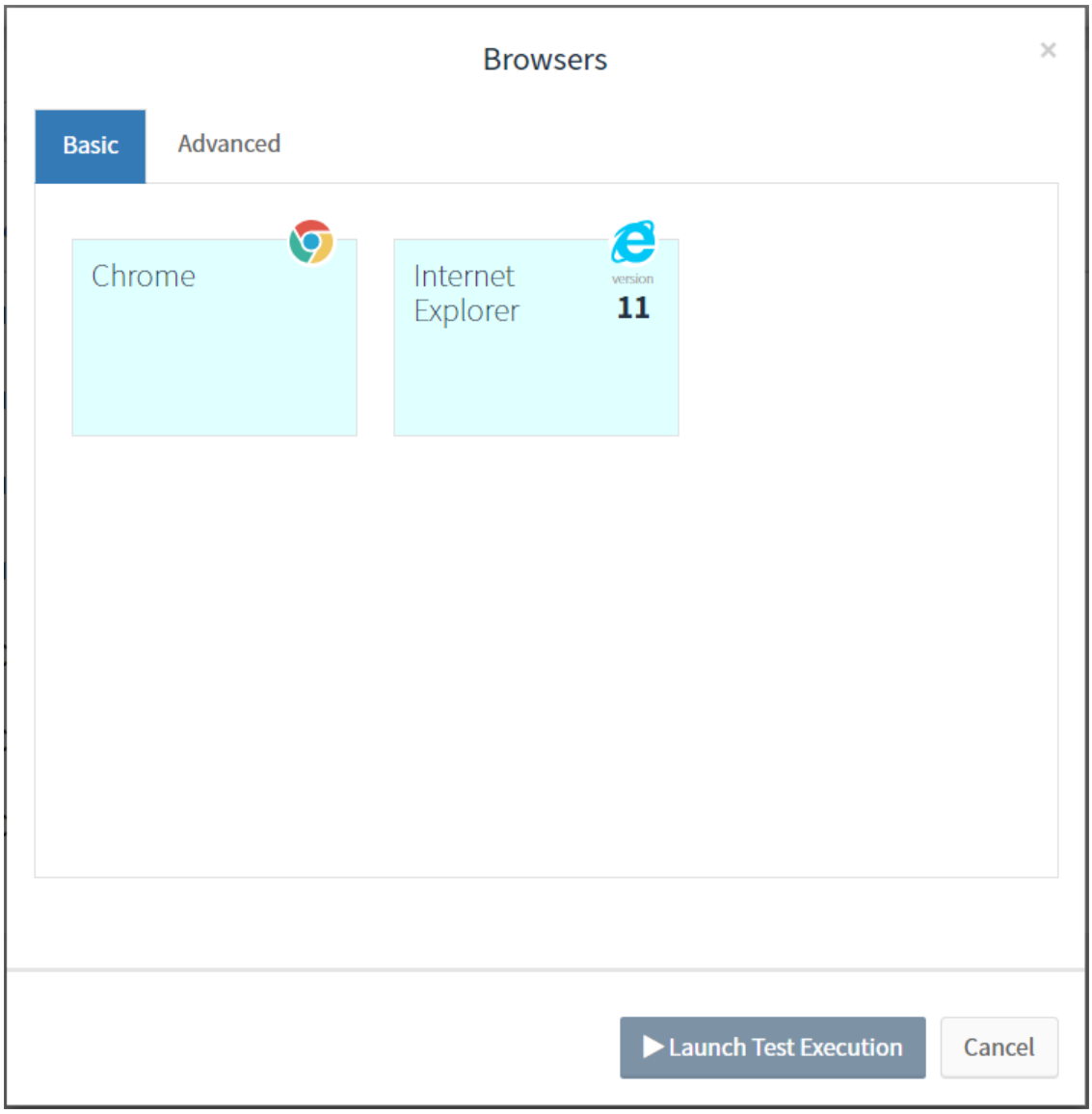
On the homepage or the page of a project, click on button `Launch test`, which opens a modal for selecting the browsers.



Select the desired browsers by clicking on them. If needed, click on the `Advanced` tab to configure a new one. To configure a browser, click on the corresponding icon and choose its properties. Then, click on `Add browser`.



The new browser is now available to be selected on the `Basic` tab.



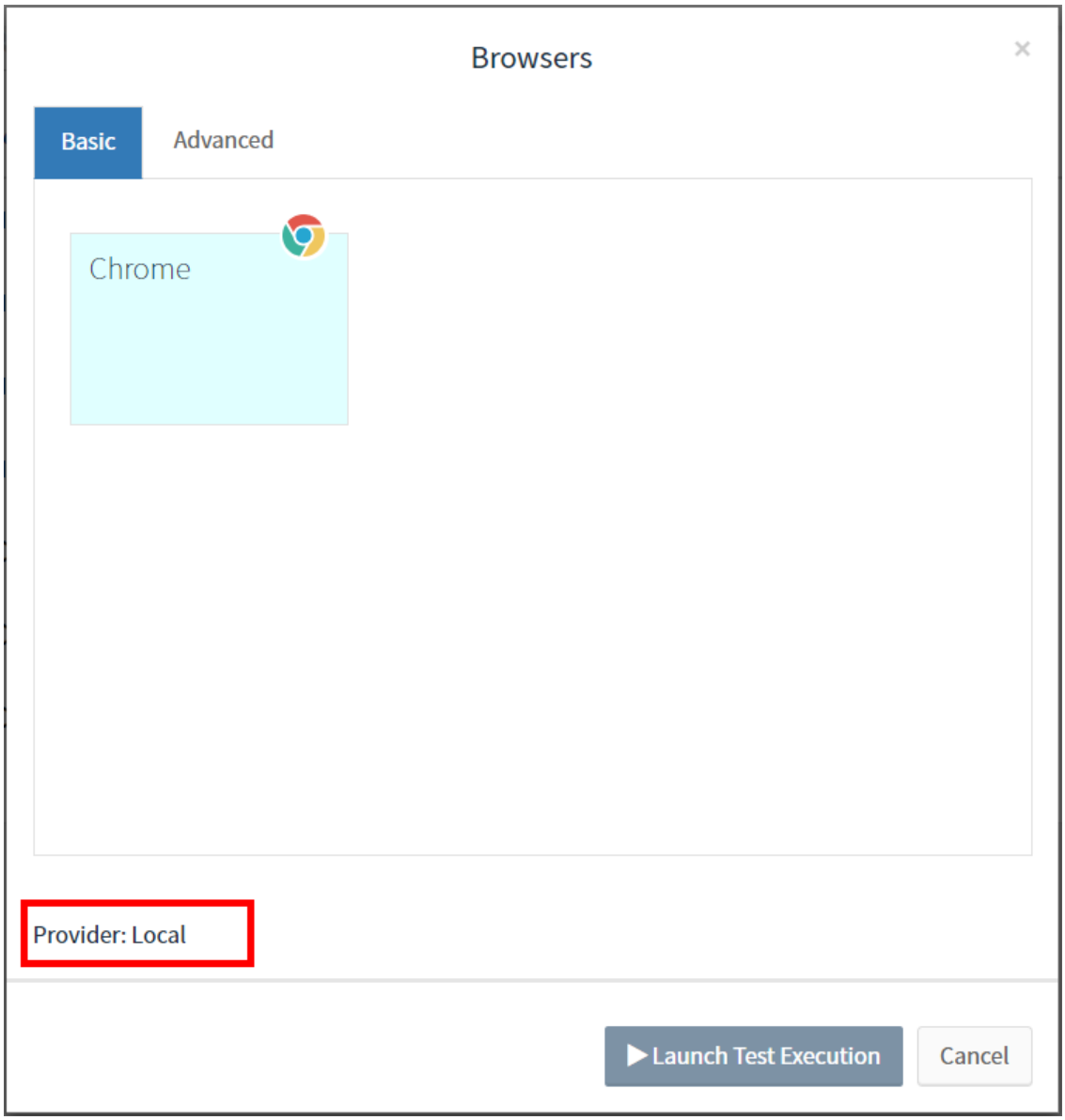
After selecting all the browsers, click on `Launch Test Execution`.

Note:

At click on button `Launch test`, the Monitoring project don't open a modal for selecting the browsers. It launchs a test immediately.

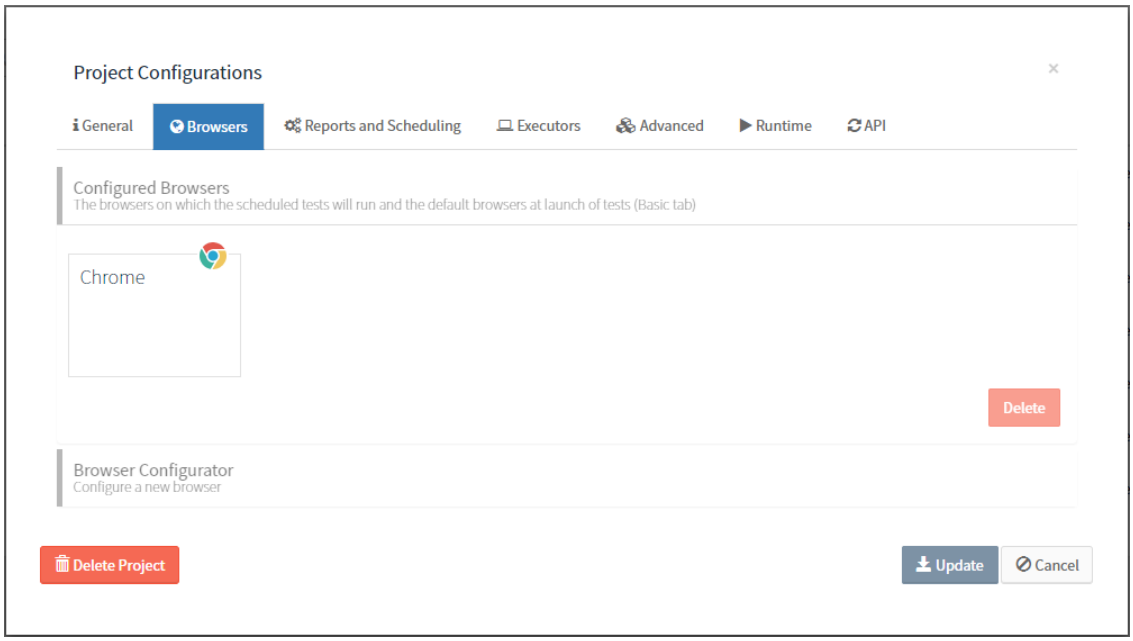
Multiple Providers

With multiple providers, the Launch modal presents the provider configured for the project.

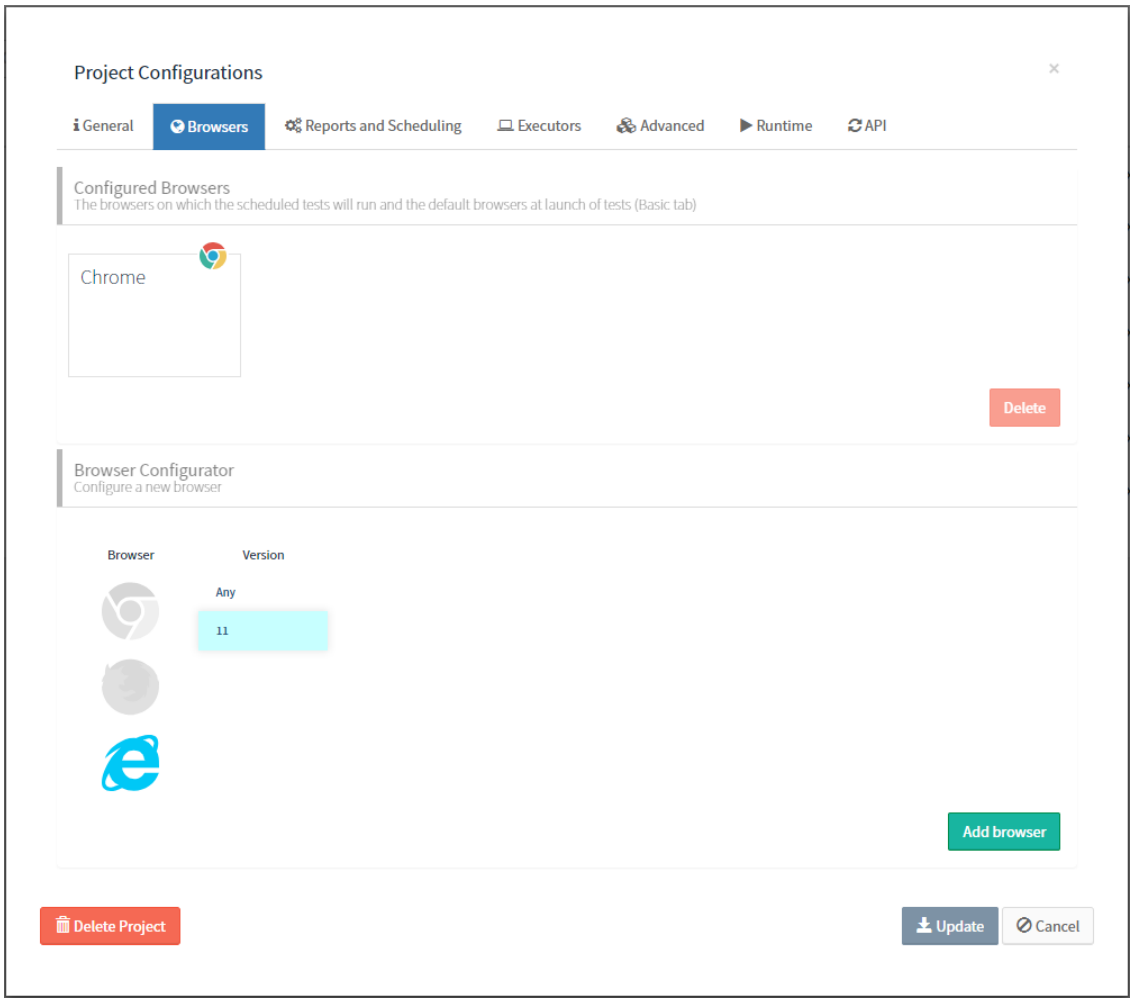


13.1.2. Schedule test executions

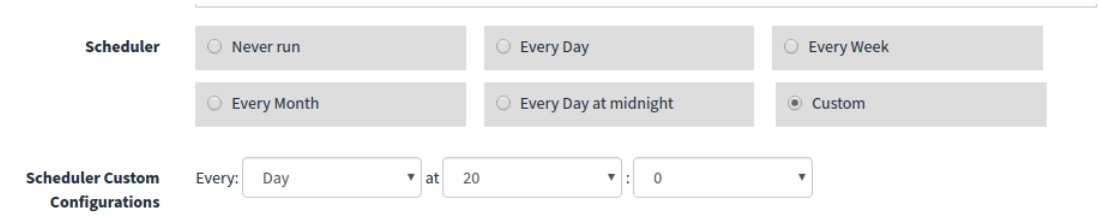
Go the configurations of a project and open the `Browsers` tab to see the currently configured browsers.



Click on `Browser Configurator` and add more browsers if needed.

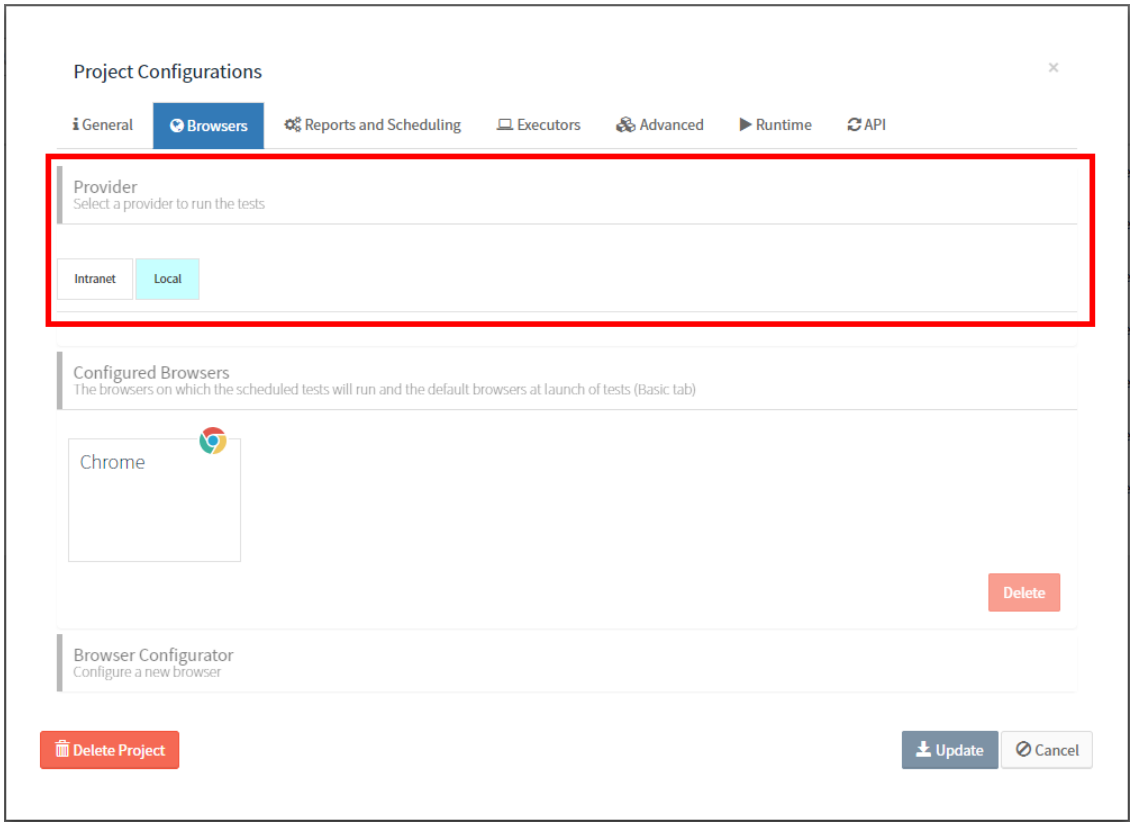


To schedule test executions on the configured browsers, switch to the `Reports and Scheduling` tab.



Multiple Providers

With multiple providers, the `Browsers` tab has an extra field to configure the default provider.

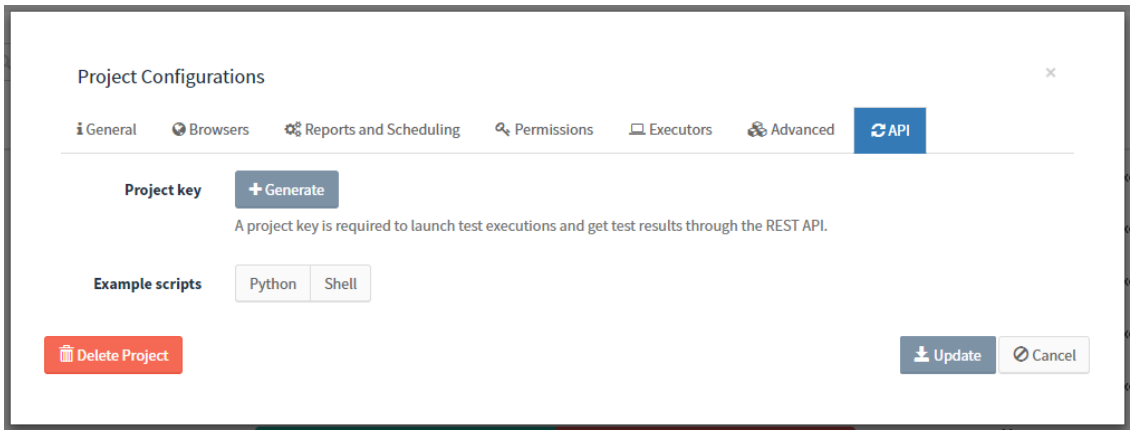


Note:

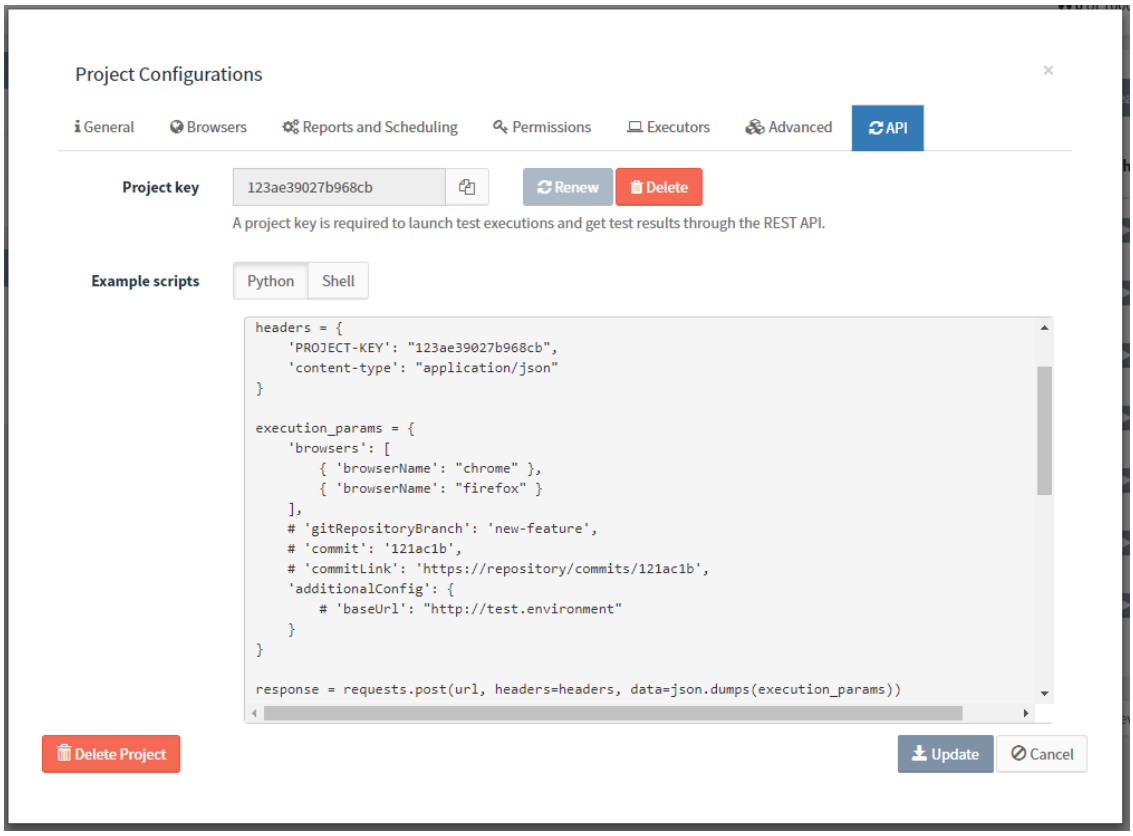
This configuration is not available at the Monitoring projects.

13.1.3. REST API

Minium Manager provides a REST API through which it is possible to launch test executions and get test results. In order to use it, an API key is required. API keys are specific of a project. To generate a key, go to the `API` tab on the configurations of a project and click on the `Generate` button.



Also on the `API` tab are some example scripts. Once the project has an API key, the scripts will be fulfilled with the project-specific data and ready to use. The first part of the script shows how to launch a test execution.



A test execution can be launched by issuing a `POST` request to `api/projects/<PROJECT-ID>/test-executions` containing the `PROJECT-KEY` header set to the project key and the `content-type` header set to `application/json`. The execution parameters are sent in JSON on the request body:

- **browsers**: list containing the capabilities of the browsers. Only the `browserName` is mandatory.
- **additionalConfig** (optional): configuration properties to be merged with the ones on the `application.yml` file of the project. Properties already defined on the

`application.yml` file are overridden.

- **gitRepositoryBranch** (optional): branch of the repository of the tests.
- **commit** (optional): the corresponding commit of the system under test, to be then included on the report of the execution.
- **commitLink** (optional): link to directly access the commit from Minium Manager.

To follow the progress of the execution until it finishes, poll the URL returned on the `Location` header of the response to the `POST` request.

```
url = response.headers['location']
response = requests.get(url, headers=headers)
while response.status_code == 404:
    time.sleep(5)
    response = requests.get(url, headers=headers)
if not response.ok: sys.exit(response.text)

execution = response.json()
while execution['state'] != "FINISHED":
    time.sleep(15)
    execution = requests.get(url, headers=headers).json()
```

While the tests are running, the response will be a JSON object containing the `state` field set to `RUNNING`, the name of the current browser and the corresponding progress in percentage, number of passed/failed and last finished feature/scenario. Example:

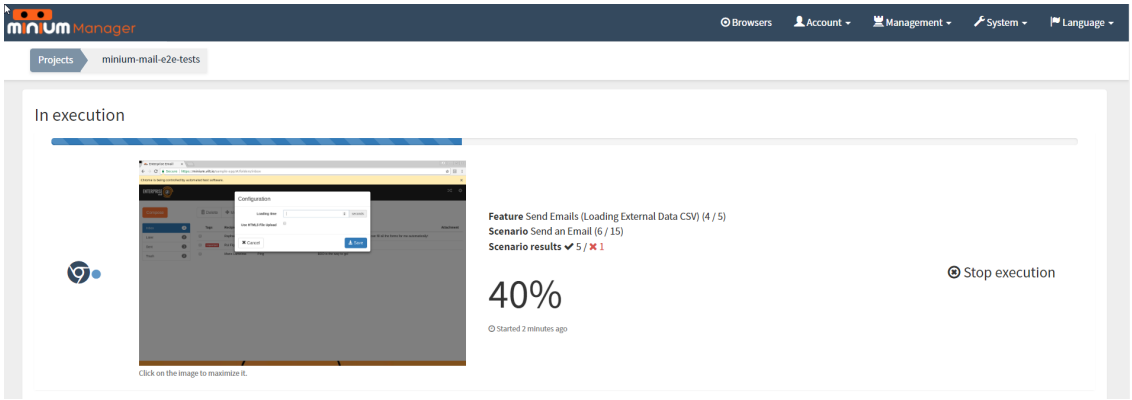
```
{
  "state": "RUNNING",
  "browser": "chrome",
  "progressInPercentage": 18,
  "failingScenariosCount": 1,
  "passingScenariosCount": 5,
  "feature": "Login",
  "scenario": "Successful login"
}
```

Once the execution finishes, the response will have `state FINISHED`, the global results and the number of passed/failed scenarios for each browser. Example:

```
{
  "state": "FINISHED",
  "globalResults": {
    "totalScenarios": 20,
    "percentageOfPassingScenarios": 95,
    "passingScenarios": 19,
    "failingScenarios": 1
  },
  "browserResults": [
    {
      "browser": {
        "browserName": "chrome"
      },
      "passedScenarios": 10,
      "failedScenarios": 0
    },
    {
      "browser": {
        "browserName": "firefox"
      },
      "passedScenarios": 9,
      "failedScenarios": 1
    }
  ]
}
```

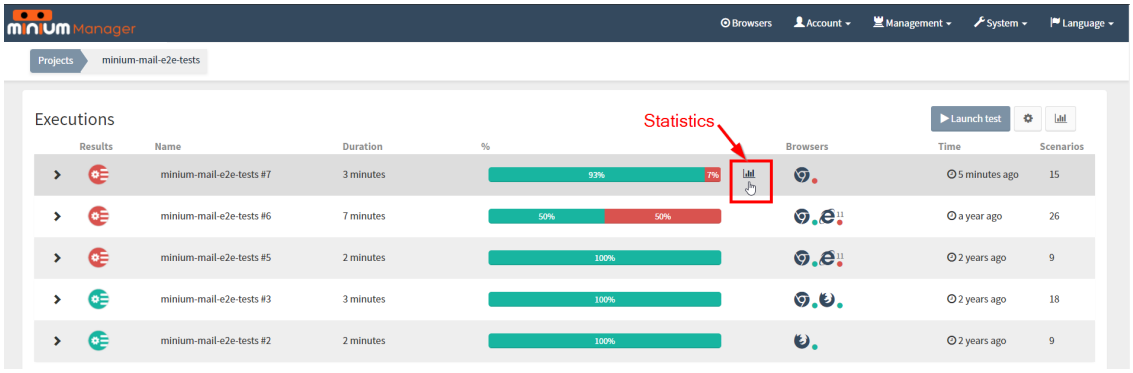

13.2. Follow the progress of a test execution

After launching a test execution, follow the progress in the project page. See the browser where the test is executing, the percentage, number of executed tests, the elapsed time and the test that is currently executing. Also, visualize the test be executed in the browser.



14. Monitor test results

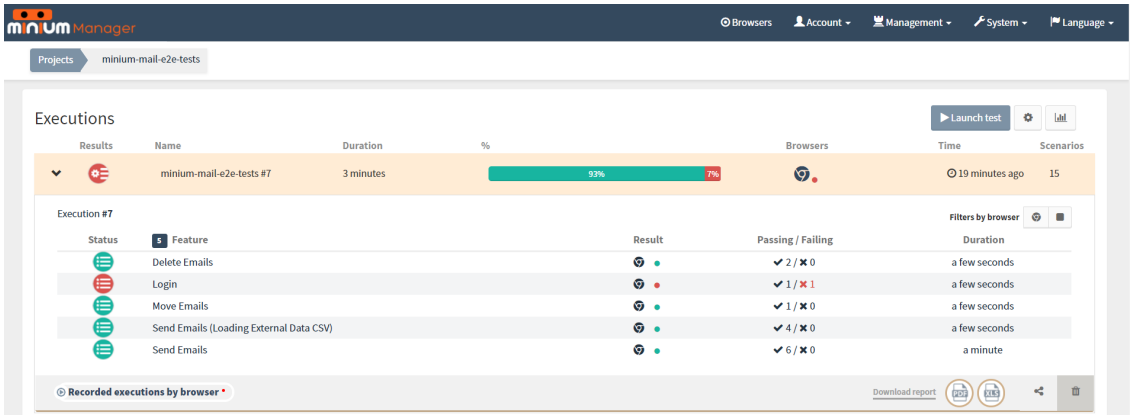
In a project page, all the launched executions are showed:



For each execution it can be seen:

- result of the execution
- name
- duration of execution
- percentage of passed and failing scenarios
- statistics of the execution (shown when the mouse is over the execution)
- browser where the execution ran
- launch time
- number of scenarios of the execution

When expands a test execution a list of executed features and their results can be seen:

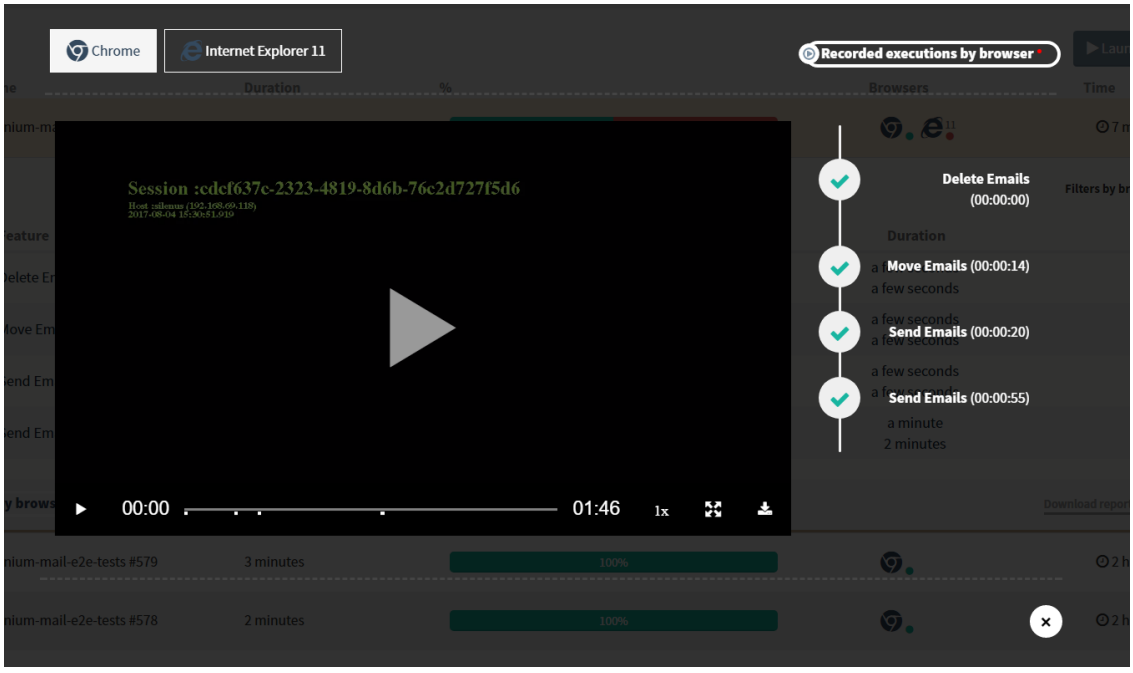


For the execution expanded it can be seen:

- features tested
- the results detailed for each feature (with filters)
- videos of the execution

- download the report of the executions (MS excel and PDF)
- remove execution

Click at "Recorded executions by browser" to see the videos of the recorded executions:

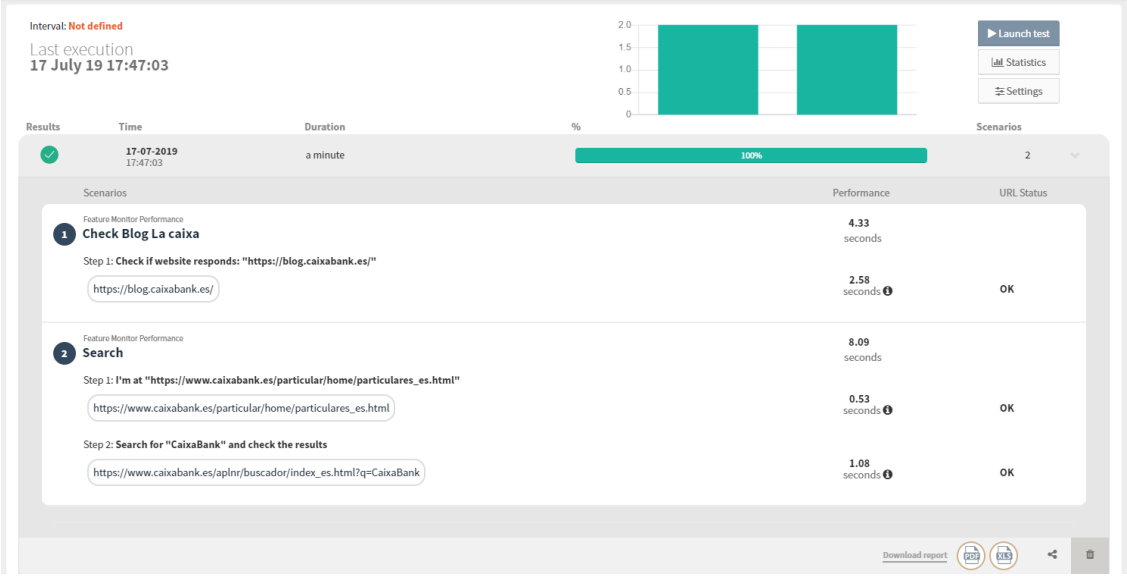


For the videos (at execution) it can be seen:

- videos for the browsers tested
- features to navigate through the video
- cue points when the feature starts on the video

Monitoring project

The result of an execution of a Monitoring project is presented at the execution page:

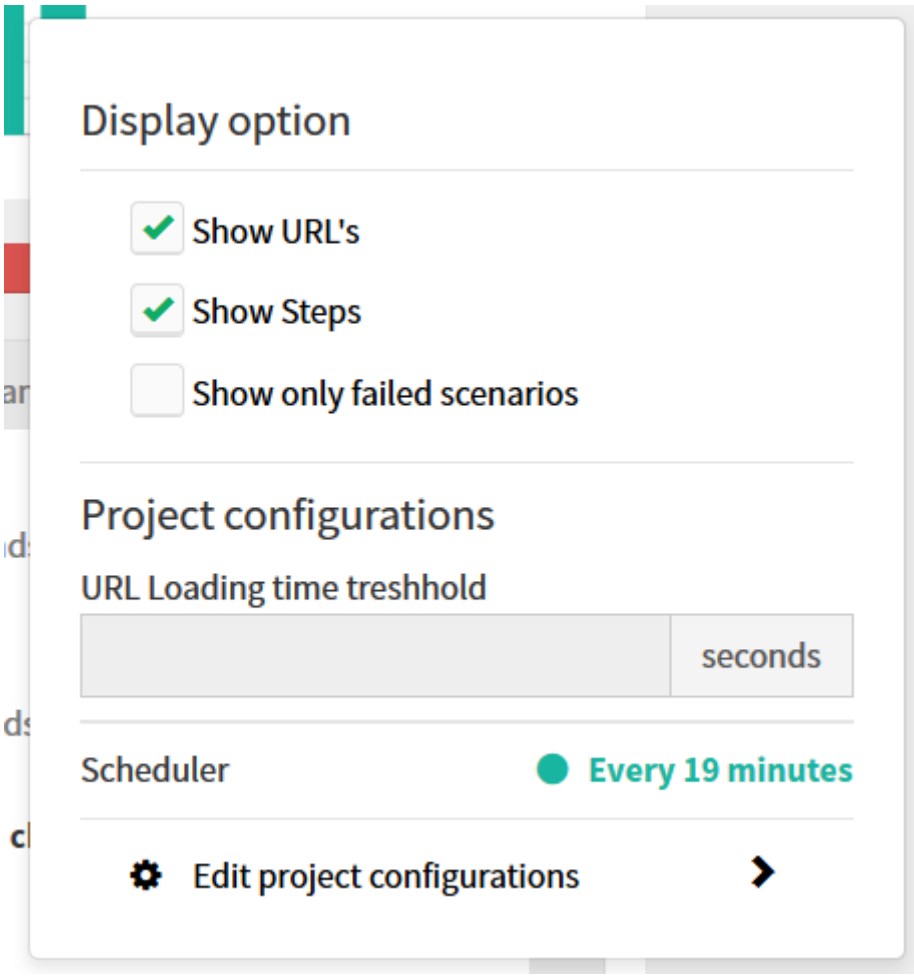


At execution page, we can see the time of each URL took to load. Also we can see the URL status and the time that took to execute the scenario.

Display options

The information at the execution page can be filtered to show/hide some details.

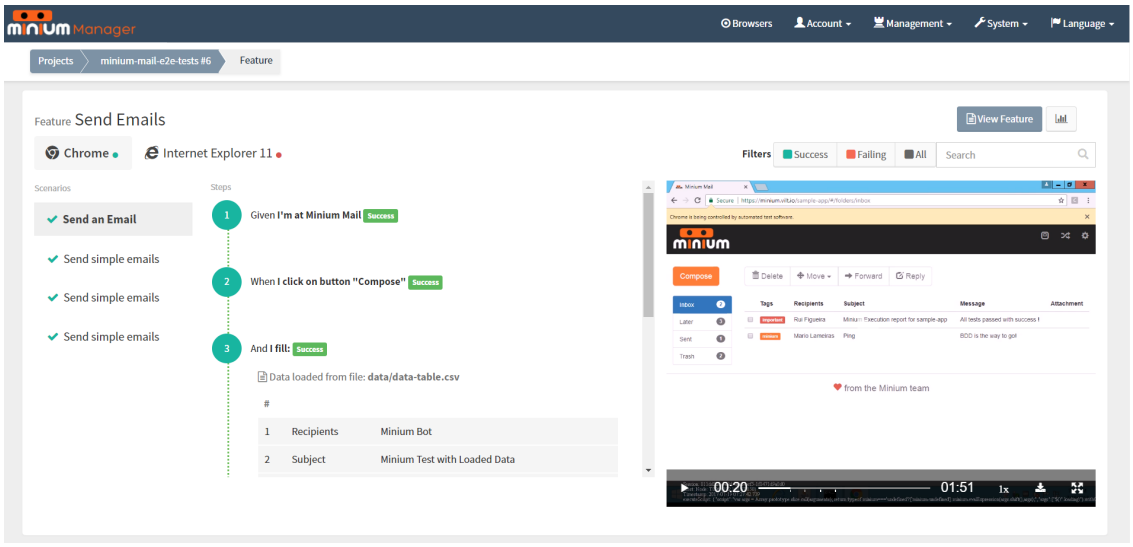
To open the `Display options` click at `Settings`:



The display options allow us to:

- Show/Hide the URL's
- Show/Hide the steps
- Show only the sailed scenarios
- Consult the URL loading time threshold (This property is configured at the project configurations at the advanced tab and warns the user (at the execution page) when the loading time of an URL is higher than the loading time threshold)
- Consult the scheduler (The scheduled is configured at the project configurations at the reports and scheduling tab)

On the feature (for web application testing) view check the scenarios of this particular feature:



When expands a scenario see all the steps and results. Is easily switch to the results of another browser. More functionalities:

- **Screenshots** - Minium Manager provides screenshots of the application at the moment of the failure. So it possible see the state of the application in the moment of failure.
- **Error messages**
- **Links to the feature or step file for each step.**
- **Evolution chart for the projects**
- **Video with the cue points when the scenarios start**

Monitoring project

Feature page of a Monitoring project scenario:

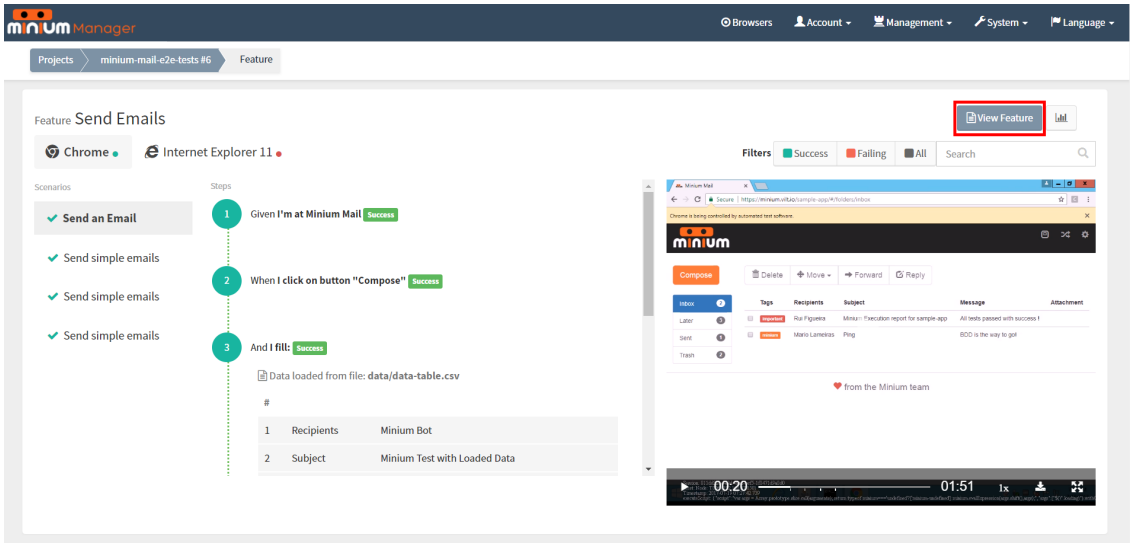
At the feature page, we can see more details about the page load performance:

- The url loaded
- The loading time with the times of the Backend Performance, Frontend Performance and DOM Content Loading
- The number of requests
- The Javascript Errors (if any)

- The URL status code

14.1. Feature overview

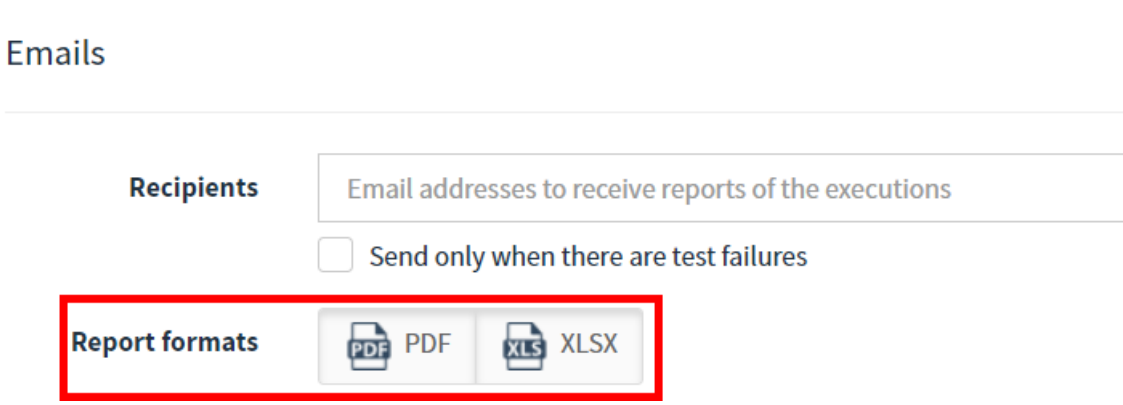
Overview of all the features, by clicking on the button View Feature.



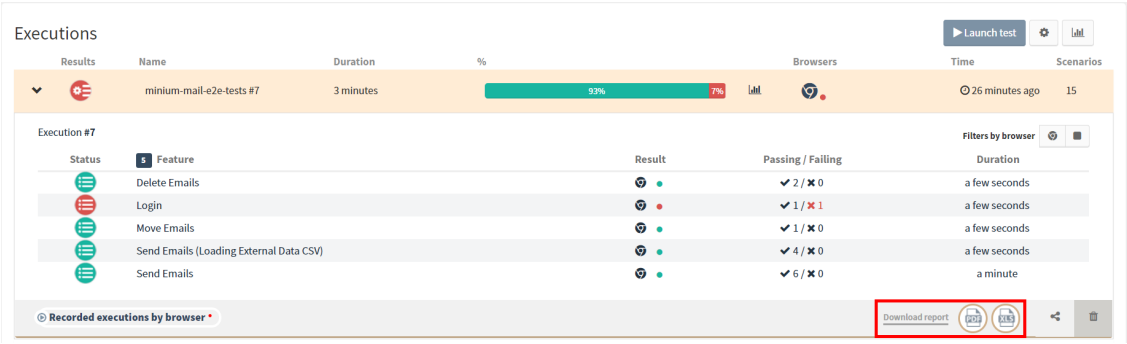
View the result of each step in each browser that the feature ran.

14.2. Reports

Receive the reports of each test execution, in `PDF` and/or `Excel` formats, via email. Alternatively, download them directly in Minium Manager. To choose the formats to receive via email, go to the configurations of the project and expand the advanced configurations:



To download a report, expand the results of a test execution and click in `Download report`:



14.3. Daily Reports

The daily report is a custom report with your favorite projects that will be sent to your email with the status of the last completed execution in the last 24 hours.

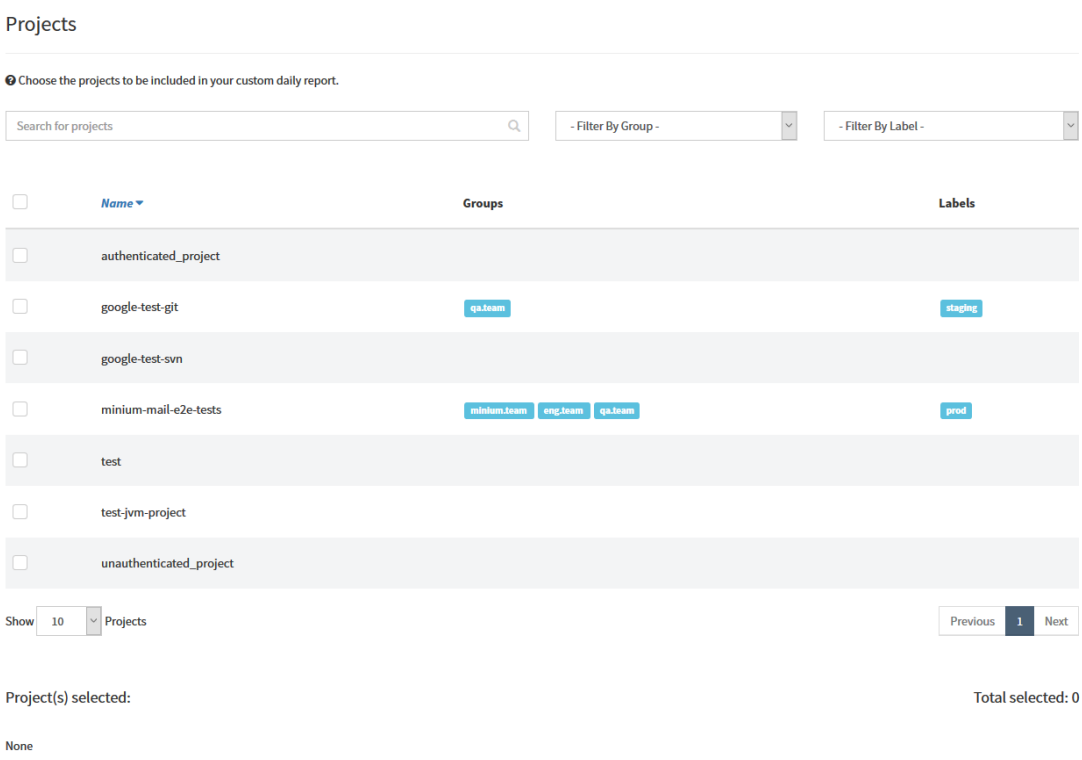
Before being able to configure a daily report, first you need to set the email configured at `Account > Settings`.

Note:

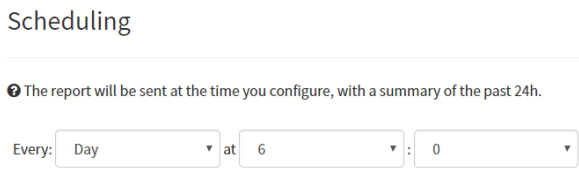
If Minium Manager is using LDAP, please contact your administrator to set your email at LDAP.

To create a daily report configuration, open the `Account` menu and click on `Daily Reports`.

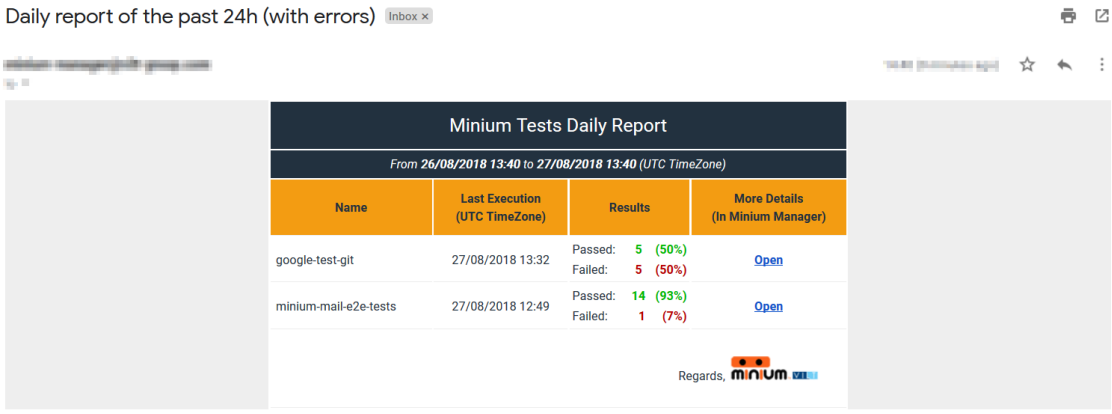
Choose the projects to be included in your custom daily report:



After, schedule the time that the email will be sent, with a summary of the past 24h.



Finally, click on the button "Create/Update Daily Report Configuration" to create/update the daily reports:



To delete the configuration, click on the button "Delete Daily Report Configuration".

The daily report was configured to be sent at **06:00** with the status of the last completed execution in the last 24 hours for the following projects:

- authenticated_project
- google-test-git
- google-test-svn
- minium-mail-e2e-tests
- test
- test-jvm-project
- unauthenticated_project

15. Roles

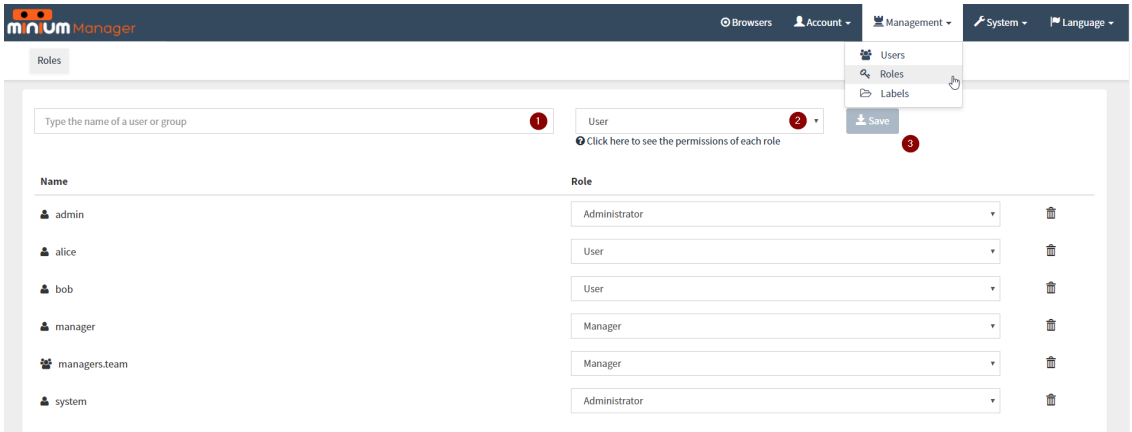
Roles allows grant different privileges to the users and the groups. The privileges associated with each role are described in the table below:

	User	Manager	Administrator
View projects	✔	✔	✔
Launch test executions	✔	✔	✔
Configure projects	✔	✔	✔
Create projects		✔	✔
Manage project labels		✔	✔
Manage users			✔
Manage roles			✔

15.1. Assign roles

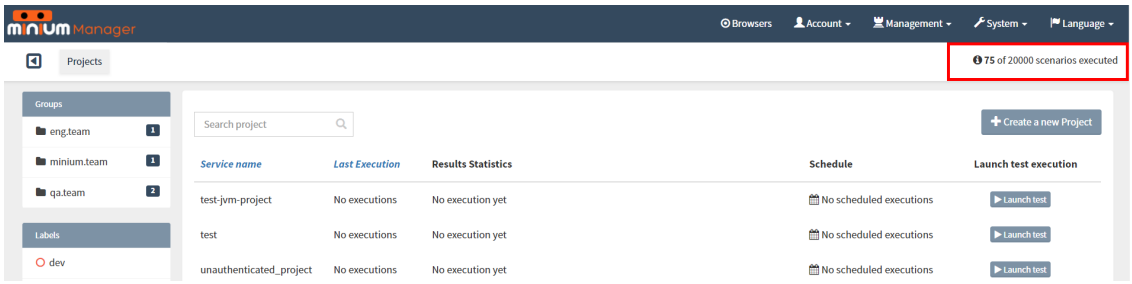
To change the role of a user or group, navigate to `Management > Roles` at the navigation bar and:

1. Introduce the name of the user or group in the search bar.
2. Select a role.
3. Click on `Save` button.

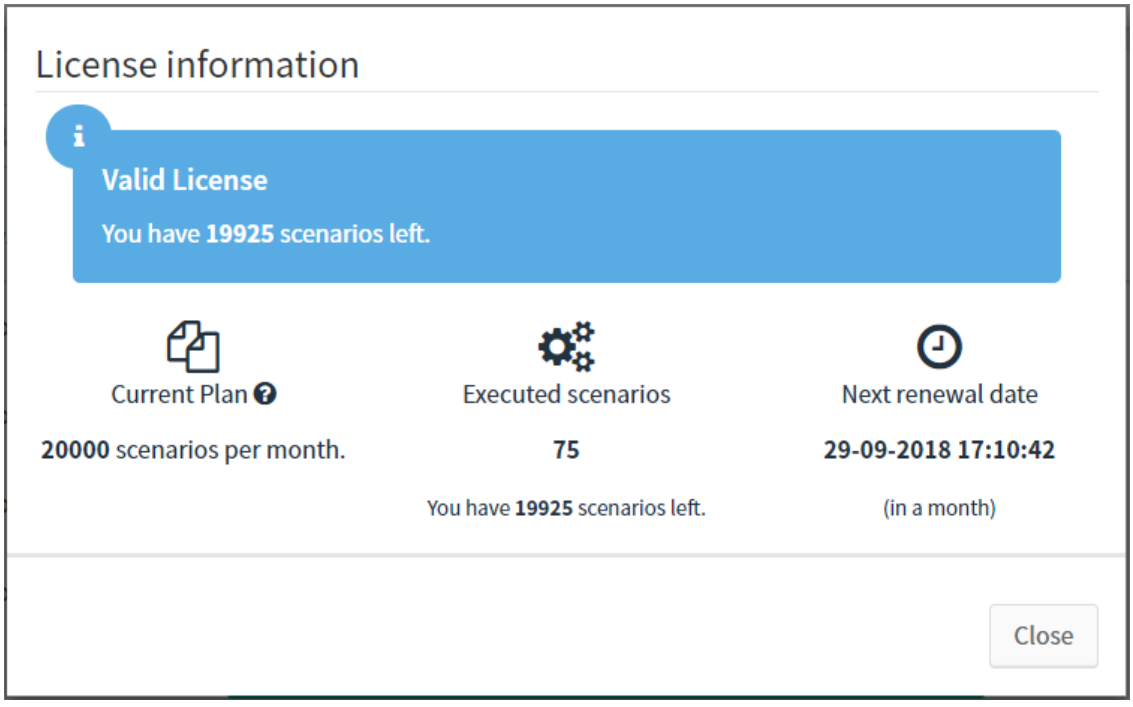


15.2. License information

At the top right of the project list page exists the license information label, that gives the number of executed scenarios in this month and the total of scenarios per month that to the current license



By clicking on the license information label, it will open a modal with more detailed information about one license.



Minium Recorder User Guide

Minium Recorder has been developed to ease and accelerate the creation of Minium tests. With the **Minium Recorder**, is no longer needed to write all the code for the step definitions by hand. Simply tell to the **Minium Recorder** to start recording the interactions with the browser, perform the actions that correspond to the defined step and an automation script is immediately available in **Minium Developer**.

But what really makes Minium Recorder different in relation to other automation script recorders is that, for each element that interacts, Minium Recorder will not generate just one CSS selector. Instead, is possible choose among a list of alternative expressions for each element that take advantage of the great Minium filtering methods that are used like for example `withLabel`, `below`, `rightOf`, etc, to build more readable and resilient expressions.

16. Configure Minium Developer

Minium Recorder is already configured at **Minium Developer Prime**, but you can configure an existing **Minium Recorder**.

To configure **Minium Developer** with the **Minium Recorder** extension, first you need to download the **Minium Recorder** extension.

Navigate to the folder where **Minium Developer** is installed, create a folder named "**extensions**", copy the **Minium Recorder** extension file to the new folder and rename the file to "minium-recorder.crx".

After, edit the `config/application-{windows,linux,macos}.yml` file in order to add the following configuration (at the chrome webdriver):

```
minium:
  developer:
    webdrivers:
      - name: chrome
        ...
        chromeOptions:
          extensions:
            - ${app.home:.}/extensions/minium-recorder.crx
          preferences:
            devtools:
              preferences:
                panel-tabOrder: "{\"chrome-
extension://ggefeclafoeoejognlebilidgmgdglogMiniumRecorder\":10,\"elements
\":20,\"console\":30,\"sources\":40,\"network\":50,\"timeline\":60,\"heap_
profiler\":70,\"resources\":80,\"security\":90,\"audits\":100}"
```

17. Launch Minium Recorder

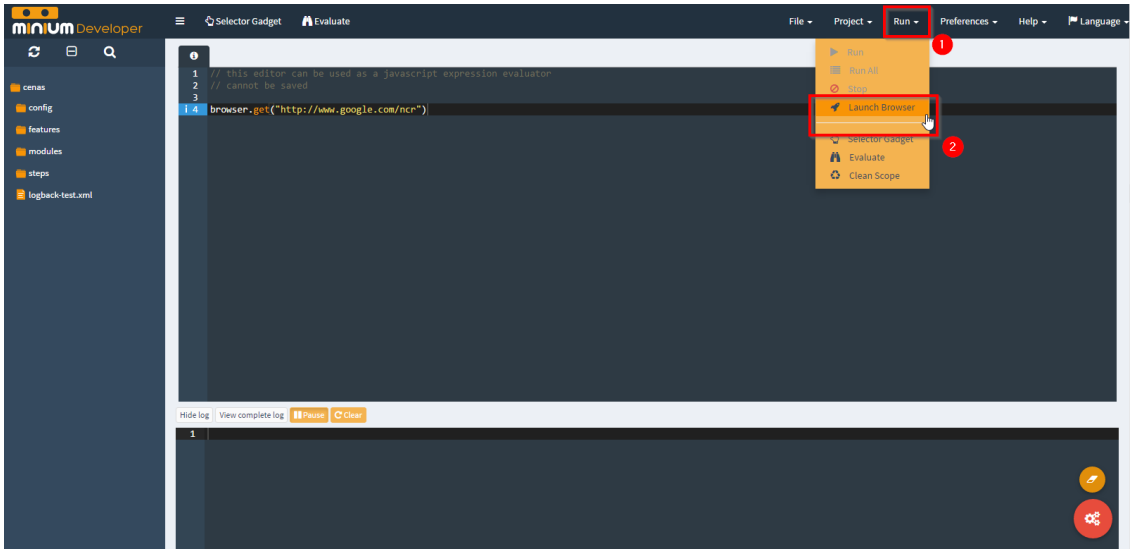
Launch **Minium Recorder** through **Minium Developer**.

Open **Minium Developer** and add the following command to the editor:

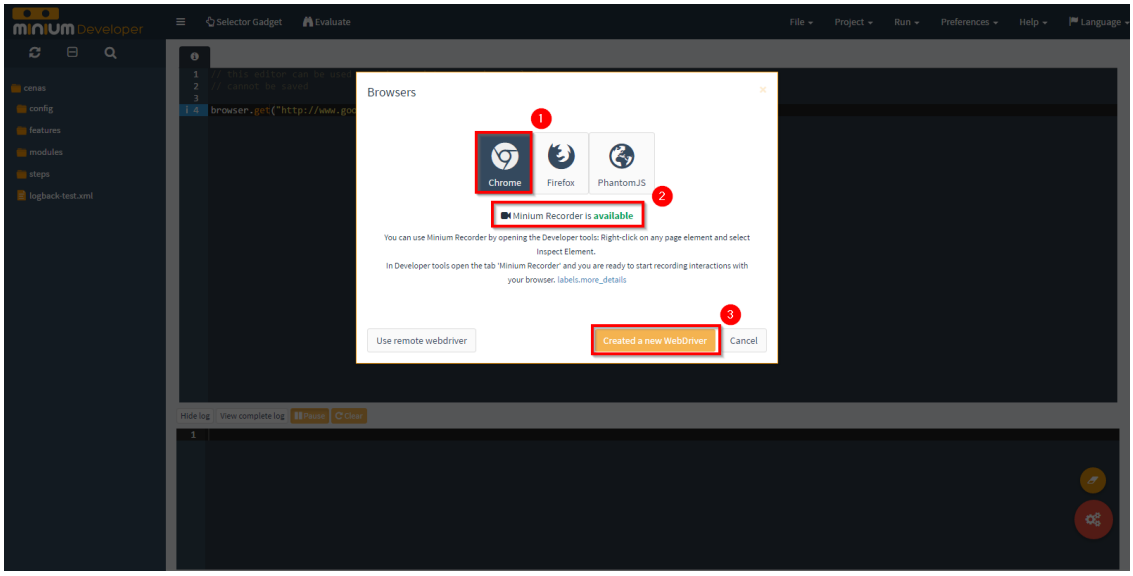
```
browser.get("http://www.google.com/ncr")
```

This will load the google search engine website, when the instance of Chrome is launched.

To launch Minium Recorder, use the (`Ctrl + Enter`) shortcut or select `Run > Launch Browser` **(1)** and **(2)** at the top-right menu:

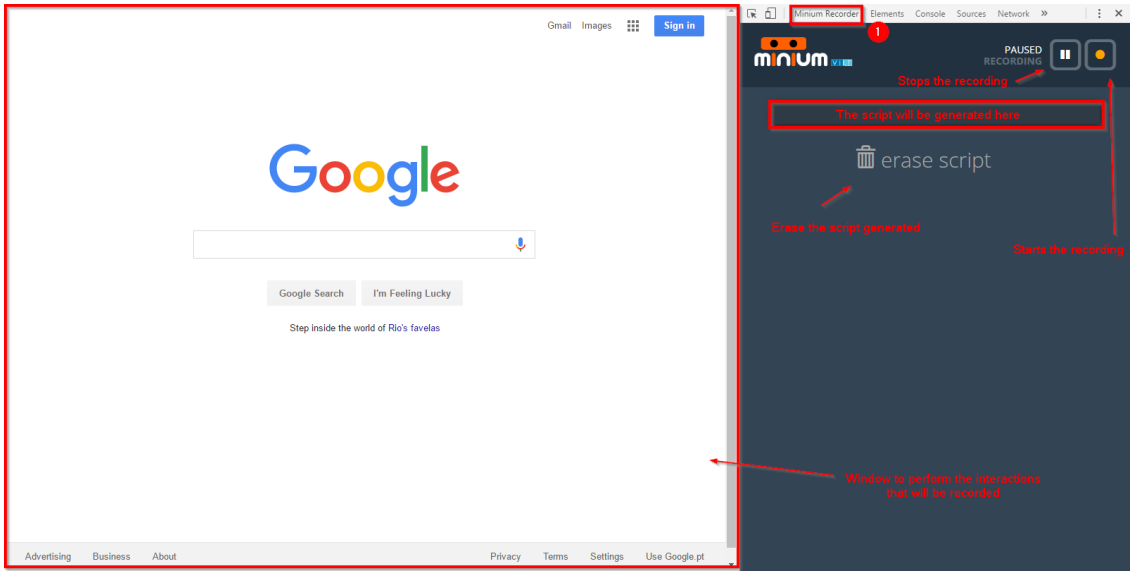


Select the `Chrome` browser **(1)**, check if Minium Recorder is available **(2)** and click `Created a new WebDriver` **(3)**:



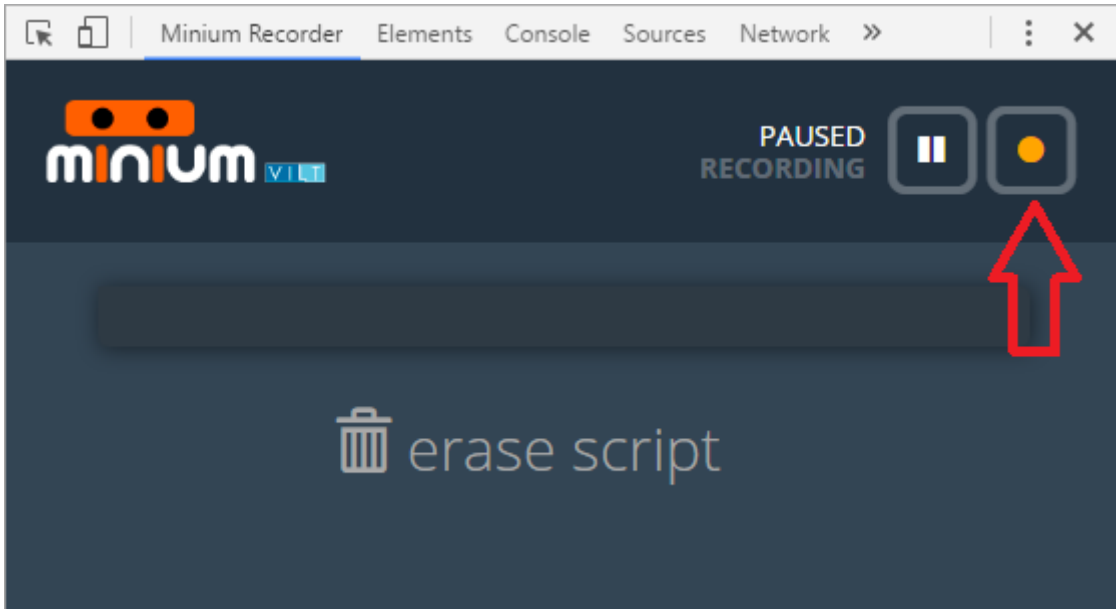
A new instance of Chrome will be launched with the Minium Recorder extension installed. In the instance of Chrome, open the `Developer tools` by using the (`F12`) shortcut (use another shortcut) or select `More Tools > Developer tools`.

After the `Developer tools` of the instance Chrome is opened, select the `Minium Recorder` tab **(1)**:



18. Start recording

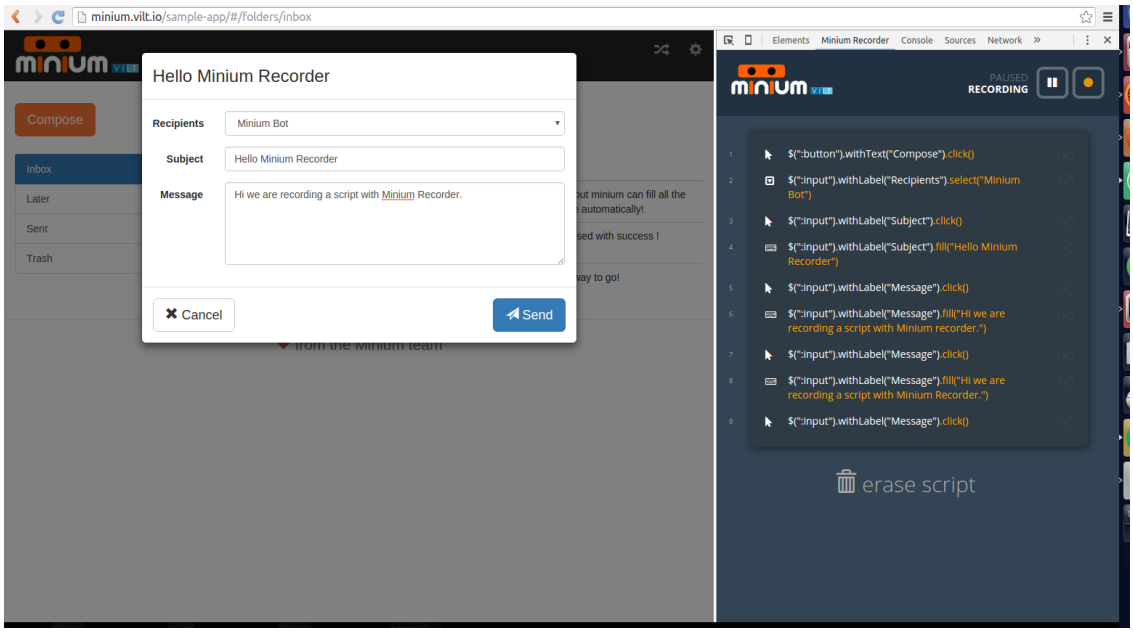
To start recording, click on the following button:



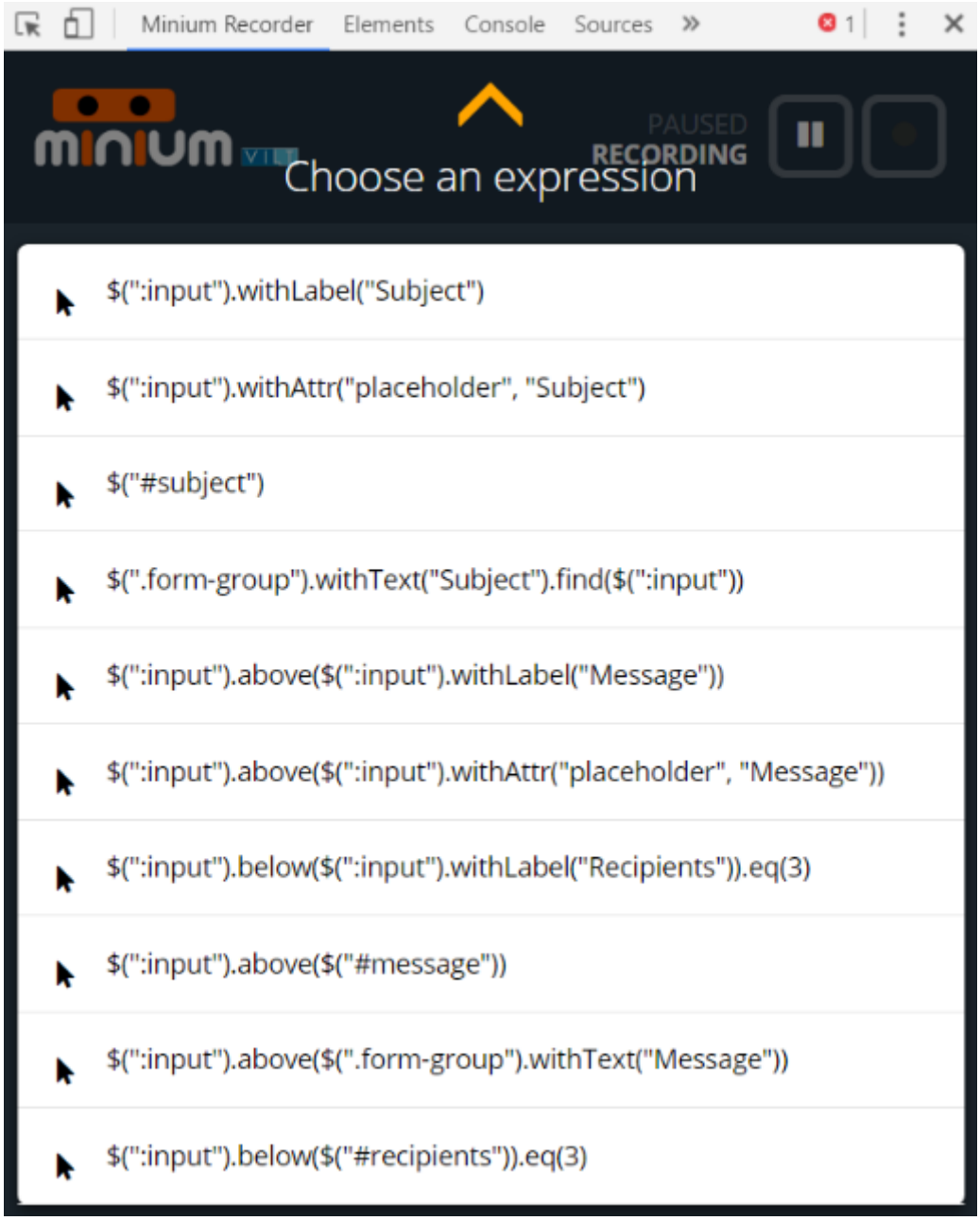
From now on, every interaction performed with the browser will be recorded.

19. Record interactions

For every action performed on the browser, like clicking on an element or filling a text input field, Minium Recorder will automatically generate the corresponding code. If typed an URL in the address bar, Minium Recorder will also generate the code to load that URL.



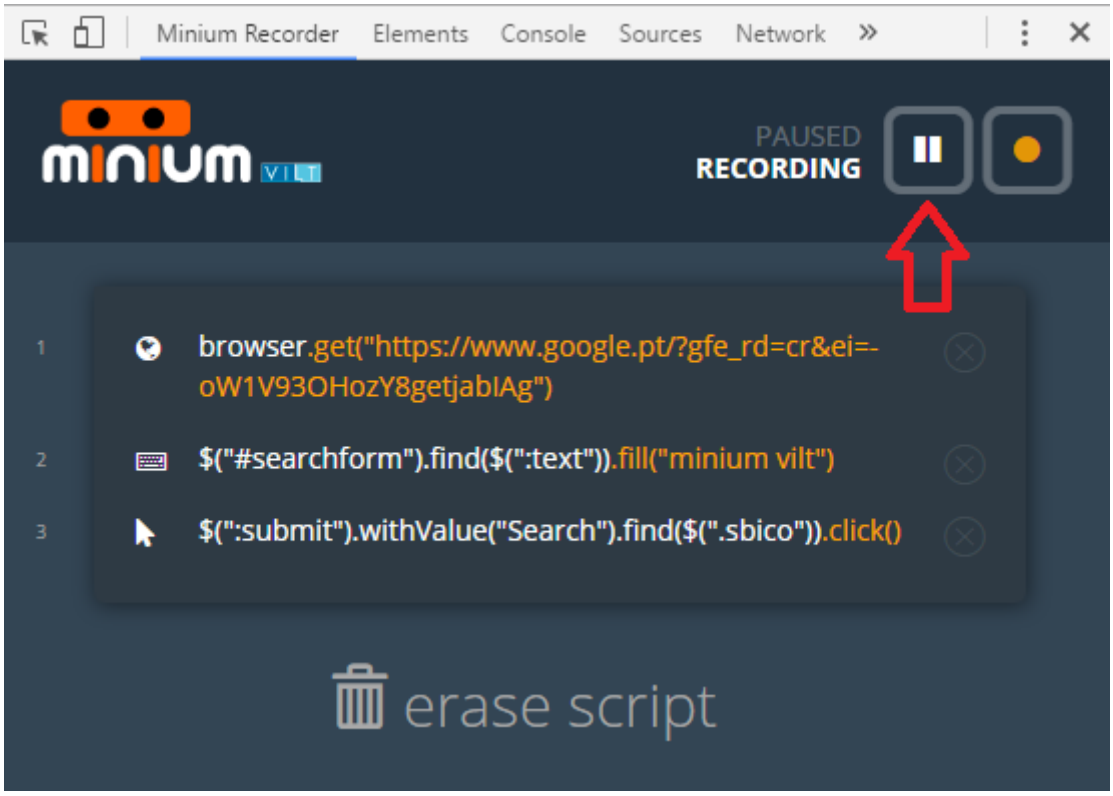
To try other expressions, click on the expression and a list of alternative expressions will be displayed:



To select an expression, just click on it.

20. Stop recording

Click on the pause button to stop a recording.



21. Import the recorded script in Minium Developer

To import a script in Minium Developer, right-click on the editor area, where is intended to put the script, and choose the option `Import recorded script`: