

# Submission and Evaluation System

This is an overview of the system that you will use to test your labs/projects and to submit them for grading.

---

**Learning Goal:** How to use the submission system.

---


## 1 Introduction

This tutorial will introduce you to a system called Jenkins, used by the submission system. You will be using this system to test your labs/projects and to submit them for grading. The labs are tested with a testbench which applies a number of tests to your design and provides you with a feedback about the features that are missing or not implemented properly. Prior to every submission, we strongly encourage you to write your own testbench, which is an essential part of debugging hardware designs. Please refer to the VHDL Testbench Tutorial for more information on testbench design.

## 2 Connection

This section will guide you through the steps to connect to the submission system. Try them.

1. Open your web browser and go to `https://ceng-labs.epfl.ch`. Note that if you are submitting from home, you must have your VPN connection to EPFL running.
2. Then, authenticate yourself using your **GASPAR username** and **password** (Figure 1).



User:

Password:

☐ Remember me on this computer

Figure 1: Connection

3. Once you are logged in, you can see the Dashboard described in Section 3.

## 3 Dashboard

The **dashboard** grants you the possibility to submit the requested files for the lab. Note that your dashboard will not look exactly as in Figure 2 as the naming convention for the submissions is `[lab_name]-[your GASPAR username]`. Guidelines on files' submission are provided in Section 4.

- Make sure you submit **all required files**. If you forget to submit a file, the system will treat it as **empty** and your score will be lower.
- You can upload a **zip file** containing **all required files**, or you can upload the files one by one.
- The system will keep all versions of each submitted file. However, only a **limited number of** submissions will be taken into account for the grade—one more reason to double check that you have uploaded **all files before submitting** them for evaluation.

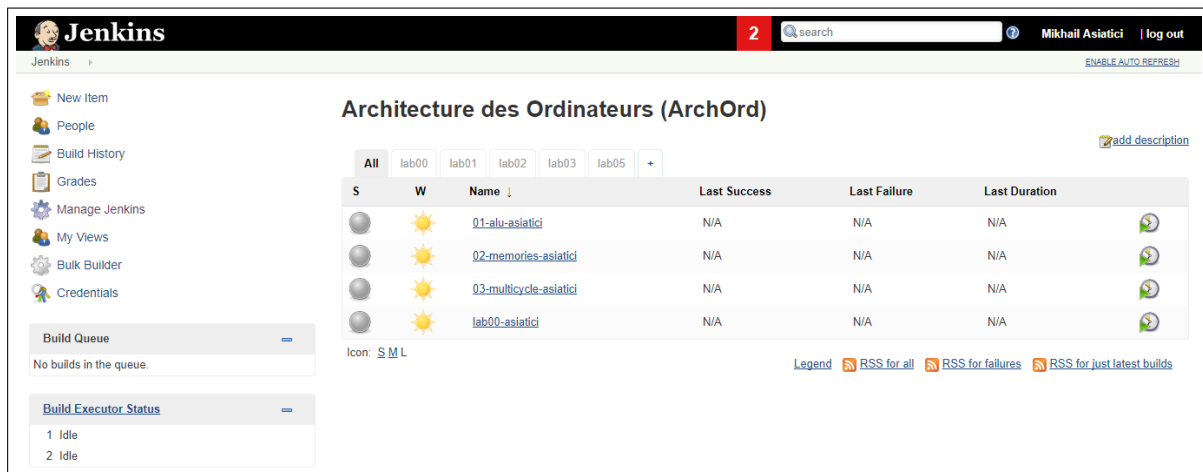



Figure 2: Dashboard

## 4 Submission / Build

This section explains how to submit the files and launch a build into Jenkins.

1. On the dashboard push the  button of a selected lab. This will redirect you to the submission page. Here is how the submission page would look if you click on the build button of the lab called **lab00-asiatici** (which, in your case, will be called **lab00-[your GASPARE username]**).

### Project lab00-asiatici

This build requires parameters:

submission.zip
 No file chosen  
Contains all files

circuit.vhd
 circuit.vhd

Figure 3: Files submission

2. Now you have the possibility to upload all required files separately **or** to upload a zip file containing all files. In this example, you would either upload the file called **circuit.vhd** or a zip file called **submission.zip**.

3. Once the files are submitted, you would click on the **Build** button. As soon as the build has finished, you could see it in the **Build History** located in the left pane of Jenkins:



Figure 4: Build History

4. If you wish to see the **Build Details** page, you should click on the date when the build was executed. Build details page allows you to see the evaluation results of your last submission.

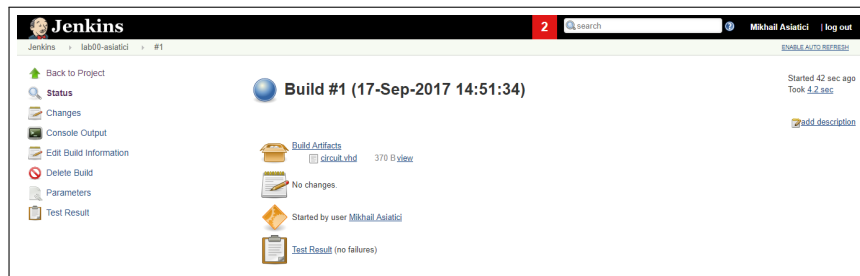

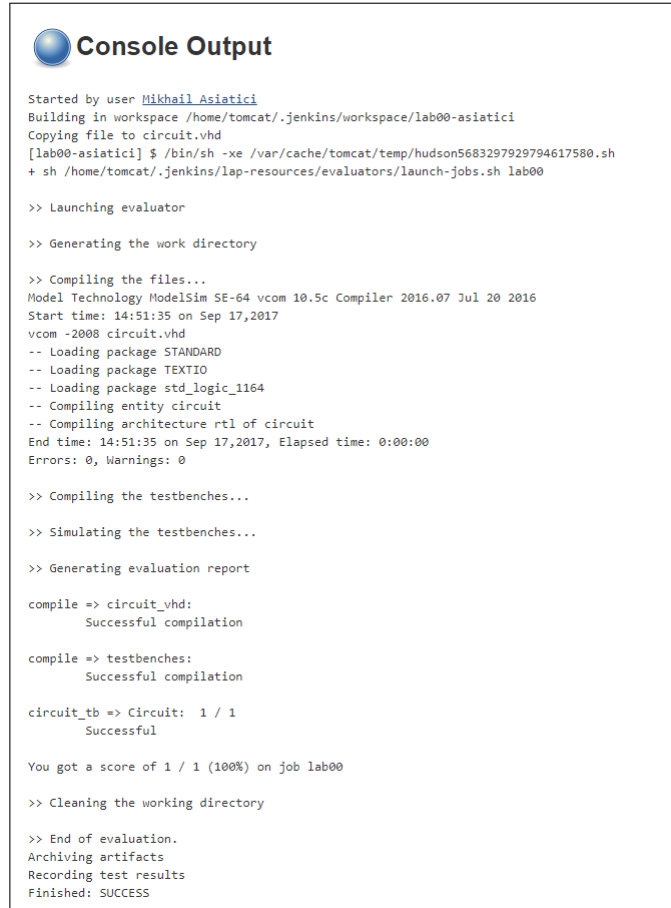


Figure 5: Build details

5. To open the detailed report of the evaluation, you should click on the **Console Output** of the current build using the  link.



**Console Output**

```

Started by user Mikhail Asiatichi
Building in workspace /home/tomcat/.jenkins/workspace/lab00-asiatici
Copying file to circuit.vhd
[lab00-asiatici] $ /bin/sh -xe /var/cache/tomcat/temp/hudson5683297929794617580.sh
+ sh /home/tomcat/.jenkins/lap-resources/evaluators/launch-jobs.sh lab00

>> Launching evaluator

>> Generating the work directory

>> Compiling the files...
Model Technology ModelSim SE-64 vcom 10.5c Compiler 2016.07 Jul 20 2016
Start time: 14:51:35 on Sep 17,2017
vcom -2008 circuit.vhd
-- Loading package STANDARD
-- Loading package TEXTIO
-- Loading package std_logic_1164
-- Compiling entity circuit
-- Compiling architecture rtl of circuit
End time: 14:51:35 on Sep 17,2017, Elapsed time: 0:00:00
Errors: 0, Warnings: 0

>> Compiling the testbenches...

>> Simulating the testbenches...

>> Generating evaluation report

compile => circuit_vhd:
    Successful compilation

compile => testbenches:
    Successful compilation

circuit_tb => Circuit: 1 / 1
    Successful


You got a score of 1 / 1 (100%) on job lab00

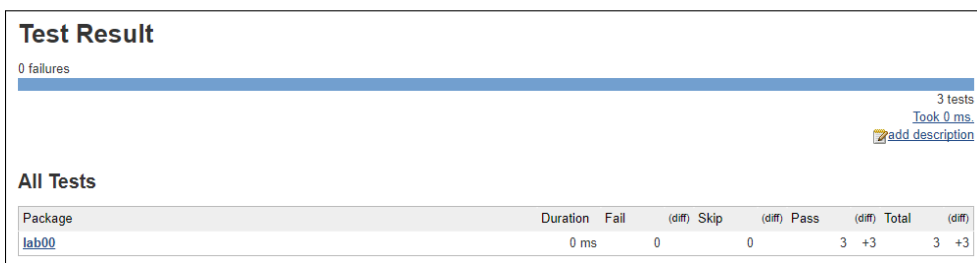
>> Cleaning the working directory

>> End of evaluation.
Archiving artifacts
Recording test results
Finished: SUCCESS

```

Figure 6: Console Output

6. To open the **Test Result** of the current build, you should click the  link. The page that appears shows what happened during the evaluation.



**Test Result**

0 failures

3 tests  
Took 0 ms.  
[add description](#)

**All Tests**

Package	Duration	Fail	(diff) Skip	(diff) Pass	(diff) Total	(diff)
<a href="#">lab00</a>	0 ms	0	0	3	+3	3 +3

Figure 7: Test Result

7. Returning to the **dashboard** after the submission allows you to see the current state of your labs and submissions.

<div> <div>All lab00 lab01 lab02 lab03 lab05 +</div> </div>						
S	W	Name ↓	Last Success	Last Failure	Last Duration	
		<a href="#">01-alu-asiatici</a>	N/A	N/A	N/A	
		<a href="#">02-memories-asiatici</a>	N/A	N/A	N/A	
		<a href="#">03-multicycle-asiatici</a>	N/A	N/A	N/A	
		<a href="#">lab00-asiatici</a>	4 min 28 sec - #1	N/A	4.2 sec	

Figure 8: Laboratories list

## 5 Exercise

To get more familiar with this submission system, try now to upload the following code as a submission to **lab00**. Make sure that after uploading you build your code and check the preliminary results.

This exercise **is not** graded.

```

library ieee;
use ieee.std_logic_1164.all;

entity circuit is
  port (
    A : in std_logic;
    B : in std_logic;
    C : in std_logic;
    D : in std_logic;
    E : in std_logic;
    F : out std_logic
  );
end circuit;

architecture rtl of circuit is
begin

  F <= A AND B when E = '0' else
       C OR D when E = '1' else
       '0';

end rtl;

```

## 6 Submission through the command line

Apart from the Web interface, we also provide a command line script for submitting each project. This script is part of the project templates that you can download from Moodle for each lab. To use the scripted submission procedure you need to:

1. Acquire an “API token” from Jenkins (explained below)
2. Install required software on your computer if you are not using the provided virtual machine.

### Acquiring an API-token from Jenkins

Follow the steps shown in Figure 9 to generate a submission token. Take note of the token and keep it until the end of the course and use it for all of your submissions through the command line. This serves as a random password for your submissions.

### Installing the required software on your machine (already installed on the VirtualBox VM)

To use the script you need to have `python3`. Install the latest version of `python3` using this link on your machine. Once you have installed `python3` open a terminal (powershell on Windows) and execute the following command.

```
> python3 -m pip install requests --user
```

Note that the acquiring a token and `python3` installation is a **one time** operation. In other words you only do need to do it in this exercise. You can reuse the token from this exercise for any other submission.

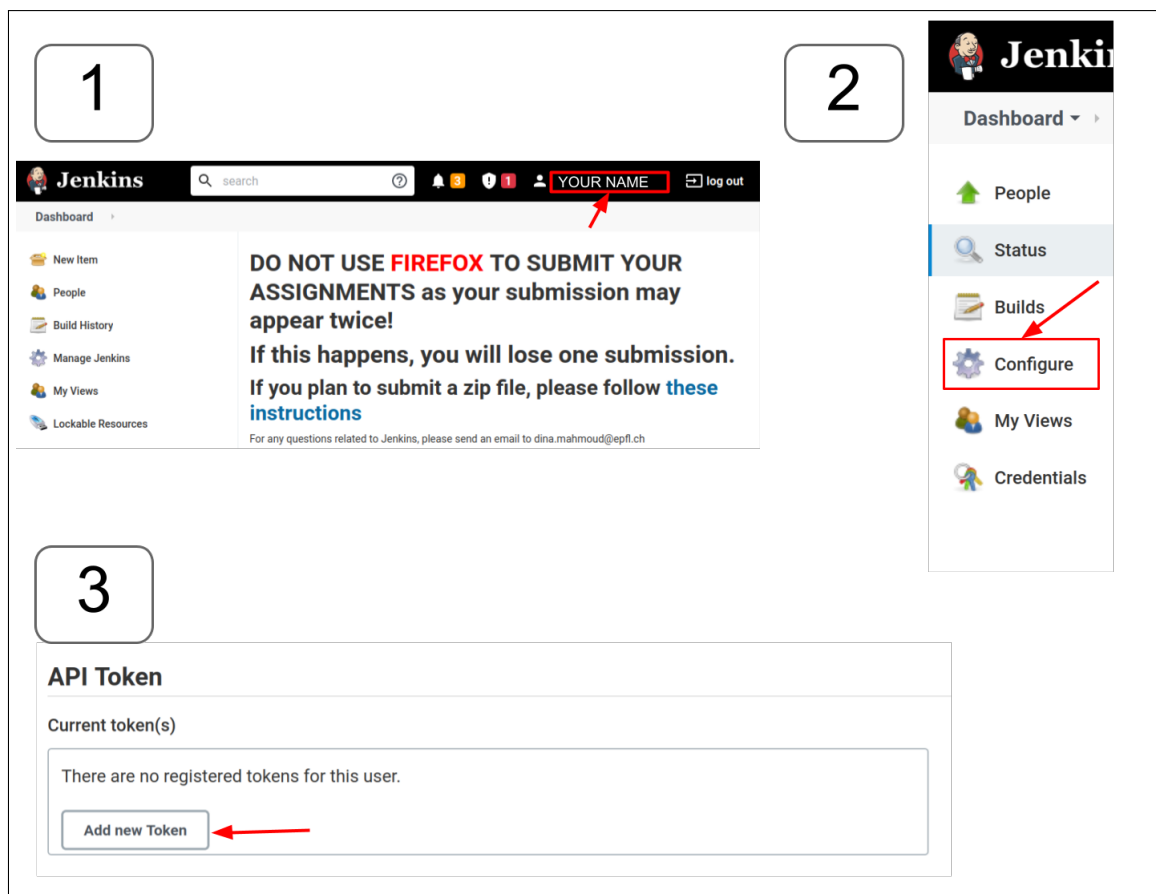


Figure 9: How to acquire an API token

Now head to the course’s Moodle page and download the *cli\_tutorial*. In it, you will find a directory the following files and folder:

```
├── JenkinsJob.py
└── submit.py
```

```
└─ vhd1
   └─ circuit.vhd
```

Go into the tutorial directory (in the terminal) and use the following command to call the `submit.py` script:

```
> python3 submit.py
```

You will be prompted for an action which is either `build` or `check`. The `build` action sends your files to Jenkins and starts the build process and `check` is used to retrieve the submission results. An example usage is shown below:

```
> python3 submit.py
Select action ? [build/check] build
GASPAR: YOUR_GASPAR_ID
token: YOUR_API_TOKEN
fetching job information https://ceng-labs.epfl.ch/job/00-tutorial-YOUR_GASPAR_ID
Adding circuit.vhd
Successfully sent build request, you can check the results later
```

Here you should use your GASPAR identity and the API token you acquired from Jenkins to submit your assignment. Once you have made your submission, you can check the results of your submission by rerunning the script with the `check` action.

```
> python3 submit.py
Select action ? [build/check] check
GASPAR: YOUR_GASPAR_ID
token: YOUR_API_TOKEN
fetching job information https://ceng-labs.epfl.ch/job/00-tutorial-YOUR_GASPAR_ID
Select a build number to see the output [1-6]:6
Fetching logs from
https://ceng-labs.epfl.ch/job/00-tutorial-YOUR_GASPAR_ID/6/logText/progressiveText?start
Show logs? [Y/n] Y
Started by user YOUR_GASPAR_ID
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/00-tutorial-YOUR_GASPAR_ID
Copying file to circuit.vhd
[00-tutorial-emami] $ /bin/bash -xe /tmp/jenkins3605494097399575367.sh
+ /opt/lap-resources/evaluators/launch-jobs.sh lab00
/opt/lap-resources/testbenches/lab00/config.sh
/opt/lap-resources/evaluators

>> Launching evaluator

>> Generating the work directory

>> Compiling the files...

linux
Model Technology ModelSim - Intel FPGA Edition vcom 10.5b Compiler 2016.10 Oct
5 2016
Start time: 15:43:51 on Sep 17, 2021
vcom -2008 -suppress 1514 circuit.vhd
-- Loading package STANDARD
-- Loading package TEXTIO
-- Loading package std_logic_1164
-- Compiling entity circuit
```

```
-- Compiling architecture rtl of circuit
End time: 15:43:51 on Sep 17,2021, Elapsed time: 0:00:00
Errors: 0, Warnings: 0

>> Compiling the testbenches...

>> Simulating the testbenches...

>> Generating evaluation report

compile => circuit_vhd:
           Successful compilation

compile => testbenches:
           Successful compilation

circuit_tb => Circuit:  1 / 1
           Successful

You got a score of 1 / 1 (100%) on job 00

>> Cleaning the working directory

>> End of evaluation.
Archiving artifacts
Recording test results
[Checks API] No suitable checks publisher found.
Finished: SUCCESS

Save to file? [Y/n] Y Saving the log to
TUTORIAL_DIRECTORY/__submission__/build_6.log
```

As you can see, you can check the result of the submission and also optionally save them into a file. This file is saved under `__submission__` directory as `build_N.log`, where `N` is the submission attempt number.

**Attention!** after submitting your assignment, you may not immediately see the new submission when you use check. This does not mean your submission did not go through, wait for a couple of minutes and check again. **Do not submit again immediately as you may end up submitting twice!**