

Ex 1/3 - Building from SCM

Configure Maven Installer

1. Use a browser to navigate to the provided public IP address for the server. Remember to add ":8080" to the end of the IP address to specify the default Jenkins port.
2. Log in with your credentials.
3. Click **Manage Jenkins**.
4. Click **Global Tool Configuration**.
5. Under *Maven installations*, click **Add Maven**.
6. In the *Name* box, enter "M3".
7. Make sure **Install automatically** is checked.
8. Click **Save**.

Configure the Build to Use Maven and Make Compile

1. Click **New Item**.
2. Enter an item name of "**mavenproject**" in the box provided.
3. Select **Freestyle project**.
4. Click **OK**.
5. Click the **Source Code Management** tab at the top of the screen.
6. Select the option for a **Git** repository.
7. Enter <https://github.com/RickHearts/mvn1>
8. Select a Build Trigger **Build after other projects are built**
Projects to watch: **Test/user_test**
9. Click the **Build** tab at the top of the screen.
10. Click **Add build step** and select the **Invoke top-level Maven targets** option.
10. Under *Maven Version*, select **M3**.
11. In the *Goals* box, enter "clean package".
12. Click **Add build step** and select the **Execute shell** option.
13. In the *Command* window, enter "compile".
14. Click **Add post-build action** and select the **Archive the artifacts** option.
15. Inside the **Archive the artifacts** box, click **Advanced...**
16. Check the option for **Fingerprint all archived artifacts**.
17. In the *Files to archive* box, enter *.
18. Click **Save**.

19. Click **Build Now**.
20. Should FAILED!
21. Refresh the window, verify the console output.
22. Install Maven on windows!
23. Install openjdk with zulu13.28.11-ca-jdk13.0.1-win_x64.msi
24. Set JAVA_HOME path
25. Download Maven Zip (already done)
26. Setup environment variable for Maven (M2_HOME)
27. Verify installation

```
mvn -version
```
28. Click **Build Now**.
29. Refresh the window, verify the console output.
30. Restart the Jenkins service
31. Log in with your credentials.
32. Click **Build Now**.
33. Refresh the window, verify the console output.
34. Should be SUCCESS!

Ex 2/3 – Create a longer “job1”

Clean Project

1. Create another job and call it job1
2. Save it empty
3. Build it
4. Open the Build History #1/Console Output
5. Build it More times (at least 5)

A Mess

1. Select **Configure / Build / Run with timeout**
2. Reduce the Timeout to 1 minutes
3. With **Execute windows batch command**, then in the text box
4. type

```
ping www.rumos.pt > out.txt
```

5. Time-out action **Fail the Build**
6. **Save** it (make sure you have 1 minute)
7. Build it 2 times
8. Change the build to

```
ping -t www.rumos.pt > out.txt
```
9. Build it
10. Change the Timeout to Deadline 0:00:10
11. Build it many times (even if it didn't finished)
12. Go the File Explorer and see the Workspace folder!
13. Go back and forth between the Configure/Save and the Build now and try all the options!
14. And you can have more options!

Ex 3/3 – Using the Jenkins REST API

1. From <http://localhost:8080/user/<your user>/configure>, browse to the **API Token** Section and click on the **Show API Token** button
2. Generate a Token
3. Install WSL and Ubuntu bash 18.04
4. If you want a list of all jobs (with a nicely formatted JSON), then you can invoke the /api/json API with a simple GET request:

```
curl -X GET http://localhost:8080/api/json?pretty=true --user <your user>:<your token>
```
5. Start the job 'job1' using Jenkins REST Api:

```
curl -X POST http://localhost:8080/job/job1/build --user <your user>:<your token>
```
6. You can also schedule the job start-up with some delay:

```
curl -X POST http://localhost:8080/job/job1/build?delay=10sec --user <your user>:<your token>
```
7. On the other hand, if you want to delete in bulk some builds, you will add the build range before the doDelete method:

```
curl -X POST http://localhost:8080/job/job1/[1-100]/doDelete --user <your user>:<your token>
```
8. If you want to delete the job 'job1' then you can do it using the doDelete method:

```
curl -X POST http://localhost:8080/job/job1/doDelete --user <your  
user>:<your token>
```