Building a Maven Project with Jenkins

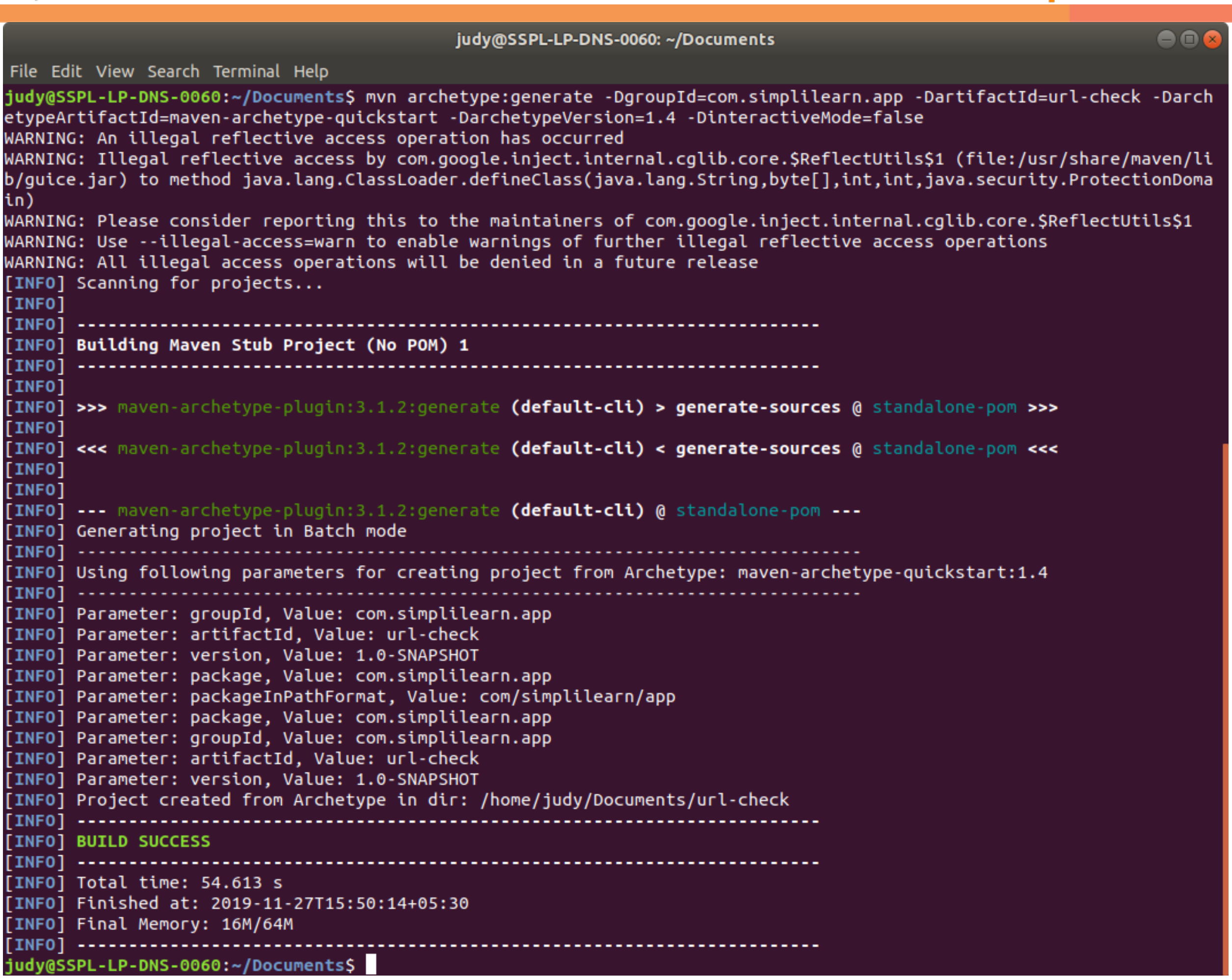
This section will guide you to:

- Install Maven and create a Maven project
- Create a Jenkins build job to build a Maven project
- Build a Maven project in Jenkins

Step 1:Install Maven and create a Maven project

- Open the terminal.
- If you do not have Maven installed already, run the command, ***sudo apt-get install maven***
- Run the command **mvn --version** to verify the installation.
- Run the following command to create a Maven project:

```
mvn archetype:generate -DgroupId=com.simplilearn.app
                      -DartifactId=url-check -DarchetypeArtifactId=maven-archetype-
                      quickstart -DarchetypeVersion=1.4 -DinteractiveMode=false
```


judy@SSPL-LP-DNS-0060: ~/Documents
File Edit View Search Terminal Help

```
judy@SSPL-LP-DNS-0060:~/Documents$ mvn archetype:generate -DgroupId=com.simplilearn.app -DartifactId=url-check -Darch
etypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4 -DinteractiveMode=false
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by com.google.inject.internal.cglib.core.$ReflectUtils$1 (file:/usr/share/maven/li
b/guice.jar) to method java.lang.ClassLoader.defineClass(java.lang.String,byte[],int,int,java.security.ProtectionDoma
in)
WARNING: Please consider reporting this to the maintainers of com.google.inject.internal.cglib.core.$ReflectUtils$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building Maven Stub Project (No POM) 1
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] >>> maven-archetype-plugin:3.1.2:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO]
[INFO] <<< maven-archetype-plugin:3.1.2:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
[INFO]
[INFO] --- maven-archetype-plugin:3.1.2:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Batch mode
[INFO] ----------------------------------------------------------------------------
[INFO] Using following parameters for creating project from Archetype: maven-archetype-quickstart:1.4
[INFO] ----------------------------------------------------------------------------
[INFO] Parameter: groupId, Value: com.simplilearn.app
[INFO] Parameter: artifactId, Value: url-check
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: com.simplilearn.app
[INFO] Parameter: packageInPathFormat, Value: com/simplilearn/app
[INFO] Parameter: package, Value: com.simplilearn.app
[INFO] Parameter: groupId, Value: com.simplilearn.app
[INFO] Parameter: artifactId, Value: url-check
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Project created from Archetype in dir: /home/judy/Documents/url-check
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 54.613 s
[INFO] Finished at: 2019-11-27T15:50:14+05:30
[INFO] Final Memory: 16M/64M
[INFO] ------------------------------------------------------------------------
judy@SSPL-LP-DNS-0060:~/Documents$
```

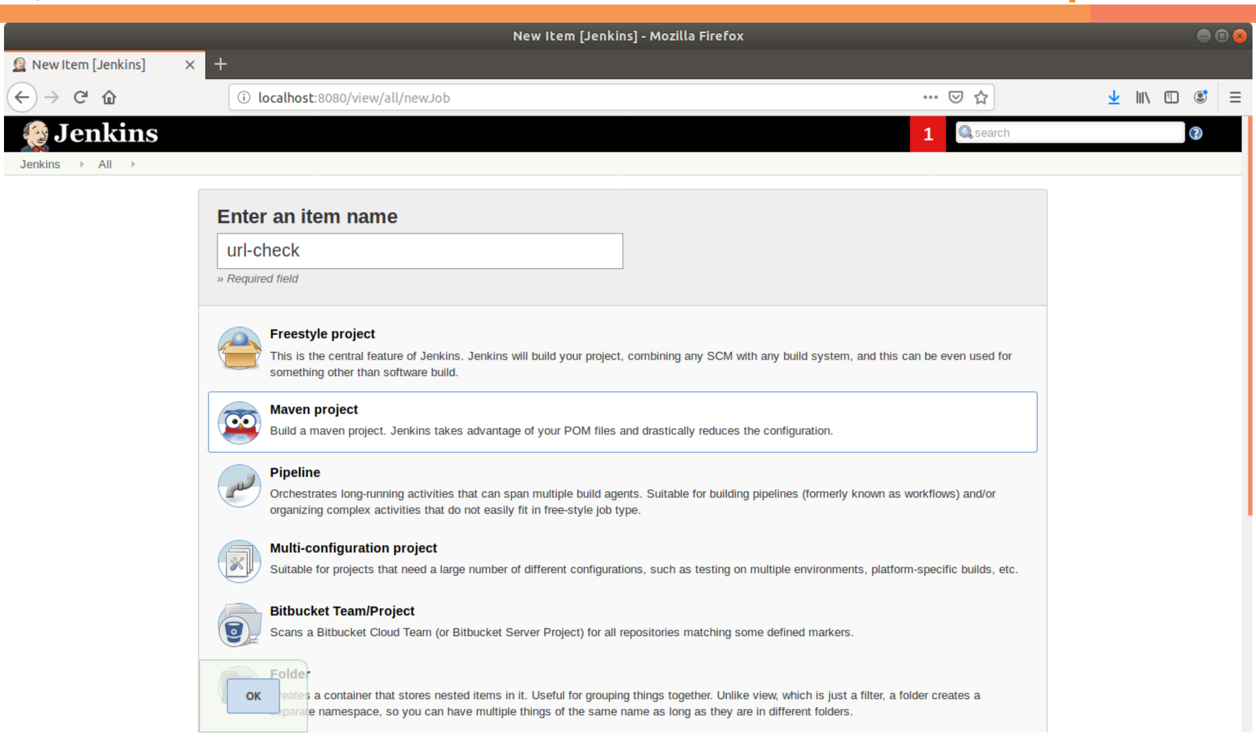
- Once the command is executed, run **cd url-check**
- Inside the project directory, run **ls** and find the *pom.xml* file.
- Run **cd src/main/java/com/simplilearn/app**
- Open App.java in a text editor. Example: nano App.java
- Replace the contents with the below code and save it:

```
package com.simplilearn.app;

import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.URL;

public class App{

    public static String checkResponse(String url) throws IOException{
```

```
    HttpURLConnection connection = (HttpURLConnection) new
URL(url).openConnection();

    connection.setRequestMethod("HEAD");
    int responseCode = connection.getResponseCode();
    String response = "Success";
    if (responseCode != 200) {
    response = "Failed";
    }
    return response;
    }

    public static void main(String[] args) throws IOException{

    System.out.println(checkResponse("https:/"+"/samples.openweathermap.org/data/2
.5/weather?q=London,uk&appid=b6907d289e10d714a6e88b30761fae22"));
    }

}
```

Step 2: Create a Jenkins build job to build a Maven project

- Click on *New Item* in the Jenkins dashboard.
- Enter a name for your Maven project.
- Select *Maven Project* as the build job type.


New Item [Jenkins] - Mozilla Firefox
New Item [Jenkins]
localhost:8080/view/all/newJob
Jenkins
search
Jenkins › All ›

Enter an item name

url-check

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

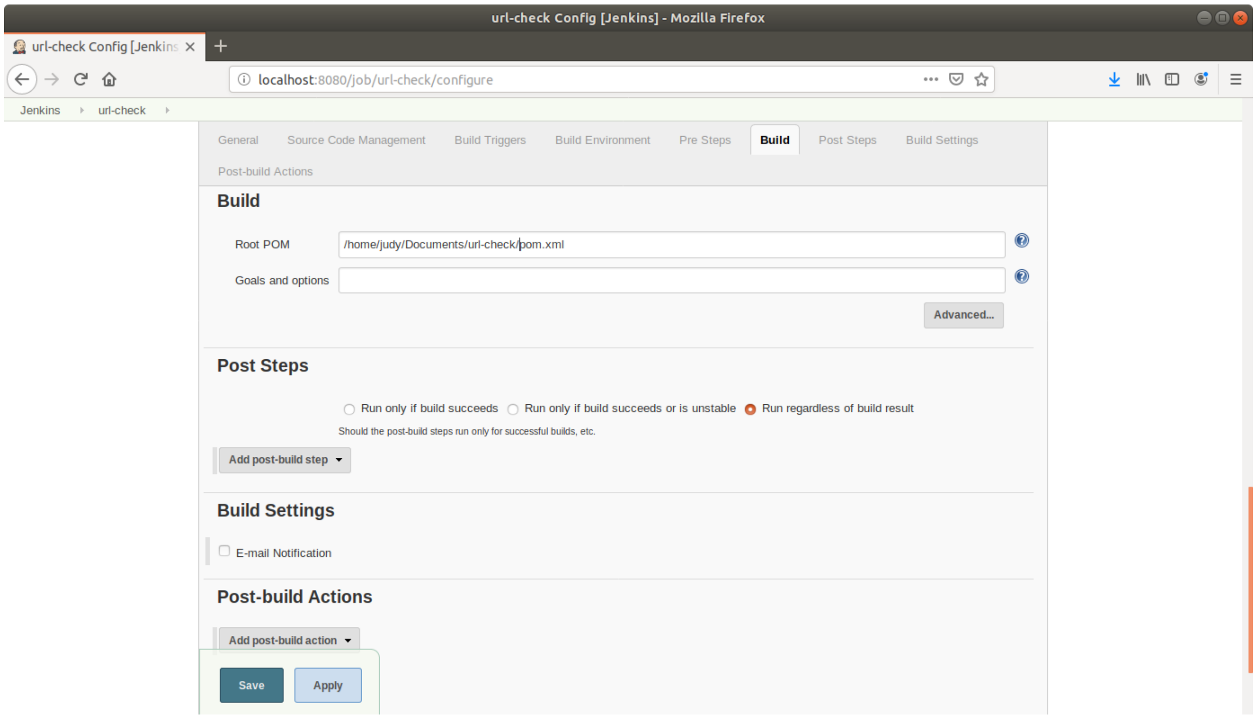
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Bitbucket Team/Project
Scans a Bitbucket Cloud Team (or Bitbucket Server Project) for all repositories matching some defined markers.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

- Click *OK*.
- Keep the defaults and scroll down to the *Build* section.
- Enter the path to the pom.xml.



Build

Root POM: /home/judy/Documents/url-check/pom.xml

Goals and options:

Post Steps

Run only if build succeeds [✓ Run regardless of build result] Run only if build succeeds or is unstable

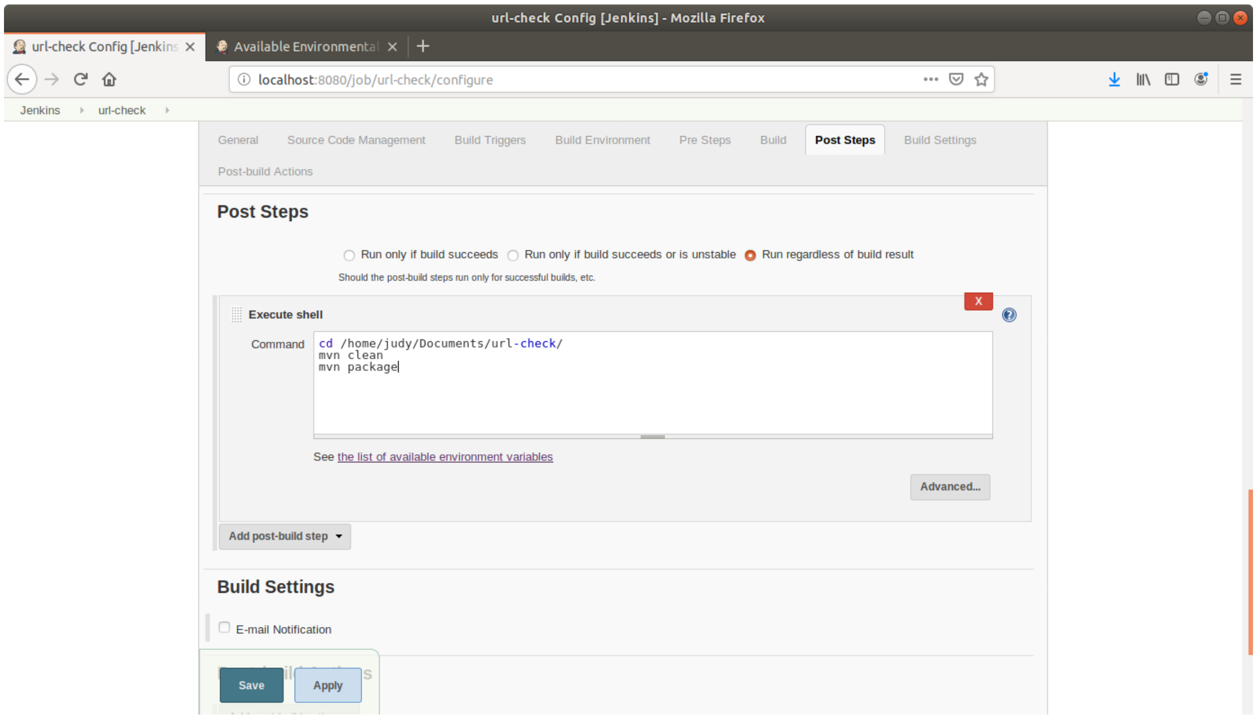
Should the post-build steps run only for successful builds, etc.

Build Settings

E-mail Notification

Post-build Actions

- Scroll down to the *Post-build Steps* section.
- Click on *Add post-build step* and choose *Execute shell* from the drop down menu.



Post Steps

○ Run only if build succeeds ○ Run only if build succeeds or is unstable [✓ ● Run regardless of build result]

Should the post-build steps run only for successful builds, etc.

Execute shell

Command

```
cd /home/judy/Documents/url-check/
mvn clean
mvn package
```

See the list of available environment variables

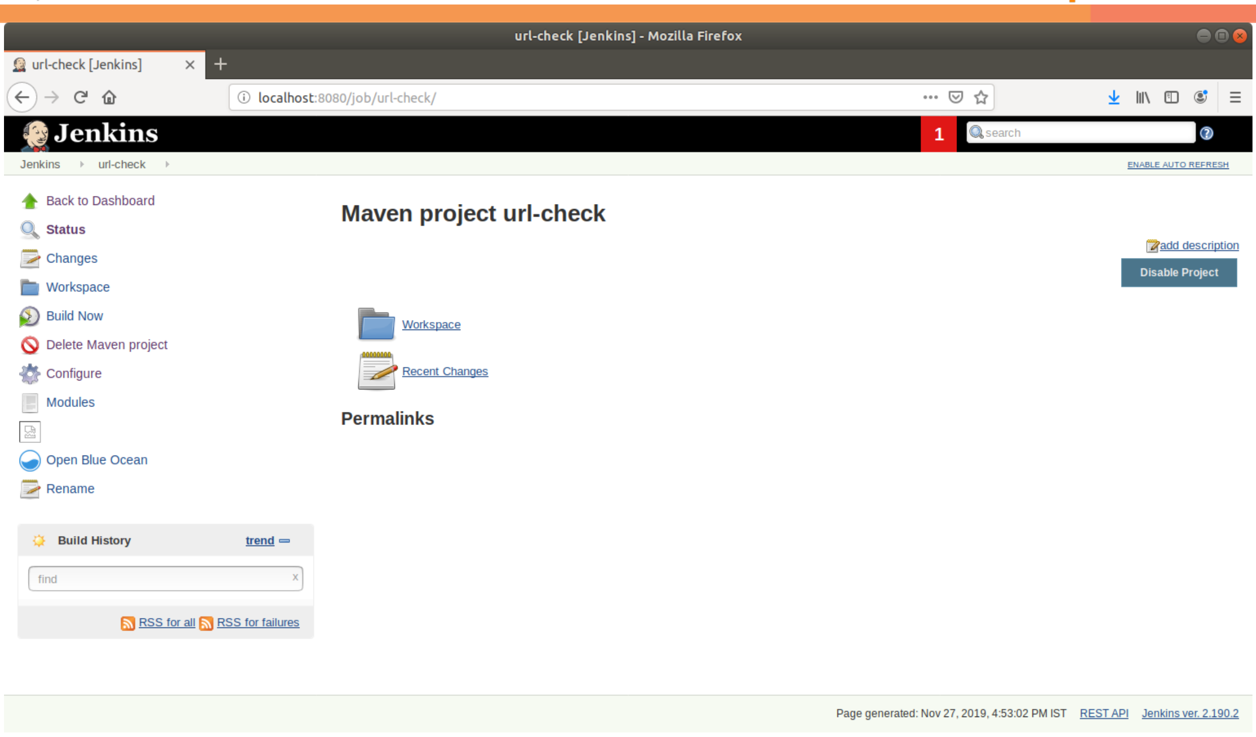
Build Settings

☐ E-mail Notification

- Enter the commands to navigate to app directory and build the package as shown above.
- Click *Save*.

Step 2.4.3: Building a Maven project in Jenkins

- Click on the project name in the Jenkins dashboard.
- Click *Build Now* in the project window. Jenkins will now build your project.

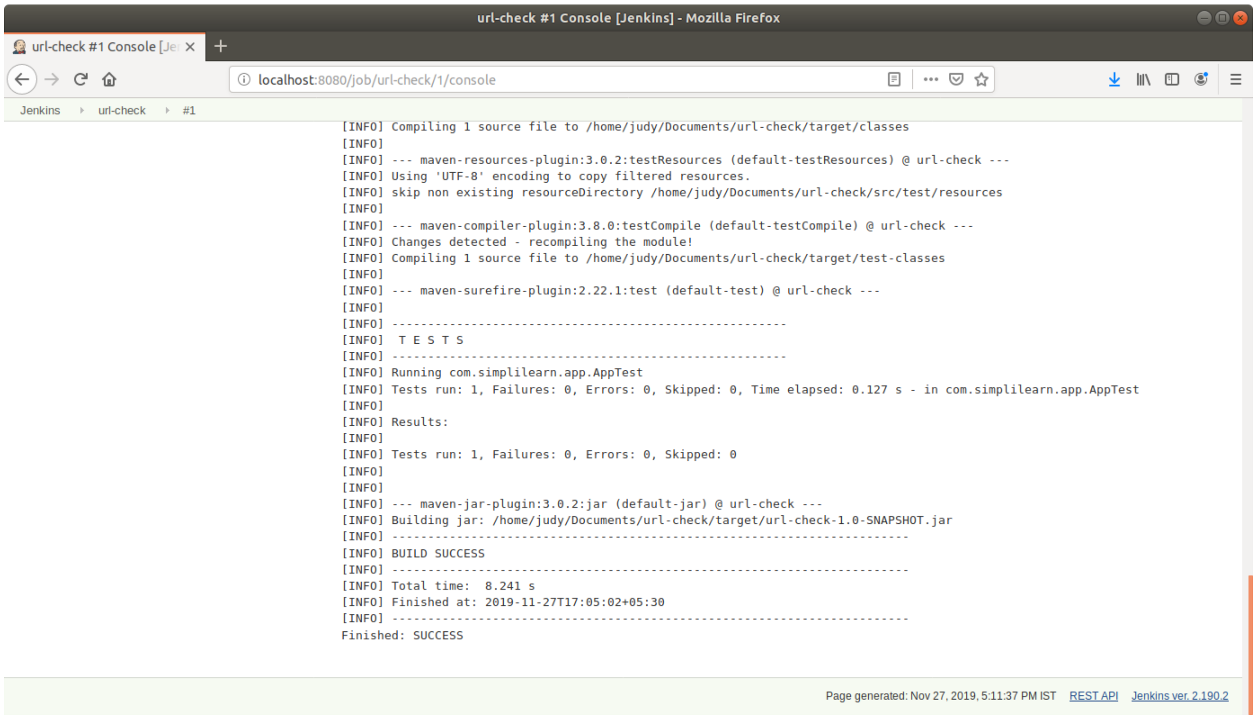


Maven project url-check

Permalinks

Page generated: Nov 27, 2019, 4:53:02 PM IST REST API Jenkins ver. 2.190.2

- Click on the *Build History* to view the build results.
- Click on the *Console Output* to view the build logs.


url-check #1 Console [Jenkins] - Mozilla Firefox
url-check #1 Console [Je
localhost:8080/job/url-check/1/console
Jenkins › url-check › #1

```
[INFO] Compiling 1 source file to /home/judy/Documents/url-check/target/classes
[INFO]
[INFO] --- maven-resources-plugin:3.0.2:testResources (default-testResources) @ url-check ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/judy/Documents/url-check/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:testCompile (default-testCompile) @ url-check ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /home/judy/Documents/url-check/target/test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.22.1:test (default-test) @ url-check ---
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running com.simplilearn.app.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.127 s - in com.simplilearn.app.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:3.0.2:jar (default-jar) @ url-check ---
[INFO] Building jar: /home/judy/Documents/url-check/target/url-check-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  8.241 s
[INFO] Finished at: 2019-11-27T17:05:02+05:30
[INFO] ------------------------------------------------------------------------
Finished: SUCCESS
```

Page generated: Nov 27, 2019, 5:11:37 PM IST REST API Jenkins ver. 2.190.2