

Test management tools

Accessing Testlink and Jenkins services

1. Accessing campus services (private network of the IT department) using SSH tunneling

This tutorial uses for example the Jenkins and Testlink services installed on machine 172.30.0.8 (private IP) and accessible (physically only connected to the network of the Campus Department) at the URLs:

<http://172.30.0.8:8080>

<http://172.30.0.8/testlink>

The SSH tunneling mechanism works similarly for any service or server hosted on a machine with a private IP address within the department's network. The necessary steps to access the two services from anywhere on the Internet, through SSH tunneling are:

Step 1. Connecting and authenticating

Students must log on to the server www.scs.ubbcluj.ro, the external SSH being open to students on port 8937. For students, the two putty commands look like this:

```
putty.exe -L 8080:172.30.0.8:8080 www.scs.ubbcluj.ro -P 8937
```

```
putty.exe -L 80:172.30.0.8:80 www.scs.ubbcluj.ro -P 8937
```

Step 2. Accessing the services

After successfully connecting and authenticating from step 1, keeping the putty connections open, the two services are available in your browser at the following addresses:

<http://localhost:8080>

<http://localhost/testlink>

Remarks:

- If only the individual use of Jenkins or Testlink is desired, only one of the putty commands from step 1 can be run individually.
- It is possible that ports 80 or 8080 on your local computer are occupied by other processes (other Web servers, various antivirus, Skype, etc.), other ports can be chosen, it is not mandatory that the local port be the same as the port on which run the service on the car you want to access.

```
putty.exe -L 8123:172.30.0.8:8080 www.scs.ubbcluj.ro -P 8937
```

with access then to

<http://127.0.0.1:8123>

```
start putty.exe -L 80:172.30.0.8:8080 avescan@www.scs.ubbcluj.ro -P 8937
```

with access then to

<http://127.0.0.1/testlink>

For Testlink

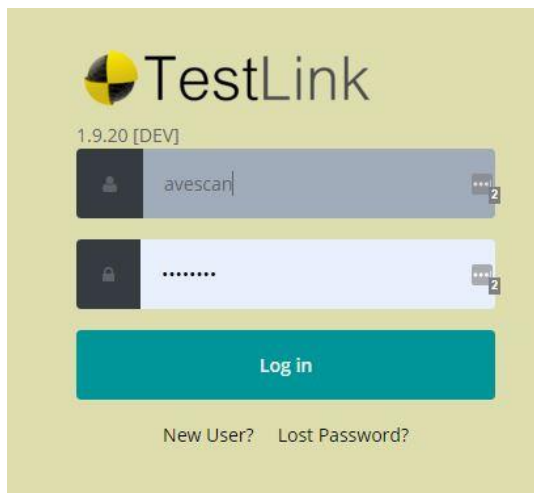
```
putty -P 8937 gdie2530@www.scs.ubbcluj.ro -L 8080:172.30.0.8:80
```

<http://127.0.0.1:8080/testlink>

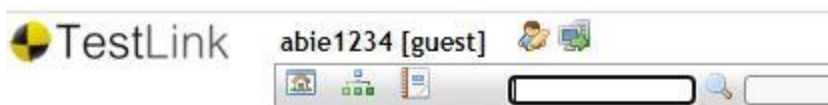
Steps for using Testlink

1. Create new user

1.1. Create a new user using your scs email



1.2. The account is created (the error should be ignored; the account is created)



Keyword Management

Browse Test Cases

Search Test Cases

1.3. Your laboratory teacher will assign you to the group project Projects93X

TestLink admin [admin] pr931- Test Project [pr931-Projects931]

User Management - Edit User

User details

Login: able1234
First Name: Ana
Last Name: Bob
Email: able1234@cs.ubbcluj.ro
Role: leader
Locale: English (wide/LK)
Authentication method: DefaultDB
Active: ☒
Expiration Date:

Save Cancel

TestLink able1234 [leader] pr931- Test Project [pr931-Projects931]

Keyword Management
Platform Management
Inventory management

Test Specification
Search Test Cases
Test Cases created per User

Current Test Plans: TestPlan_Name1Name2 OK

Test Plan Management
Builds / Releases

Execute Tests
Test Reports and Metrics
Metrics Dashboard

Add / Remove Test Cases
Assign Test Case Execution

Powered by TestLink 1.9.20 [DEV]

Important remark 1. Only admin user can create project. For each group we have created a project, thus each student will use the project name from his group.

The already created projects are:

- Projects931
- Projects932
- Projects933
- Projects934
- Projects935
- Projects936
- Projects937
- ProjectsOthers (for the students from other years of study or/and reenrolment)

Important remark 1. To each project there are two custom fields assigned:

- JavaClassName
- JavaTestMethodName

TestLink avescan [admin] prOthers- Test Project [prOthers-ProjectsOthers]

Assign custom fields - Test Project: ProjectsOthers

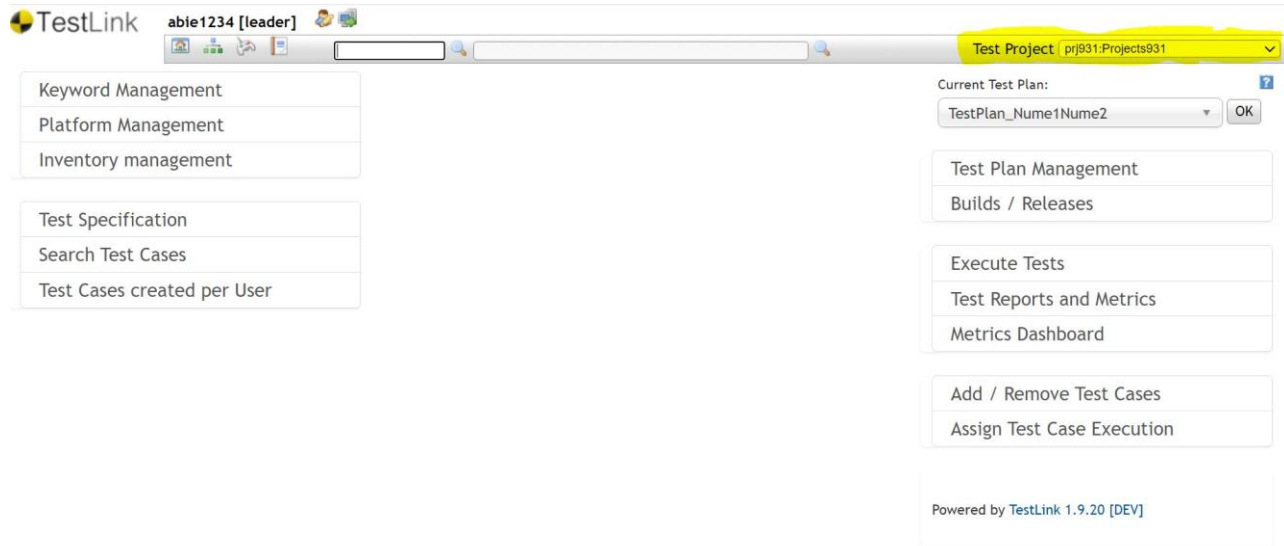
Available custom fields

Name	Label	Type	Available for
<input checked="" type="checkbox"/> JavaClassName	name of the autotest class	string	Test Case
<input checked="" type="checkbox"/> JavaTestMethodName	name of the test method	string	Test Case

Assign

2. Select the project that you are assigned to

- Project Menu
- In the upper right corner select from the combo box your Test Project.



3. Create a Test Plan

Each team will create his/her own Test Plan, the name of the test plan will be composed by groupNumber followed by the names of the team members concatenated with TestPlan.

93X_Name01Name02_TestPlan

Select “Test Plan Management” and create the test plan using the naming convention above.

TestLink abie1234 [leader] prj931-

Test Plan Management - Test Project Projects931

Name 931_Name01Name02_TestPlan

Description

Source

Format B I U S X₂ X² [Icons]

Active ☒

Public ☒

Create Cancel

TestLink abie1234 [leader] prj931- Test Project prj931: Projects931

Keyword Management
Platform Management
Inventory management

Test Specification
Search Test Cases
Test Cases created per User

Current Test Plan:
931_Name01Name02_TestPlan OK

Test Plan Management
Builds / Releases

Execute Tests
Test Reports and Metrics
Metrics Dashboard

Add / Remove Test Cases
Assign Test Case Execution

Powered by TestLink 1.9.20 [DEV]

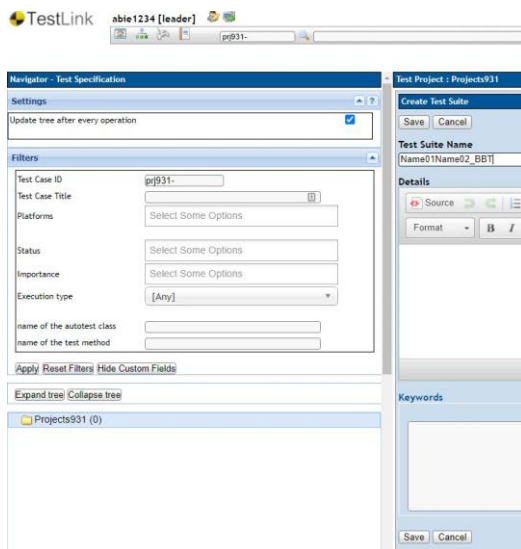
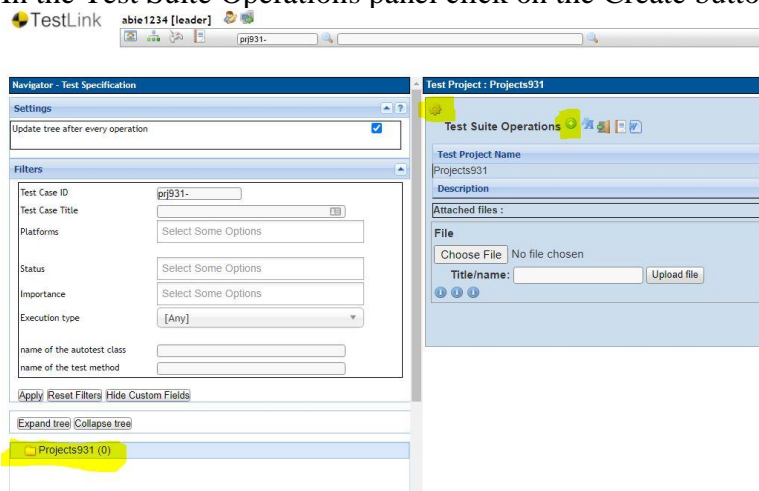
4. Create Test Suites/ Test Cases

Each student will create in total 3 test suites:

- 1 test suite from laboratory 2 - black-box testing
 - Test Suite Name = **Name01Name02_BBT**
- 1 test case from laboratory 3 - white-box testing
 - Test Suite Name = **Name01Name02_WBT**
- 1 test case from laboratory 4 - integration testing.
 - Test Suite Name **Name01Name02_IntegrationT**

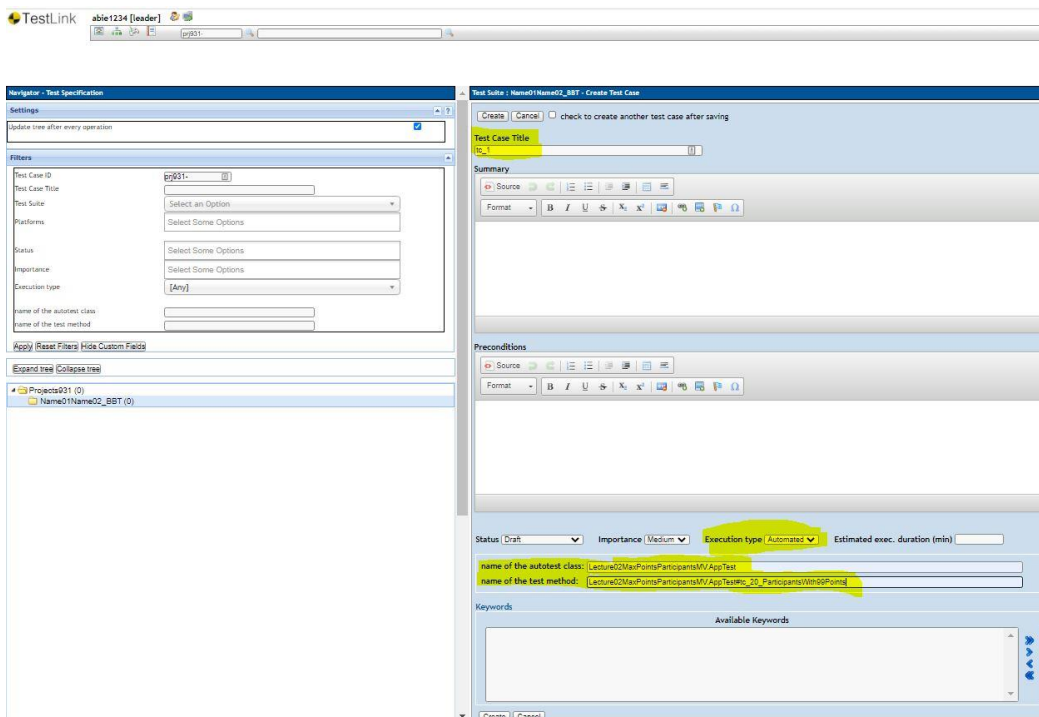
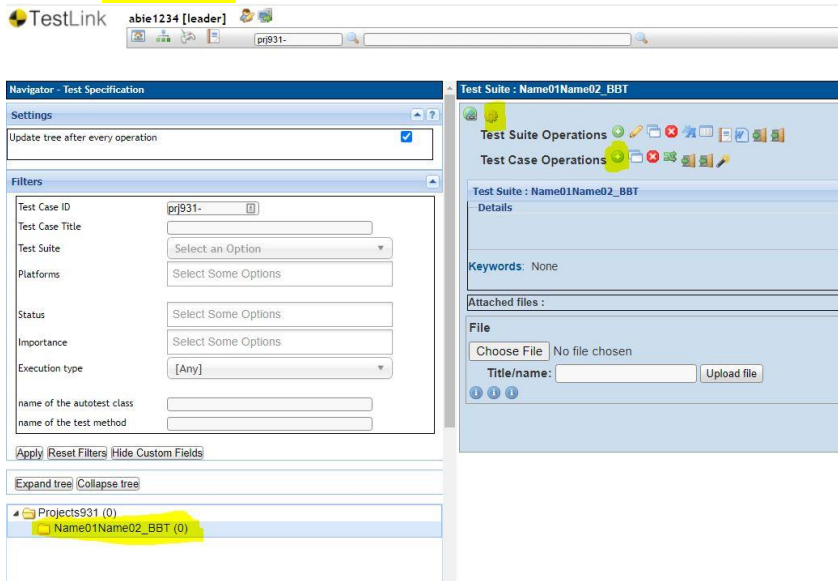
Create Test Suites - STEPS

- Project/Desktop menu --> Test Specification section --> click on Test Specification
- Select your test project in navigation tree (left side screen)
- In the Test Suite Operations panel click on the Create button.

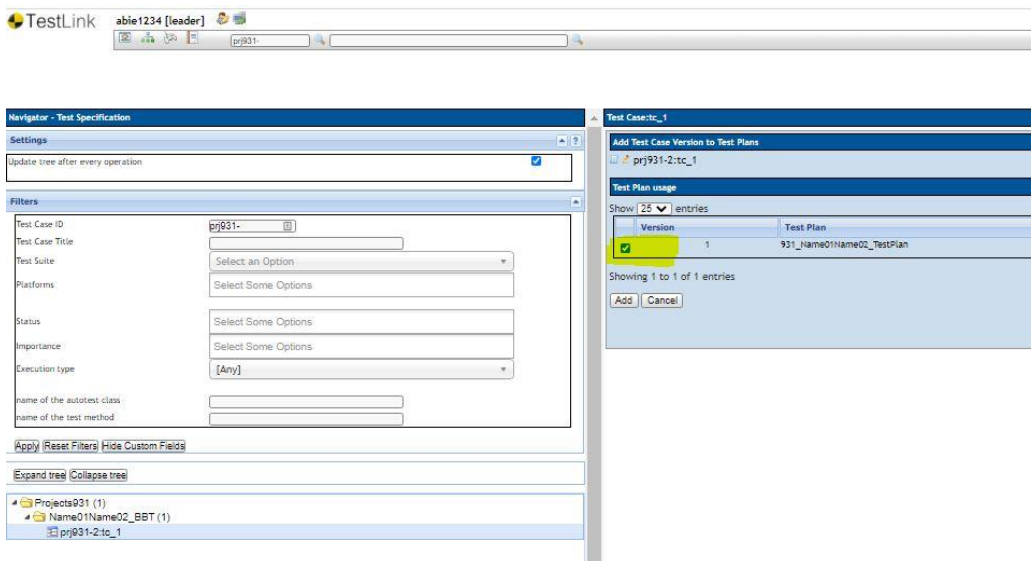
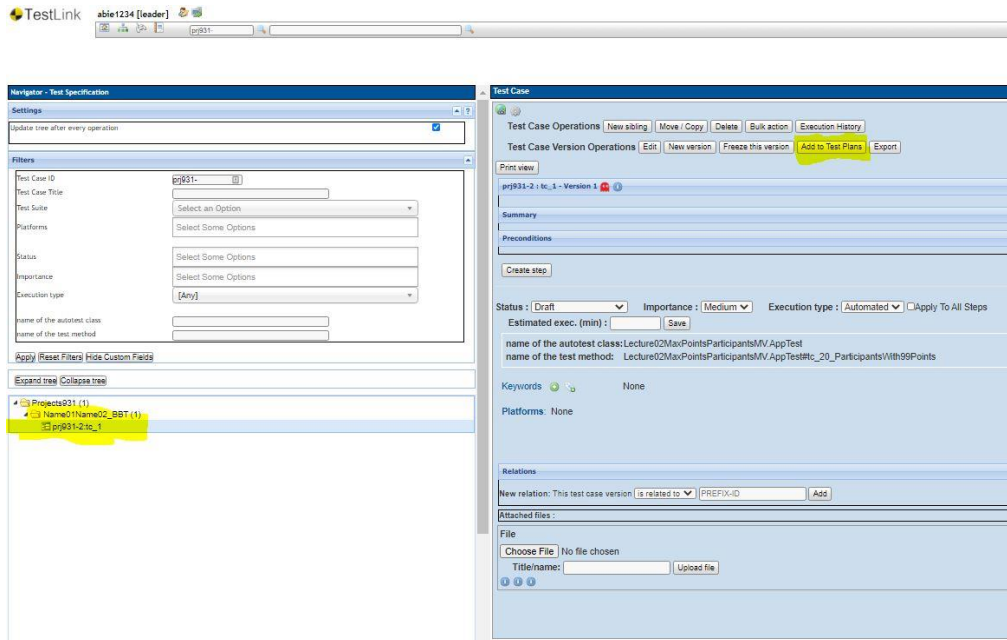


5. Create Test Cases - STEPS

- Select your test suite in navigation tree (left side screen)
- In the Test Suite Operations/ Test Case Operations panel click on the Create button (from Test Case Operation). Figure 6a and Figure 6b. Remember to modify Execution type to automated.



6. Add a Test Case to a Test Plan



Steps for using Jenkins

1. Create new user

1.1. Select “create an account”



Welcome to Jenkins!

Please sign in below or [create an account](#).

1.2 Create the new account using your scs email

Create an account!

If you already have a Jenkins account, [please sign in](#).

Username
able1234

Full name
Aaaaaa Bbbbbb

Email
able1234@scs.ubbcluj.ro

Password ☐ Show

Strength: **Moderate**

A strong password is a long password that's unique for every site. Try using a phrase with 5-6 words for the best security.

Create account

1.3. The account is created

 **Jenkins**

Dashboard > Aaaaaa Bbbbbb

 People

 Status

 Builds

 Configure

 My Views

 Credentials

Aaaaaa Bbbbbb

Jenkins User ID: able1234

2. Create a new Job

Each team will create his/her own **Job (for the TestPlan created in Testlink)**, the name of the job will be composed of: groupNumber followed by the names of the team members concatenated with the word Job. (**93X_Name01Name02_Job**)

- for example, for one of the team in group 931 Pop and Popescu the name of the job will be:
 - **931_PopPopescu_Job.**

Jenkins

Dashboard > All >

Enter an item name

93X1Name01Name02_Job

» 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 ev

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workf

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

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder on long as they are in different folders.

GitHub Organization
Scans a GitHub organization (or user account) for all repositories matching some defined markers.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

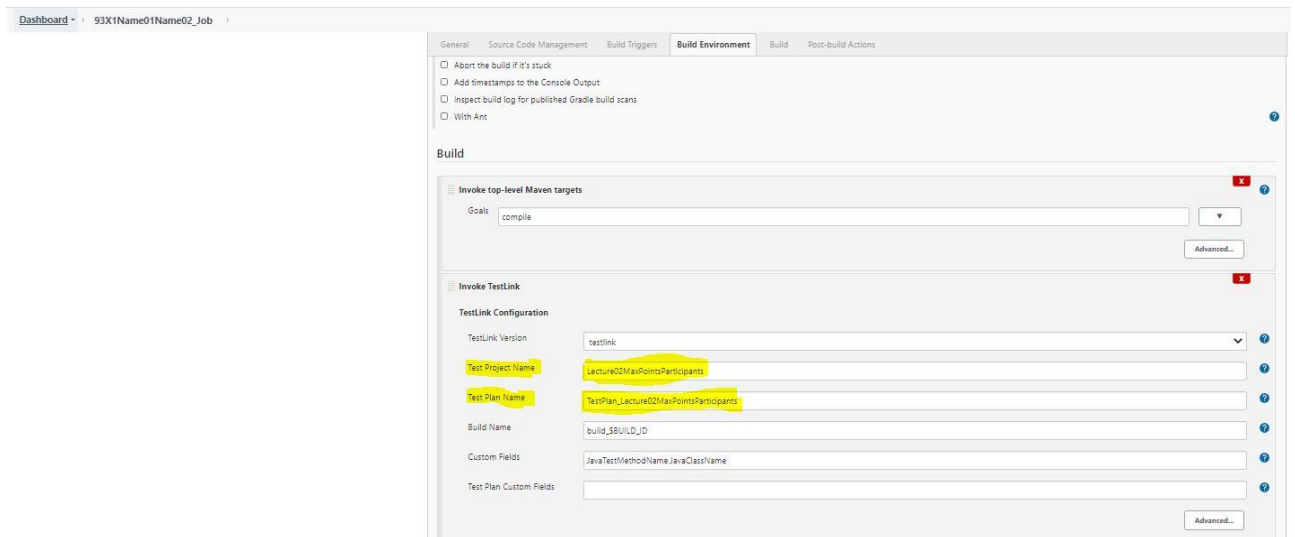
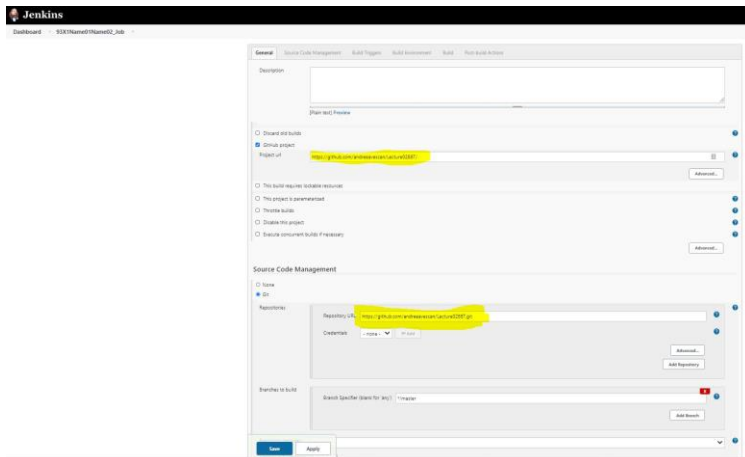
If you want to create a new item from other existing, you can use this option:

Copy from Lecture02B8T

OK

Create a new Job from other existing job

1. Use “Copy from”: Lecture02BBT
2. Modify in your Job the following elements
 - a. Github link with your Github link
 - b. Test Project and Test Plan with
 - c. Lecture02MaxPointsParticipantswith Projects93X
 - d. TestPlan_Lecture02MaxPointsParticipants with 93X_Name01Name02_TestPlan (created in Testlink).



The screenshot shows the Jenkins configuration page for a job named '93X1Name01Name02_Job'. The 'Build Environment' tab is selected. Under the 'Build' section, the 'Invoke top-level Maven targets' goal is set to 'compile'. The 'Invoke TestLink' section is expanded, showing the 'TestLink Configuration' with the following values: 'TestLink Version' is 'testlink', 'Test Project Name' is 'Project931', 'Test Plan Name' is '931_Name01Name02_TestPlan', 'Build Name' is 'build_\${BUILD_ID}', 'Custom Fields' is 'JavaTest/Method/Name/JavaClassName', and 'Test Plan Custom Fields' is empty. The 'Test Project Name' and 'Test Plan Name' fields are highlighted with a yellow box.

3. Build

- Select the Job
- Select Build Now

The screenshot shows the Jenkins main interface. The 'Jenkins' logo is at the top left. Below it, the breadcrumb 'Dashboard > 93X1Name01Name02_Job' is visible. On the left sidebar, the 'Build Now' button is highlighted with a blue border. Other buttons in the sidebar include 'Back to Dashboard', 'Status', 'Changes', 'Workspace', 'Configure', 'Delete Project', 'TestLink results', 'GitHub', and 'Rename'. The main content area shows the 'Project 93X1Name01Name02_Job' title, a 'Workspace' folder icon, a 'Recent Changes' icon, and a 'Permalinks' section. At the bottom, there is a 'Build History' section with a search bar and a table showing build history.

○ Console Output

Dashboard
93X1Name01Name02_Job
#2

Back to Project

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete build '#2'

Git Build Data

TestLink

Previous Build

Console Output

Started by user Aaaaaa Bbbbbb

Running as SYSTEM

Building in workspace /var/lib/jenkins/workspace/93X1Name01Name02_Job

The recommended git tool is: NONE

No credentials specified

```
> git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/andreeavescan/Lecture02BOT.git # timeout=10
Fetching upstream changes from https://github.com/andreeavescan/Lecture02BOT.git
> git --version # timeout=10
> git --version # 'git version 2.25.1'
> git fetch --tags --force --progress -- https://github.com/andreeavescan/Lecture02BOT.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 0b173888c809a44d23358b0520136ec59788a3d1 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 0b173888c809a44d23358b0520136ec59788a3d1 # timeout=10
Commit message: "Initial commit"
> git rev-list --no-walk 0b173888c809a44d23358b0520136ec59788a3d1 # timeout=10

[93X1Name01Name02_Job] $ mvn compile
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by com.google.inject.internal.cglib.core.$ReflectUtils$1 (file:/usr/share/maven/lib/guice.jar) to method java.lang.ClassLoader
WARNING: Please consider reporting this to the maintainers of com.google.inject.internal.cglib.core.$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
[21:34nINFO][m] Scanning for projects...
[21:34nINFO][m] [1m--< [0:36mLecture02MaxPointsParticipantsMV:Lecture02MaxPointsParticipants[0:1m >---[m
[21:34nINFO][m] [1mBuilding Lecture02MaxPointsParticipants 1.0-SNAPSHOT[m
[21:34nINFO][m] [1m-----[ jar ]-----[m
[21:34nINFO][m]
[21:34nINFO][m] [1m--- [0:32maven-resources-plugin:3.0.2:resources[m @ [1m(default-resources)[m @ [0:36mLecture02MaxPointsParticipants[0:1m ---[m
[21:34nINFO][m] Using 'UTF-8' encoding to copy filtered resources.
[21:34nINFO][m] skip non existing resourceDirectory /var/lib/jenkins/workspace/93X1Name01Name02_Job/src/main/resources
[21:34nINFO][m]
[21:34nINFO][m] [1m--- [0:32maven-compiler-plugin:3.8.0:compile[m @ [1m(default-compile)[m @ [0:36mLecture02MaxPointsParticipants[0:1m ---[m
[21:34nINFO][m] Nothing to compile - all classes are up to date
[21:34nINFO][m] [1m-----[m
[21:34nINFO][m] [1:32mBUILD SUCCESS[m
[21:34nINFO][m] [1m-----[m
[21:34nINFO][m] Total time: 0.569 s
[21:34nINFO][m] Finished at: 2021-03-06T07:26:33+02:00
[21:34nINFO][m] [1m-----[m
Preparing TestLink client APT.
Using TestLink URI: http://localhost/testlink/lib/api/xmlrpc/v1/xmlrpc.php

Found 1 automated test cases in TestLink.
```

13

4. Sending results from Jenkins to Testlink

After successfully running the job, in Testlink it is updated automatically the status of the test cases in Not Run, Passed, Failed, or Blocked.

5. View the state of a Test case in Testlink

In the Test Specification menu, within the chosen project, eg Projects931, a desired Test Suite is selected, e.g., abie1234_BBT, and a test case.

The screenshot displays the TestLink web interface. The left sidebar shows the 'Execute Tests' section with filters for Test Case ID (pg931), Test Suite (931_Name01Name02_TestPlan), and Build (build_2). The main content area shows the 'Test Results on Build build_2' for the test case 'pg931-2'. The test case is marked as 'Passed' on 06/03/2021 at 07:26:38. The test case details include the execution type (Automated), estimated execution duration (1 min), and the test method (Lecture02MaxPointsParticipantsMVI.AppTest). The test case is currently assigned to no tester.

Date	Build	Tested by	Status
06/03/2021 07:26:38	build_2	admin	Passed

Test Case pg931-2 : Version : 1 : to : 1
No tester assigned

Execution type : Automated
Estimated exec. duration (min) : 1
name of the autotest class : Lecture02MaxPointsParticipantsMVI.AppTest
name of the test method : Lecture02MaxPointsParticipantsMVI.AppTestMVI_20_ParticipantsWith99Points

Notes / Description

File (0) (0) (0)
Choose Files No file chosen

Important Notice: Once a Result is updated from 'Not Run' to another value, you cannot set it back to 'Not Run'. You can still set the Result to any other value.