

Conținut

Lista de figuri	1
1. Accesare Jenkins	2
2. Crearea unui cont în Jenkins.....	2
3. Crearea unui <i>Job</i>	2
4. Configurarea unui <i>Job</i>	3
5. Configurarea <i>Custom Fields</i> în TestLink pentru fiecare Test Case.....	9
6. Execuția unui <i>Job</i>	10
7. Transmiterea rezultatelor din Jenkins în TestLink.....	11
8. Vizualizarea stării unui Test Case în TestLink	11

Lista de figuri

Figure 1. Crearea unui nou <i>Job</i> în Jenkins	3
Figure 2. Opțiunea <i>Configure</i> pentru un <i>Job</i>	3
Figure 3. Setarea repository-ului	4
Figure 4. Setarea proiectului GitHub și a Repository-ului	4
Figure 5. Secțiunea Build, opțiunea Maven.....	5
Figure 6. Setarea comenzilor Maven.....	5
Figure 7. Secțiunea Build, opțiunea TestLink.....	5
Figure 8. Configurarea opțiunii TestLink.....	6
Figure 9. Secțiunea Build, opțiunea Test Execution	7
Figure 10. Configurarea opțiunii Test Execution (1)	7
Figure 11. Configurarea opțiunii Test Execution (2)	8
Figure 12. Configurarea opțiunii Test Execution (3)	8
Figure 13. Secțiunea Build, opțiunea Result Seeking Strategy	8
Figure 14. Configurarea opțiunii Result Seeking Strategy	9
Figure 15. Secțiunea Post-build Actions, opțiunea Archive the artefacts	9
Figure 16. Configurarea opțiunii Archive the artefacts. Salvarea configurărilor jobului.....	9
Figure 17. Configurarea <i>Custom Fields</i> pentru <i>Test Case</i> -urile create în TestLink	10
Figure 18. Execuția unui <i>job</i> in Jenkins.....	10
Figure 19. Vizualizarea statusului unui Test Case (not run/passed/failed/blocked)	11
Figure 20. Vizualizarea istoricului stărilor unui Test Case în TestLink	11

1. Accesare Jenkins

1. **Jenkins 2.263.4** este instalat pe o mașină virtuală a serverului SCS.
2. Accesul la Jenkins din afara Campusului FSEGA se poate face prin **SSH Tunneling** (vezi [tutorial SSH Tunneling](#)), utilizând comanda de conectare:
`putty.exe -L 8080:172.30.0.8:8080 www.scs.ubbcluj.ro -P 8937`
 - urmată de autentificare folosind user-ul si parola SCS;
 - după conectare la rețeaua din Campus, link-ul de acces la Jenkins poate fi:
 - <http://localhost:8080/login?from=%2F>, dacă s-a folosit folosit local portul 8080;
 - <http://localhost:8081/login?from=%2F>, dacă s-a folosit local portul 8081; este recomandată folosirea altui port, e.g., 8081, în cazul în care există aplicații care folosesc deja portul 8080.

2. Crearea unui cont în Jenkins

1. **Fiecare student** își va crea un cont pe platforma Jenkins.
2. Pentru userul cu adresa de e-mail xyir1234@scs.ubbcluj.ro se va crea un cont cu user name-ul **xyir1234**.

3. Crearea unui Job

1. Fiecare echipă va crea câte un **Job** pentru fiecare plan de testare creat în TestLink, i.e, **xyir1234_BBT_TP**, **xyir1234_WBT_TP**, **xyir1234_IntT_TP**;
2. Pentru echipa userul [xyir1234](#), numele job-urilor create vor fi:
 - **xyir1234Job_BBT** pentru rularea testelor din [Lab02](#);
 - **xyir1234Job_WBT** pentru rularea testelor din [Lab03](#);
 - **xyir1234Job_IntT** pentru rularea testelor de integrare din [Lab04](#).
3. Din meniul **Jenkins** ---> **New Item**; se completează câmpurile:
 - **Name:** **xyir1234Job_AAA**
 - se selectează opțiunea **Freestyle project**
4. **OK** pentru a finaliza crearea Jobului (vezi Figure 1).

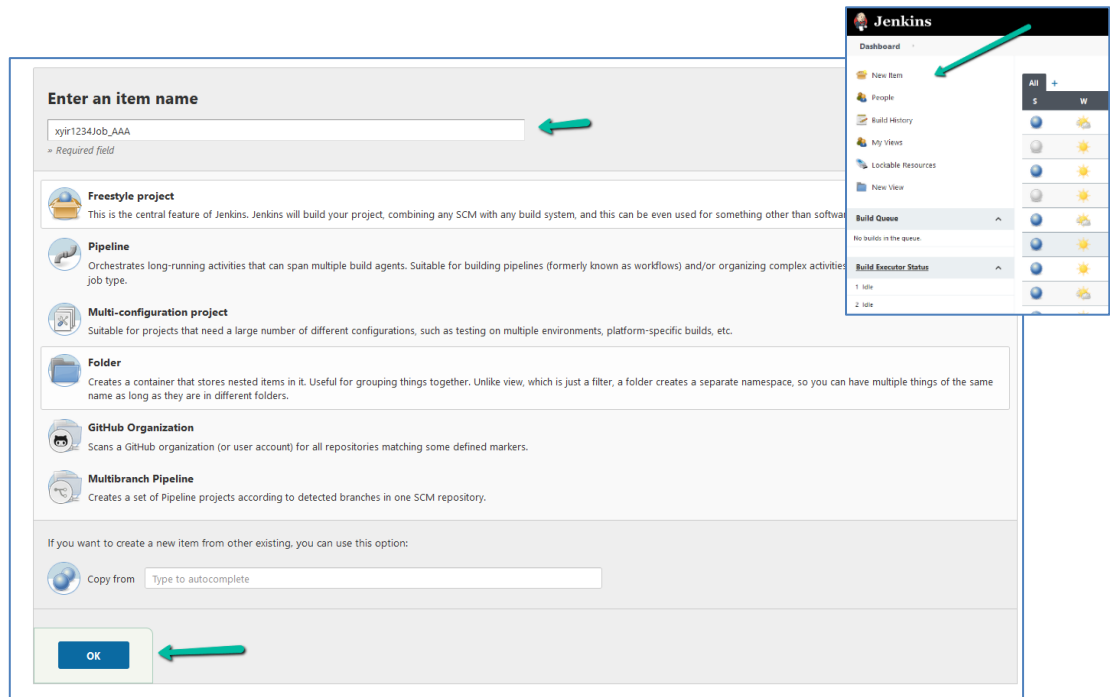


Figure 1. Crearea unui nou Job în Jenkins

5. La crearea unui job nou, **fereastra de configurare se deschide implicit**. Pentru **modificarea** configurărilor unui job existent se urmează pașii din **Secțiunea 4. Configurarea unui Job**.

4. Configurarea unui Job

- din fereastra principală se selectează jobul dorit din lista de joburi existente;
- din meniul din partea stângă, în alege opțiunea **Configure** (vezi Figure 2);

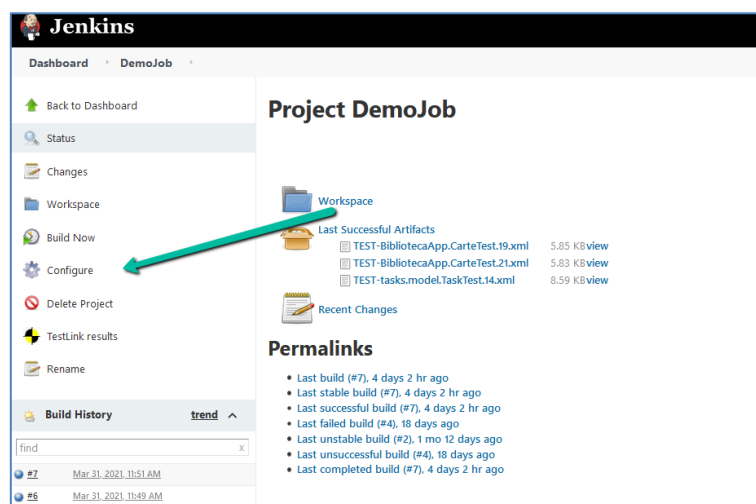


Figure 2. Opțiunea Configure pentru un Job

- în tab-ul General, în secțiunea **Source Code Management** (vezi Figure 3):
 - se alege tipul de Repository: **Git**;
 - se completează **Repository URL**:
<https://vvss-UBB@bitbucket.org/vvss-UBB/xyir1234.git>

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

☐ GitHub project
☐ This build requires lockable resources
☐ This project is parameterized
☐ Throttle builds
☐ Disable this project
☐ Execute concurrent builds if necessary

Source Code Management

☐ None
☒ Git

Repositories

Repository URL:
 Credentials: [Add](#)

Branches to build

Branch Specifier (blank for 'any'):

Repository browser:

Additional Behaviours: [Add](#)

Figure 3. Setarea repository-ului

- Dacă se folosește un repository GitHub, atunci se completează numele proiectului GitHub și numele repository-ului, vezi Figure 4;

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

☒ GitHub project

Project url:

☐ This build requires lockable resources
☐ This project is parameterized
☐ Throttle builds
☐ Disable this project
☐ Execute concurrent builds if necessary

Source Code Management

☐ None
☒ Git

Repositories

Repository URL:
 Credentials: [Add](#)

Branches to build

Branch Specifier (blank for 'any'):

Figure 4. Setarea proiectului GitHub și a Repository-ului

4. În secțiunea **Build**:

- click **Add build step** și se alege opțiunea **Invoke top-level Maven targets** (vezi Figure 5):
 - se completează (vezi Figure 6):
 - **Maven Version:** **mvn**
 - **Goals:** **compile**

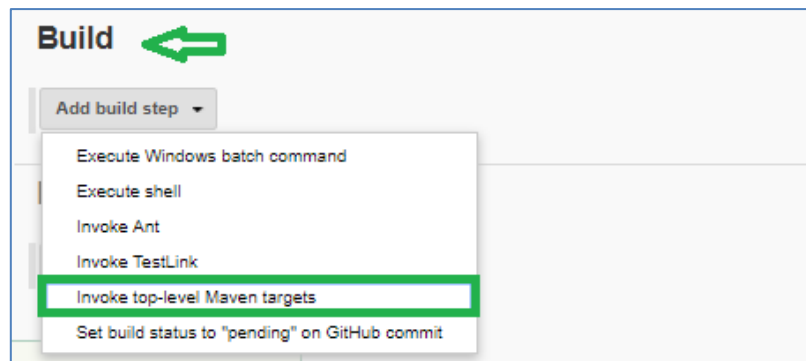


Figure 5. Secțiunea Build, opțiunea Maven

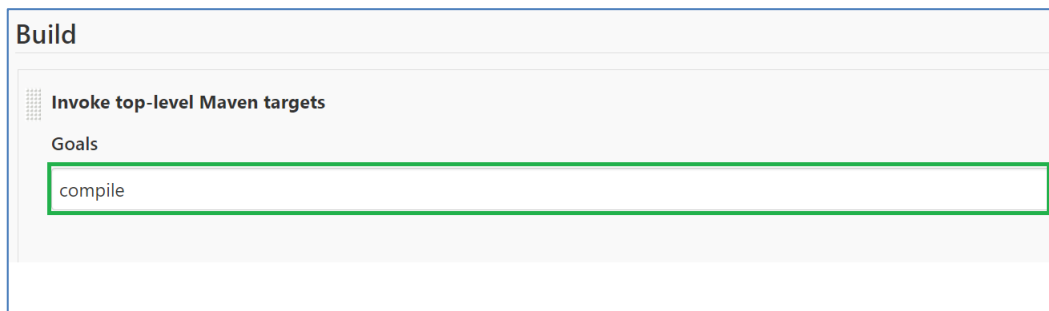


Figure 6. Setarea comenzilor Maven

- click **Add build step** și se alege opțiunea **Invoke TestLink** (vezi Figure 7);

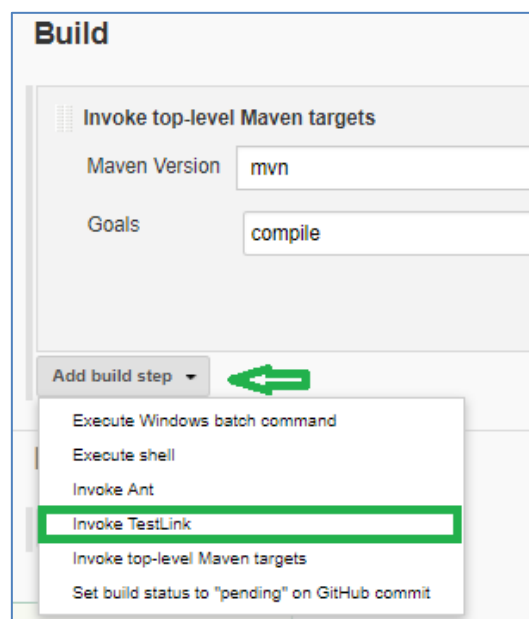


Figure 7. Secțiunea Build, opțiunea TestLink

- se continuă configurarea pentru opțiunile **TestLink Configuration**, **Test Execution** și **Result Seeking Strategy**;
 - [1]. opțiunea **TestLink Configuration** (vezi Figure 8):
 - **TestLink Version:** **testlink**
 - **Test Project Name:**
 - numele proiectului TestLink folosit, e.g., **Prj231**
 - **Test Plan Name:**

- numele planului de testare folosit, e.g., **xyir1234_BBT_TP**
- **Build Name:** **build_\${BUILD_ID}**
- **Custom Fields:**
 - se completează **ambii identificatori** ai câmpurilor cheie definiți, care au fost deja asociați proiectului TestLink, i.e., **JavaClassName, JavaTestMethodName**

Figure 8. Configurarea opțiunii TestLink

[2]. opțiunea **Test Execution** (vezi Figure 9):

- pentru opțiunea **Iterative Test Build Steps** click **Add action** și se alege **Invoke top-level Maven targets**:
 - se completează (vezi Figure 10, Figure 11, Figure 12):
 - **Maven Version:** **mvn**
 - **Goals:** **verify**
 - în secțiunea **Advanced**:
 - **Properties:** **test=\${TESTLINK_TESTCASE_JavaTestMethodName}**
 - click **Add action** și se alege opțiunea **Execute Shell**:
 - **Command:**

```
#!/bin/bash
mv target/surefire-reports/TEST-${TESTLINK_TESTCASE_JAVACLASSNAME}.xml
target/surefire-reports/TEST-
${TESTLINK_TESTCASE_JAVACLASSNAME}.${TESTLINK_TESTCASE_ID}.xml
```

Dacă nu se copiază corect codul de mai sus, încercați cu cel de jos, exact așa cum este.

```
#!/bin/bash
mv target/surefire-reports/TEST-${TESTLINK_TESTCASE_JAVACLASSNAME}.xml target/surefire-reports/TEST-${TESTLINK_TESTCASE_JAVACLASSNAME}.${TESTLINK_TESTCASE_ID}.xml
```

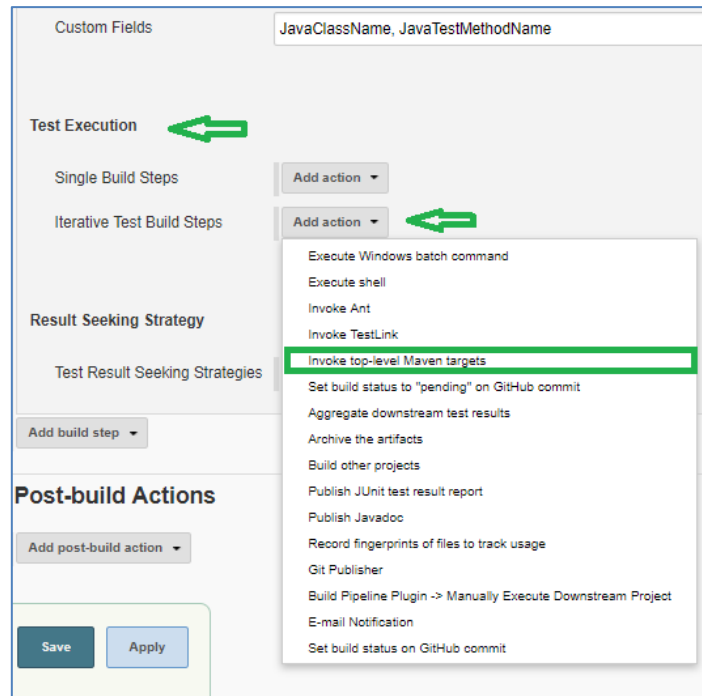


Figure 9. Secțiunea Build, opțiunea Test Execution

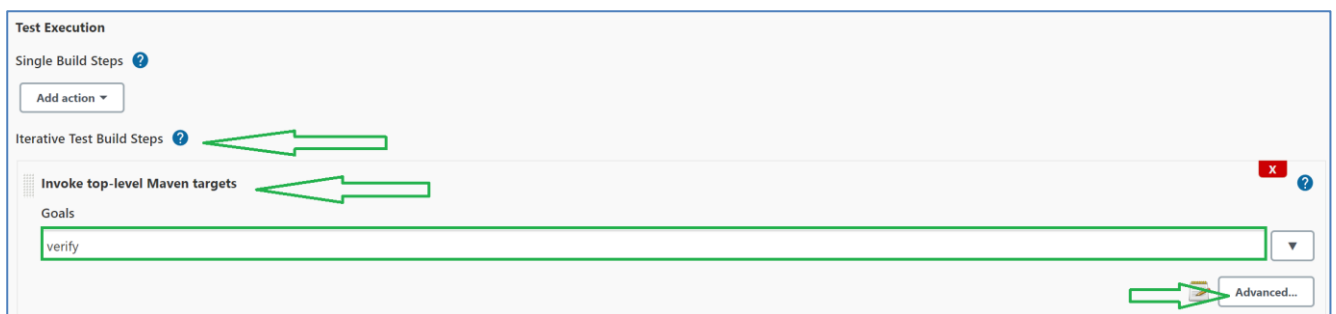


Figure 10. Configurarea opțiunii Test Execution (1)

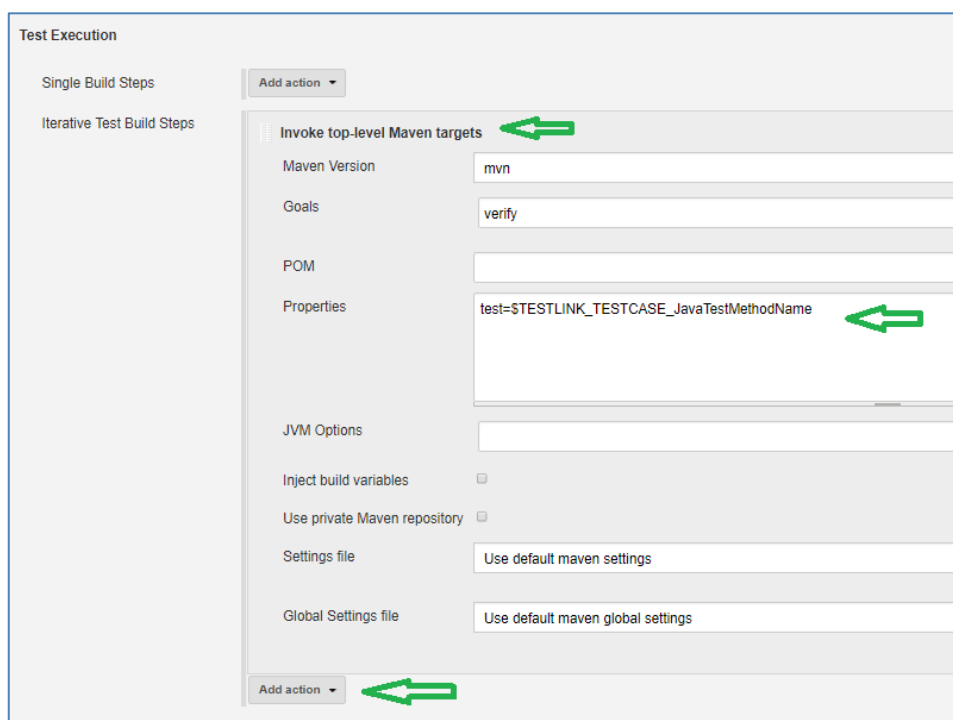


Figure 11. Configurarea opțiunii Test Execution (2)

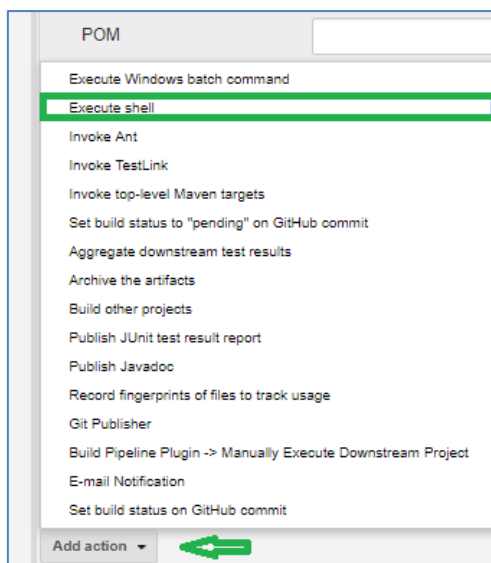


Figure 12. Configurarea opțiunii Test Execution (3)

- [3]. opțiunea **Result Seeking Strategy** (vezi Figure 13):
- click **Add strategy** se alege **JUnit method name**:
 - se completează (vezi Figure 14):
 - **Include Pattern**: **** /TEST-*.xml**
 - **Key Custom Field**: **JavaTestMethodName**
 - se bifează **Include test notes**;

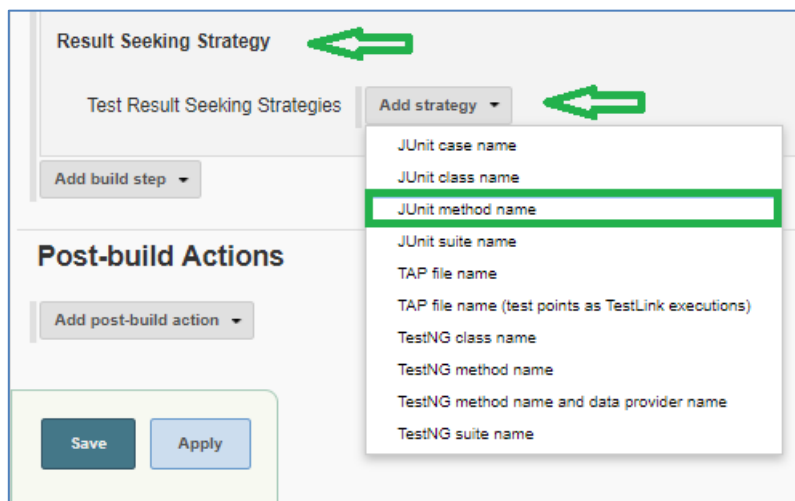


Figure 13. Secțiunea Build, opțiunea Result Seeking Strategy

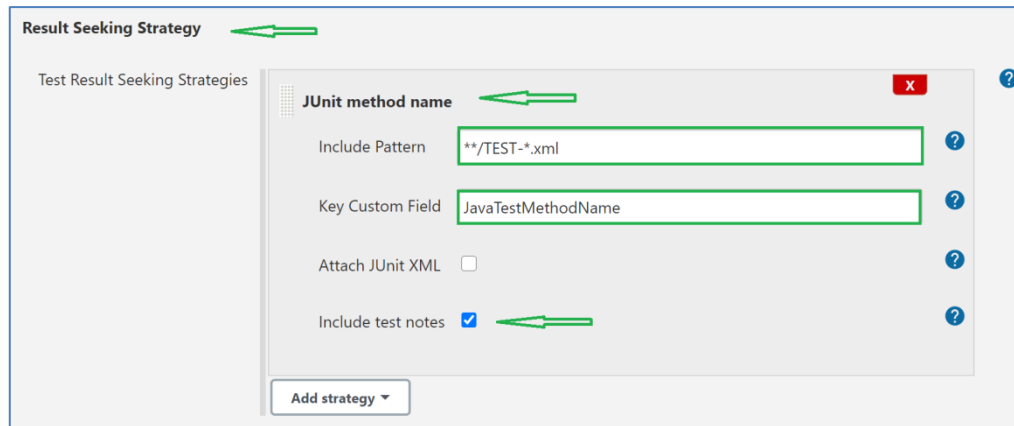


Figure 14. Configurarea opțiunii Result Seeking Strategy

5. În secțiunea **Post-build Actions**:

- click **Add post-build action** și se alege opțiunea **Archive the artifacts** (vezi Figure 15):
 - se completează (vezi Figure 16):
 - **Files to archive:** ****/TEST-*.xml**

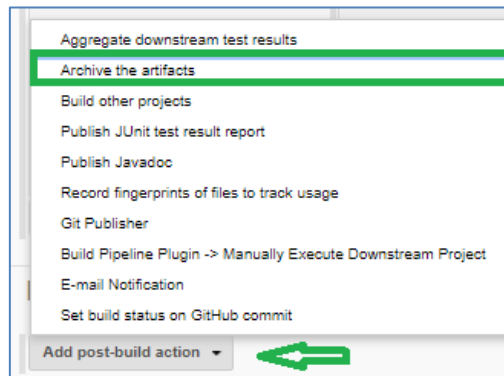


Figure 15. Secțiunea Post-build Actions, opțiunea Archive the artefacts

6. **Save** pentru a finaliza configurarea jobului (vezi Figure 16).

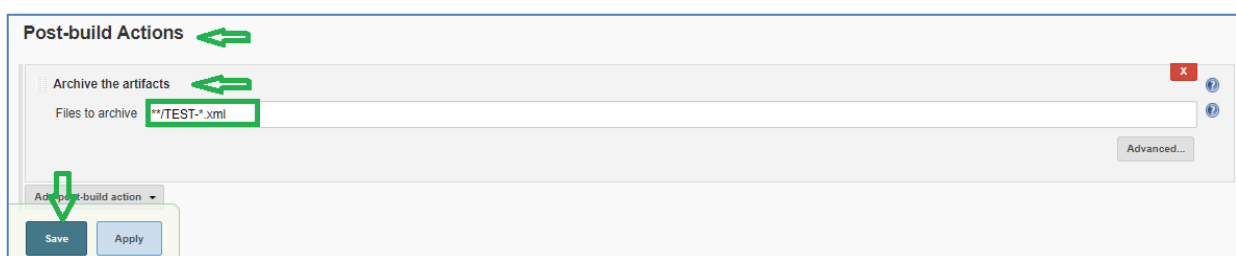


Figure 16. Configurarea opțiunii Archive the artefacts. Salvarea configurărilor jobului

5. Configurarea *Custom Fields* în TestLink pentru fiecare Test Case

- Se verifică în TestLink dacă **Custom Fields**, i.e., variabilele care permit comunicarea între TestLink și Jenkins, asociate fiecărui test case indică clasa de testare concretă și testul corespunzător (vezi Figure 17).
- **Este necesar să se aplice Observația de la pagina 14 din Tutorialul TestLink.**

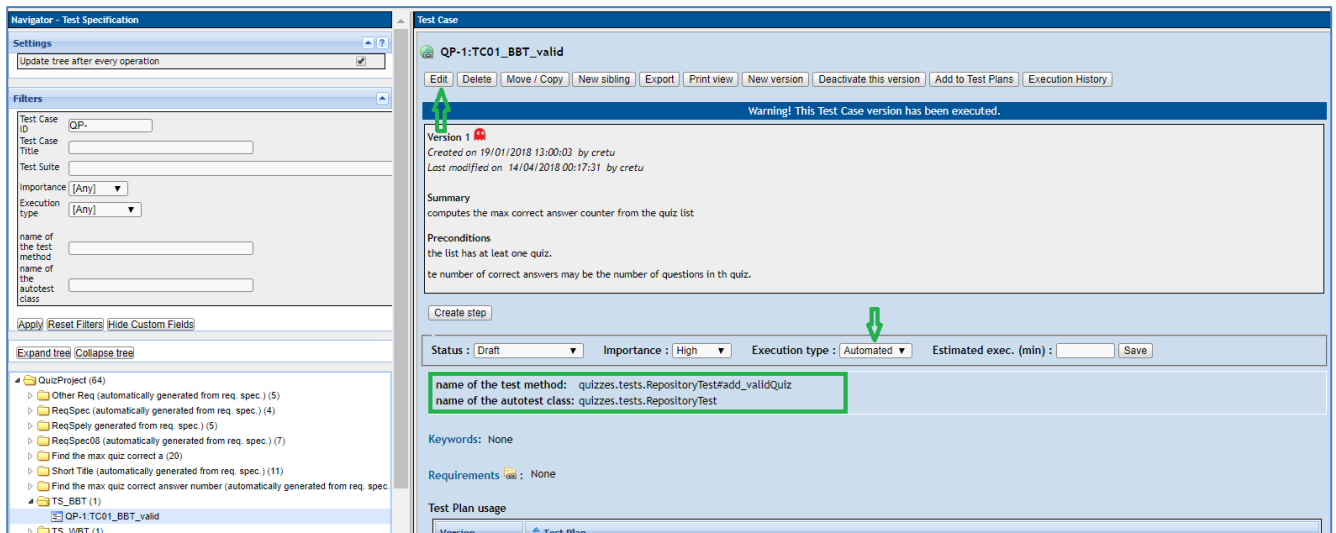


Figure 17. Configurarea Custom Fields pentru Test Case-urile create în TestLink

6. Execuția unui Job

1. din fereastra principală se selectează jobul dorit din lista de joburi (proiecte) existente;
2. din meniul din partea stângă se alege opțiunea **Build Now** și se așteaptă finalizarea execuției (vezi Figure 18) sau se poate selecta **Build**-ul curent al jobului pentru a vizualiza secțiunea **Console Output** în timpul execuției jobului;
3. în secțiunea **Build History**, când mouse-ul se află deasupra unui build executat prin click apare un meniu din care se alege opțiunea **Console Output**;
 - dacă execuția Build-ului s-a realizat cu succes se alege opțiunea **TestLink results** din meniul din stânga pentru a vizualiza rezultatele.

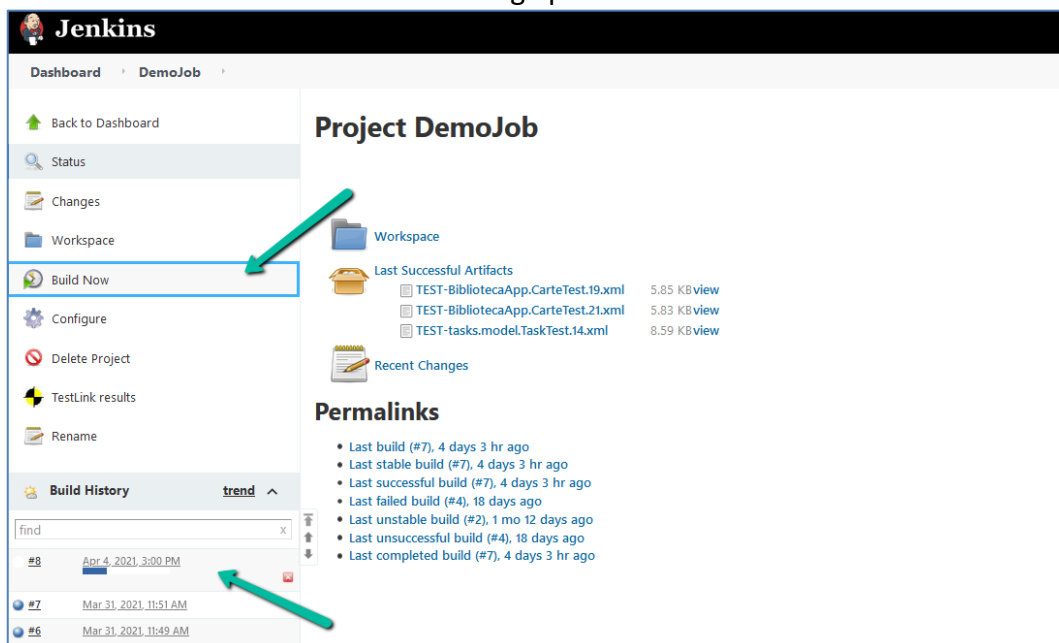


Figure 18. Execuția unui job in Jenkins

7. Transmiterea rezultatelor din Jenkins în TestLink

- După execuția cu succes a unui Job, testele gestionate de TestLink își actualizează starea automat în **Not Run**, **Passed**, **Failed** sau **Blocked** (vezi Figure 19).

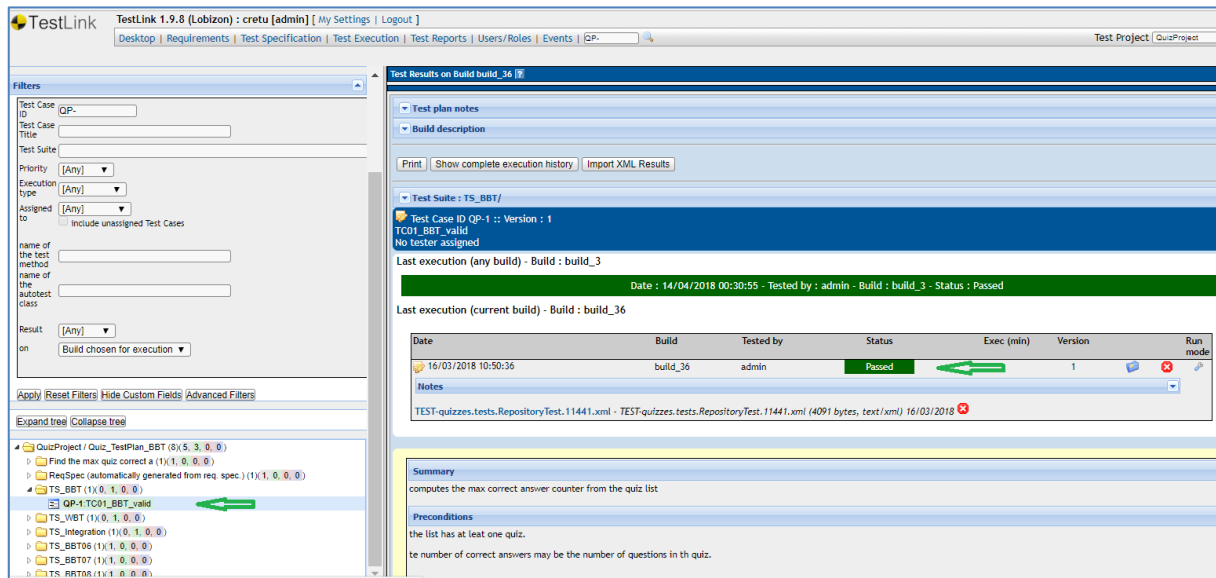


Figure 19. Vizualizarea statusului unui Test Case (not run/passed/failed/blocked)

8. Vizualizarea stării unui Test Case în TestLink

- În meniul **Test Specification**, în cadrul proiectului ales, e.g., **Prj231**, se selectează un **Test Suite** dorit, e.g., **xyir1234_BBT**, și un test case;
- în frame-ul din partea dreaptă se alege opțiunea **Execution History** (vezi Figure 20).

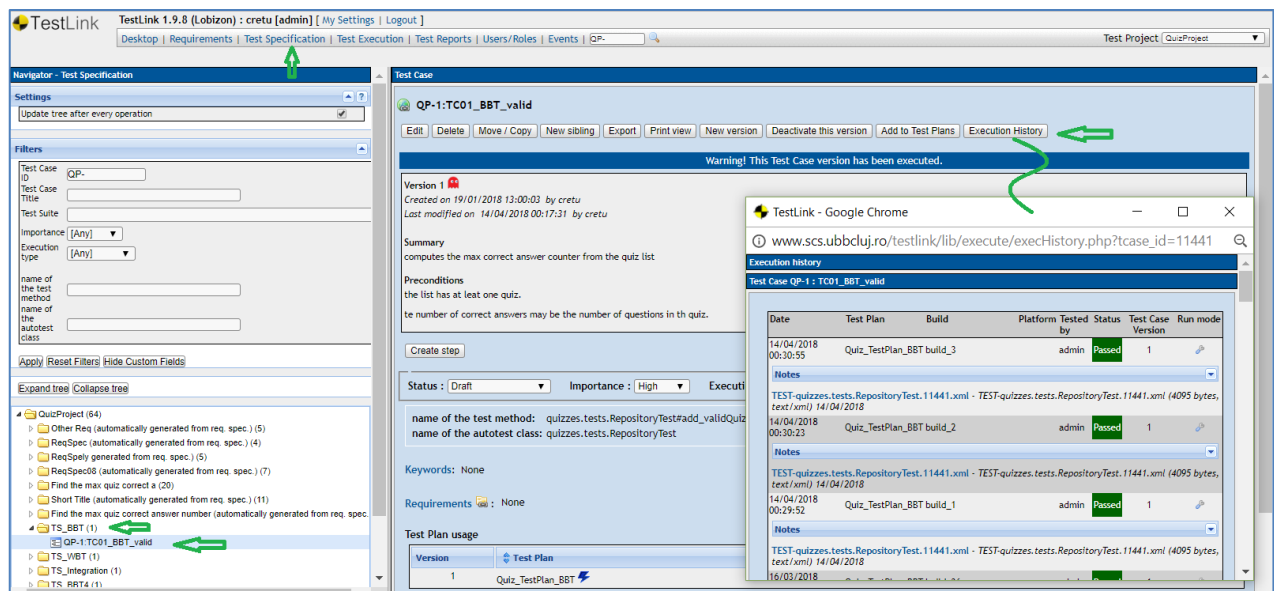


Figure 20. Vizualizarea istoricului stărilor unui Test Case în TestLink