



# Testing in DevOps

DOu – Certified Tester in DevOps (CTD)  
Exercise Solutions

## HO-2.2.2(HO-1)

Exercise - Perform a guided exercise of creating tests using the xUnit framework against a given code

- Add junit stage in jenkins pipeline and trigger the jenkins build. Identify the reason why Junit test is failed.

## HO-2.2.2(HO-1)

Exercise Solution - Perform a guided exercise of creating tests using the xUnit framework against a given code

- Make sure you are already logged in Jenkins and be on Dashboard UI
- Add Unit Test stage in Jenkinsfile.txt on local machine:
  - Clean(already added)
  - PMD(already added)
  - Compile(already added)
  - Code Analysis(already added)
  - **Unit Test**

## HO-2.2.2(HO-1)

### Add Stage – Unit Test

1. Unit Test defines a stage that appears on the Jenkins UI
2. **sh** runs the Maven command to run the unit tests

```
//Code starts for stage Unit Test
stage('Unit Test') {                               (1)
    steps {
        sh 'mvn test'                               (2)
    }
    post {
        always {
            junit 'target/surefire-reports/*.xml'
        }
    }
}
//Code ends for stage Unit Test
```

## HO-2.2.2(HO-1)

Exercise Solution - Perform a guided exercise of creating tests using the xUnit framework against a given code

- Saving Jenkinsfile.txt & Commit it on GitHub
- Running The Pipeline
  - Go to Jenkins Dashboard
  - Build will be triggered automatically because of webhook
  - Click Build no. link
  - Click Console Output link
  - Verify the build result is failed
- 3 Unit Tests will be failed
- Let's understand why they have failed
- Unit Test 1 Explanation:
  - The reason of failing "saveUserTest" test in "UserServiceTest.java" is:
  - we are trying to set username = Uth in "saveUser" method app code whereas username is already set to "SUN" in the "saveUserTest" test code, so it found mismatch in both usernames

## HO-2.2.2(HO-1)

Exercise Solution - Perform a guided exercise of creating tests using the xUnit framework against a given code

- src/main/java:

- package - com.expense.service.impl
- class - UserServiceImpl
- method - saveUser

```
@Override
public User saveUser(User user) {

    // Buggy Code (UserServiceTest.saveUserTest())
    user.setUsername("Uth");
    return user;

    // Correct Code
    //user.setPassword(passwordEncoder.encode(user.getPassword()));
    //return userRepository.save(user);

}
```

- src/test/java:

- package - com.expense.service
- class - UserServiceTest
- method - saveUserTest

```
@Test
public void saveUserTest() {

    User user = new User();
    user.setUsername("SUN");

    when(userRepository.save(Mockito.any(User.class))).thenReturn(user);
    User created = userService.saveUser(user);

    assertThat(created.getUsername(), is("SUN"));

}
```

## HO-2.2.2(HO-1)

Exercise Solution - Perform a guided exercise of creating tests using the xUnit framework against a given code

- Now see the reason of failing for remaining other 2 unit tests
- Fix the code for all 3 unit tests
- Saving Jenkinsfile.txt & Commit it on GitHub
- Running The Pipeline
  - Go to Jenkins Dashboard
  - Build will be triggered automatically because of webhook
  - Click Build no. link
  - Click Console Output link
  - Verify that all unit tests pass and then build result is successful

## HO-2.2.2(HO-1)

### Unit Test Result

- You can see result at below location in container:  
**`/var/jenkins_home/workspace/<your pipeline project name>/target/surefire-reports/`**
- You can open any junit report(txt file) by using cat command

E.g. **`cat  
com.expense.controller.ExpenseControllerTest.txt`**

```
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 11, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:04 min
```