

## JUnit5: Standard Tests

Step 1: Configuring pom.xml to add JUnit5 dependencies

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>UsingJUnit</groupId>
  <artifactId>UsingJUnit</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <dependencies>
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter-engine</artifactId>
      <version>5.4.2</version>
    </dependency>

    <dependency>
      <groupId>org.junit.platform</groupId>
      <artifactId>junit-platform-launcher</artifactId>
      <version>1.2.0</version>
    </dependency>
  </dependencies>
</project>
```

Step 2: Creating a StandardTestClass as a standard template for test

classes

```
package com.ecommerce.tests;
```

```
import static org.junit.jupiter.api.Assertions.fail;  
import static org.junit.jupiter.api.Assumptions.assumeTrue;
```

```
import org.junit.jupiter.api.AfterAll;  
import org.junit.jupiter.api.AfterEach;  
import org.junit.jupiter.api.BeforeAll;  
import org.junit.jupiter.api.BeforeEach;  
import org.junit.jupiter.api.Disabled;  
import org.junit.jupiter.api.DisplayName;  
import org.junit.jupiter.api.Test;  
import org.junit.platform.runner.JUnitPlatform;  
import org.junit.runner.RunWith;
```

```
@DisplayName("JUnit 5 Standard Test Class Example")  
@RunWith(JUnitPlatform.class)  
public class StandardTestClass {
```

```
    @BeforeAll  
    static void initAll() {  
        System.out.println("initAll()");  
    }
```

```
    @BeforeEach  
    void init() {  
        System.out.println("init()");  
    }
```

```
    @Test  
    void succeedingTest() {  
        System.out.println("succeedingTest()");  
    }
```

```

@Test
void failingTest() {
    System.out.println("failingTest()");

    fail("a failing test");
}

@Test
@Disabled("for demonstration purposes")
void skippedTest() {
    // not executed
}

@Test
void abortedTest() {
    System.out.println("abortedTest()");

    assertTrue("abc".contains("Z"));
    fail("test should have been aborted");
}

@AfterEach
void tearDown() {
    System.out.println("tearDown()");
}

@AfterAll
static void tearDownAll() {
    System.out.println("tearDownAll()");
}
}

```