Testing

Unit testing - test individual class, done by developer at the time of coding

             - method by method we are going to test

JUnit framework

Junit 5 = Junit jupiter + mockito junit jupiter - min with jdk8 with 3 jar files

  Junit 5 = Junit platform + Junit Jupiter + Junit Vintage

Junit platform

   - Launch testing framework

   - Define test engines

   - Console launcher

   - Integration with 3rd party

Junit Jupiter

   - core of junit 5, new programming model + extension to write test

Junit Vintage

   - To run test that are written with junit4 and junit5

Junit 4 = junit 4.13 + mockito-core 3.2.4 - min with jdk5 with sinlge jar file

Junit4                         Junit5

@Before                        @BeforeEach

@After                         @AfterEach

@BeforeClass                   @BeforeAll

@AfterClass                    @AfterAll

@Test                          @Test

@Ignore                        @Disabled

Mockito framework

    - open source framework in java to create mock object

    - Every application we design it have controller, service and dao layer. Whenever we request it first hits the controller method, then it goes to service and from service it go to dao layer and from dao it hits database

   - So at time of testing hitting db is not good practice, so we create mock object

controller-service-dao(mock) -db

Advantages

1. Saves developers from writing mock objects on their own

2. supports exception handling - we can check for negative scenario for exception

3. Annotation support

4. it can check order of method calls and no of times method called

1. Add dependency Spring-boot-starter-test - Junit, AssertJ, Hamcrest, Mockito, JsonPath, JsonAssert, Spring Test

@RunWith(SpringRunner.class) - make class as test runner

@SpringBootTest - used as alternative for @ContextConfiguration - used for creating ApplicationContext

mvn clean install - to compile and download dependency

mvn test - test in command prompt

mvn site:site - used to document entire dependency in html format. After that in target/site folder we can see dependency.html files