

```
=====
What is Junit ?
=====
```

=> Free & open source java based framework

=> It is used for unit testing

Note: It is used only for java projects unit testing

```
=====
What is Unit Testing ?
=====
```

=> Testing individual components of the application is called as Unit testing.

```
class UserService {

    m1() {
    }

    m2(){
    }

    m3(){
    }
}
```

=> Unit testing is used to identify whether code is working as expected or not.

=> Unit testing is used to identify bugs in the code.

=> With the help of unit testing we can identify bugs at early stage.

=> By performing unit testing we can provide quality code to higher environments
(dev, qa, uat and prod)

Note: Developer is responsible to perform unit testing in the project

```
=====
What is isolated unit testing ?
=====
```

=> Controller methods are depends on service methods and service methods depends on dao/repo methods.

=> When we are performing unit testing for controller method that time only controller method should be executed (service method shouldn't be called).

Ex:

```
m1() {

    //logic

    m2 ( )

    // logic

}
```

```
m2(){
    //logic
}
```

Note: When we are unit testing m1() method only m1() logic should be executed without calling real m2() method.

=> To perform isolated unit testing we will use Mocking.

```
=====
What is Mocking ?
=====
```

=> The process of creating substitute object for real-object is called as Mocking.

- 1) Mockito
- 2) Easy Mock
- c) Power Mock
- d) JMock

```
=====
What is code coverage ?
=====
```

=> The process of identifying which lines of code is unit tested and which lines of code is missed in unit testing.

=> Industry standard is minimum 80% code coverage for the project.

=> To get code coverage report we can use 'jacocco' plugin.

```
m2(boolean value){
    // logic

    if(c1){
        //10 lines
    }else if(c2){
        // 15 lines
    }else if(c3){
        // 20 lines
    }else{
        // 10 lines
    }
}
```

testM2_c1(a) ----> with c1 input

testM2_c2(a) ----> with c2 input

testM2_c3(a) ----> with c3 input

testM2_c4(a) ----> with input whic don't satisfy any condition

```
=====
SpringBoot Rest API + Junit + Mocking + Jacoco
=====
```

Git repo : https://github.com/ashokitschool/sb_rest_api_junit_app.git

@Test : To represent our method as unit test method (test case)

Note: @Test is used at method level.

@MockBean : It is used to create mock object for real bean

@InjectMocks : It is used to inject mock beans in our target

@SpringBootTest : It is used to represent our class as SpringBoot test class.

@WebMvcTest : To represent unit testing for RestController class

```
=====
Jacoco plugin configuration in pom.xml
=====
```

=> Add below plugin and execute 'mvn test goal.

Note: Code Coverage report we can find in "target/site/jacoco/index.html"

=> When our classes doesn't have business logic then we don't perform unit testing and we will exclude such classes from code coverage report.

EX: Start class, pojo classes, entities and form bindings, Config, Constants

```
-----
<plugin>
  <groupId>org.jacoco</groupId>
  <artifactId>jacoco-maven-plugin</artifactId>
  <version>0.8.11</version>
  <configuration>
    <excludes>
      <exclude>in/ashokit/model/**</exclude>
      <exclude>in/ashokit/Application**</exclude>
    </excludes>
  </configuration>
  <executions>
    <execution>
      <goals>
        <goal>prepare-agent</goal>
      </goals>
    </execution>
    <execution>
      <id>report</id>
      <phase>test</phase>
      <goals>
        <goal>report</goal>
      </goals>
    </execution>
  </executions>
</plugin>
-----
```