Types of testing:

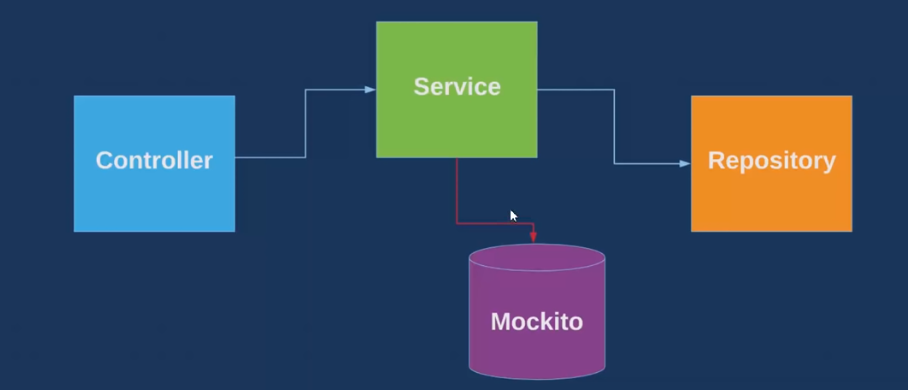
1. Client side testing
2. Server side testing

Server-side Testing:

In this case, you normally mock the requests, and you want to check how your server logic reacts.

there are two main strategies we can identify in Spring:

1. writing Controller tests using the Mockito approach, or
2. making use of RestTemplate.



Pom.xml

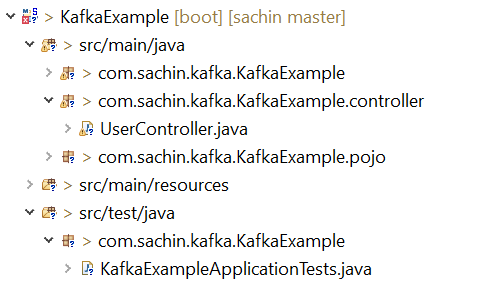
<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>



**import** org.junit.Test;

**import** org.junit.runner.RunWith;

**import** org.springframework.boot.test.context.SpringBootTest;

**import** org.springframework.test.context.junit4.SpringRunner;

@RunWith(SpringRunner.**class**)

@SpringBootTest

**public** **class** KafkaExampleApplicationTests {

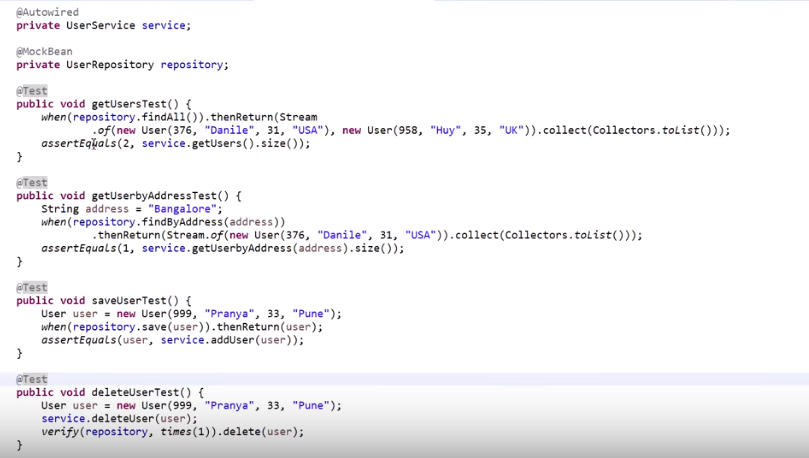
@Test

**public** **void** contextLoads() {

}

}





PoweMock API:

<https://www.youtube.com/watch?v=WBZHm4UlPPE>

Using this we can mock any of static , final or private methods, this was limitation of Mockito

Uses Reflection API

Example for Static method:

