```
==============================pom.xml==============================
<?xml version="1.0" encoding="UTF-8"?>
<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
    https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.3.0.RELEASE</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.ameya</groupId>
    <artifactId>009-SpringBootJUnitMockito</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>009-SpringBootJUnitMockito</name>
    <description>Demo project for Spring Boot</description>
    <properties>
        <java.version>1.8</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
            <!--
            <exclusions>
                <exclusion>
                    <groupId>org.junit.vintage</groupId>
                    <artifactId>junit-vintage-engine</artifactId>
                </exclusion>
            </exclusions>
             -->
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>
==============================MyDAO.java==============================
package com.ameya.daos;

import org.springframework.stereotype.Repository;

@Repository
public class MyDAO {

    public int[] retrieveAllData() {
        return new int[] {1,2,3,4,5};
    }
}
==============================MyService.java==============================
package com.ameya.services;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.ameya.daos.MyDAO;

@Service
```

```
public class MyService {
    @Autowired
    private MyDAO dataService;

    public int findTheGreatest() {
        int[] data=dataService.retrieveAllData();
        int greatest=Integer.MIN_VALUE;
        for(int value : data) {
            if(value > greatest) {
                greatest=value;
            }
        }
        return greatest;
    }
}
```

=======================MyServiceMockSpringContextTest.java=========================

```
package com.ameya.unittests;

import static org.junit.Assert.assertEquals;
import static org.mockito.Mockito.when;

import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.boot.test.mock.mockito.MockBean;
import org.springframework.test.context.junit4.SpringRunner;

import com.ameya.daos.MyDAO;
import com.ameya.services.MyService;

@RunWith(SpringRunner.class)
@SpringBootTest
public class MyServiceMockSpringContextTest {

    @MockBean
    MyDAO dataServiceMock;
    @Autowired
    MyService businessImpl;

    @Test
    public void testFindTheGreatestFromAllData() {
        when(dataServiceMock.retrieveAllData()).thenReturn(new int[] {24,15,3});
        assertEquals(24,businessImpl.findTheGreatest());
    }
    @Test
    public void testFindTheGreatestFromAllDataForOneValue() {
        when(dataServiceMock.retrieveAllData()).thenReturn(new int[] {15});
        assertEquals(15,businessImpl.findTheGreatest());
    }
    @Test
    public void testFindTheGreatestFromAllDataForNoValues() {
        when(dataServiceMock.retrieveAllData()).thenReturn(new int[] {});
        assertEquals(Integer.MIN_VALUE,businessImpl.findTheGreatest());
    }
}
```

==============================MyServiceMockTest.java=========================

```
package com.ameya.unittests;

import static org.junit.Assert.assertEquals;
import static org.mockito.Mockito.when;

import org.junit.Test;
import org.junit.runner.RunWith;
import org.mockito.InjectMocks;
import org.mockito.Mock;
import org.mockito.junit.MockitoJUnitRunner;

import com.ameya.daos.MyDAO;
import com.ameya.services.MyService;

@RunWith(MockitoJUnitRunner.class)
public class MyServiceMockTest {
```

```
145
146         @Mock
147         MyDAO dataServiceMock;
148         @InjectMocks
149         MyService businessImpl;
150         @Test
151         public void testFindTheGreatestFromAllData() {
152             when(dataServiceMock.retrieveAllData()).thenReturn(new int[] {24,15,3});
153             assertEquals(24,businessImpl.findTheGreatest());
154         }
155         @Test
156         public void testFindTheGreatestFromAllDataForOneValue() {
157             when(dataServiceMock.retrieveAllData()).thenReturn(new int[] {15});
158             assertEquals(15,businessImpl.findTheGreatest());
159         }
160         @Test
161         public void testFindTheGreatestFromAllDataForNoValues() {
162             when(dataServiceMock.retrieveAllData()).thenReturn(new int[] {});
163             assertEquals(Integer.MIN_VALUE,businessImpl.findTheGreatest());
164         }
165     }
166
```