# Spring-boot 1.5.9.RELEASE

Mockmvc use

org.junit.Test;

org.junit.runner.Runner;

## Pom.xml

<**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-test</**artifactId**>  
 <**scope**>test</**scope**>  
 <**exclusions**>  
 <**exclusion**>  
 <**groupId**>junit</**groupId**>  
 <**artifactId**>junit</**artifactId**>  
 </**exclusion**>  
 </**exclusions**>  
</**dependency**>  
<**dependency**>  
 <**groupId**>org.junit.jupiter</**groupId**>  
 <**artifactId**>junit-jupiter</**artifactId**>  
 <**version**>5.4.2</**version**>  
 <**scope**>test</**scope**>  
</**dependency**>  
<**dependency**>  
 <**groupId**>org.junit.platform</**groupId**>  
 <**artifactId**>junit-platform-commons</**artifactId**>  
 <**version**>1.5.2</**version**>  
 <**scope**>test</**scope**>  
</**dependency**>  
<**dependency**>  
 <**groupId**>org.junit.platform</**groupId**>  
 <**artifactId**>junit-platform-runner</**artifactId**>  
 <**version**>1.5.2</**version**>  
 <**scope**>test</**scope**>  
</**dependency**>

## Class

**package** com.znv.cop.controller;  
  
**import** com.znv.cop.service.impl.DeviceMeteService;  
**import** com.znv.cop.service.impl.MeteCollectionService;  
**import** com.znv.cop.service.impl.MeteTemplateService;  
**import** com.znv.cop.vo.entity.MeteEntity;  
**import** org.hamcrest.Matchers;  
**import** org.junit.Test;  
**import** org.junit.jupiter.api.BeforeAll;  
**import** org.junit.runner.RunWith;  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;  
**import** org.springframework.boot.test.autoconfigure.web.servlet.WebMvcTest;  
**import** org.springframework.boot.test.mock.mockito.MockBean;  
**import** org.springframework.http.MediaType;  
**import** org.springframework.test.context.ContextConfiguration;  
**import** org.springframework.test.context.junit4.SpringRunner;  
**import** org.springframework.test.web.servlet.MockMvc;  
  
**import** java.util.ArrayList;  
**import** java.util.Arrays;  
**import** java.util.List;  
  
**import static** org.mockito.Matchers.*any*;  
**import static** org.mockito.Mockito.*when*;  
**import static** org.springframework.test.web.servlet.request.MockMvcRequestBuilders.*get*;  
**import static** org.springframework.test.web.servlet.result.MockMvcResultMatchers.*jsonPath*;  
  
@RunWith(SpringRunner.**class**)  
@AutoConfigureMockMvc  
@ContextConfiguration(classes = {  
 ApiMeteController.**class**,  
 MeteTemplateService.**class**,  
 MeteCollectionService.**class**,  
 DeviceMeteService.**class**})  
@WebMvcTest  
**public class** ApiMeteControllerTest {  
  
 **private** Integer **id**;  
 **private** List<MeteEntity> **entities**;  
  
 @BeforeAll  
 **public void** setUp() {  
 **id** = 50;  
 MeteEntity entity = **new** MeteEntity();  
 entity.setId(**id**);  
 entity.setTmplId(60);  
 entity.setMeteId(**"0481305001"**);  
 entity.setMeteKind(3);  
 entity.setMeteName(**"自动手动开关机控制"**);  
 entity.setPort(**"4"**);  
 **entities** = Arrays.*asList*(entity);  
 }  
  
 @Autowired  
 **private** MockMvc **mockMvc**;  
  
 @MockBean  
 **private** MeteTemplateService **meteTemplateService**;  
  
 @MockBean  
 MeteCollectionService **meteCollectionService**;  
  
 @MockBean  
 DeviceMeteService **deviceMeteService**;  
  
 @Test  
 **public void** isOK() **throws** Exception {  
 *when*(**meteTemplateService**.tmplMeteList(*any*())).thenReturn(**entities**);  
 **mockMvc**.perform(*get*(**"/mete/template/detail?id={}"**, **id**)  
 .contentType(MediaType.***APPLICATION\_JSON***)  
 .accept(MediaType.***APPLICATION\_JSON***)  
 ).andExpect(*jsonPath*(**"$.data[0].meteName"**, Matchers.*is*(**"自动手动开关机控制"**)));  
 }  
  
}

# Spring-boot 2.1.6.RELEASE

## Pom.xml

<**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-test</**artifactId**>  
 <**exclusions**>  
 <**exclusion**>  
 <**groupId**>junit</**groupId**>  
 <**artifactId**>junit</**artifactId**>  
 </**exclusion**>  
 </**exclusions**>  
 <**scope**>test</**scope**>  
</**dependency**>  
<**dependency**>  
 <**groupId**>org.junit.jupiter</**groupId**>  
 <**artifactId**>junit-jupiter</**artifactId**>  
 <**version**>5.4.2</**version**>  
 <**scope**>test</**scope**>  
</**dependency**>

## Class

**package** com.znv.cop.controller;  
  
**import** com.znv.cop.inf.rsp.MeteCollectionListRspDto;  
**import** com.znv.cop.service.impl.DeviceMeteService;  
**import** com.znv.cop.service.impl.MeteCollectionService;  
**import** com.znv.cop.service.impl.MeteTemplateService;  
**import** com.znv.cop.vo.entity.MeteEntity;  
**import** org.hamcrest.Matchers;  
**import** org.junit.Before;  
**import** org.junit.Test;  
**import** org.junit.jupiter.api.BeforeAll;  
**import** org.junit.runner.RunWith;  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;  
**import** org.springframework.boot.test.autoconfigure.web.servlet.WebMvcTest;  
**import** org.springframework.boot.test.mock.mockito.MockBean;  
**import** org.springframework.http.MediaType;  
**import** org.springframework.test.context.ContextConfiguration;  
**import** org.springframework.test.context.junit4.SpringRunner;  
**import** org.springframework.test.web.servlet.MockMvc;  
  
**import** java.util.ArrayList;  
**import** java.util.Arrays;  
**import** java.util.List;  
  
**import static** org.mockito.Matchers.*any*;  
**import static** org.mockito.Mockito.*when*;  
**import static** org.springframework.test.web.servlet.request.MockMvcRequestBuilders.*get*;  
**import static** org.springframework.test.web.servlet.result.MockMvcResultMatchers.*jsonPath*;  
**import static** org.springframework.test.web.servlet.result.MockMvcResultMatchers.*status*;  
  
@RunWith(SpringRunner.**class**)  
@AutoConfigureMockMvc  
@ContextConfiguration(classes = {  
 ApiMeteController.**class**,  
 MeteTemplateService.**class**,  
 MeteCollectionService.**class**,  
 DeviceMeteService.**class**})  
@WebMvcTest  
**public class** ApiMeteControllerTest {  
  
 **private** Integer **id**;  
 MeteCollectionListRspDto **meteCollectionListRspDto**;  
  
 @Before  
 **public void** setUp() {  
 **id** = 50;  
 MeteEntity entity = **new** MeteEntity();  
 entity.setId(**id**);  
 entity.setTmplId(60);  
 entity.setMeteId(**"0481305001"**);  
 entity.setMeteKind(3);  
 entity.setMeteName(**"自动手动开关机控制"**);  
 entity.setPort(**"4"**);  
 List<MeteEntity> entities = Arrays.*asList*(entity);  
 **meteCollectionListRspDto** = **new** MeteCollectionListRspDto();  
 **meteCollectionListRspDto**.setStaffId(**"00496"**);  
 **meteCollectionListRspDto**.setStaffName(**"king"**);  
 **meteCollectionListRspDto**.setYt(entities);  
 }  
  
 @Autowired  
 **private** MockMvc **mockMvc**;  
  
 @MockBean  
 **private** MeteTemplateService **meteTemplateService**;  
  
 @MockBean  
 MeteCollectionService **meteCollectionService**;  
  
 @MockBean  
 DeviceMeteService **deviceMeteService**;  
  
 @Test  
 **public void** isOK() **throws** Exception {  
 *when*(**meteTemplateService**.tmplMeteTree(**id**)).thenReturn(**meteCollectionListRspDto**);  
 **mockMvc**.perform(*get*(**"/mete/template/detail?id=50"**)  
 .contentType(MediaType.***APPLICATION\_JSON***)  
 .content(**"\"id\":\"50\""**)  
 .accept(MediaType.***APPLICATION\_JSON***)  
 )  
 .andExpect(*status*().isOk())  
 .andExpect(*jsonPath*(**"$.data.yt[0].meteName"**, Matchers.*is*(**"自动手动开关机控制"**)));  
 }  
  
}

class DeviceImportServiceTest {  
  
 @InjectMocks  
 DeviceImportService importService;  
  
 @Mock  
 DeviceDao deviceDao;  
  
 @Mock  
 GeographyCodeDao geographyCodeDao;  
  
 @BeforeEach  
 void setUp() {  
 MockitoAnnotations.*initMocks*(this);  
 }  
  
 @Test  
 void batchAdd() {  
 String batchAddMethodName = "batchAdd";  
 Method batchAddMethod = ReflectionSupport  
 .*findMethod*(DeviceImportService.class, batchAddMethodName, List.class, String.class)  
 .orElseThrow(() -> {  
 return new BusinessException(new ErrorCode(-1, "method not found"));  
 });  
 *assertEquals*(1,  
 ((BatchImportRspDto) (ReflectionSupport.*invokeMethod*(batchAddMethod, importService, new ArrayList<>(Arrays.*asList*(new HashMap())), "airmedia"))).getTotal());  
 }

}

<https://www.bytestree.com/spring/spring-boot-2-junit-5-rest-api-unit-testing/>

For JUnit 4 adding the dependency for spring-boot-starter-test was enough. But for Junit 5 you need to add following dependencies:

* spring-boot-starter-test
* junit-jupiter-api
* junit-jupiter-engine
* maven-surefire-plugin