



链滴

# 如何使用 JUnit 自动化测试 mvc 接口 (model)

作者: [TWanGT](#)

原文链接: <https://ld246.com/article/1570852078686>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 如何使用 JUnit 自动化测试mvc接口(model)

一般正常的 `restful` 接口我们可以很方便的写出测试用例, 但是遇到有 `Model` 的 `Controller` 方法就比难过了, 我之前找了很多方法(方向是如何构造Model), 但是都没找到解决方案, 知道我偶然发现 `Mockvc` 这个东西, 于是就有了解决方案, 本文就讲解如何使用 `Junit` 进行 `mvc` 接口的测试

JUnit: 个回归测试框架, 方便我们对代码进行测试, 特别是在重构(修改)后验证代码的功能逻辑是否正确, 被开发者用于实施对应用程序的单元测试, 加快程序编制速度, 同时提高编码的质量

如果还不知道什么是 `JUnit` 请移步

## 简单讲解

难点就在与这个 `Model` 对象的构造, 无法使用 `Spring` 的自动注入来初始化 `Model`, 使用 `new` 也不实(这个是个接口), 所以对于这种 `controller` 方法要么就只测试内部的 `service` 逻辑, 要么就改写一下成无 `Model` 参数的(破坏代码结构, 不优雅), 现在有了 `MockMvc` 就能自动帮我们解决这个 `Model` 的题, 而且还能将生成的视图输出到 `console`

我们要做的就是单元测试调用 `controller` 方法钱将 `MockMvc` 构造出来, 然后利用 `MockMvc` 内部方法进行 `controller` 方法的调用

## 关键源码

### 引入部分

```
import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.get;
import static org.springframework.test.web.servlet.result.MockMvcResultHandlers.print;
import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.status;
import static org.springframework.test.web.servlet.setup.MockMvcBuilders.webAppContextSetup;
```

### 代码部分

- `@Before`: 测试前需要自动执行的注解(方法上)
- `MockMvc`: 我们需要用到的模拟 `MVC` 对象

```
private MockMvc mockMvc;

@Before
public void setup() {
    // webAppContextSetup 注意上面的static import
```

```
// webApplicationContextSetup 构造的WEB容器可以添加fileter 但是不能添加listenCLASS
// WebApplicationContext context =
// ContextLoader.getCurrentWebApplicationContext();
// 如果控制器包含如上方法 则会报空指针
this.mockMvc = webApplicationContextSetup(this.wac).build();
}
```

## 接口调用写法

```
mockMvc.perform((get("/要测试的接口地址"))).andExpect(status().isOk()).andDo(print());
```