



链滴

# Spring Boot 构建 RESTful API

作者: [Javen](#)

原文链接: <https://ld246.com/article/1494807793727>

来源网站: [链滴](#)

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

首先，回顾并详细说明一下在快速入门中使用的 @Controller、@RestController、@RequestMapping 注解。如果您对 Spring MVC 不熟悉并且还没有尝试过快速入门案例，建议先看一下快速入门的内容。

@Controller: 修饰 class, 用来创建处理 http 请求的对象

@RestController: Spring4 之后加入的注解, 原来在 @Controller 中返回 json 需要 @ResponseBody 来配合, 如果直接用 @RestController 替代 @Controller 就不需要再配置 @ResponseBody, 默认返回 json 格式。

@RequestMapping: 配置 url 映射

下面我们尝试使用 Spring MVC 来实现一组对 User 对象操作的 RESTful API, 配合注释详细说明在 Spring MVC 中如何映射 HTTP 请求、如何传参、如何编写单元测试。

RESTful API 具体设计如下:

User 实体定义:

```
public class User {
    private Long id;
    private String name;
    private Integer age;
    // 省略setter
    // 省略getter
}
```

实现对 User 对象的操作接口

```
@RestController
@RequestMapping(value="/users") // 通过这里配置使下面的映射都在/users下
public class UserController {
    // 创建线程安全的Map
    static Map<Long, User> users = Collections.synchronizedMap(new HashMap<Long, User>());
    @RequestMapping(value="/", method=RequestMethod.GET)
    public List<User> getUserList() {
        // 处理"/users/"的GET请求, 用来获取用户列表
        // 还可以通过@RequestParam从页面中传递参数来进行查询条件或者翻页信息的传递
        List<User>
```

```

    &gt; r = new ArrayList<User>(users.values());
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">        return r;
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">    }
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">
    </span> </span> <span class="highlight-line"> <span class="highlight-cl"> @RequestMapping
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">     public String
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         // 处理"/us
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         // 除了@
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         users.put(
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         ser.getId(), user);
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         return "su
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         cess";
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">     }
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">
    </span> </span> <span class="highlight-line"> <span class="highlight-cl"> @RequestMapping
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">     public User g
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         // 处理"/us
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         // url中的i
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         return user
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         .get(id);
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">     }
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">
    </span> </span> <span class="highlight-line"> <span class="highlight-cl"> @RequestMapping
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">     public String
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         // 处理"/us
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         User u = u
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         u.setName
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         u.setAge(u
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         er.getAge());
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         users.put(i
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         , u);
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         return "su
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         cess";
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">     }
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">
    </span> </span> <span class="highlight-line"> <span class="highlight-cl"> @RequestMapping
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">     public String
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         // 处理"/us
    </span> </span> <span class="highlight-line"> <span class="highlight-cl">         rs/{id}"的DELETE请求，用来删除User

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">        users.rem
ve(id);
</span></span><span class="highlight-line"><span class="highlight-cl">        return "su
cess";
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code></pre>
<p>下面针对该 Controller 编写测试用例验证正确性，具体如下。当然也可以通过浏览器插件等进
请求提交验证。</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> @RunWith(Spri
gJUnit4ClassRunner.class)
</span></span><span class="highlight-line"><span class="highlight-cl"> @SpringApplica
tionConfiguration(classes = MockServletContext.class)
</span></span><span class="highlight-line"><span class="highlight-cl"> @WebAppConf
guration
</span></span><span class="highlight-line"><span class="highlight-cl"> public class App
icationTests {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> private Moc
Mvc mvc;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> @Before
</span></span><span class="highlight-line"><span class="highlight-cl"> public void s
tUp() throws Exception {
</span></span><span class="highlight-line"><span class="highlight-cl">     mvc = Mo
kMvcBuilders.standaloneSetup(new UserController()).build();
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> @Test
</span></span><span class="highlight-line"><span class="highlight-cl"> public void t
stUserController() throws Exception {
</span></span><span class="highlight-line"><span class="highlight-cl">     // 测试Use
Controller
</span></span><span class="highlight-line"><span class="highlight-cl">     RequestBui
lder request = null;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> // 1、 get
一下user列表，应该为空
</span></span><span class="highlight-line"><span class="highlight-cl">     request =
get("/users/");
</span></span><span class="highlight-line"><span class="highlight-cl">     mvc.perfo
rm(request)
</span></span><span class="highlight-line"><span class="highlight-cl">         .andE
spect(status().isOk())
</span></span><span class="highlight-line"><span class="highlight-cl">         .andE
spect(content().string(equalTo("[]")));
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> // 2、 pos
提交一个user
</span></span><span class="highlight-line"><span class="highlight-cl">     request =
post("/users/")

```

```

</span></span><span class="highlight-line"><span class="highlight-cl"> .para
("id", "1")
</span></span><span class="highlight-line"><span class="highlight-cl"> .para
("name", "测试大师")
</span></span><span class="highlight-line"><span class="highlight-cl"> .para
("age", "20");
</span></span><span class="highlight-line"><span class="highlight-cl"> mvc.perfo
m(request)
</span></span><span class="highlight-line"><span class="highlight-cl"> .andE
pect(content().string(equalTo("success")));
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> // 3、 get
取user列表， 应该有刚才插入的数据
</span></span><span class="highlight-line"><span class="highlight-cl"> request =
et("/users/");
</span></span><span class="highlight-line"><span class="highlight-cl"> mvc.perfo
m(request)
</span></span><span class="highlight-line"><span class="highlight-cl"> .andE
pect(status().isOk())
</span></span><span class="highlight-line"><span class="highlight-cl"> .andE
pect(content().string(equalTo("[{"id":1,"name":"测试大师","age":20}]]));
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> // 4、 put
改id为1的user
</span></span><span class="highlight-line"><span class="highlight-cl"> request =
ut("/users/1")
</span></span><span class="highlight-line"><span class="highlight-cl"> .para
("name", "测试终极大师")
</span></span><span class="highlight-line"><span class="highlight-cl"> .para
("age", "30");
</span></span><span class="highlight-line"><span class="highlight-cl"> mvc.perfo
m(request)
</span></span><span class="highlight-line"><span class="highlight-cl"> .andE
pect(content().string(equalTo("success")));
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> // 5、 get
个id为1的user
</span></span><span class="highlight-line"><span class="highlight-cl"> request =
et("/users/1");
</span></span><span class="highlight-line"><span class="highlight-cl"> mvc.perfo
m(request)
</span></span><span class="highlight-line"><span class="highlight-cl"> .andE
pect(content().string(equalTo("[{"id":1,"name":"测试终极大师","age":30}]]));
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> // 6、 del
除id为1的user
</span></span><span class="highlight-line"><span class="highlight-cl"> request =
elete("/users/1");
</span></span><span class="highlight-line"><span class="highlight-cl"> mvc.perfo
m(request)
</span></span><span class="highlight-line"><span class="highlight-cl"> .andE
pect(content().string(equalTo("success")));
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> // 7、 get

```

一下user列表, 应该为空

```
</span></span><span class="highlight-line"><span class="highlight-cl"> request =
et("/users/");
</span></span><span class="highlight-line"><span class="highlight-cl"> mvc.perfo
m(request)
</span></span><span class="highlight-line"><span class="highlight-cl"> .andE
pect(status().isOk())
</span></span><span class="highlight-line"><span class="highlight-cl"> .andE
pect(content().string(equalTo("[]")));
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span></code></pre>
```

至此, 我们通过引入 web 模块 (没有做其他的任何配置), 就可以轻松利用 Spring MVC 的功, 以非常简洁的代码完成了对 User 对象的 RESTful API 的创建以及单元测试的编写。其中同时介绍了 Spring MVC 中最为常用的几个核心注解: @Controller,@RestController,RequestMapping 以及一些参数绑定的注解: @PathVariable,@ModelAttribute,@RequestParam 等。