

1. @PathVariable: 映射url片段到java方法的参数

2. 在url声明中使用正则表达式

```
1 @RestController
2 public class UserController {
3
4     // @PathVariable把url中的变量值传递到java方法中的参数里来
5     // 针对id写一个正则表达式 :\\d+ 表示只能接收数字
6     @RequestMapping(value = "/user/{id:\\d+}", method = RequestMethod.GET)
7     public User getInfo(@PathVariable(name = "id") String id) {
8         User user = new User();
9         user.setUsername("tom");
10        return user;
11    }
12 }
```

```
1 @RunWith(SpringRunner.class)
2 @SpringBootTest
3 public class UserControllerTest {
4
5     @Autowired
6     private WebApplicationContext wac;
7
8     private MockMvc mockMvc;
9
10    @Before
11    public void setup() {
12        mockMvc = MockMvcBuilders.webAppContextSetup(wac).build();
13    }
14
15    @Test
16    public void whenGetInfoSuccess() throws Exception {
17        mockMvc.perform(get("/user/1")
18            .contentType(MediaType.APPLICATION_JSON_UTF8))
19            .andExpect(status().isOk())
20            .andExpect(jsonPath("$.username").value("tom"));
21    }
22
23    @Test
24    public void whenGetInfoFail() {
25        try {
```

```

26 mockMvc.perform(get("/user/a")
27 .contentType(MediaType.APPLICATION_JSON_UTF8))
28 .andExpect(status().is4xxClientError());
29 } catch (Exception e) {
30 // TODO Auto-generated catch block
31 e.printStackTrace();
32 }
33 }
34 }

```

3. @JsonView控制json输出内容

@JsonView使用步骤

1. 使用接口来声明多个视图
2. 在值对象的get方法上指定视图

```

1 public class User {
2
3 public interface UserSimpleView {};
4 public interface UserDetailsView extends UserSimpleView{};
5
6 private String username;
7 private String password;
8
9 @JsonView(UserSimpleView.class)
10 public String getUsername() {
11 return username;
12 }
13 public void setUsername(String username) {
14 this.username = username;
15 }
16 @JsonView(UserDetailsView.class)
17 public String getPassword() {
18 return password;
19 }
20 public void setPassword(String password) {
21 this.password = password;
22 }
23 }

```

3. 在Controller方法上指定视图

```
1 @RestController
2 public class UserController {
3
4     @RequestMapping(value = "/user", method = RequestMethod.GET)
5     @JsonView(User.UserSimpleView.class)
6     public List<User> query(UserQueryCondition userQueryCondition, @PageableDefault(page = 2, size = 17, sort = "username,asc") Pageable pageable){
7
8         System.out.println(ReflectionToStringBuilder.toString(userQueryCondition, ToStringStyle.MULTI_LINE_STYLE));
9
10        System.out.println(pageable.getPageSize());
11        System.out.println(pageable.getPageNumber());
12        System.out.println(pageable.getSort());
13        List<User> users = new ArrayList<User>();
14        users.add(new User());
15        users.add(new User());
16        users.add(new User());
17        return users;
18    }
19
20    // @PathVariable把url中的变量值传递到java方法中的参数里来
21    // 针对id写一个正则表达式 :\\d+ 表示只能接收数字
22    @RequestMapping(value = "/user/{id:\\d+}", method = RequestMethod.GET)
23    @JsonView(User.UserDetailView.class)
24    public User getInfo(@PathVariable(name = "id") String id) {
25        User user = new User();
26        user.setUsername("tom");
27        return user;
28    }
29 }
```