

编写第一个Restful API

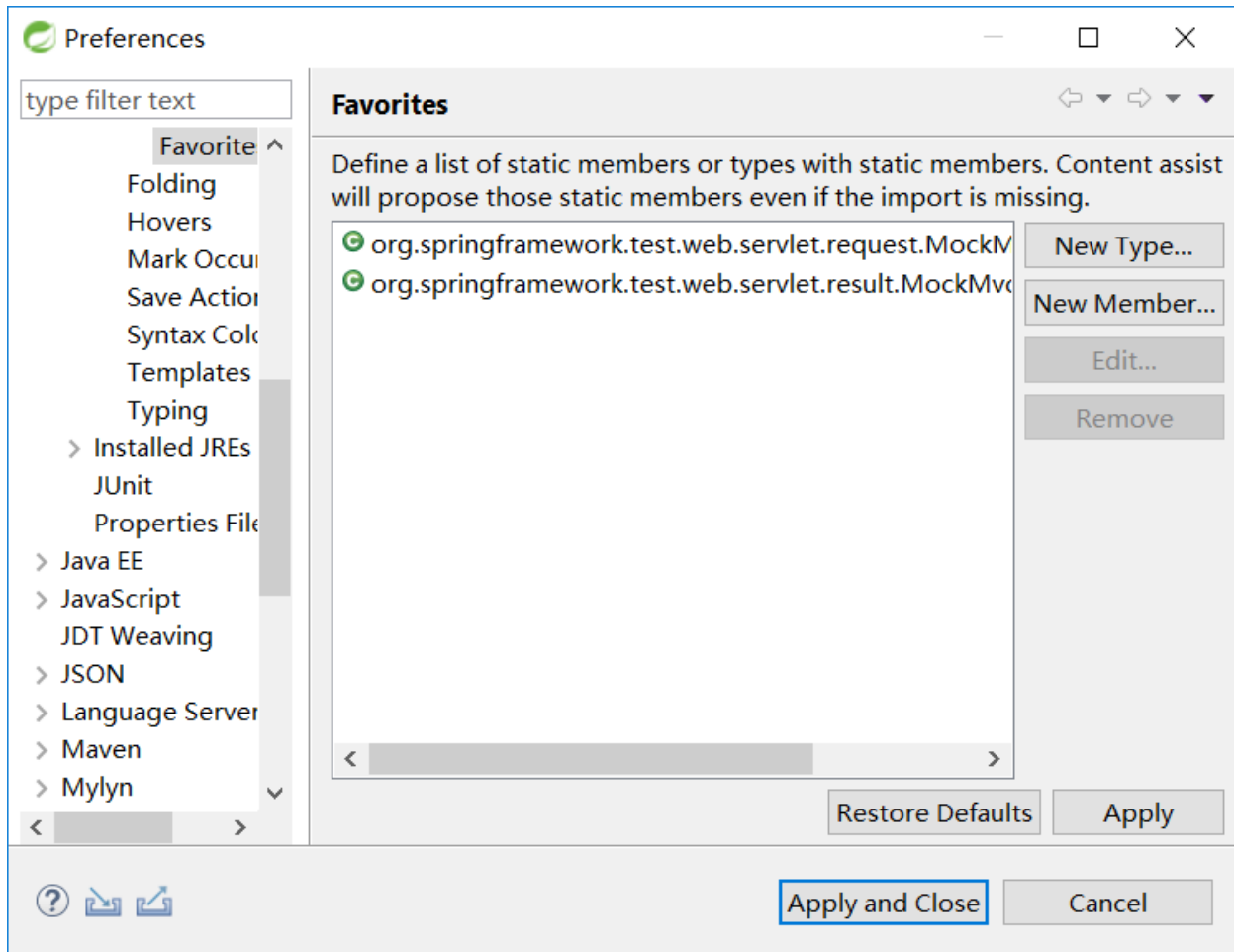
编写针对Restful API的测试用例

使用注解声明RestfulAPI

在RestfulAPI中传递参数

```
1  /**
2   *
3   */
4  package com.xiaoming.web.controller;
5
6  import static org.springframework.test.web.servlet.request.MockMvcRequest
    Builders.get;
7  import static org.springframework.test.web.servlet.result.MockMvcResultMa
    tchers.jsonPath;
8  import static org.springframework.test.web.servlet.result.MockMvcResultMa
    tchers.status;
9
10 import org.junit.Before;
11 import org.junit.Test;
12 import org.junit.runner.RunWith;
13 import org.springframework.beans.factory.annotation.Autowired;
14 import org.springframework.boot.test.context.SpringBootTest;
15 import org.springframework.http.MediaType;
16 import org.springframework.test.context.junit4.SpringRunner;
17 import org.springframework.test.web.servlet.MockMvc;
18 import org.springframework.test.web.servlet.request.MockMvcRequestBuilde
    rs;
19 import
    org.springframework.test.web.servlet.result.MockMvcResultMatchers;
20 import org.springframework.test.web.servlet.setup.MockMvcBuilders;
21 import org.springframework.web.context.WebApplicationContext;
22
23 /**
24  * @author 江小明
25  *
26  */
27 @RunWith(SpringRunner.class)
28 @SpringBootTest
29 public class UserControllerTest {
30
31     @Autowired
32     private WebApplicationContext wac;
```

```
33
34 private MockMvc mockMvc;
35
36 @Before
37 public void setup() {
38     mockMvc = MockMvcBuilders.webApplicationContextSetup(wac).build();
39 }
40
41 @Test
42 public void whenQuerySuccess() throws Exception {
43     //perform执行
44     mockMvc.perform(get("/user")//伪造请求
45         .param("username", "xiaoming")
46         .param("age", "18")
47         .param("xxx", "yyy")
48         .param("ageTo", "15")
49         // .param("page", "3") //当前页
50         // .param("size", "15") //页大小
51         // .param("sort", "age,desc") //排序
52         .contentType(MediaType.APPLICATION_JSON_UTF8))//参数格式为UTF8
53         .andExpect(status().isOk()) //期望 状态是OK
54         .andExpect(jsonPath("$.length()").value(3));//返回的json内容 是个集合长度
    为3
55 }
56 }
57
```



配置爱好，这样就不用每次都去写这个类，可以直接调用这个类的方法

常用注解

@RestController: 标明此Controller提供RestAPI

@RequestMapping及其变体: 映射http请求url到java方法

@RequestParam: 映射请求参数到java方法的参数

@PageableDefault: 指定分页参数默认值

UserController类

```
1 package com.xiaoming.web.controller;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 import org.apache.commons.lang.builder.ReflectionToStringBuilder;
7 import org.apache.commons.lang.builder.ToStringStyle;
8 import org.springframework.data.domain.Pageable;
9 import org.springframework.data.web.PageableDefault;
10 import org.springframework.web.bind.annotation.RequestMapping;
```

```

11 import org.springframework.web.bind.annotation.RequestMethod;
12 import org.springframework.web.bind.annotation.RequestParam;
13 import org.springframework.web.bind.annotation.RestController;
14
15 import com.xiaoming.dto.User;
16 import com.xiaoming.dto.UserQueryCondition;
17
18 /**
19  * @author 江小明
20  *
21  */
22 @RestController
23 public class UserController {
24
25     @RequestMapping(value = "/user", method = RequestMethod.GET)
26     public List<User> query(UserQueryCondition userQueryCondition, @Pageable
Default(page = 2, size = 17, sort = "username,asc") Pageable pageable){
27         System.out.println(ReflectionToStringBuilder.toString(userQueryConditio
n, ToStringStyle.MULTI_LINE_STYLE));
28         System.out.println(pageable.getPageSize());
29         System.out.println(pageable.getPageNumber());
30         System.out.println(pageable.getSort());
31         List<User> users = new ArrayList<User>();
32         users.add(new User());
33         users.add(new User());
34         users.add(new User());
35         return users;
36     }
37 }

```

UserQueryCondition类

```

1 package com.xiaoming.dto;
2
3 public class UserQueryCondition {
4
5     private String username;
6     private int age;
7     private int ageTo;
8     private String xxx;
9     public String getUsername() {
10         return username;

```

```

11  }
12  public void setUsername(String username) {
13      this.username = username;
14  }
15  public int getAge() {
16      return age;
17  }
18  public void setAge(int age) {
19      this.age = age;
20  }
21  public int getAgeTo() {
22      return ageTo;
23  }
24  public void setAgeTo(int ageTo) {
25      this.ageTo = ageTo;
26  }
27  public String getXxx() {
28      return xxx;
29  }
30  public void setXxx(String xxx) {
31      this.xxx = xxx;
32  }
33
34
35  }

```

User类

```

1  /**
2   *
3   */
4  package com.xiaoming.dto;
5
6  /**
7   * @author 江小明
8   *
9   */
10 public class User {
11
12
13     private String username;

```

```
14  private String password;
15  public String getUsername() {
16  return username;
17  }
18  public void setUsername(String username) {
19  this.username = username;
20  }
21  public String getPassword() {
22  return password;
23  }
24  public void setPassword(String password) {
25  this.password = password;
26  }
27
28
29 }
```