

接口测试自动化基础

HtmlUnit

HtmlUnit

HtmlUnit相比HttpClient功能更加强大，就像一个浏览器，HtmlUnit是Junit的扩展测试框架之一，该框架模拟浏览器的行为，开发者可以使用其提供的API对页面的元素进行操作。HtmlUnit支持HTTP，HTTPS，COOKIE，表单的POST和GET方法，能够对HTML文档进行包装，页面的各种元素都可以被当作对象进行调用，对JavaScript的支持也比较好。

HtmlUnit

```
// 定义一个WebClient
WebClient webClient = new WebClient(BrowserVersion.FIREFOX_45);
// 从指定URL获取HtmlPage
HtmlPage page = webClient.getPage("https://www.baidu.com/");
// 将HtmlPage转换成字符串打印出来
System.out.println(page.asText());
// 关闭客户端
webClient.close();
```

GET请求

```
//1.创建WebClient对象
WebClient client = new WebClient();
//2.创建WebRequest对象
WebRequest request = new WebRequest(new URL(url),HttpMethod.GET);
//3.设置header属性
client.addRequestHeader("Content-Type", "application/json");
//4.执行请求
Page p =client.getPage(request);
//5.获得响应
WebResponse response = p.getWebResponse();
//6.获得响应正文
System.out.println(response.getStatusCode());
System.out.println(response.getContentAsString());
//7.断开连接
client.close();
```

POST请求

```
@Test
public void login() throws FailingHttpStatusCodeException, IOException{
//    1.创建webClient对象
    WebClient client =new WebClient();
//    2.创建请求WebRequest
    WebRequest request = new WebRequest(new URL(url),
        HttpMethod.POST);
//    3.设置Header
    client.addRequestHeader("Content-Type","application/json");
//    4.设置请求体
    request.setRequestBody("{\"phoneArea\":\"86\",\"phoneNumber\":\"20000000000\",\"pass
//    5.执行请求
    Page p =client.getPage(request);
//    6.获得响应
    WebResponse response=p.getWebResponse();
//    7.获得响应正文
    String result=response.getContentAsString();
    System.out.println(result);
//    8.断开连接
    client.close();

}
```

思考

- 如果请求的参数是JSON，怎么处理？
- 封装get方法为doGet(String url,Map para)
- 封装post方法为doPost(String url,JSNOObject para)

Cookie的管理

- 获得Cookie

```
Set<Cookie> cookies =  
client1.getCookieManager().getCookies();
```

- 使用Cookie

```
Iterator<Cookie> i =cookies .iterator();  
    while (i.hasNext())  
    {  
        client2.getCookieManager().addCookie(i.next());  
    }
```