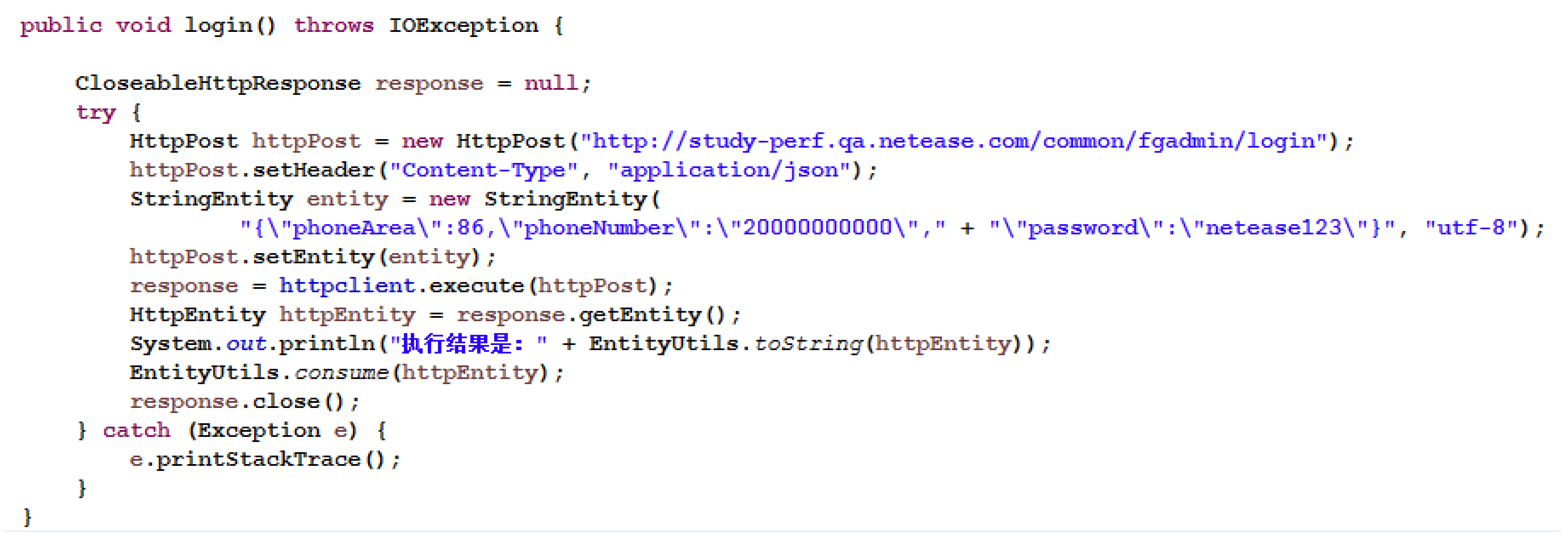
接口测试自动化

面向场景的接口测试

用例回顾

场景2：完整下单流程(未登录、有收货地址)	1、登录成功	POST /common/fgadmin/login	Content-Type=application/json	phoneArea="86" phoneNumber="20000000000" password="netease123"	code : 200 message : "success"
	2、查询收货地址	GET /fgadmin/address/list	Content-Type=application/json	无	code : 200 message : “success” result : “list”:[{“id”:“1”,“receiverName”:“张三”,“cellPhone”:“20000000000”,“addressDetail”:“河北师范大学","province":"河北省","city":"石家庄市","area":"裕华区"}]
	3、计算运费	GET /common/getTransportFee	Content-Type=application/json	id=1 addressDetail="河北省_石家庄市_裕华区"	code : 200 message : "success" result : 6.0
	4、提交订单	POST /fgadmin/orders/submit	Content-Type=application/json	skuIds=“1” receiverName=“张三” cellPhone=“20000000000” addressDetail=“河北师范大学" province="河北省" city="石家庄市" area="裕华区" transportFee=6.0	code : 200 message : "success" result : {"id":"1","totalFee":"205.0","createTime":"2016-11-11 20:00:00"}

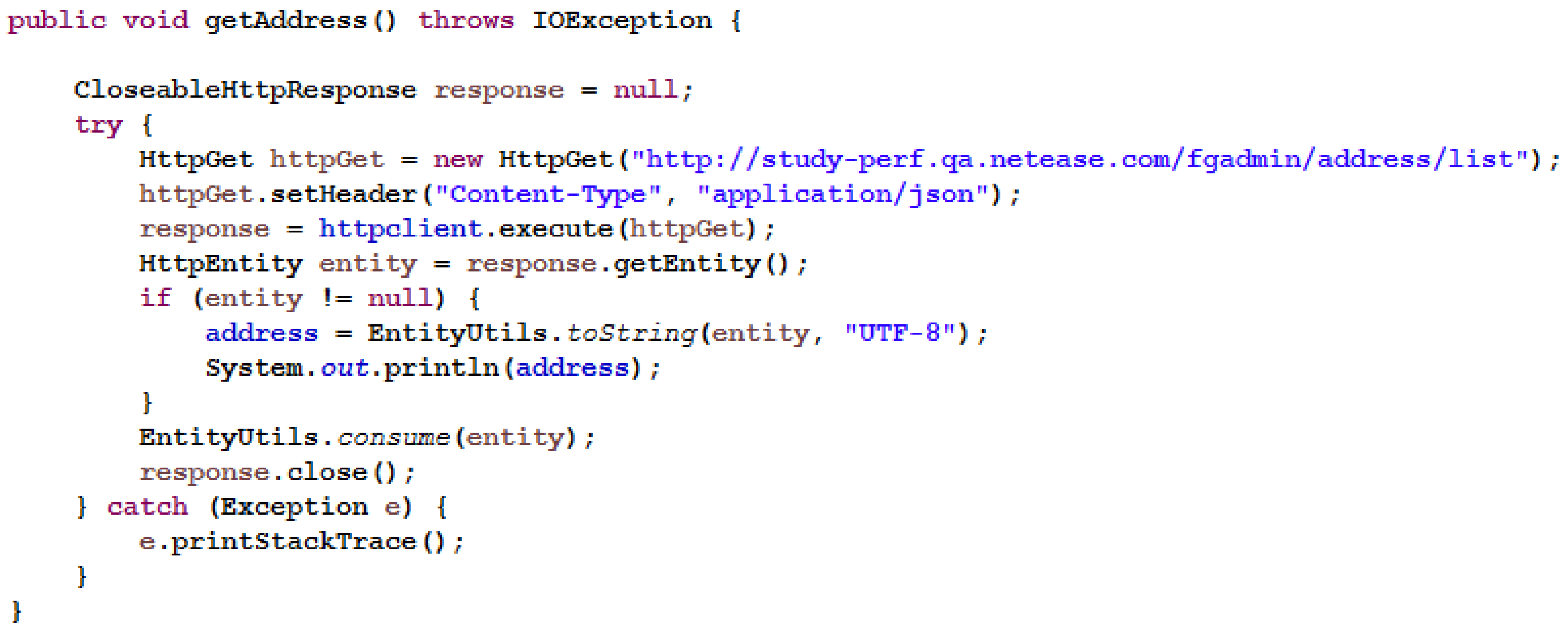
用例步骤-登录成功



```java
public void login() throws IOException {

    CloseableHttpResponse response = null;
    try {
        HttpPost httpPost = new HttpPost("http://study-perf.qa.netease.com/common/fgadmin/login");
        httpPost.setHeader("Content-Type", "application/json");
        StringEntity entity = new StringEntity(
                "{\"phoneArea\":86,\"phoneNumber\":\"20000000000\"," + "\"password\":\"netease123\"}", "utf-8");
        httpPost.setEntity(entity);
        response = httpclient.execute(httpPost);
        HttpEntity httpEntity = response.getEntity();
        System.out.println("执行结果是: " + EntityUtils.toString(httpEntity));
        EntityUtils.consume(httpEntity);
        response.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

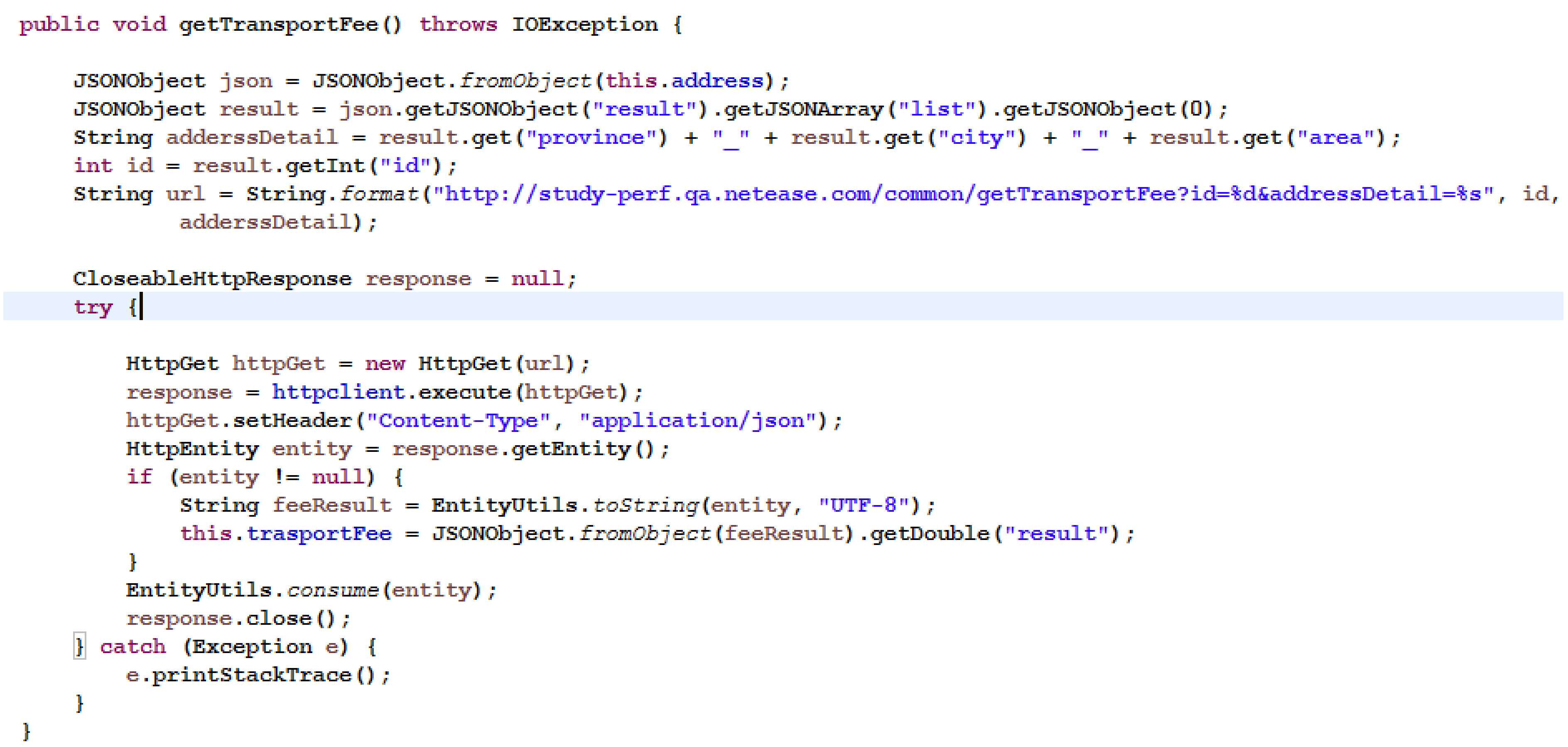
用例步骤-查询收货地址



```
public void getAddress() throws IOException {

    CloseableHttpResponse response = null;
    try {
        HttpGet httpGet = new HttpGet("http://study-perf.qa.netease.com/fgadmin/address/list");
        httpGet.setHeader("Content-Type", "application/json");
        response = httpclient.execute(httpGet);
        HttpEntity entity = response.getEntity();
        if (entity != null) {
            address = EntityUtils.toString(entity, "UTF-8");
            System.out.println(address);
        }
        EntityUtils.consume(entity);
        response.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

用例步骤-计算运费



```java
public void getTransportFee() throws IOException {

    JSONObject json = JSONObject.fromObject(this.address);
    JSONObject result = json.getJSONObject("result").getJSONArray("list").getJSONObject(0);
    String adderssDetail = result.get("province") + "_" + result.get("city") + "_" + result.get("area");
    int id = result.getInt("id");
    String url = String.format("http://study-perf.qa.netease.com/common/getTransportFee?id=%d&addressDetail=%s", id,
            adderssDetail);

    CloseableHttpResponse response = null;
    try {

        HttpGet httpGet = new HttpGet(url);
        response = httpclient.execute(httpGet);
        httpGet.setHeader("Content-Type", "application/json");
        HttpEntity entity = response.getEntity();
        if (entity != null) {
            String feeResult = EntityUtils.toString(entity, "UTF-8");
            this.trasportFee = JSONObject.fromObject(feeResult).getDouble("result");
        }
        EntityUtils.consume(entity);
        response.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

用例步骤–提交订单

```java
JSONObject json = JSONObject.fromObject(address);
JSONObject result = json.getJSONObject("result").getJSONArray("list").getJSONObject(0);

String receiverName = result.getString("receiverName");
String cellPhone = result.getString("cellPhone");
String addressDetail = result.getString("addressDetail");
String province = result.getString("province");
String city = result.getString("city");
String area = result.getString("area");
double Fee = this.trasportFee;
JSONObject jsonPost = new JSONObject();
jsonPost.element("skuIds", "2");
jsonPost.element("skuNumbers", "1");
jsonPost.element("stockIds", "74966312");
jsonPost.element("receiverName", receiverName);
jsonPost.element("cellPhone", cellPhone);
jsonPost.element("addressDetail", addressDetail);
jsonPost.element("province", province);
jsonPost.element("city", city);
jsonPost.element("area", area);
jsonPost.element("voiceStatus", 0);
jsonPost.element("needInvoice", 0);
jsonPost.element("invoiceHead", "");
jsonPost.element("transportFee", Fee);
jsonPost.element("logisticsCompanyId", 1);
jsonPost.element("accessSource", "noSource");
jsonPost.element("accessDevice", 0);
```

用例步骤–提交订单（续）

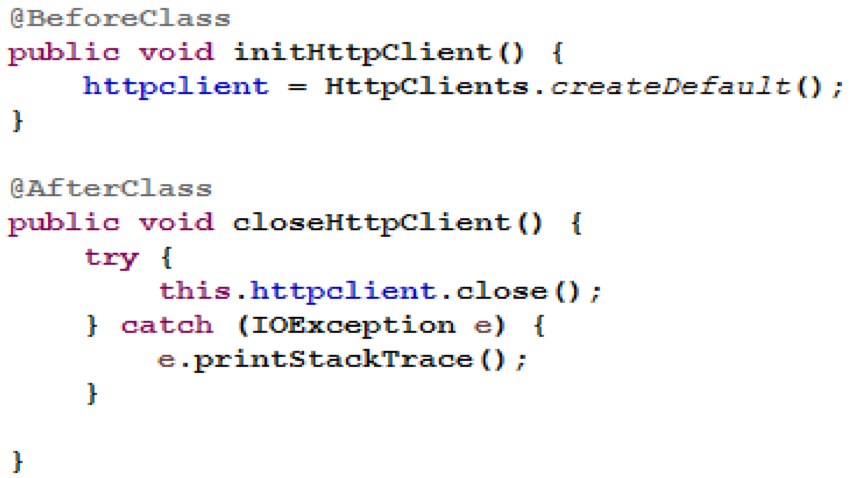
```
CloseableHttpResponse response = null;
try {

    HttpPost httpPost = new HttpPost("http://study-perf.qa.netease.com/fgadmin/orders/submit");
    httpPost.setHeader("Content-Type", "application/json");
    StringEntity entity = new StringEntity(jsonPost.toString(), "utf-8");
    httpPost.setEntity(entity);
    httpPost.setHeader("csrfToken", "csrfToken");

    response = httpclient.execute(httpPost);
    HttpEntity httpEntity = response.getEntity();
    String submitResult = EntityUtils.toString(httpEntity);
    System.out.println(submitResult);
    JSONObject jsonResult = JSONObject.fromObject(submitResult);
    Assert.assertEquals(jsonResult.getInt("code"), 200);
    EntityUtils.consume(httpEntity);
    response.close();
} catch (Exception e) {
    e.printStackTrace();
}
```

场景测试总结

- 注意事项：
  - 使用TestNG的BeforeClass和AfterClass去初始化和关闭HttpClient对象
  - 使用dependsOnMethods属性关联场景的接口



```
@BeforeClass
public void initHttpClient() {
    httpclient = HttpClients.createDefault();
}

@AfterClass
public void closeHttpClient() {
    try {
        this.httpclient.close();
    } catch (IOException e) {
        e.printStackTrace();
    }

}
```

场景测试总结

- 注意事项：
  - 根据具体场景分析关联的接口，使用TestNG功能来安排用例执行顺序
  - 使用成员变量来传递接口返回数据