

## \* 网络协议

### 1. cookie、session、token区别

cookie是一门客户端技术，一般是由服务器生成返回给浏览器客户端来保存的，并且cookie是以键值对的形式保存在浏览器客户端的，每一个cookie都会有名称，值，过期时间...。cookie有很多使用场景，在项目中比较常见的有：

- 1.登录记住用户名
- 2.记录用户浏览记录

Http协议是无状态的，Session是一种让请求从无状态变成有状态的机制，Session是服务端的会话技术，当用户登录了系统，服务器端的就会创建一个会话，此会话中可以保存登录用户的信息。

token是令牌技术，服务器端生成的一段字符串，使用token做鉴权不需要将用户信息保存在服务器。

### 2. get/post请求区别

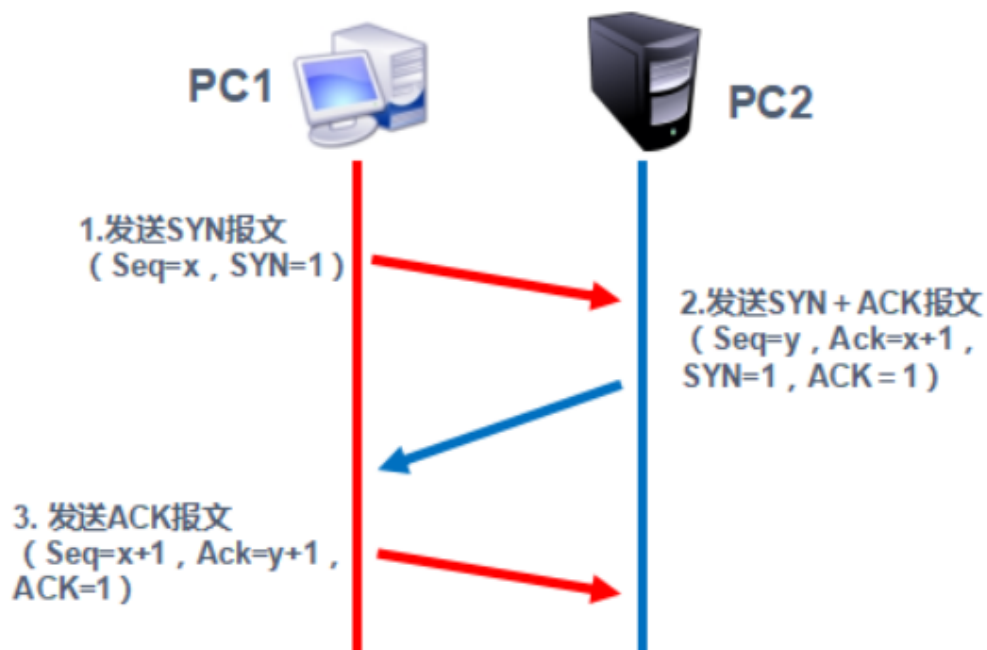
Get:

- 1.一般对于从服务器取数据的请求可以设置为get方式
- 2.Get方式在传递参数的时候，一般都会把参数直接拼接在url上（例如：<http://xxx?id=1&type=2>）
- 3.Get请求方法可以传递的参数数据量是有限的（因为一般参数是拼接在url上，而浏览器对于url的长度是有限制的）
- 4.Get请求因为是直接将数据拼接在url上，所以安全性不如post强（相对），但是get的执行效率要快于post

Post:

- 1.一般往服务器提交数据的请求会设置为post方式
- 2.Post方式传递参数一般都会把参数放在请求体，而不会拼接在url
- 3.Post请求方法可以提交的数据量不受限制
- 4.Post请求传参相对于get要安全（相对不是绝对），但是post的执行效率不如get。

### 3. TCP/IP三次握手，四次挥手

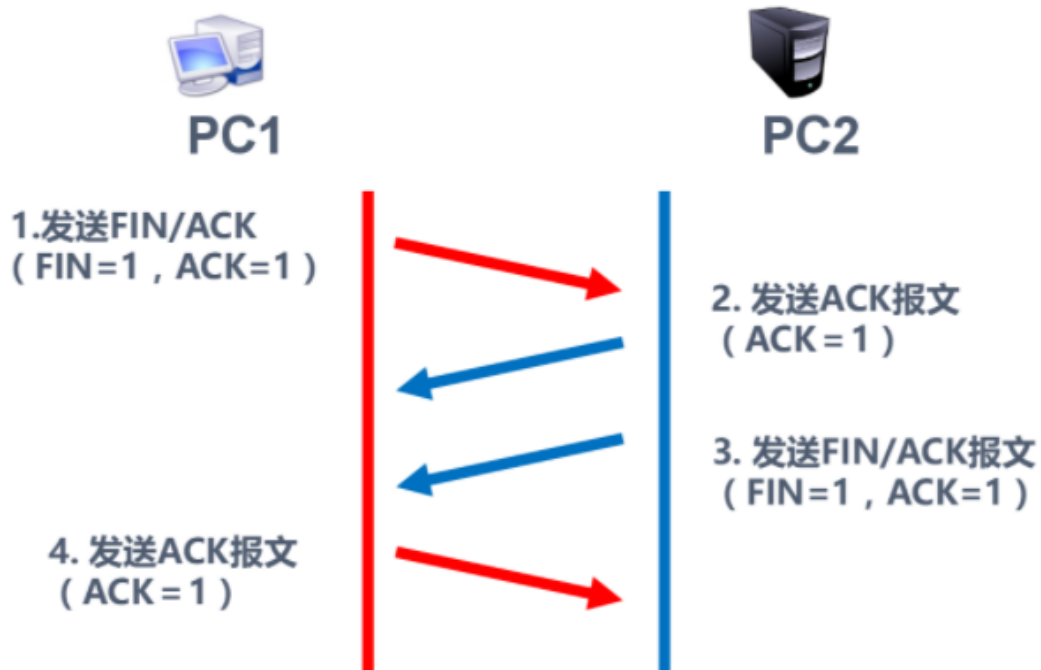


三次握手是指建立TCP连接协议时，需要在客户端和服务端之间发送三个包，握手过程中传送的包里不包含数据，三次握手完毕后，客户端与服务器才正式开始传送数据。

第一次握手：客户端和服务端都处于CLOSED关闭状态，客户端将标志位SYN置为1，随机产生一个值seq=x，并将该数据包发给服务器，客户端进入SYN-SENT（同步已发送）状态；

第二次握手：服务端收到数据包后由标志物SYN=1得知客户端请求建立连接，服务端将标志位SYN和ACK都设置为1，确认号ack = x+1,随机产生一个值seq = y，并将该数据包发送给客户端以确认连接请求，服务端进入SYN-RCVD（同步收到）状态；

第三次握手：客户端收到确认后，检查确认号ack是否为x+1，ACK是否为1，如果正确则将标志位ACK置为1，ack = y+1，并将该数据包发送给服务器，服务器检查ack是否为y+1，ACK是否为1，如果正确则连接建立成功，客户端和服务端进入ESTABLISHED（已建立连接）状态，完成三次握手，随后客户端和服务端就可以开始传输数据。



第一次挥手：客户端发起中断连接请求，也就是发送FIN报文，告诉服务端我的数据已经全部发送完了，如果你还没有准备好，可以继续发送数据。客户端进入FIN-WAIT-1（终止等待1）状态，等待服务端确认；

第二次挥手：服务端发送ACK，告诉客户端，请求我收到了，但我还没有准备好，请你等我的消息。这个时候服务端进入CLOSE-WAIT（关闭等待）状态，此时的TCP处于半关闭状态。客户端收到服务端的确认后进入FIN\_WAIT（终止等待2）状态，等待服务端发送连接释放报文段；

第三次挥手：服务端没有要向客户端发送的数据了，服务端发出连接释放报文段，服务器进入LAST-ACK（最后确认）状态，等待客户端的确认；

第四次挥手：客户端收到服务端发送的连接释放报文段后，对此发出确认报文段，客户端进入TIME-

WAIT（时间等待）状态。此时TCP未释放掉，需要经过时间等待计时器设置的时间2MSL后，客户端才进入CLOSED（关闭）状态。

## 4. OSI七层网络模型

第一层：物理层

电缆连线连接器，网卡等,物理层的任务就是为它的上一层提供一个物理连接，以及它们的机械、电气、功能和过程特性。

第二层：数据链路层

交换机,控制物理层和网络层之间的通讯，把网络层的数据分割成物理层可以传输的帧。

第三层：网络层

路由器,将网络地址翻译成对应的物理地址，并决定如何将数据从发送方路由到接收方。

第四层：传输层

负责建立端到端的连接，保证报文在端到端之间的传输。

第五层：会话层

负责建立、管理和终止表示层实体之间的会话连接

第六层：表示层

数据的解码和编码、加密和解密、压缩和解压缩

第七层：应用层

负责对软件提供接口以使程序能使用网络服务

## 5. 简述一次浏览器请求过程

一、域名解析

二、建立连接（TCP三次握手建立连接）

三、发起HTTP请求

四、服务器响应请求返回结果

五、浏览器得到HTML、js、css

六、浏览器解析HTML、js、css等资源

七、浏览器对页面进行渲染

八、断开连接（TCP四次挥手断开连接）

## 接口自动化相关

### 1. 接口自动化测试框架设计

开发语言：Java

项目构建工具：Maven

测试框架：TestNG管理/执行用例

底层接口自动化支持库：RestAssured

可视化报表：Allure

设计思路：

1、用例通过Excel管理，用例编辑方便直观

2、数据驱动DataProvider

3、参数化设计-解决接口依赖，参考Postman类似设计，将接口依赖数据提取出来存放到全局共享池中，需要时通过引用即可

4、JDBC操作数据库二次封装，数据库断言更方便

5、随机工具类实现，用例前置所需数据方便获取

6、日志集成

7、集成Allure报表

8、Jenkins持续集成

- ▶ common
- ▶ data
- ▶ pojo
- ▶ testcases
- ▶ utils

common: 公共类封装, 基于REST-assured封装get/post/put/delete等请求, 响应断言与数据库断言公共方法封装, 参数化替换实现

data: 参数化数据保存

pojo: 实体类

testcases: 测试用例, 通过TestNG框架注解管理, DataProvider实现数据驱动

utils: 工具类, Excel操作工具类、JDBC操作工具类、随机数据生成工具类、正则表达式工具类...

## 2. 依赖第三方接口如何进行测试

通过mock服务模拟接口返回数据, 比如Java的开源项目moco

## 3. 接口自动化中前后接口依赖怎么处理的

- 1、将依赖接口返回的需要数据取出来
- 2、作为参数传给下个接口

## 4. Java如何连接数据库进行操作?

JDBC(Java Database Connectivity),使用apache提供的开源dbutils工具类库使得操作数据库变得更加简单, 提供了QueryRunner核心类进行增删改查等操作。

## 5. 如何构造测试数据?

- 通过UI页面操作生成  
缺点: 页面稳定性差, 维护成本高  
优点: 相对直接
- 通过调用接口API生成  
缺点: 无法保证被调接口是否正常, 如果该生成数据接口异常, 多个调用方使用都会导致失败。  
优点: 使用简单, 直接调用即可
- 通过操作数据库直接构造测试数据  
缺点: 需要测试人员深入理解数据库表结构与业务逻辑, 对测试要求高, 建议只查询, 不做增删改  
优点: 最直接原始, 能够保证隔离性

## 6. 接口出现异常, 如何分析?

REST-assured框架/Postman/Jmeter工具的请求和响应log信息, 分析请求参数、请求头、地址是否与接口文档一致

分析服务端日志

## 7. 接口自动化如何断言?

使用TestNG的Assert类

- http响应状态码
- 响应头、响应体字段比对: REST-assured内置Gpath (支持对json、xml、html字段提取)

- 数据库断言：JDBC操作数据库

## 8. 接口自动化取得了哪些成果

- 1、应用场景：回归 or 冒烟？
- 2、接口自动化测试覆盖率（接口自动化用例/接口功能用例），优先覆盖核心业务接口模块
- 3、节省的人力成本

## 9. 自动化如何落地实施

## 10. 实施自动化过程中遇到了哪些问题？

流程\文档

框架设计/用例实现（测试数据管理问题）

接口自动化执行（不稳定）