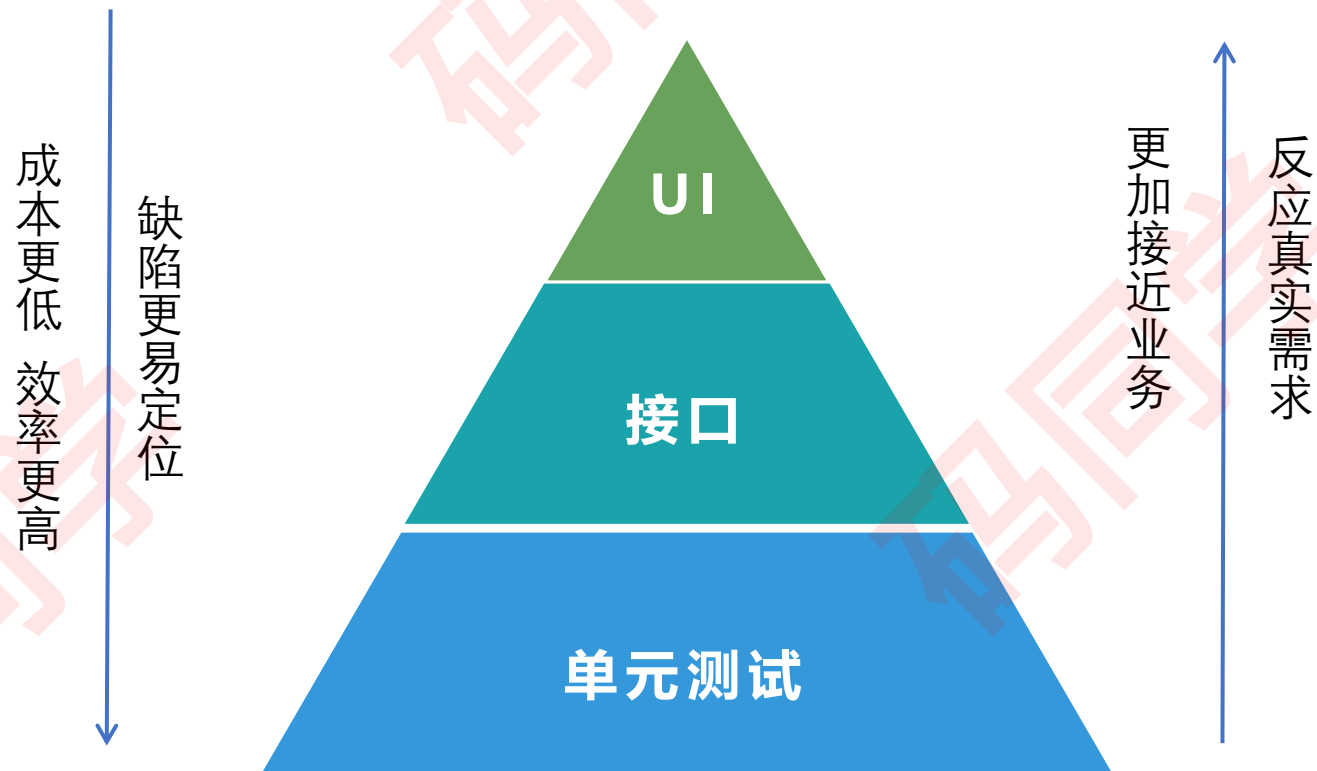




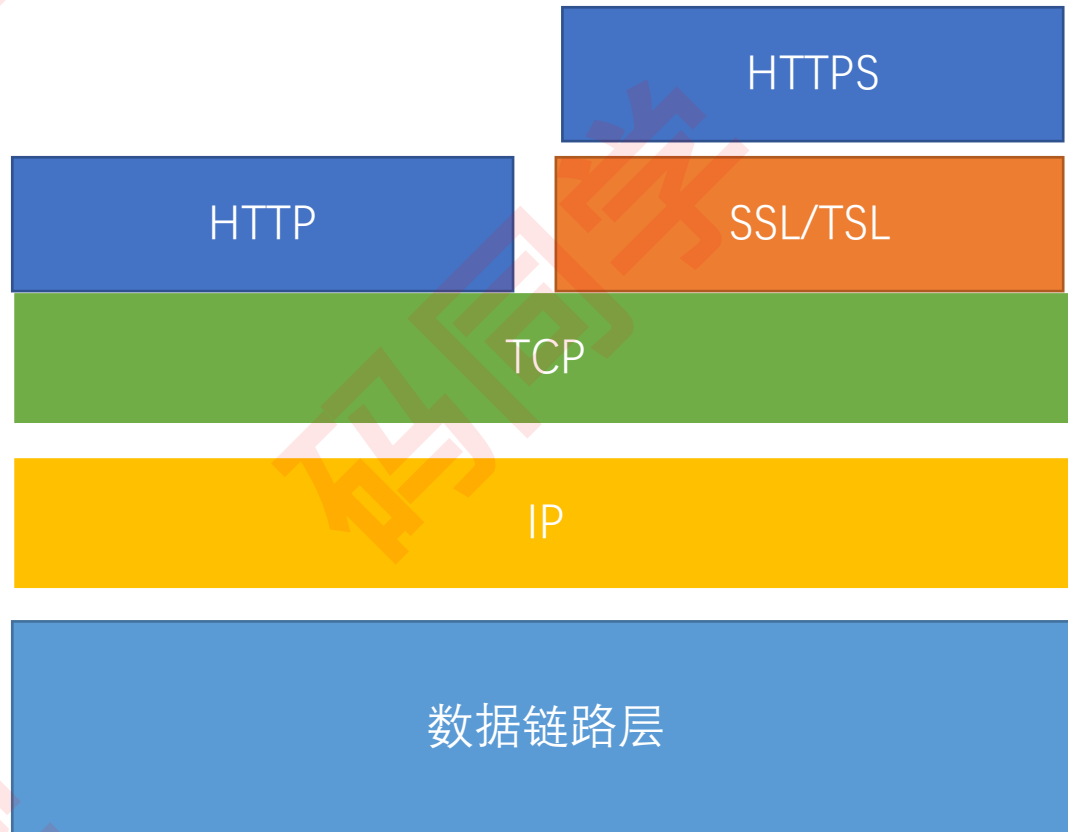
接口测试基础

授课老师：沙陌老师
微信：Matongxue_2

- 接口泛指实体把自己提供给外界的一种抽象化物（可以为另一实体），用以由内部操作分离出外部沟通方法，使其能被内部修改而不影响外界其他实体与其交互的方式。
- 人类与电脑等信息机器或人类与程序之间的接口称为用户界面。电脑等信息机器硬件组件间的接口叫硬件接口。电脑等信息机器软件组件间的接口叫软件接口。
- 在计算机中，接口是计算机系统中两个独立的部件进行信息交换的共享边界。这种交换可以发生在计算机软、硬件，外部设备或进行操作的人之间，也可以是它们的结合。



- UI 元素的展示以及和用户交互的正确性
- API自动化偏重于Http的交互，通过组合不同的数据发送Http请求，获取结果并验证响应
- API测试编写和维护都比UI测试更容易，业界常常采用API和UI组合实现自动化测试不同层次的覆盖



HTTP与HTTPS的区别

- HTTPS协议需要到ca申请证书，一般免费证书很少，需要交费。
- HTTP是超文本传输协议，信息是明文传输，HTTPS则是具有安全性的ssl加密传输协议。
- HTTP和HTTPS使用的是完全不同的连接方式用的端口也不一样,前者是80,后者是443。
- HTTPS协议是由SSL+HTTP协议构建的可进行加密传输、身份认证的网络协议，要比http协议安全

HTTP--Hyper Text Transfer Protocol, 超文本传输协议, 是一种建立在TCP上的无状态连接, 整个基本的工作流程是客户端发送一个HTTP请求, 说明客户端想要访问的资源 and 请求的动作, 服务端收到请求之后, 服务端开始处理请求, 并根据请求做出相应的动作访问服务器资源, 最后通过发送HTTP响应把结果返回给客户端。其中一个请求的开始到一个响应的结束称为事务, 当一个事物结束后还会在服务端添加一条日志条目

HTTP/0.9

只有一个命令GET

没有HEADER等描述数据的信息

服务器发送完毕，就关闭TCP连接

HTTP/1.0

增加了很多命令,比如POST, PUT等

增加status code和header

多字符集支持、多部分发送、权限、缓存等

HTTP/1.1

持久连接

pipeline,可以在同一个连接里面发送多个请求（同时只能存在一个http请求）。但在服务端还是要按照顺序一个进行返回。假如两个请求，第一个处理得较慢，第二个请求已经处理完成了，但还是要等第一个请求返回完之后再返回第二个请求

增加host和其他一些命令，host的出现可以让同一个物理服务器跑多个web服务

HTTP2

所有数据以二进制传输

同一个连接里面发送多个请求不再需要按照顺序来

头信息压缩以及推送等提高效率的功能

主要特点：

- 1、简单快速：客户向服务器请求服务时，只需传送请求方法(method)和路径(url)。请求方法常用的有GET,POST,PUT,DELETE。每种方法规定了客户与服务器联系的类型不同。由于HTTP协议简单，使得HTTP服务器的程序规模小，因而通信速度很快。
- 2、灵活：HTTP允许传输任意类型的数据对象。传输类型由Content-Type加以标记。
- 3、支持B/S及C/S模式
- 4、无连接：无连接的含义是限制每次连接只处理一个请求。服务器处理完客户的请求，并收到客户的应答后，即断开连接。采用这种方式可以节省传输时间。(HTTP1.1开始支持长连接，需要服务端配置相应的参数)
- 5、无状态：HTTP协议是无状态协议。无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则它必须重传，这样可能导致每次连接传送的数据量增大。
另一方面，在服务器不需要先前信息时它的应答就较快。

HTTP是基于客户/服务器模式，且面向连接的。典型的HTTP事务处理有如下的过程：

- (1) 客户与服务器建立连接；
- (2) 客户向服务器提出请求；
- (3) 服务器接受请求，并根据请求返回相应的文件作为应答；
- (4) 客户与服务器关闭连接。

HTTP报文由从客户机到服务器的请求和从服务器到客户机的响应构成。请求报文格式如下：

请求行 - 通用信息头 - 请求头 - 实体头 - 报文主体

响应报文格式如下：

状态行 - 通用信息头 - 响应头 - 实体头 - 报文主体

请求头：

accept：请求报头域，指定客户端可以接收的类型有哪些

accept-encoding：指定客户端可接收的内容编码

accept-language：指定客户端可接收的编码语言

host：用户指定请求资源的主机和端口号

cookies：用于存储本地的数据

refer：用于标识这个请求是从哪个页面来的，如登录前是从主页来的，就显示主页的信息

user-agent (UA)：用于识别用户使用的客户端版本等信息

content-type：用于标识具体请求的媒体类型

| 方法 | 描述 |
|---------|-------------------------------------------------------------------------|
| GET | 请求指定的页面信息，并返回实体主体 |
| HEAD | 类似于 GET 请求，只不过返回的响应中没有具体的内容，用于获取报头 |
| POST | 向指定资源提交数据进行处理请求（例如提交表单或者上传文件）。数据被包含在请求体中。POST 请求可能会导致新的资源的建立和/或已有资源的修改。 |
| PUT | 从客户端向服务器传送的数据取代指定的文档的内容。 |
| DELETE | 请求服务器删除指定的页面。 |
| CONNECT | HTTP/1.1 协议中预留给能够将连接改为管道方式的代理服务器。 |
| OPTIONS | 允许客户端查看服务器的性能。 |
| TRACE | 回显服务器收到的请求，主要用于测试或诊断。 |
| PATCH | 是对 PUT 方法的补充，用来对已知资源进行局部更新。 |

HTTP状态码的英文为HTTP Status Code，有如下几类：

1xx: 表示HTTP请求已经接受，继续处理请求

2xx: 表示HTTP请求已经处理完成

3xx: 表示把请求访问的URL重定向到其他目录

4xx: 表示客户端出现错误

5xx: 表示服务端出现错误

常见的状态码如下：

| 状态码 | 状态码英文名称 | 中文描述 |
|-----|------------------------|-----------------------------------------------------------------------|
| 400 | Bad Request | 客户端请求的语法错误，服务器无法理解 |
| 401 | Unauthorized | 请求要求用户的身份认证 |
| 403 | Forbidden | 服务器理解请求客户端的请求，但是拒绝执行此请求 |
| 404 | Not Found | 服务器无法根据客户端的请求找到资源（网页） |
| 415 | Unsupported Media Type | 服务器无法处理请求附带的媒体格式 |
| 500 | Internal Server Error | 服务器内部错误，无法完成请求 |
| 503 | Service Unavailable | 由于超载或系统维护，服务器暂时的无法处理客户端的请求。延时的长度可包含在服务器的Retry-After头信息中 |
| 200 | OK | 请求成功。一般用于GET与POST请求 |
| 301 | Moved Permanently | 永久移动。请求的资源已被永久的移动到新URI，返回信息会包括新的URI，浏览器会自动定向到新URI。今后任何新的请求都应使用新的URI代替 |
| 302 | Found | 临时移动。与301类似。但资源只是临时被移动。客户端应继续使用原有URI |
| 304 | Not Modified | 服务器返回此状态码时，不会返回任何资源。客户端通常会缓存访问过的资源 |

Content-Type（内容类型），用于定义接口请求参数的类型和接口返回内容的类型

以application开头的媒体格式类型：

- application/xhtml+xml：XHTML格式
- application/xml：XML数据格式
- application/atom+xml：Atom XML聚合格式
- application/json：JSON数据格式
- application/pdf：pdf格式
- application/msword：Word文档格式
- application/octet-stream：二进制流数据（如常见的文件下载）
- application/x-www-form-urlencoded：<form encType=" " >中默认的encType，form表单数据被编码为key/value格式发送到服务器（表单默认的提交数据的格式）

常见的媒体格式类型如下：

- text/html : HTML格式
- text/plain : 纯文本格式
- text/xml : XML格式
- image/gif : gif图片格式
- image/jpeg : jpg图片格式
- image/png : png图片格式

另外一种常见的媒体格式是上传文件之时使用的：

multipart/form-data : 需要在表单中进行文件上传时，就需要使用该格式

- Soap协议规范:

简单对象访问协议是交换数据的一种协议规范, 是一种轻量的、简单的、基于XML (标准通用标记语言下的一个子集) 的协议, 它被设计成在WEB上交换结构化的和固化的信息

- SOAP与XML的关系: 所有的SOAP消息都使用XML形式编码

- SOAP简单的理解就是这样的一个开放协议 $SOAP = RPC + HTTP + XML$: 采用HTTP作为底层通讯协议; RPC作为一致性的调用途径, XML作为数据传送的格式, 允许服务提供者和服务客户经过防火墙在INTERNET进行通讯交互。

一个SOAP方法可以简单地看作遵循SOAP编码规则的HTTP请求和响应, 一个 SOAP终端则可以看作一个基于HTTP的URL, 它用来识别方法调用的目标。像CORBA/IIOP一样, SOAP不需要具体的对象绑定到一个给定的终端, 而是由具体实现程序来决定怎样把对象终端标识符映像到服务器端的对象

练习网站: http://www.webxml.com.cn/zh_cn/index.aspx 提供了一些免费的soap接口

RESTFUL是一种网络应用程序的设计风格 and 开发方式，基于HTTP，可以使用XML格式定义或JSON格式定义。RESTFUL适用于移动互联网厂商作为业务使能接口的场景，实现第三方OTT调用移动网络资源的功能，动作类型为新增、变更、删除所调用资源

REST（英文：Representational State Transfer，简称REST）描述了一个架构样式的网络系统，
比如 web 应用程序

REST相比于SOAP（Simple Object Access protocol，简单对象访问协议）以及XML-RPC更加简单明了，无论是对URL的处理还是对Payload的编码，REST都倾向于用更加简单轻量方法设计和实现。值得注意的是REST并没有一个明确的标准，而更像是一种设计的风格

RESTFUL特点包括：

- 1、每一个URI代表1种资源；
- 2、客户端使用GET、POST、PUT、DELETE4个表示操作方式的动词对服务端资源进行操作：
GET用来获取资源，POST用来新建资源（也可以用于更新资源），PUT用来更新资源，DELETE用来删除资源；
- 3、通过操作资源的表现形式来操作资源；
- 4、资源的表现形式是XML或者HTML；
- 5、客户端与服务端之间的交互在请求之间是无状态的，从客户端到服务端的每个请求都必须包含理解请求所必需的信息

1. 接口协议
2. 接口功能
3. 接口数据流转
4. 接口的加解密方式
5. 接口测试的范围
6. 接口测试工具选择

1. 时间
2. 人力
3. 范围
4. 里程碑
5. 风险

1. 正常
2. 异常
3. 字段
-

1. 第一轮
2. 第二轮
3. 回归

1. jenkins集成
2. 邮件
3. 报告

接口需求调研

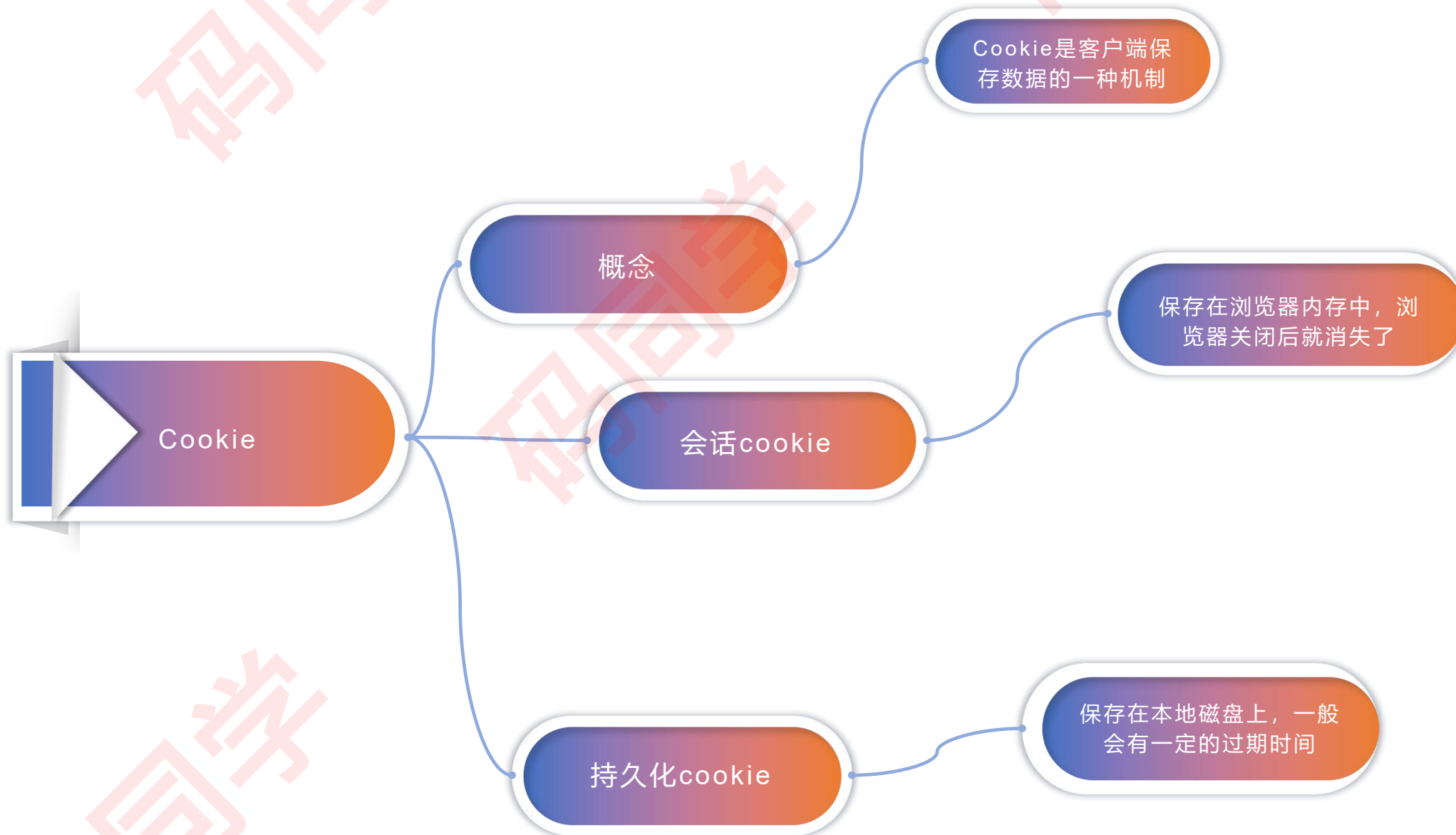
接口测试计划

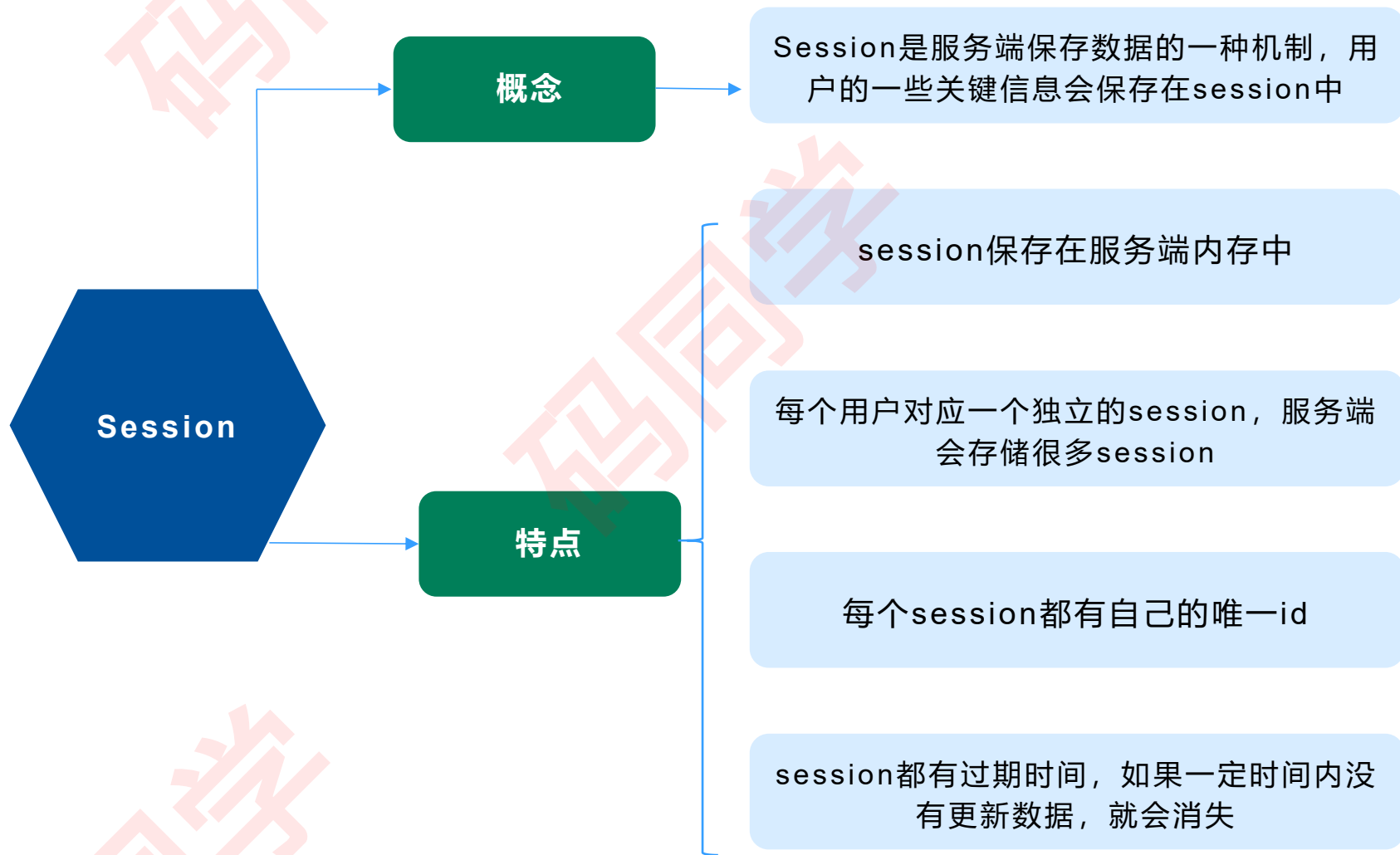
接口用例设计
和评审

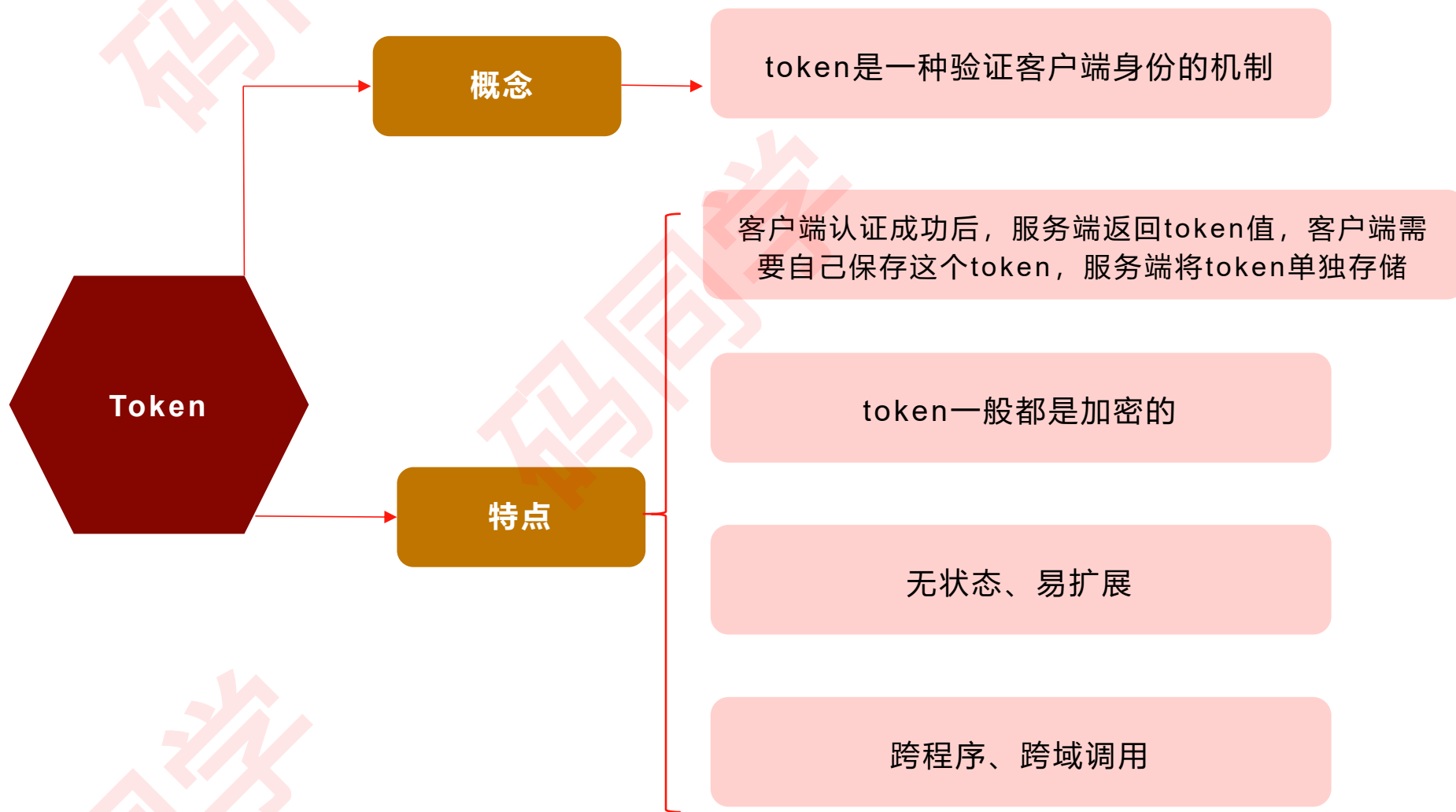
接口测试执行

接口测试自动
化持续集成











THANK YOU

授课老师：沙陌