

web测试

课程安排：

第一天：web系统介绍，html+css+javascript（开发语言）

第二天：网络协议（计算机之间相互通信时遵守的约定，抽象），web测试技术（功能性，兼容性，易用性，性能，安全等）

html+css+javascript用来做什么？为什么要学习html+css+javascript？

html+css+javascript是web系统前端开发的三大语言。

把网页比喻成人体，html作用对于网页的作用就像在定义人体的器官，比如有两只眼睛，一个头.....，那html的作用就是在定义网页的内容；

css作用就好比定义人体的外观，比如大眼睛，双眼皮，头圆圆的，css就用来定义html的样式；

javascript作用好比定义人体的动作，眼睛眨，胳膊可以跳舞.....那javascript的作用就是定义网页的动作，让网页具有交互性。

- 1) 了解网站前端怎么开发出来的，可以对以后进行web系统测试，能够方便去分析定位bug的原因。
- 2) 后面要学习web自动化测试，为后面自动化测试课程打基础。

1.web系统介绍

1.什么是web系统？

人话：需要通过浏览器来进行访问各种网站类的系统，比如百度网站，淘宝网站，工商银行个人网银，页游等。

2.三种常见网络架构

1) B/S (browser/server 浏览器/服务器)===web系统

特点：通过浏览器来访问，浏览器上的网页就是系统前端

前端：跟用户进行交互，用户能看得见摸得着

后端：有服务器软件，比如tomcat，mysql等，还有开发代码，作用是：业务逻辑处理，数据存储等。

比如百度网站，淘宝网站，工商银行个人网银，页游。

2) C/S (client/server 客户端/服务器) ==移动端，PC端等需要安装客户端软件

客户端：是一个客户端软件，需要用户自己安装

比如：手机上的淘宝app，微信app，电脑上的微信程序，360软件，网络电视上的优酷软件

B/S和C/S优缺点：

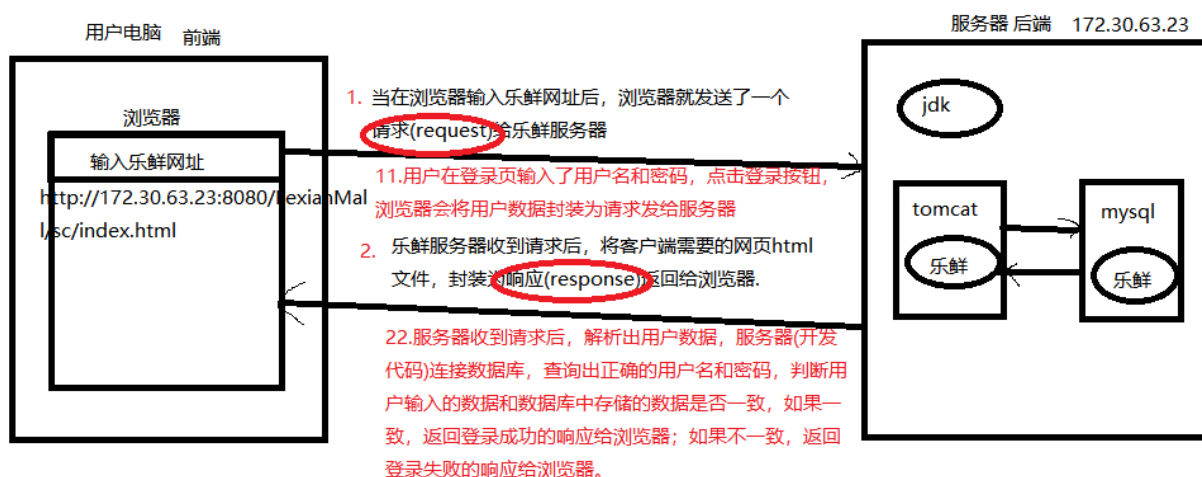
B/S:不需要安装，可以直接打开浏览器访问，比较方便，扩展，维护方便。访问速度会受网络影响，所有的操作都是用鼠标操作，速度慢。

C/S:需要用户安装客户端软件，需要一些配置，不方便，如果扩展新功能，发布新版本，用户需要安装新的客户端(更新)。对于新旧多个版本维护成本高。客户端环境(android和ios)开发成本高。可以支持个性化定制。

3) P2P (point to point 点对点)

比如：飞秋，迅雷

3.web系统工作原理



4.url组成

url(uniform resource location统一资源定位器)==俗称网址

<http://172.30.63.23:8080/LexianMall/sc/index.html>

<https://www.baidu.com/>

<https://search.jd.com/Search?keyword=iphone14&enc=utf-8>

url的组成有5个部分：

1) 协议：http或者https

作用：约定浏览器与服务器之间通信的统一格式，成为协议

http:hypertext transport protocol 超文本传输协议 在浏览器和服务器之间进行超文本传输通信

https:安全的http=http+ssl(安全套接层)，ssl可以对传输的内容进行加密，还能够对服务器进行身份认证。

http和https的区别：

1.http是明文传输，不安全的

2.https是加密传输，相对安全，同时可以认证服务器身份，可以防止伪造服务器的攻击者攻击。

3.http协议默认端口是80，https默认端口号是443，如果使用默认端口号，可以在网址中省略。

4.如果使用https协议，需要申请证书(可以理解为密钥文件)，需要跟专门机构CA来申请，需要一定费用。

2) 主机：服务器的ip地址或者域名

172.30.63.23

www.baidu.com 需要申请 在浏览器访问时候，需要先访问DNS服务器，获取服务器域名对应的ip地址

DNS：域名解析，作用时将域名翻译成ip地址

3) 端口

计算机需要上网(访问其他远程计算机，进行通信)需要网卡，网卡有进行一些虚拟的编号，成为端口号，一块网卡可以有65536个端口号，0-65535 作用：当计算机上面启动一个程序，会占用一个端口号，我们通信时，可以根据端口号来找到对应的应用程序。

http协议默认端口是80，https默认端口号是443

tomcat默认端口号是8080，可以自己配置为其他端口

4) 页面路径

5) 请求参数

以?开头，表示用户的数据，可以有请求参数，也可以没有的。

多个请求参数用&连接，每个请求参数是以键值对形式 比如 ?键1=值1&键2=值2

2.html

html (hypertext markup language 超文本标记语言) 使用各种标记来定义网页的内容，比如文本，图片，链接，视频，音频等。

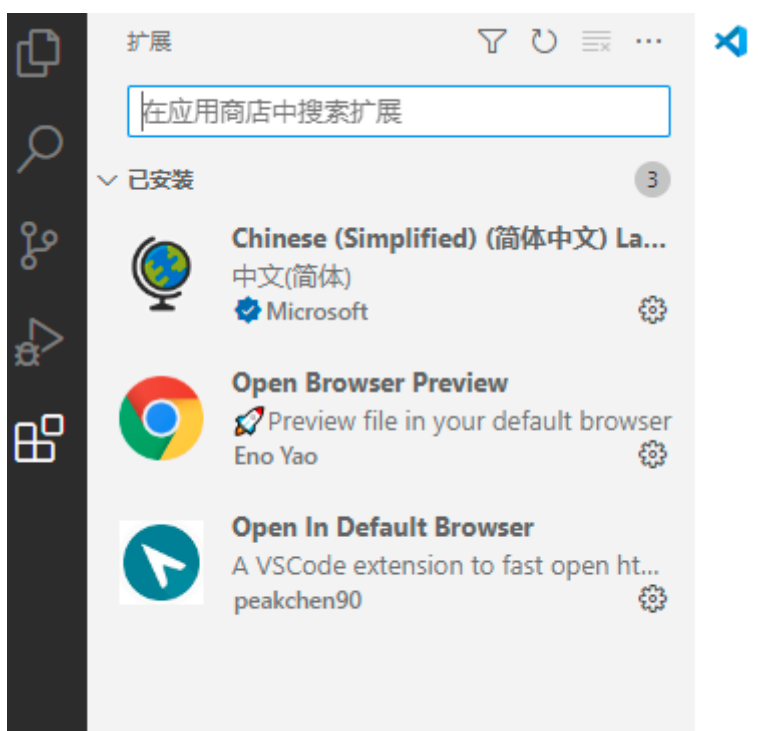
标记格式：<标记名></标记名>

html代码保存的文件后缀名 .html

html代码不区分大小写。

html代码编辑，可以直接用记事本来写，也可以用专门的编辑工具(集成开发环境，IDE)，比如vs code，pycharm，notepad++等。

使用vscode是，安装一下三个插件，使用方便。



1.html文档基本格式

```
<html>
  <!--注释内容 -->
  <!--head头标记，主要来定义浏览器所需要的内容，比如网页标题，编码方式，外部关联文件等-->
  <head>
    <title>这是我的网页</title>  <!--title标记定义网页标题-->
  </head>
  <body> <!-- body标记定义网页显示的内容，比如文字，图片等-->
    这是我的第一个网页，非常简单。
  </body>
</html>
```



```
</html>
```

2) 图片标签

img标签 自闭合标签

src属性 ==定义图片文件的路径 ==必须有的属性

alt属性 ==定义当图片加载失败，显示的替代文字

title属性 ===定义图片的标题，当鼠标放在图片时提示的信息

height属性==定义图片的高度

width==定义图片的宽度

```

```

3) 链接标签

作用：点击链接时会跳转到另外一个页面。

a标签

href属性 ===定义需要跳转到的地址 ==必须有的属性

(1) 文本链接

```
<a href="http://www.baidu.com">百度</a>
```

(2) 图片链接

```
<a href="http://www.baidu.com"></a>
```

4) 表单

表单：将用户在页面上录入的数据提交服务器进行处理。

form标签==表单的大标签

action属性==数据要提交给服务器的哪个程序进行处理

method属性==表示数据提交的方式，分为get和post两种

input标签 ==定义表单中各种各样的元素，比如文本输入框，密码输入框，单选框，复选框，按钮，文件上传等

type属性==定义表单元素的类型

type='text' 文本输入框

type='password' 密码输入框

type='radio' 单选框

type='checkbox' 复选框

type='file' 文件上传框

type='submit' 提交按钮 ==作用：当点击提交按钮时，会将表单中所有的数据提交到服务器

type='button' 普通按钮 需要开发工程师自定义需要实现的功能

.....

name属性 ==定义提交的数据的名称，在表单提交数据时一起提交给服务器，用于让服务器区分不同的数据

value属性 ==表示提交的数据，一般输入框不用定义

```
<!-- 定义一个登录表单 -->
<form action="login" method="get">
  账号: <input type="text" name="account"><br>
  密码: <input type="password" name="passwd"><br>
  <input type="submit" value="登录">
</form>
```

说明：

- 1.如果表单请求方式时get，表单数据放在url中提交
- 2.对于按钮来说，value属性用来定义按钮上显示的文字。

3.css

css (cascading style sheet层叠样式表) 对html中的标签进行样式的修饰，比如定义文本的颜色，字体大小，图片的大小，对齐方式等。主要的作用就是用来美化，布局网页。

css代码文件后缀名时.css

css代码不区分大小写

注释格式： /* 注释内容 */

css代码基本格式：

```
选择器{
    属性1:值1;
    属性2:值2;
    .....
}
```

说明：

- 1.CSS代码中的样式都需要用{}括起来
- 2.{}中的每一条都表示一条样式，修饰比如html标签的颜色，大小等
- 3.所有大的样式都以分号结尾 每一条样式都以键值对形式来定义 属性:值
- 4.{}前面称为选择器，作用选择哪些标签来修饰样式。

1.三种存在形式

1) 行内样式表

在标签的行内，通过style属性来定义css样式

作用域：只能针对这一个标签生效样式。

```
<p style="color: blueviolet;font-size: 50px;text-align: center;font-style: italic;">这是博为峰57期</p>
```

2) 内部样式表

在html文档的内部，在head标签内，通过一个style标签来定义

作用域：可以针对html文档内指定标签来生效

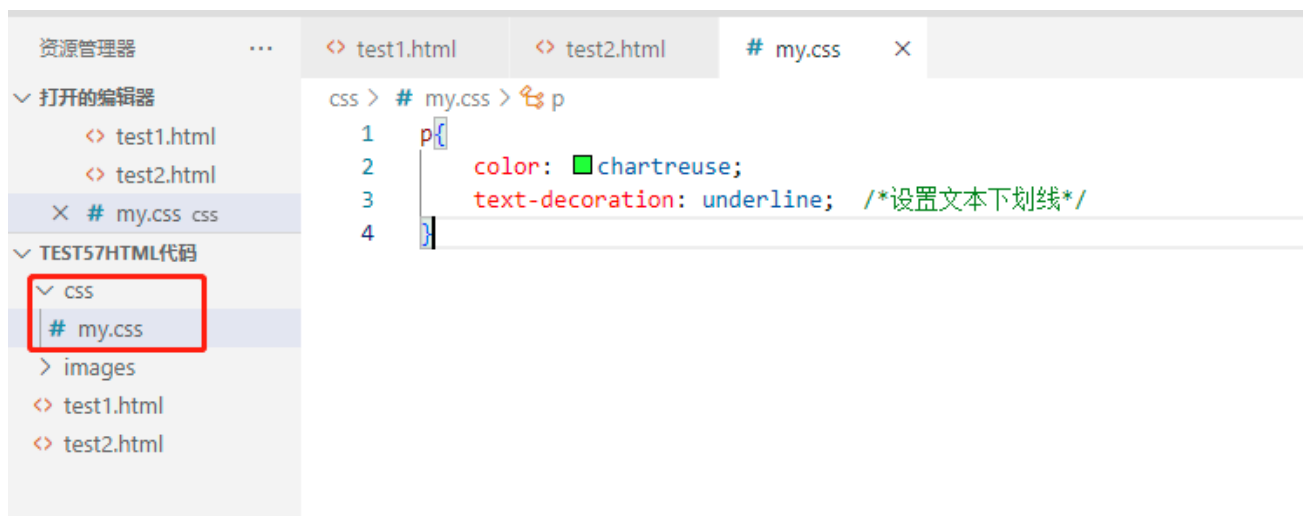
```
<head>
    <title>css</title>
    <style>
        p{
            font-style: italic; /*设置斜体字*/
            font-weight: 900; /*设置字体粗细*/
            font-size: 30; /*设置字体大小*/
            color: blue; /*设置字体颜色*/
        }
    </style>
</head>
```


3) 外部样式表

在html文档的外部，定义一个单独css文件来写css代码

作用域：可以针对多个html去修饰他们的样式。

外部单独CSS文件：



html中如何引用外部的css代码：

在html文档的head标签中用link标签来引用外部css文件

实例：

```
<head>
  <title>css</title>
  <link rel="stylesheet" href="css/my.css">
</head>
```

说明：

link标签中rel="stylesheet"表示关联的是样式表文件

href属性用来定义外部样式表文件的路径

如果某一个标签被多种样式表来修饰，采用就近原则。谁离标签更近，就采用哪种样式表。

2.常用选择器

选择器作用：选择哪些标签来修饰样式。

1) 标签选择器

直接使用标签名作为选择器，选择这一种标签来修饰样式。

```
p{
    font-style: italic; /*设置斜体字*/
    font-weight: 900; /*设置字体粗细*/
    font-size: 30; /*设置字体大小*/
    color: blue; /*设置字体颜色*/
}
```

2) id选择器

对于html标签可以定义id属性，就好比人的身份证号的作用一样，用来唯一区分标签。

使用标签的id属性来选择对应的标签来修饰样式。

格式 #id属性

实例：

```
<p >这是博为峰57期</p>
<p id="t1">学习css</p>
```

css代码：

```
#t1{
    color: darkorange;
}
```

3) 类选择器

在html标签中可以针对标签定义class属性，用来对标签进行分类，一般与css结合使用。

使用class属性来选择对应的标签来修饰样式。

格式 .class属性

css代码：

```
.c1{
    color: darkgreen;
}
.c2{
    color: deeppink;
}
```

html代码：

```
<p class="c1">这是博为峰57期</p>
<p class="c1">学习css</p>
<p class="c2">学习内部样式表</p>
```

如果多个选择器选择到了同一个标签，如何生效？

优先级：

id选择器>类选择器>标签选择器

3.javascript

javascript是一种脚本语言，可以直接被浏览器解释执行。简称js

javascript与java关系？

没有任何关系。发布前夕改名字，为了蹭热度。

js代码文件后缀名是.js

js代码每一句都以分号结尾

js代码区分大小写。

注释格式 //注释内容

1.在html内部嵌入js代码

在html内部使用script标签来嵌入js代码，script标签可以在head中也可以在body中。

```
<html>
  <head>
    <title>js</title>
  </head>
  <body>
    <p>学习js</p>
    <script>
      alert('今天周五，真开心');    //alert() 是js中的内置函数 在浏览器显示一个弹框
    </script>
  </body>
</html>
```

2.关联html外部单独的js文件

html关联外部js文件，在head标签中通过script标签来进行关联。

外部js文件：



html关联外部js文件:

```
<head>
  <title>js</title>
  <script src="js/my.js"></script>
</head>
```

说明: src属性用来定义外部js文件的路径

3.通过事件处理程序执行js代码

事件: 网页中得动作, 比如网页中鼠标点击, 双击, 按键, 鼠标移入, 鼠标移出等

事件处理程序: 当网页发生所定义得事件后执行得js代码。==js中得函数调用

事件定义得一般格式 事件名='事件处理程序' 事件名一般以on开头

比如一个按钮定义一个点击事件

```
<input type='button' value='计算' onclick='事件处理程序'>
```

示例:

```
<html>
  <head>
    <title>js</title>
    <script src="js/my.js"></script>
  </head>
  <body>
    <!-- 在网页上定义一个加法计算器 两个输入框 一个按钮 当点击按钮时，将输入框中
    两个数进行加法运算，然后弹框显示结果-->
    <input type="text" id="num1">
    <input type="text" id="num2">
    <input type="button" value="计算" onclick="add()" >
  </body>
</html>
```

```
//定义一个加法函数
function add() {
  //1.先从输入框中读出两个运算数据
  //var 表示定义变量关键字
  //document表示整个html文档对象
  //getElementById() 该方法通过id定位标签
  //input标签(输入框)中输入得内容存储在value属性中
  //以下代码得意思：在html文档中找到id为num1得标签，
  //获取该标签得value属性值保存在变量n1中
  //parseInt() 函数作用为转整形
  var n1=parseInt(document.getElementById('num1').value);
  var n2=parseInt(document.getElementById('num2').value);
  //2.进行加法
  var result=n1+n2;
  //3.弹框显示加法结果
  alert(result);
}
```

4.函数

函数：一段具有特定功能得代码块，可以被反复调用。目的：提高代码复用率，提高编程效率。

特点：在函数定义时，代码不会执行。

在函数调用时，代码才会执行。

1.函数定义

语法：

```
function 函数名 (形参1, 形参2.....)
{
    函数代码块
}
```

2.函数调用

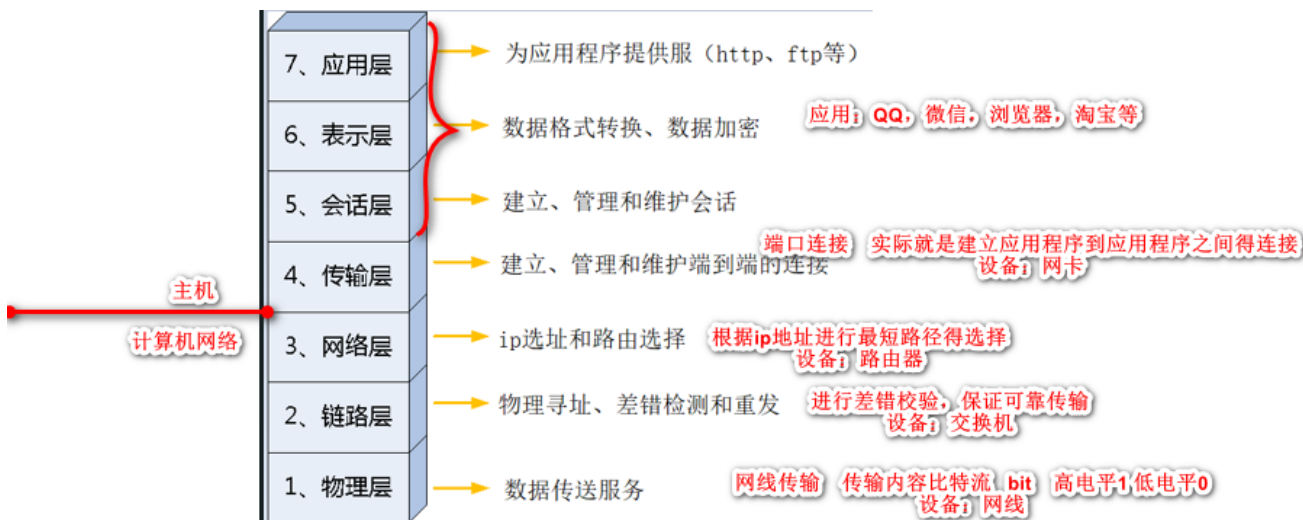
语法：

```
函数名 (实参1, 实参2.....) ;
```

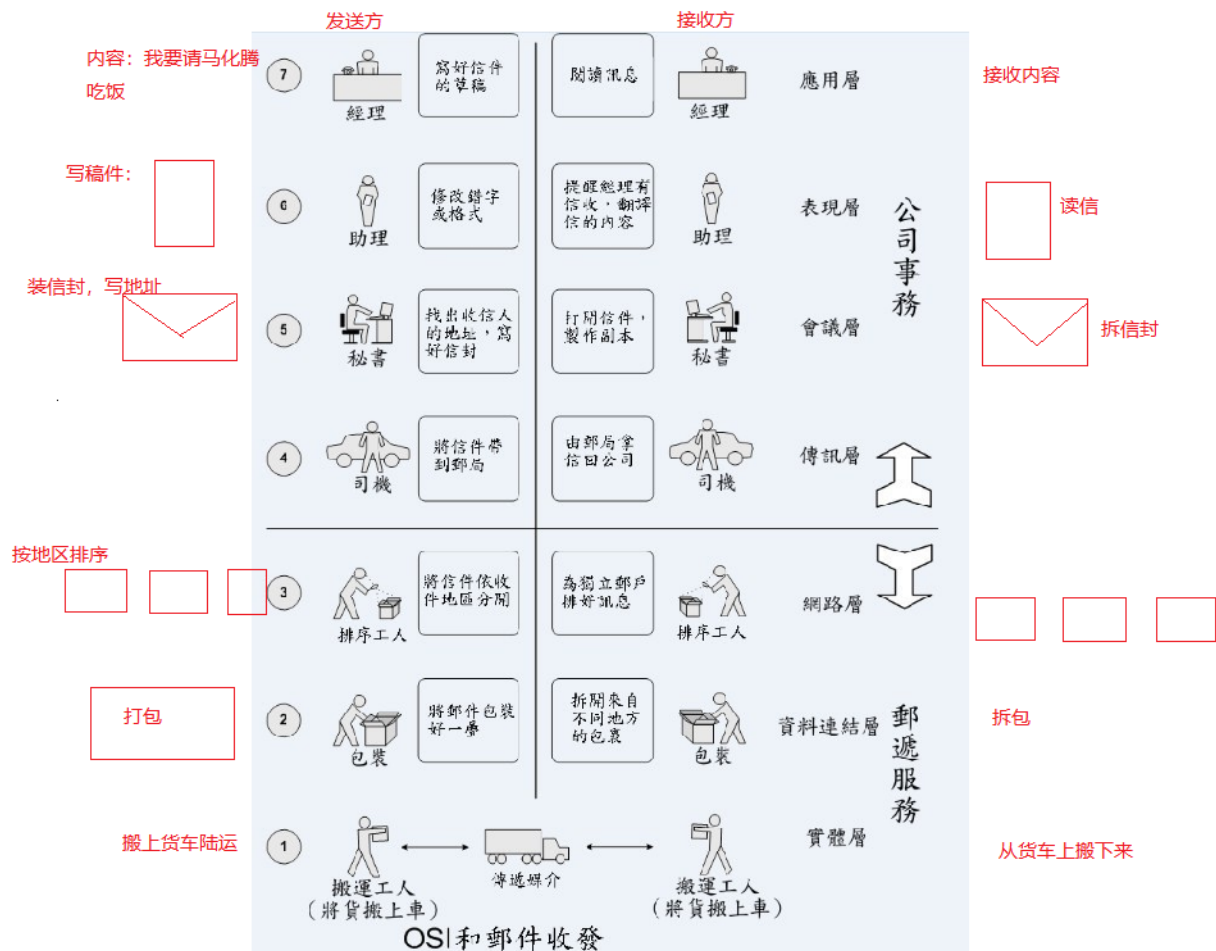
4.网络协议

1.OSI

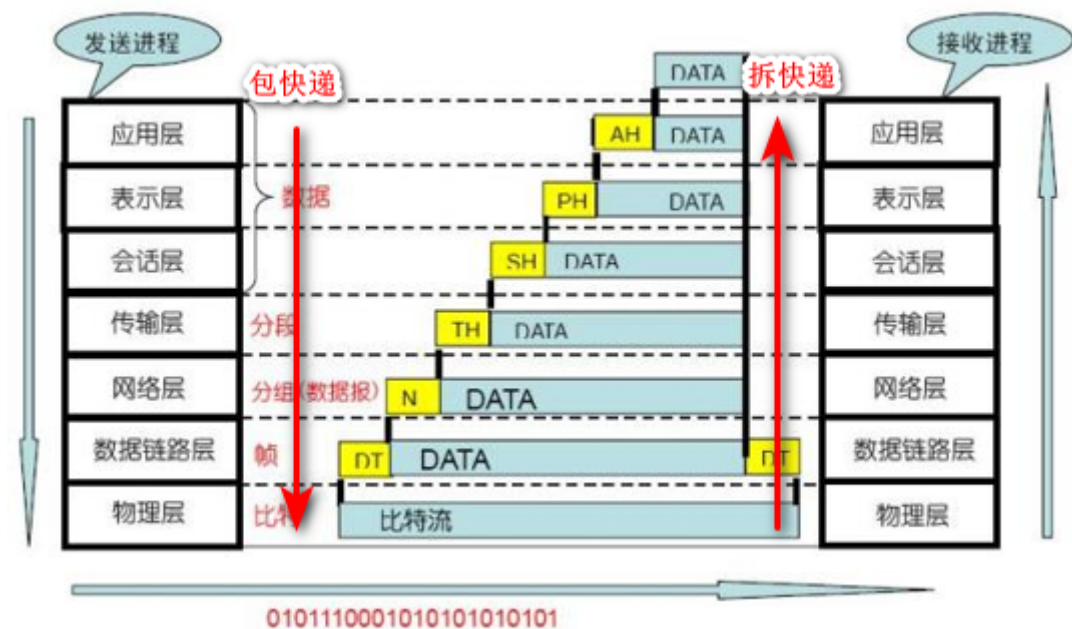
Open System Interconnection reference model 开放式系统互联参考模型



2.通信过程生活示例==邮寄

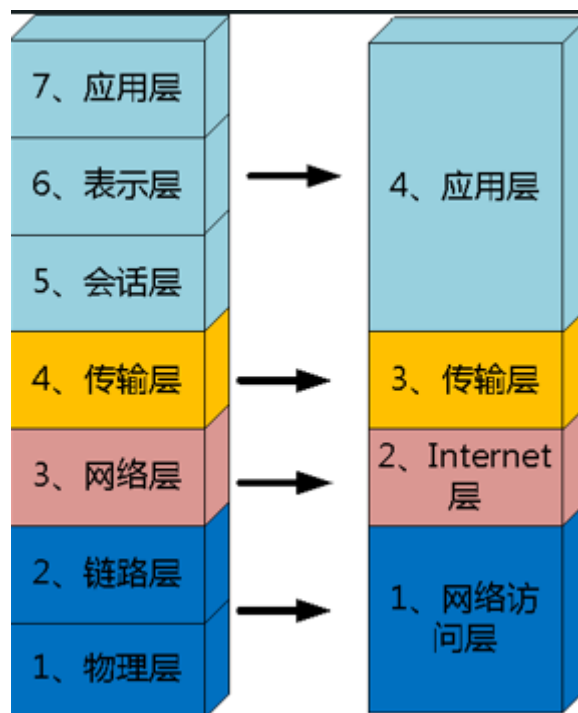


3.封装和解封装



4.TCP/IP模型

实际应用中，OSI模型太复杂，所以做了简化，将7层OSI模型简化为4层的TCP/IP模型



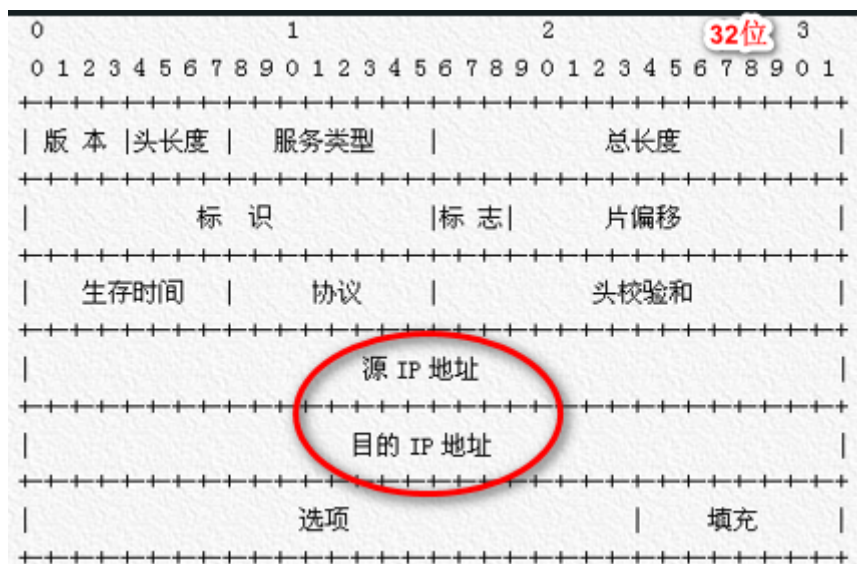
TCP协议 transmission control protocol 传输控制协议

IP协议 internet protocol 互联网协议

5.IP协议

IP协议工作在第二层：网络层

IP协议头：



无线局域网适配器 WLAN:

```

连接特定的 DNS 后缀 . . . . . :
本地连接 IPv6 地址. . . . . : fe80::deb7:fe27:ec96:aec6%11
IPv4 地址 . . . . . : 172.30.62.110
子网掩码 . . . . . : 255.255.255.0
默认网关. . . . . : 172.30.62.1
  
```


十进制转二进制：

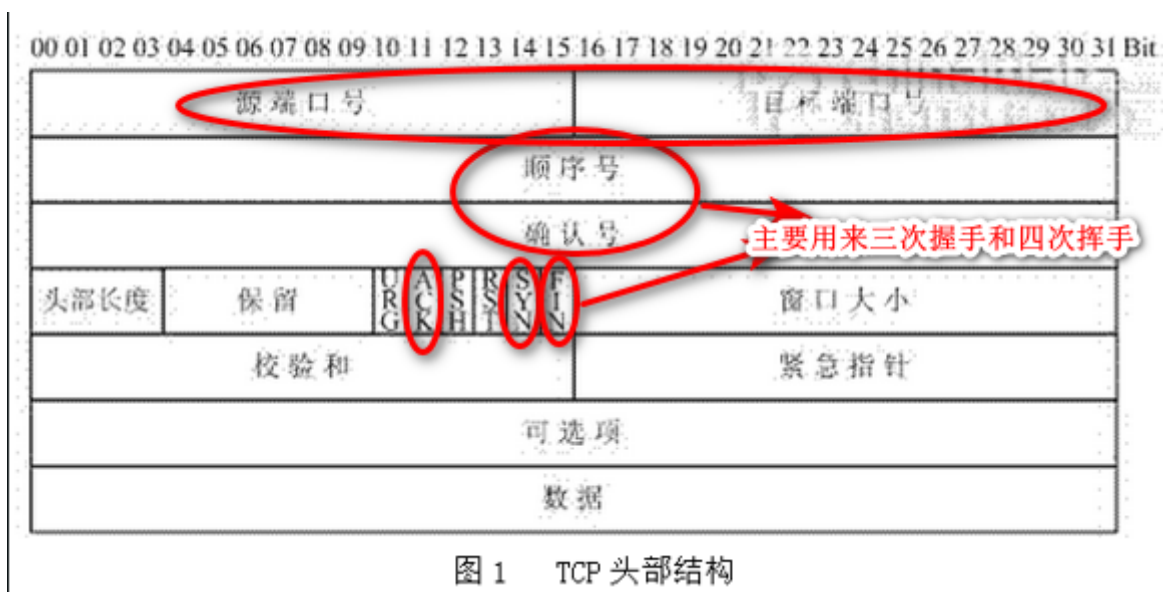
11001101.01100111.10101110.1011110

6.TCP协议

TCP协议工作在第三层：传输层

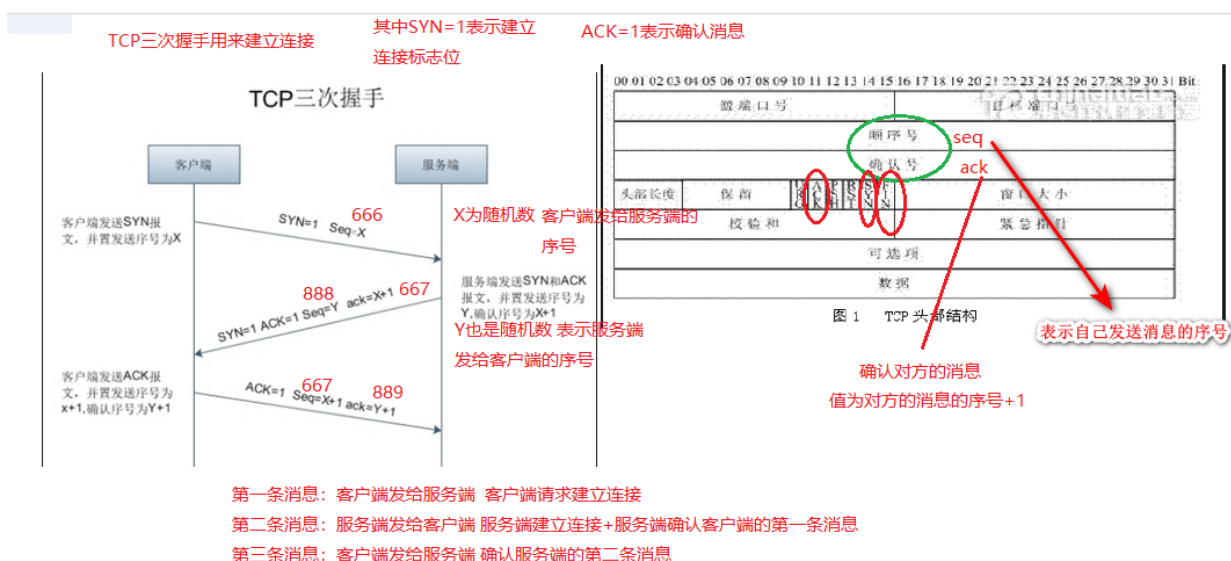
作用：建立端到端得连接 。TCP 协议提供了一种端到端的、基于连接的、可靠的通信服务。

tcp协议头：



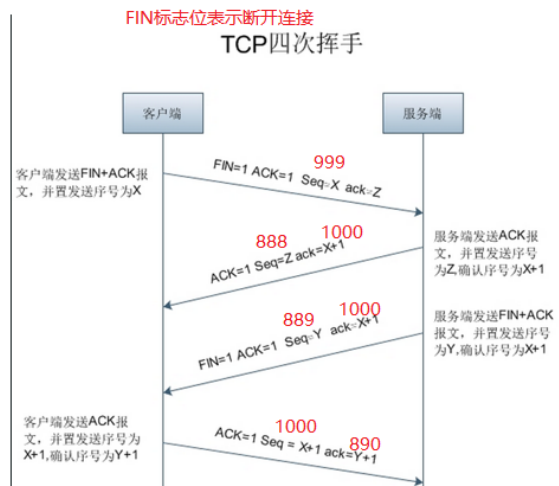
7.TCP协议的三次握手

三次握手用来建立连接，SYN标志位表示建立连接



8.TCP协议的四次挥手

四次挥手用来断开连接，FIN标志表示断开连接



第一次：客户端发给服务端 客户端请求断开连接

第二次：服务端发给客户端 确认客户端的第一条消息+发送剩余的数据给客户端

第三次：服务端发给客户端 服务端请求断开连接+确认客户端的第一条消息

第四条：客户端发给服务端 确认服务端的第三条消息

9.UDP协议

UDP协议也是工作在传输层。

定义：UDP（User Data Protocol，用户数据报协议）它是面向非连接的协议，它不与对方建立连接，而是直接就把数据包发送过去！UDP适用于一次只传送少量数据、对可靠性要求不高的应用环境。比如网络电视，实时游戏等。

TCP协议好比打电话，需要建立连接和断开连接；UDP协议好比发短信，不需要建立连接。

10.tcp协议和udp协议区别

- 1、TCP面向连接（三次握手和四次挥手）；UDP是无连接的，即发送数据之前不需要建立连接
- 2、TCP提供可靠的服务。通过TCP连接传送的数据，无差错，不丢失，不重复，且按序到达；UDP尽最大努力交付，即不保证可靠交付
- 3、TCP面向字节流，实际上是TCP把数据看成一连串无结构的字节流；UDP是面向报文的，UDP没有拥塞控制，因此网络出现拥塞不会使源主机的发送速率降低
- 4、每一条TCP连接只能是点到点的；UDP支持一对一，一对多，多对一和多对多的交互通信

总结：TCP面向连接，可靠的；udp是无连接，不可靠的

11.http协议

http协议工作在应用层协议。

http:hypertext transport protocol 超文本传输协议

定义：规定了浏览器和www服务器之间互相通信的规则，通过Internet传送www文档的数据传送协议。

http协议特点和原理：

HTTP 协议是一种**请求-响应**式的协议。

最显著的特点是客户端发送的每次请求都需要服务器回送响应，在请求结束后，会主动释放连接。从建立连接到关闭连接的过程称为“一次连接”。

原理：

- 1)短连接：用完就释放。
- 2)需要不断向服务器发起连接请求来保持在线状态。
- 3)若服务器长时间无法收到客户端的请求，则认为客户端“下线”
- 4)若客户端长时间无法收到服务器的回复，则认为网络已经断开。

12.fiddler工具

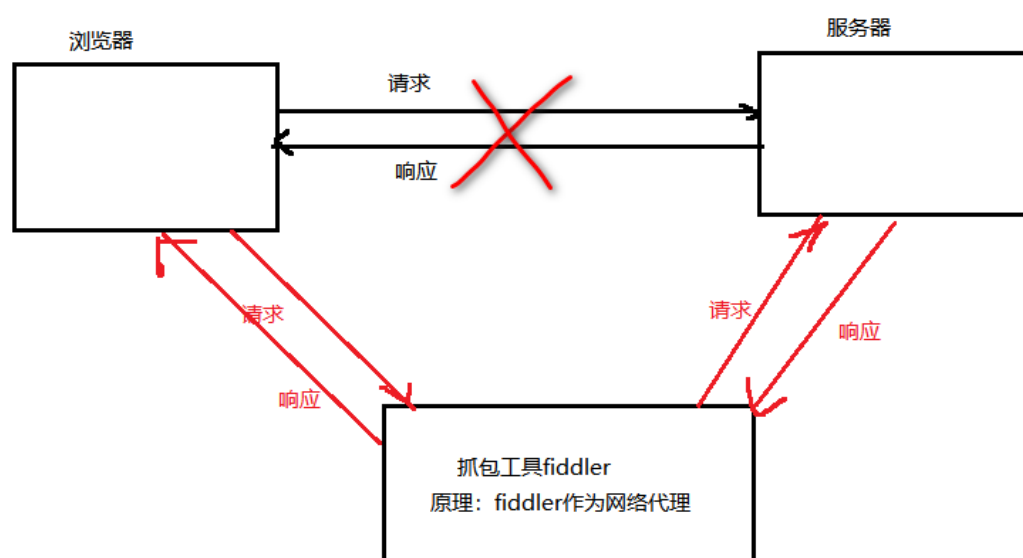
fiddler工具用于进行抓包，可以看到浏览器和服务器的请求和响应的内容。

fiddler:可以用来抓应用层http协议。

wireshark:可以抓到所有层的内容。可以看到传输层，网络层，链路层等数据。

charles:可以用来抓应用层http协议。可以应用在macOS平台上。

fiddler工作原理：



fiddler工具使用：

Progress Idenk Fiddler Web Debugger

File Edit Rules Tools View Help

WinConfig Replay Go Stream Decode | Keep: All sessions

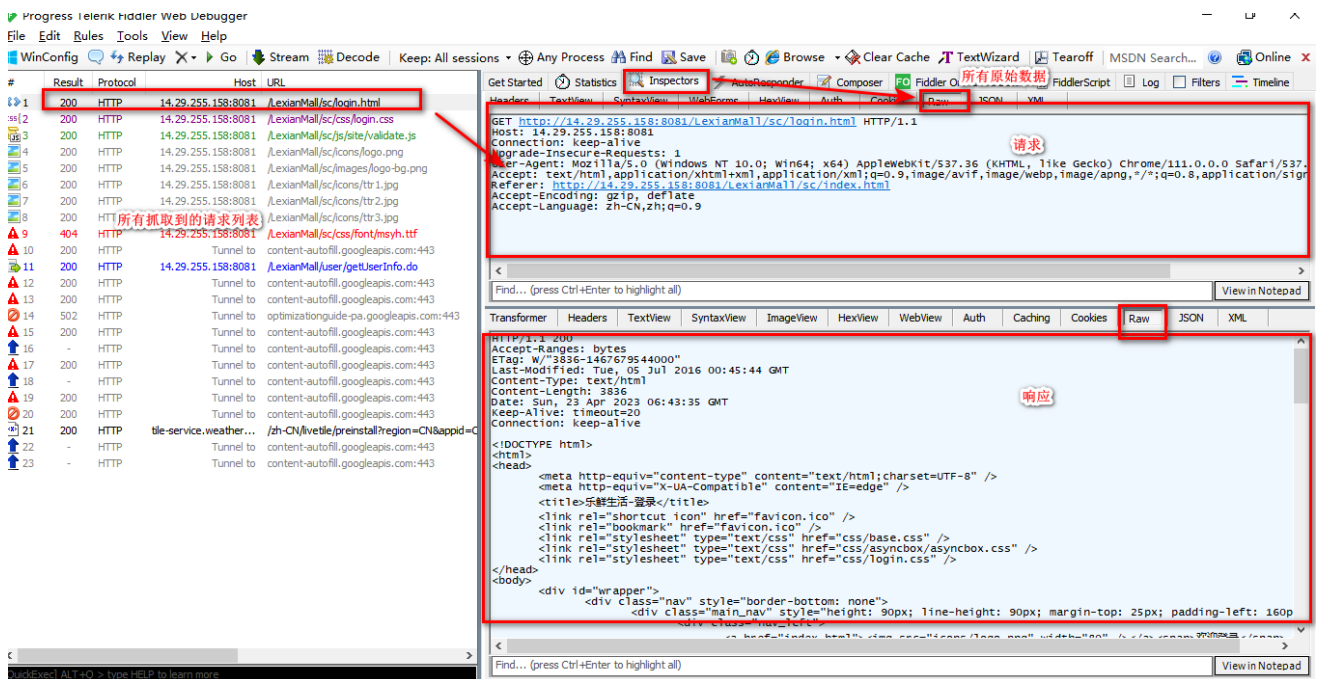
#	Result	Protocol	Host	URL
1	502	HTTPS	www.fiddler2.com	/UpdateCheck.aspx?isBeta=False
2	200	HTTP	fiddler2.com	/content/GetArticles?clientId=F9BF7FF56502
3	200	HTTP	fiddler2.com	/content/GetBanner?clientId=F9BF7FF56502
4	200	HTTP	Tunnel to	www.bing.com:443
5	302	HTTPS	www.bing.com	/RelatedSearch?addfeaturesnoexpansion=re
6	200	HTTP	Tunnel to	cn.bing.com:443
7	200	HTTPS	cn.bing.com	/RelatedSearch?addfeaturesnoexpansion=re
8	502	HTTP	Tunnel to	www.google.com:443
9	200	HTTP	m6815.talk99.cn	/monitor/s?c=e&i=20001818&v=f01966ed52
10	200	HTTP	Tunnel to	flowus.cn:443
11	200	HTTPS	flowus.cn	/api/spaces/43079a95-e0de-47db-a63f-e5ce
12	200	HTTP	m6815.talk99.cn	/monitor/s?c=e&i=20001818&v=f01966ed52
13	502	HTTP	Tunnel to	optimizationguide-pa.googleapis.com:443
14	200	HTTP	m6815.talk99.cn	/monitor/s?c=e&i=20001818&v=f01966ed52
15	200	HTTP	Tunnel to	v10.events.data.microsoft.com:443
16	200	HTTP	Tunnel to	v10.events.data.microsoft.com:443
17	200	HTTP	Tunnel to	v10.events.data.microsoft.com:443
18	200	HTTP	Tunnel to	v10.events.data.microsoft.com:443
19	200	HTTP	m6815.talk99.cn	/monitor/s?c=e&i=20001818&v=f01966ed52
20	200	HTTP	Tunnel to	clientservices.googleapis.com:443
21	304	HTTPS	clientservices.googl...	/chrome-variations/seed?osname=win&chanr
22	200	HTTP	m6815.talk99.cn	/monitor/s?c=e&i=20001818&v=f01966ed52
23	200	HTTP	Tunnel to	v10.events.data.microsoft.com:443

表示处于抓包状态，点一下，capturing会消失，表示关闭抓包状态

QuickEx... ALT+Q > type HELP to learn more

Capturing All Processes 23 CustomRules.js was loaded at: Sun Apr 23 14:

fiddler抓包内容查看：

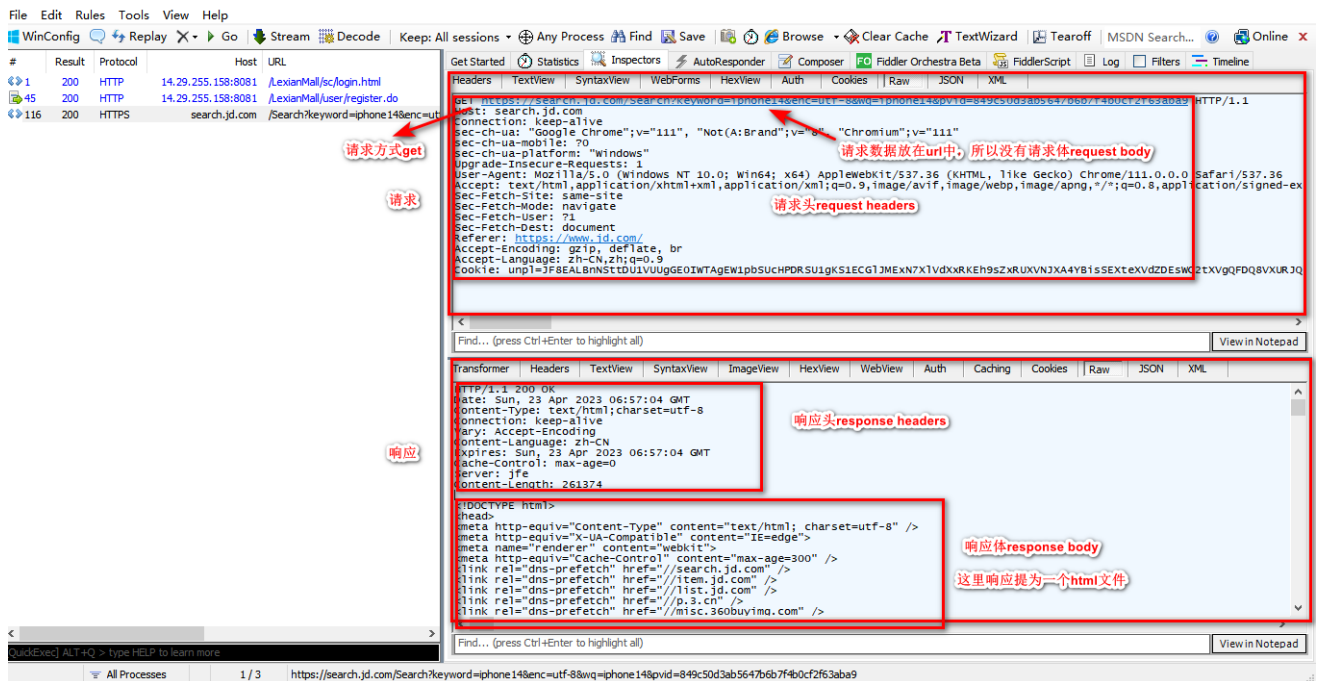


13.http请求

请求：Request，由客户端发送给服务器端

请求分类：GET和POST请求GET请求主要是数据的获取POST请求主要是数据的提交

1) get请求



2) post请求



14.http响应



响应包含 正常的响应 和 异常的响应。

HTTP协议通过响应的状态码来进行定义：1xx, 2xx, 3xx（正常）, 4xx, 5xx（异常）

1xx 信息，服务器收到请求，需要请求者继续执行操作

2xx 成功，操作被成功接收并处理

3xx 重定向，需要进一步的操作以完成请求

4xx 客户端错误，请求包含语法错误或无法完成请求

5xx 服务器错误，服务器在处理请求的过程中发生了错误

15.get和post的区别

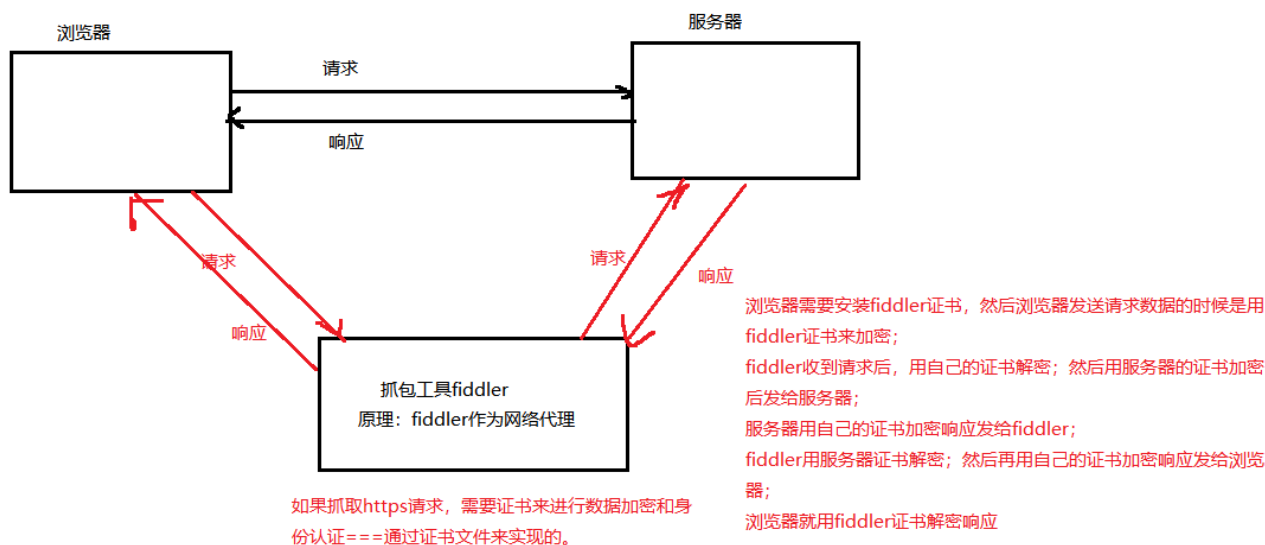
1.get请求方法的请求参数是直接放在url中。

2.Url的长度是浏览器限制，不同浏览器限制url长度稍有不同。比如ie限制url在2kb内，chrome限制8182个字符。不适合传输大数据量请求。

3.Get请求参数暴露在url中，明文传输，安全性较低。

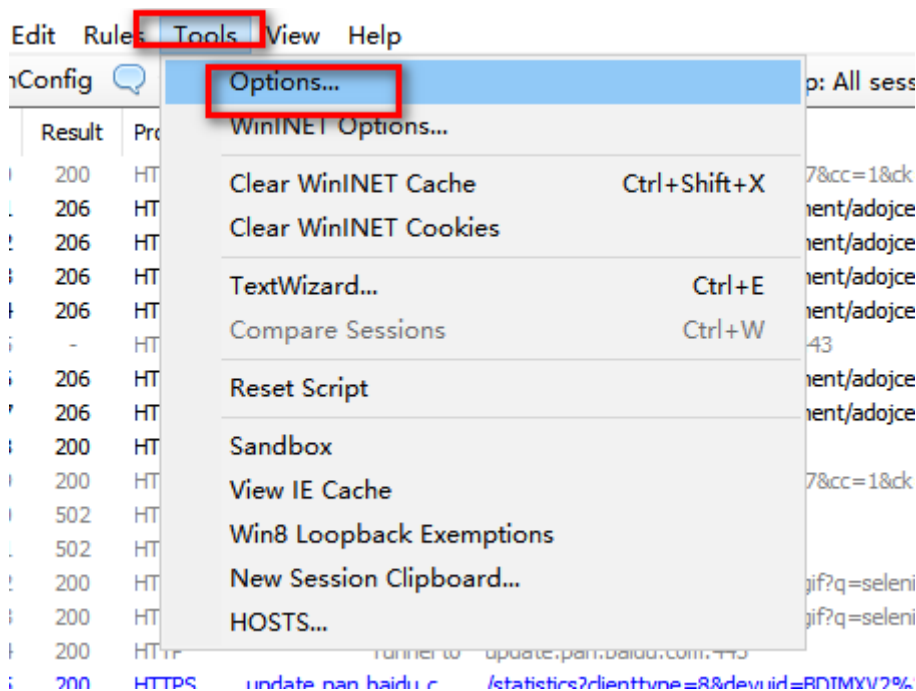
4.post请求参数放在请求body中，不会有长度限制Post参数在请求体中，可以明文传输也可以密文传输，相对安全。

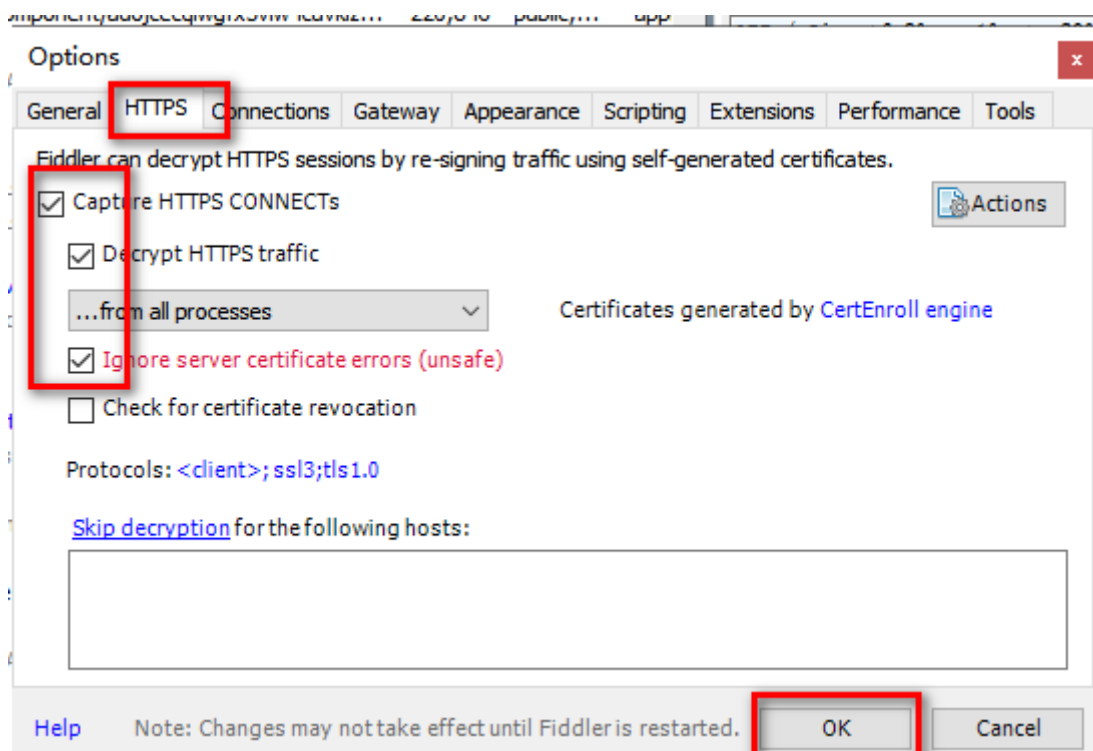
16.https抓包



通过以上图示可以看出，浏览器需要fiddler证书。

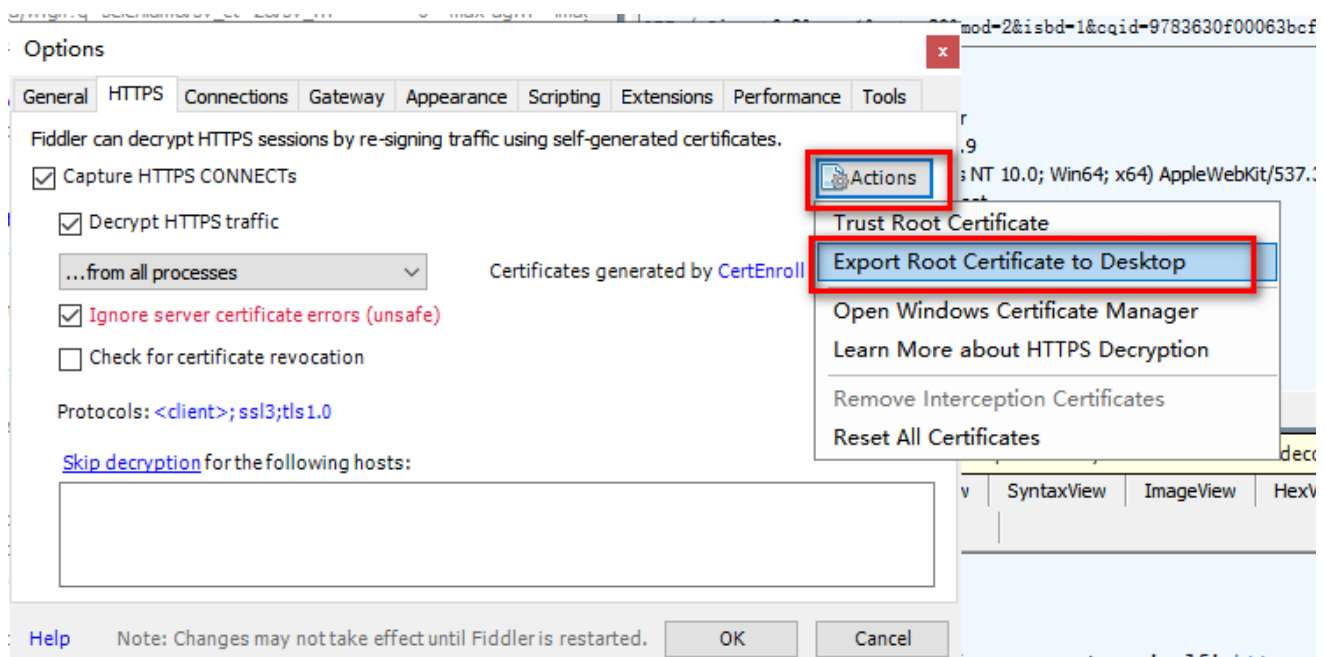
1)fiddler设置





2) 安装fiddler证书

首先到处证书到桌面

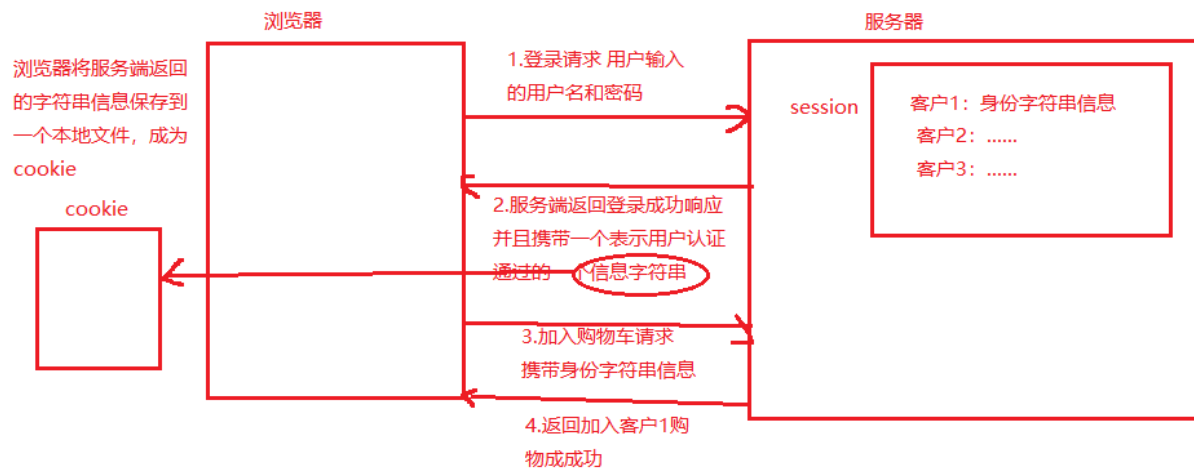


然后安装证书

步骤：双击桌面FiddlerRoot.cer，然后点击安装证书，选择安装到本地计算机，然后选择将证书安装到以下存储，选择受信任的根证书颁发机构，最后确定，完成证书安装。

最后重启fiddler和浏览器，就可以抓到https请求。

17.Session和Cookie



总结: 1.session存储在服务器, 每一个用户都有一个sessionid
2.cookie存储在用户本地计算机, cookie值由服务器生产返回的

