# HTTP概述

# 介绍

## HTTP是Hyper Text Transfer Protocol的缩写，即超文本传输协议。

## HTTP是一种请求/响应式的协议，客户端在与服务器端建立连接后，就可以向服务器端发送请求，这种请求被称作HTTP请求，服务器端接收到请求后会做出响应，称为HTTP响应。

## 这个请求包含一个请求页面的名字和请求页面的信息等

## 返回的响应包含被请求的页面和被请求页面的信息以及服务器的一些信息等

## HTTP协议具有很多特点，具体如下：

### 支持客户端(浏览器就是一种Web客户端)/服务器模式。

### 简单快速：客户端向服务器请求服务时，只需传送请求方式和路径。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。

### 灵活：HTTP允许传输任意类型的数据，正在传输的数据类型由Content-Type加以标记。

### 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，如果后续处理需要前面的信息，则它必须重传。

# HTTP1.0和HTTP1.1

## HTTP1.0：基于HTTP1.0协议的客户端与服务器在交互过程中需要经过建立连接、发送请求信息、回送响应信息、关闭连接四个步骤。发送请求,建一次连接,获得一个web资源，连接断开

## HTTP1.1协议支持持久连接，也就是说在一个TCP连接上可以传送多个HTTP请求和响应，从而减少了建立和关闭连接的消耗和延时。发送请求，创建一次连接，获得多个web资源,连接断开 。

# HTTP消息

## 当用户在浏览器中访问某个URL地址、单击网页的某个超链接或者提交网页上的form表单时，浏览器都会向服务器发送请求数据，即HTTP请求消息。

## 服务器接收到请求数据后，会将处理后的数据回送给客户端，即HTTP响应消息。

# HTTP请求消息 与 HTTP请求行

## HTTP请求行位于请求消息的第一行，它包括三个部分，分别是请求方式、资源路径以及所使用的HTTP协议/版本。

## 例如：POST /examples/LoginServlet HTTP/1.1

## HTTP的八种请求方式中，最常用的就是GET和POST方式

### GET方式

#### 当用户在浏览器地址栏中直接输入某个URL地址或者单击网页上一个超链接时，或者将网页上的form表单的method属性设置为“GET”或者不设置method属性(默认值是GET)，用户提交表单时，浏览器都将使用GET方式发送请求。

#### 如果浏览器请求的URL中有参数部分，在浏览器生成的请求消息中，参数部分将附加在请求行中的资源路径后面。例如：http://www.itcast.cn/javaForum?name=lee&psd=hnxy

#### 当浏览器向服务器发送请求消息时，上述URL中的参数部分会附加在要访问的URI资源后面，具体如下所示：

**GET /javaForum?name=lee&psd=hnxy HTTP/1.1**

#### 注意的是，使用GET方式传送的数据量有限，最多不能超过1KB。

### POST方式

#### 如果网页上form表单的method属性设置为“POST”，当用户提交表单时，浏览器将使用POST方式提交表单内容，并把各个表单元素及数据作为HTTP消息的实体内容发送给服务器，而不是作为URI地址的参数传递。

#### 在实际开发中，通常都会使用POST方式发送请求，其原因主要有两个，具体如下：

##### POST传输数据大小无限制。由于GET请求方式是通过请求参数传递数据的，因此最多可传递1KB的数据。而POST请求方式是通过实体内容传递数据的，因此可以传递数据的大小没有限制。

##### POST比GET请求方式更安全。由于GET请求方式的参数信息都会在URL地址栏明文显示，而POST请求方式传递的参数隐藏在实体内容中，用户是看不到的，因此，POST比GET请求方式更安全。

# HTTP请求消息头

## 在HTTP请求消息中，请求行之后，便是若干请求消息头。

## 请求消息头主要用于向服务器端传递附加消息

## 例如：Content-Type: application/x-www-form-urlencoded

## 例如：User-Agent: Mozilla/5.0 ...

## 例如：Host: www.baidu.com

## 例如：Accept-Language: zh-CN

# 消息体

## 通常情况下,只有post请求方式才会使用到请求体,请求体中都是用户表单提交的数据 ，每一项数据都使用键值对 (k=v),多组值使用&相连。

## 例如：uid=admin&password=123456

# HTTP响应消息与HTTP响应状态行

## HTTP响应状态行位于响应消息的第一行，它包括三个部分，分别是HTTP协议版本、一个表示成功或错误的整数代码（状态码）和对状态码进行描述的文本信息。

## 例如：HTTP/1.1 200 OK

# 状态码

## 1xx：表示请求已接收，需要继续处理

## 2xx：表示请求已成功被服务器接收、理解、并接受

## 3xx：为完成请求，客户端需进一步细化请求

## 4xx：客户端的请求有错误

## 5xx：服务器端出现错误

## 200 - OK 客户端的请求成功，响应消息返回正常的请求结果

## 301 - Moved Permanently 永久移动

## 302 - 请求重定向

## 304 - 请求资源没有改变,访问本地缓存

## 400 - Bad Request 请求无效

## 401 - Unauthorized 未经授权

## 403 - Forbidden 禁止

## 404 - Not Found 找不到，表示服务器上不存在客户端请求的资源。通常是 用户路径编写错误,也可能是服务器资源己删除。

## 405 - Method Not Allowed 不允许此请求方式

## 500 - Internal Server Error 内部服务器错误

## 503 - Service Unavailable 服务不可用

## HTTP响应消息头

## 在HTTP响应消息中，第一行为响应状态行，紧接着的是若干响应消息头，服务器端通过响应消息头向客户端传递附加信息，包括服务程序名、被请求资源需要的认证方式、客户端请求资源的最后修改时间、重定向地址等信息

## 例如：Content-Type: text/html; charset=utf-8

## 例如：Content-Length: 8058

## 例如：Date: hu, 08 Jan 2015 06:18:40 GMT

## 例如：Server: Apache-Coyote/1.1

## 消息体

## 响应体,就是服务器发送给浏览器的正文 。

## 例如：<html> ... </html>

# HTTP其他头字段与通用头字段

## 在HTTP消息中，有些头字段既适用于请求消息也适用于响应消息，这样的字段被称为通用头字段。

## Cache-Control

### 在一个Cache-Control头字段中可以设置多个值

## Connection

### Connection头字段用于指定处理完本次请求/响应后，客户端和服务器端是否还要继续保持连接。

## Date

### Date头字段用于表示HTTP消息产生的当前时间

## Pragma

### Pragma头字段主要在HTTP1.0中通知代理服务器和客户端如何使用缓存页面，它的值只能固定设置为no-cache

## Transfer-Encoding

### Transfer-Encoding头字段指定响应消息的实体内容采用哪种传输编码方式，目前标准设置值只有chunked。

## Trailer

### 一些头字段可以放置在整个HTTP消息的尾部，也就是可以在实体内容部分之后放置头字段信息。对于放置在尾部的头字段，需要在消息头中使用Trailer字段说明

## Upgrade

### Upgrade头字段用在客户端，用于指定客户端想要从当前协议切换的新的通信协议。

## Via

### Via头字段用于指定HTTP消息所途经的代理服务器所使用的协议和主机名称。

## Warning

### Warning头字段主要用于说明其它头字段和状态码不能说明的一些附加警告信息。

# 实体头字段

## Allow

### Allow头字段指定了请求资源所支持的请求方式（如GET、POST等），用于通知客户端应该严格按照指定的方式请求资源。

## Content-Language

### Content-Language用于指定返回网页文档的国家语言类型，其设置值是zh-cn、en-us、ja等国家语言的标准名称。

## Content-Length

### Content-Length头字段用于表示实体内容的长度(字节数)。

## Content-Location

### Content-Location头字段用于指定响应消息中实体内容的实际位置路径。

## Content-Range

### Content-Range头字段用于指定服务器返回的部分实体内容的位置信息。

## Content-MD5

### Content-MD5头字段用于提供对实体内容的完整性检查，它的值是对实体内容MD5数字摘要后再进行BASE64编码的结果。

## Content-Type

### Content-Type用于指出实体内容的MIME类型。

### 用于定义网络文件的类型以及网页字符的编码。用于决定浏览器以什么形式、什么编码读取这个文件。

## Content-Encoding

### Content-Encoding头字段用于指定实体内容的压缩编码方式。

## Expires

### Expires头字段用于指定当前文档的过期时间，浏览器在这个时间以后不能再继续使用本地缓存，而需要向服务器发出新的访问请求。

## Last-Modified

### Last-Modified头字段用于指定文档最后的更改时间，设置值为GMT格式的时间

# 常见的MIME类型

## 超文本标记语言hext/html

## 普通文本text/plain

## Microsoft Wordapplication/msword

## PDF文档application/pdf

## AVI文件video/x-msvideo