Android网络编程

HTTP

HTTP简介

HTTP是一个属于应用层的面向对象的协议,由于其简捷、快速的方式,适用于分布式超媒体信息系统。它于1990年提出,经过几年的使用与发展,得到不断地完善和扩展。

HTTP协议的主要特点

- 1. 支持C/S (客户/服务器) 模式。
- 2. **简单快速**:客户向服务器请求服务时,**只需传送请求方法和路径**。 **请求方法**常用的有**GET、HEAD、POST**,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**。
- 3. **灵活**: HTTP允许**传输任意类型**的数据对象。正在传输的类型由Content-Type加以标记。
- 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**。
- 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 URL 的格式

http://host[":"port][abs_path]

http表示要通过HTTP协议来定位网络资源;

host表示合法的Internet主机域名或者IP地址;

port指定一个端口号, 为空则使用默认端口80;

abs path指定请求资源的URI(Web上任意的可用资源)。

HTTP有两种报文分别是请求报文和响应报文,让我们先来看看请求报文。

HTTP的请求报文先来看看请求报文的一般格式:



通常来说一个HTTP请求报文由**请求行、请求报头、空行、请求数据**4个部分组成。

① 请求行

请求行由**请求方法**、URL字段、HTTP协议的版本组成,格式如下:

Method Request-URI HTTP-Version CRLF

Method表示请求方法;

Request-URI是一个统一资源标识符;

HTTP-Version表示请求的HTTP协议版本;

CRLF表示回车和换行(除了作为结尾的CRLF外,不允许出现单独的CR或LF字符)。

HTTP请求方法

HTTP请求方法有8种,分别是GET、POST、DELETE、PUT、HEAD、TRACE、CONNECT、OPTIONS。

其中PUT、DELETE、POST、GET分别对应着增删改查,对于移动开发最常用的就是POST和GET了。

- 1. GET:请求获取Request-URI所标识的资源。
- 2. POST:在Request-URI所标识的资源后附加新的数据。
- 3. HEAD: 请求获取由Request-URI所标识的资源的响应消息报头。
- 4. PUT: 请求服务器存储一个资源,并用Request-URI作为其标识。
- 5. DELETE: 请求服务器删除Request-URI所标识的资源。
- 6. TRACE: 请求服务器回送收到的请求信息, 主要用于测试或诊断。
- 7. 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。
- 8. OPTIONS: 请求查询服务器的性能,或者查询与资源相关的选项和需求。

② 请求报头

在请求行之后会有**0个或者多个请求报头,每个请求报头**都包含**一个名字和一个值**,它们之间用":"分 割。

关于请求报头,会在后面的消息报头一节做统一的解释。

③ 空行

请求头部会以一个空行,发送回车符和换行符,通知服务器以下不会有请求头。

④ 请求数据

请求数据不在GET方法中使用,而是**在POST方法中使用**。

POST方法适用于需要客户填写表单的场合,与请求数据相关的最常用的请求头是Content-Type和Content-Length。

HTTP的响应报文

先来看看响应报文的一般格式:

版本	空格	状态	码	空格	原因短语	回车符	换行符	
头部域名称			:	头部域值		回车符	换行符	
头部域名称			:	头部域值		回车符	换行符	
回车符	换行符	F						
响应正文								

HTTP的响应报文由**状态行、消息报头、空行、响应正文**组成。响应报头后面会讲到,响应正文是服务器返回的资源的内容,先来看看状态行。

① 状态行

状态行格式如下:

HTTP-Version Status-Code Reason-Phrase CRLF

HTTP-Version表示服务器HTTP协议的版本; Status-Code表示<mark>服务器发回的响应状态代码</mark>; Reason-Phrase表示<mark>状态代码的文本描述</mark>。

状态代码

状态代码有三位数字组成,第一个数字定义了响应的类别,且有五种可能取值:

• 100~199: 指示信息,表示请求已接收,继续处理

• 200~299: 请求成功,表示请求已被成功接收、理解、接受

• 300~399: 重定向, 要完成请求必须进行更进一步的操作

• 400~499: 客户端错误,请求有语法错误或请求无法实现

• 500~599: 服务器端错误,服务器未能实现合法的请求

常见的状态码如下:

• 200 OK: 客户端请求成功

• 400 Bad Request:客户端请求有语法错误,不能被服务器所理解

• 401 Unauthorized:请求未经授权,这个状态代码必须和WWW-Authenticate报头域一起使用

• 403 Forbidden: 服务器收到请求, 但是拒绝提供服务

• 500 Internal Server Error: 服务器发生不可预期的错误

• 503 Server Unavailable:服务器当前不能处理客户端的请求,一段时间后可能恢复正常

② 消息报头

③ 空行

④ 响应正文

HTTP的消息报头

消息报头分为**通用报头、请求报头、响应报头、实体报头**等。 消息头由**键值对**组成,每行一对,**关键字和值用英文冒号":"分隔。**

通用报头

既可以出现在**请求**报头,**也可**以出现在**响应**报头中

• Date: 表示消息产生的日期和时间

• Connection: **允许发送指定连接的选项** 例如指定连接是连续的,或者指定"close"选项,通知服务器,在响应完成后,关闭连接

• Cache-Control:用于**指定缓存指令**,缓存指令是单向的(响应中出现的缓存指令在请求中未必会出现),且是独立的(一个消息的缓存指令不会影响另一个消息处理的缓存机制)

请求报头

请求报头通知服务器关于客户端请求的信息,典型的请求头有:

• Host: 请求的主机名,允许多个域名同处一个IP地址,即虚拟主机

• User-Agent: 发送请求的浏览器类型、操作系统等信息

• Accept: 客户端可识别的内容类型列表,用于指定客户端接收那些类型的信息

• Accept-Encoding: 客户端可识别的数据编码

• Accept-Language: 表示浏览器所支持的语言类型

• **Connection**: 允许客户端和服务器**指定与请求/响应连接有关的选项**,例如这是为Keep-Alive则表示保持连接。

• Transfer-Encoding: 告知接收端为了保证报文的可靠传输,对报文采用了什么编码方式。

响应报头

用于服务器传递自身信息的响应, 常见的响应报头:

• Location: 用于**重定向**接受者到一个新的位置,常用在更换域名的时候

• Server:包含可服务器用来处理请求的系统信息,与User-Agent请求报头是相对应的

实体报头

实体报头用来定义被传送资源的信息,**既可**以用于**请求也可**用于**响应**。 请求和响应消息都可以传送一个实体,常见的实体报头为:

• Content-Type: 发送给接收者的实体正文的媒体类型

• Content-Lenght: 实体正文的长度

• Content-Language: 描述资源所用的自然语言,没有设置则该选项则认为实体内容将提供给所有的语言阅读

• Content-Encoding: 实体报头被用作媒体类型的修饰符,它的值指示了已经被应用到实体正文的附加内容的编码,因而要获得Content-Type报头域中所引用的媒体类型,必须采用相应的解码机制。

• Last-Modified: 实体报头用于指示资源的最后修改日期和时间

• Expires: 实体报头给出响应过期的日期和时间