复习：

HTTP协议是TCP/IP协议族中应用层的协议之一：

请求消息(Message)

响应消息(Message)

消息的格式：

1. 起始行CRLF
2. 消息头CRLF
3. 消息体

请求消息的格式：

GET /index.php?a&b=c HTTP/1.1

Host: [www.tedu.cn](http://www.tedu.cn)

User-Agent:

Client-IP:

Accept: text/html,\*/\*

Cookie:

Referer: <http://127.0.0.1/1.html>

Connection: keep-alive

Date: ....GMT

Content-Type: text/plain、multipart/form-data

Content-Length:

username=tom&upwed=123456

响应消息的格式：

HTTP/1.1 200 OK

Server：Apache Httpd2.2

Content-Type: image/png

Cotnent-Length: 7855

Location:

Content-Type:

Refresh: 3;url=””

010101101010010101101010100101010

1xx:

2xx:

3xx: Location: <http://www.oracle.com>

4xx:

5xx:

1. 常见是响应消息头部

通用头部：Connection / Date / Cache-Control / Pragma

响应消息专用头部：Server / Age /

响应主体描述头部：Expires / Content-Type / Content-Length / Last-Modified / Location

扩展头部：

2.(面试题)对于前段工程师来说，Web优化从哪些角度考虑：

(1)数据库结构优化

(2)数据库访问优化

(3)Web服务器端优化——启用服务器端数据缓存

(4)HTTP请求和响应优化

1)减少DNS次数

2)使用持久连接

3)减少请求次数，设置资源的合理的Expires时间、缓存时长

4)缩短响应处理时间——使用数据库数据缓存、动态页面静态化、AJAX

5)减少响应数据的长度——启用数据压缩

(5)HTML优化

(6)CSS优化

(7)JavaScript优化

<html>

<head>

<script src=”http://119.206.2.48/code/jquery1.7.js”/>

</head>

</html>

2.与缓存控制相关的响应头

Cache-Control: max-age=3600 //告诉客户端缓存此次响应数据的时长

Cache-Control: max-age=0

Cache-Control: no-cache

Cache-Control: must-revalidate //告诉客户端在提供缓存之前必须跟服务器重新验证一次

Pragma: no-cache

Expires: Fri, 15 March 2015 12:12:22 GMT //指定响应数据的确切过期时间

Expires: 过去的时间/0 //告诉客户端不要缓存响应数据

项目中控制响应消息头的方法：

1. 修改Web服务器的配置文件，会影响到其中所有的虚拟主机的所有资源
2. 前端工程师：可以修改单个HTML文件的<meta/>标签来控制响应头

<head>

<meta http-equiv=”Cache-Control” content=”no-cache” />

<meta http-equiv=”Expires” content=”0”/>

<meta http-equiv=”Pragma” content=”no-cache”/>

<meta http-equiv=”Content-Type” content=”text/html;charset=UTF-8”/>

</head>

1. 后端工程师：可以直接使用代码控制某个资源的响应头

PHP: <?php header(‘Cache-Control: no-cache’); ?>

JSP: <% response.setHeader(‘Expires’, ‘0’); %>

这种控制HTTP响应头的方式优先级要高于使用<meta http-equiv=””/>指定的值

3.Web应用中的状态保持

状态保持：服务器可以得知当前客户端之前曾经的访问历史

HTTP协议本身是“Stateless”——无状态的。

|  |
| --- |
| HTTP/1.1是一个基于请求/响应模型的、无状态的、面向持久连接的一个TCP/IP协议栈中应用层协议。 |

为了实现“状态保持”功能，有三种技术可以使用：

1. Cookie：把数据存储于客户端磁盘中——是不可靠的状态保持
2. HTML5 Local Storage：把数据存储于客户端磁盘中
3. Session：把数据存储在服务器端——是可靠的状态保持技术

4.HTTP和HTTPS

HTTP：80端口，请求和响应消息都是明文传输，可能被窃听、篡改、伪造

HTTPS： HTTP over SSL/TLS，443端口，请求和响应消息都是加密传输的

5.Web项目的国际化

i18n： internationalization，Web项目的国际化，就是针对不同地区的客户端提供不同语言版本的内容。

Web项目实现国际化有两种方案：

(1)伪国际化：页面中让用户选择语言，根据用户选择跳转到不同版本的HTML页面

(2)真国际化：同一个可以根据客户端浏览器首选语言的不同，而给出不同的文字。

6.常见的字符集及对应的编码方案

|  |  |  |
| --- | --- | --- |
| **字符编码名称** | **字符集包括哪些字符** | **编码规则** |
| ASCII | 大小写英文字母、数字、标点符号 | 8位的数字 |
| GB2312 | 常用的简体汉字 | 16bit |
| GBK | 常用的简体/繁体汉字 | 16bit |
| GB18030 | 常用/不常用的简/繁体汉字 | 8/16/32bit |
| Big5 | 港澳台地区使用繁体汉字 | 16bit |
| Latin-1  别名ISO8859-1 | 西欧字符集 | 8bit |
| UTF-8 | Unicode字符集(万国码)，对世界上主要的文字的常用符号都囊括 | 8/16/24bit |
| UTF-16 | Unicode字符集 |  |
| UTF-32 | Unicode字符集 |  |

与字符编码方案相关HTTP响应头：

Content-Type: text/html;charset=UTF-8

Content-Language: zh-CN 指定响应主体内容的受众语言

与字符编码方案相关HTTP请求头：

Accept-Charset: UTF-8, GBK

Accept-Language：zh-CN 客户端告诉服务器自己首选的语言