核心编程第3天

[学习目标 2](#_Toc521345226)

[昨日回顾 2](#_Toc521345227)

[PHP数据分页 3](#_Toc521345228)

[1、创建连接数据库的公共文件conn.php 3](#_Toc521345229)

[2、显示学生信息列表list.php 4](#_Toc521345230)

[3、分页原理 4](#_Toc521345231)

[4、仿百度分页 5](#_Toc521345232)

[（1）读取分页数据 5](#_Toc521345233)

[（2）仿百度分页 6](#_Toc521345234)

[HTTP协议概述 7](#_Toc521345235)

[1、B/S网络结构 7](#_Toc521345236)

[2、HTTP协议概述 7](#_Toc521345237)

[3、HTTP协议的特点 7](#_Toc521345238)

[HTTP协议之URL 8](#_Toc521345239)

[1、什么是URL？ 8](#_Toc521345240)

[2、URL各部分含义 8](#_Toc521345241)

[HTTPWatch抓包工具使用 8](#_Toc521345242)

[1、HTTPWatch描述 8](#_Toc521345243)

[2、HTTPWatch的安装和授权 9](#_Toc521345244)

[3、HTTPWatch的简单使用 9](#_Toc521345245)

[4、Firefox抓包工具(firebug) 9](#_Toc521345246)

[HTTP协议之请求 10](#_Toc521345247)

[1、HTTP请求的构成 10](#_Toc521345248)

[2、HTTP请求行的格式 10](#_Toc521345249)

[3、HTTP请求头含义(部分) 11](#_Toc521345250)

[4、HTTP请求正文 11](#_Toc521345251)

[5、请问：访问以下文件,http需要发送几次请求？ 12](#_Toc521345252)

[HTTP协议之响应 13](#_Toc521345253)

[1、HTTP响应的构成 13](#_Toc521345254)

[2、HTTP响应行的格式 13](#_Toc521345255)

[3、HTTP响应状态码(重点) 13](#_Toc521345256)

[4、302状态码演示 14](#_Toc521345257)

[5、304状态码演示 14](#_Toc521345258)

[6、403状态码演示 15](#_Toc521345259)

[7、404状态码演示 16](#_Toc521345260)

[8、HTTP响应头含义(部分) 16](#_Toc521345261)

[9、刷新并跳转 16](#_Toc521345262)

[综合案例：实现视频文件下载 17](#_Toc521345263)

[1、下载的静态页面download.html 17](#_Toc521345264)

[2、下载程序处理download.php 18](#_Toc521345265)

# 学习目标

* PHP数据分页
* HTTP协议之URL
* HTTP协议之请求
* HTTP协议之响应

# 昨日回顾



# PHP数据分页

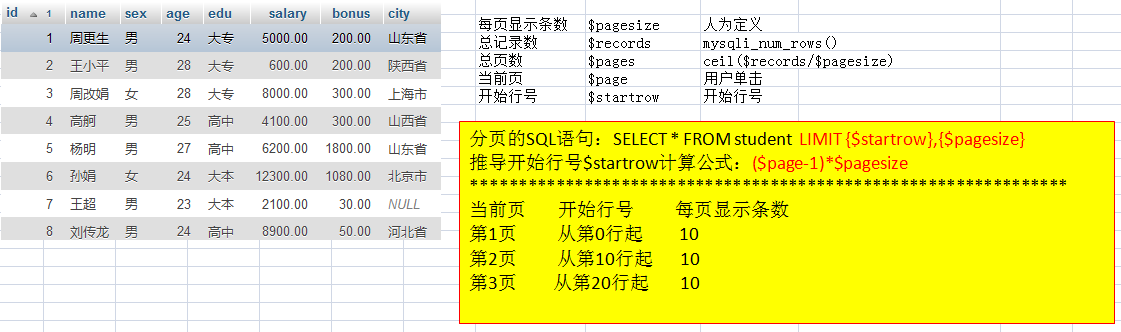
## 1、创建连接数据库的公共文件conn.php



## 2、显示学生信息列表list.php



## 3、分页原理

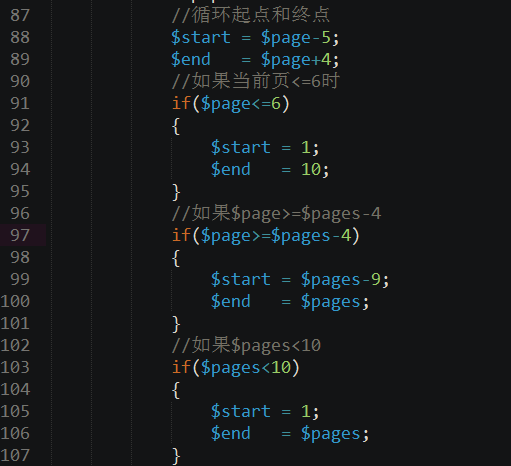


## 4、仿百度分页

### （1）读取分页数据



### （2）仿百度分页



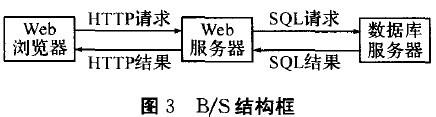




# HTTP协议概述

## 1、B/S网络结构

B/S结构（Browser/Server , 浏览器/服务器模式），是WEB兴起后的一种网络结构模式，**WEB浏览器是客户端最主要的应用软件**。**这种模式统一了客户端，将系统功能实现的核心部分集中到了服务器上**，简化了系统的开发、维护和使用，同时也大大简化了客户端的电脑载荷，减轻了系统维护与升级的成本和工作量。



## 2、HTTP协议概述

* **超文本传输协议**（HTTP，HyperText Transfer Protocol)是互联网上应用最为广泛的一种网络协议。所有的WWW文件都必须遵守这个标准。
* HTTP是一个客户端与服务器端请求和应答的标准。
* HTTP是网络传输的一种标准、规则。

## 3、HTTP协议的特点

* **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、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。
* **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。
* **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。
* **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。缺少记忆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协议之URL

## 1、什么是URL？

* URL(Uniform Resource Locator)统一资源定位符。
* URL是对从互联网上得到的资源的位置和访问方法的一种简洁的表示方法。
* 格式：protocol://hostname[:port]/directory/filename?name=value#anchor
* 举例：http://www.itcast.cn:80/include/itcast.php?p=5&u=root#top

## 2、URL各部分含义

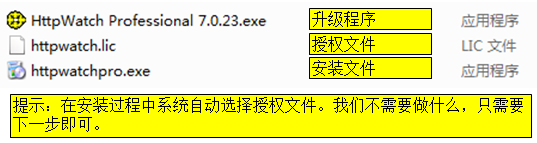
* protocol：它告诉浏览器打开文件使用的协议。如：http、https、ftp、telnet、mailto、file等
* hostname：指定服务器的主机名称，也可以是IP地址；
* port：指定访问的端口号，默认为80；安全协议的默认端口443，FTP默认端口21。
* directory：指定访问的资源目录名称；
* filename：指定访问的资源名称；
* ?name=value：指定访问资源时，附带的参数部分；
* #anchor：指定访问资源时的锚点名称。

# HTTPWatch抓包工具使用

## 1、HTTPWatch描述

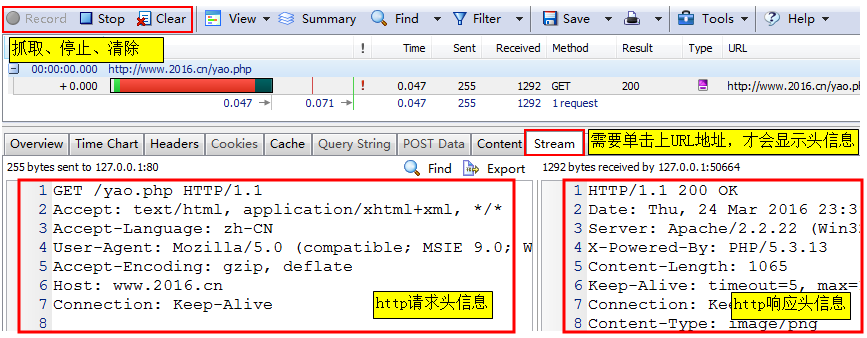
* HttpWatch 网页数据分析工具，可分析打开网页时网页上每个元素所用的时间以及具体的交互记录，包括Cookies、消息头、字符查询等信息，是分析网站服务器性能的必备工具。
* HttpWatch 用来分析、优化网页结构非常不错，支持HTTPS及分析报告输出为XML、CSV等格式。HttpWatch 支持IE6至IE9的浏览器，也支持火狐浏览器2.0至7.0版本。

## 2、HTTPWatch的安装和授权



## 3、HTTPWatch的简单使用

* HttpWatch是当成IE浏览器的一个插件来使用的；
* 依次打开“查看”—“浏览器栏”—“HttpWatch”，启动httpwatch；
* 快捷键：shift+F2



## 4、Firefox抓包工具(firebug)

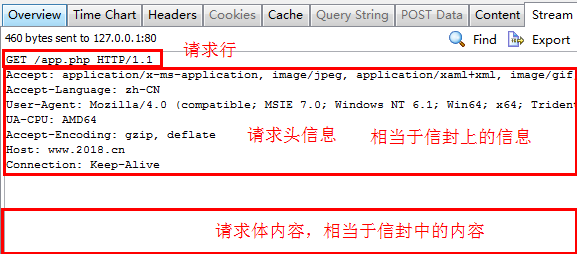
需要安装Firefox的抓包工具firebug。



# HTTP协议之请求

## 1、HTTP请求的构成

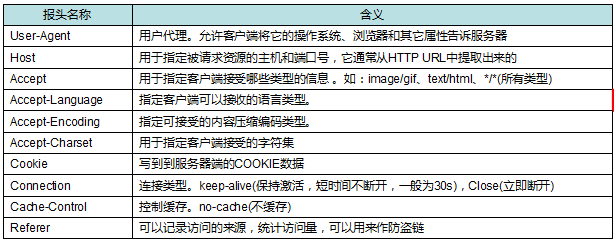
http请求由三部分组成，分别是：**请求行**、**请求头**、**请求正文** 。



## 2、HTTP请求行的格式

* 格式：**Method Request-URI HTTP-Version**
* 参数：
  + Method：请求方法，必须大写。
  + Request-URI：是一个统一资源标识符(URI)，不含主机部分。
  + HTTP-Version：表示请求的HTTP协议版本。HTTP/1.0短连接，HTTP/1.1长连接。
* 什么是长连接？数据传输完成，保持连接通道不断开，等待同域名下继续使用该通道传输数据。

## 3、HTTP请求头含义(部分)

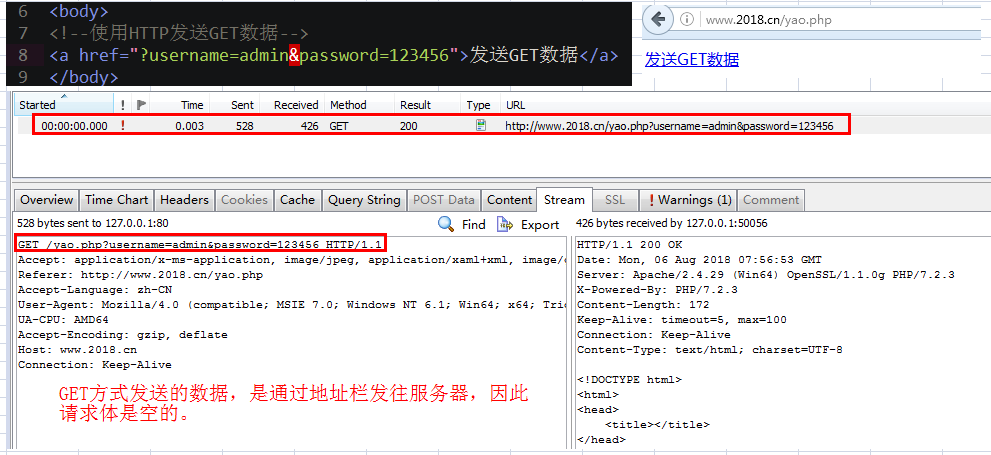


**提示：发送不同请求，请求头信息不太一样。**

## 4、HTTP请求正文

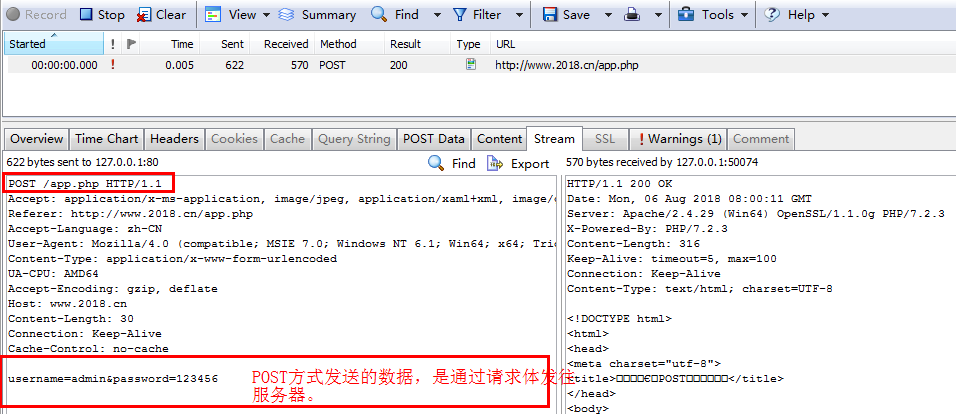
* GET方式，没有消息体，数据附在URI之后传递到服务器；
* POST方式，有消息体，数据放在消息体中传递到服务器；
* 消息体和消息头之间有一空行，不能省略；

**使用HTTP协议发送GET数据**

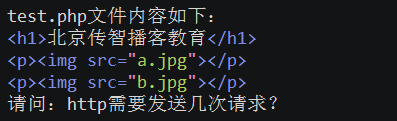


**使用HTTP协议发送POST数据**

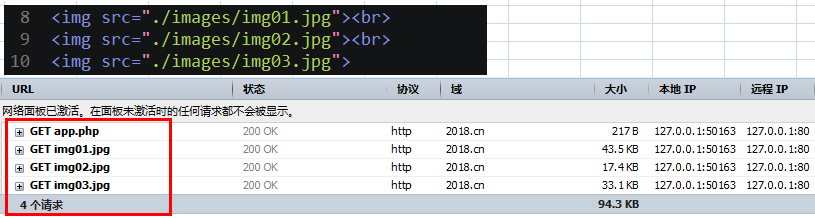




## 5、请问：访问以下文件,http需要发送几次请求？



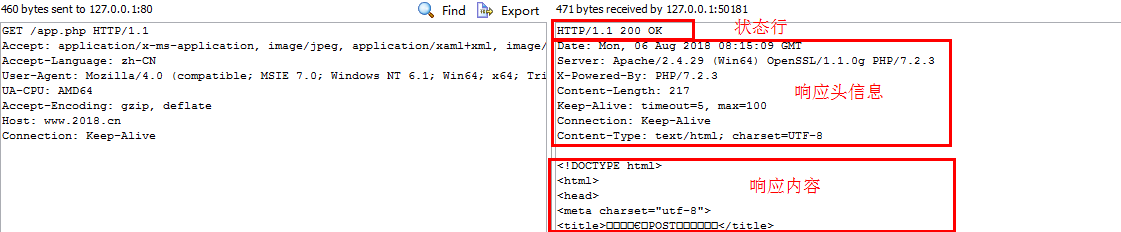
**<img>、<link>、<script>、<frame>都是自动向服务器发请求。**



# HTTP协议之响应

## 1、HTTP响应的构成

HTTP响应也是由三个部分组成，分别是：**状态行**、**响应头**、**响应正文；**



## 2、HTTP响应行的格式

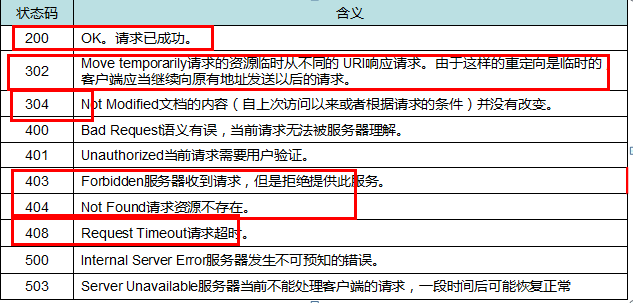
* 格式：**HTTP-Version Status-Code Reason-Phrase**
* 参数：
  + HTTP-Version表示服务器HTTP协议的版本；
  + Status-Code表示服务器发回的响应状态代码；
  + Reason-Phrase表示状态代码的文本描述。

## 3、HTTP响应状态码(重点)

**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**：

* 1xx：指示信息--表示请求已接收，继续处理；
* 2xx：成功--表示请求已被成功接收、理解、接受；
* 3xx：重定向--要完成请求必须进行更进一步的操作；
* 4xx：客户端错误--请求有语法错误或请求无法实现；
* 5xx：服务器端错误--服务器未能实现合法的请求。

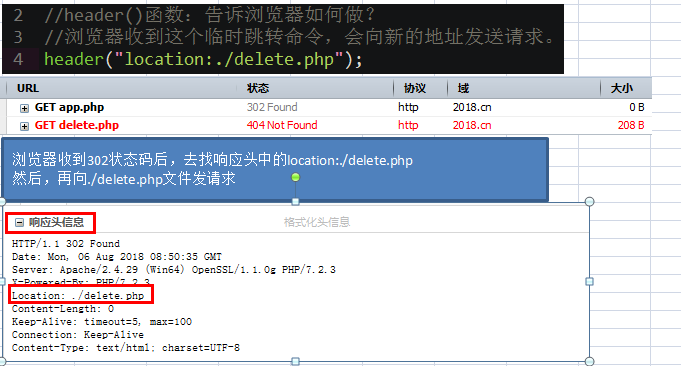
**常见HTTP响应状态码含义：**



## 4、302状态码演示

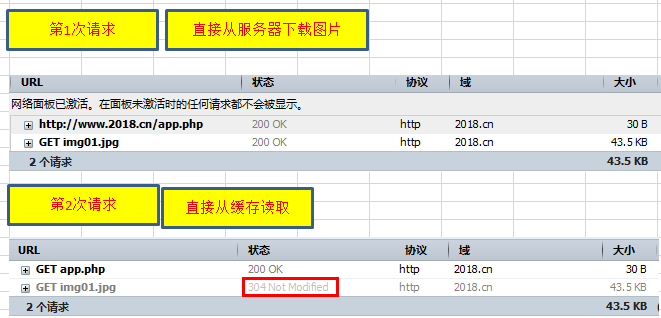
301永久重定向：更改服务器配置，重启服务器，由旧域名转成新域名。

302临时重定向：不需要服务器配置，直接在PHP中修改，不是重大改变用302。



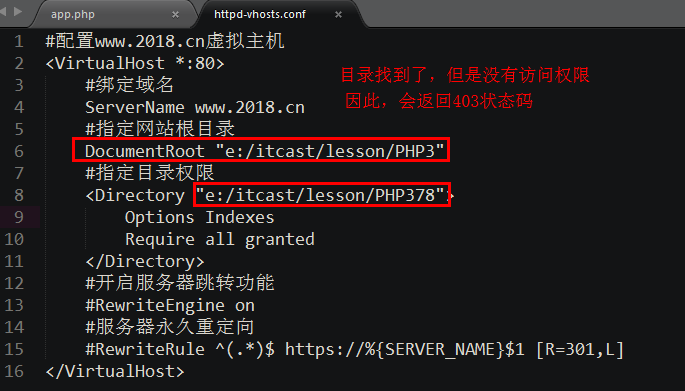
## 5、304状态码演示

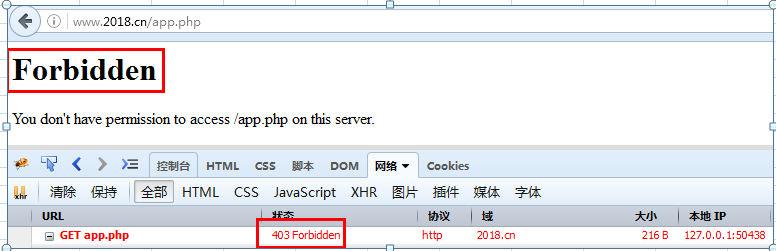
**304的含义：文件内容没有修改，不需要重新从服务器下载数据，直接从缓存读取。**



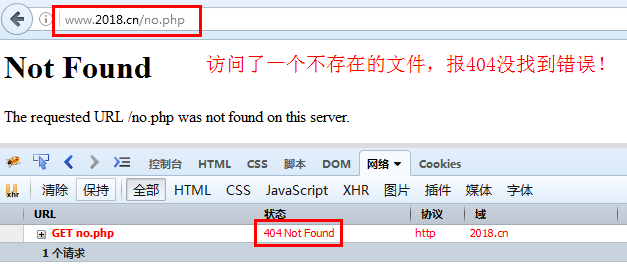
## 6、403状态码演示

403含义：请求接收，但无权访问或禁止访问。





## 7、404状态码演示



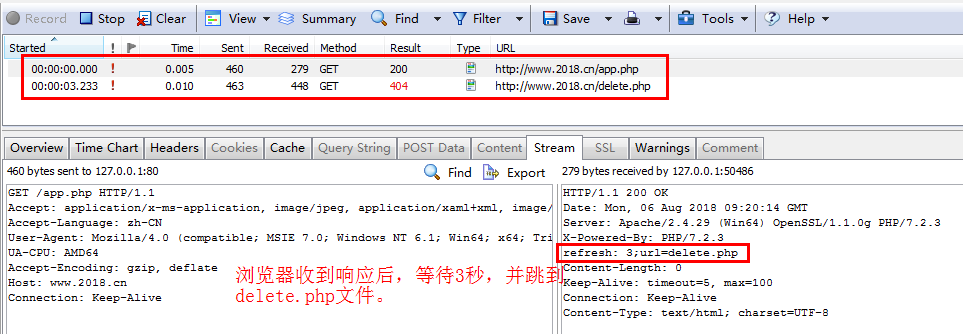
## 8、HTTP响应头含义(部分)

一个HTTP响应代表服务器给浏览器回送的数据，同时告诉浏览器应当怎样处理数据。



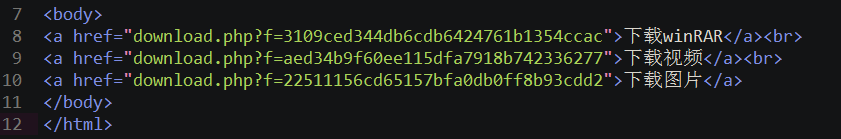
## 9、刷新并跳转





# 综合案例：实现视频文件下载

## 1、下载的静态页面download.html





## 2、下载程序处理download.php

