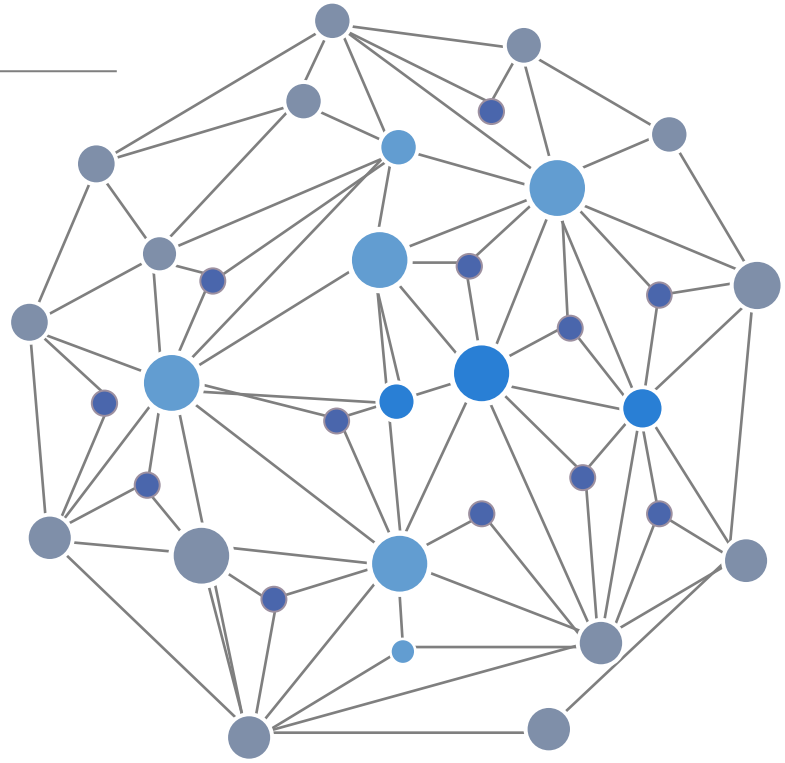
Web 程序设计

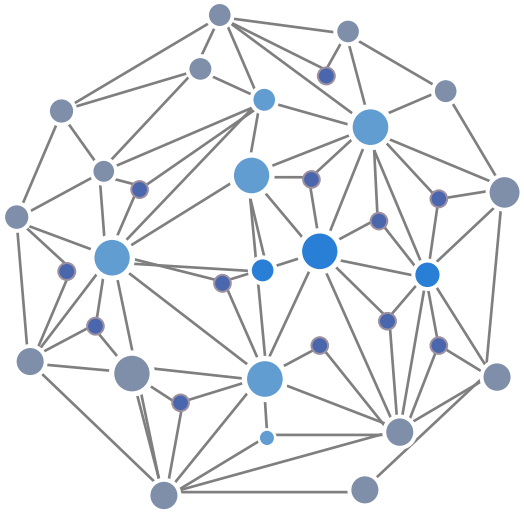


第十讲 Web 开发技术进阶

福州大学 计算机与大数据学院
软件工程系 陈昱

内容提要

- AJAX
- 面向对象的 PHP
- Web Service
- Web 开发框架



AJAX

什么是 AJAX

- AJAX = Asynchronous JavaScript And XML
- 这个名字由 Jesse James Garrett 提出
  - February 18, 2005
- 但 Google 公司很早就开始应用这个技术
  - GMail, Google Maps, Google Suggest

Google Suggest

ajax

ajax
ajax fc
ajax jquery
ajax 跨域
ajax技术
ajax post
ajaxform
ajax datatype
ajaxsubmit
ajaxsetup

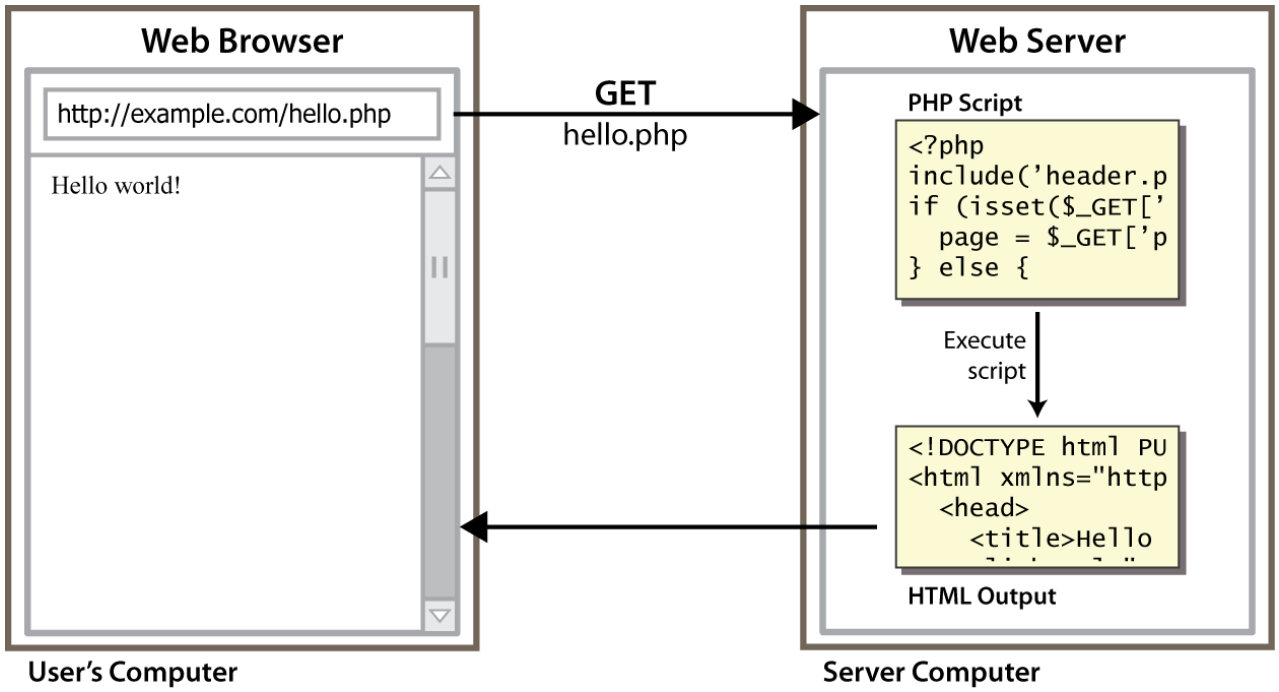
Google 大全

权政策和条款

Google Maps

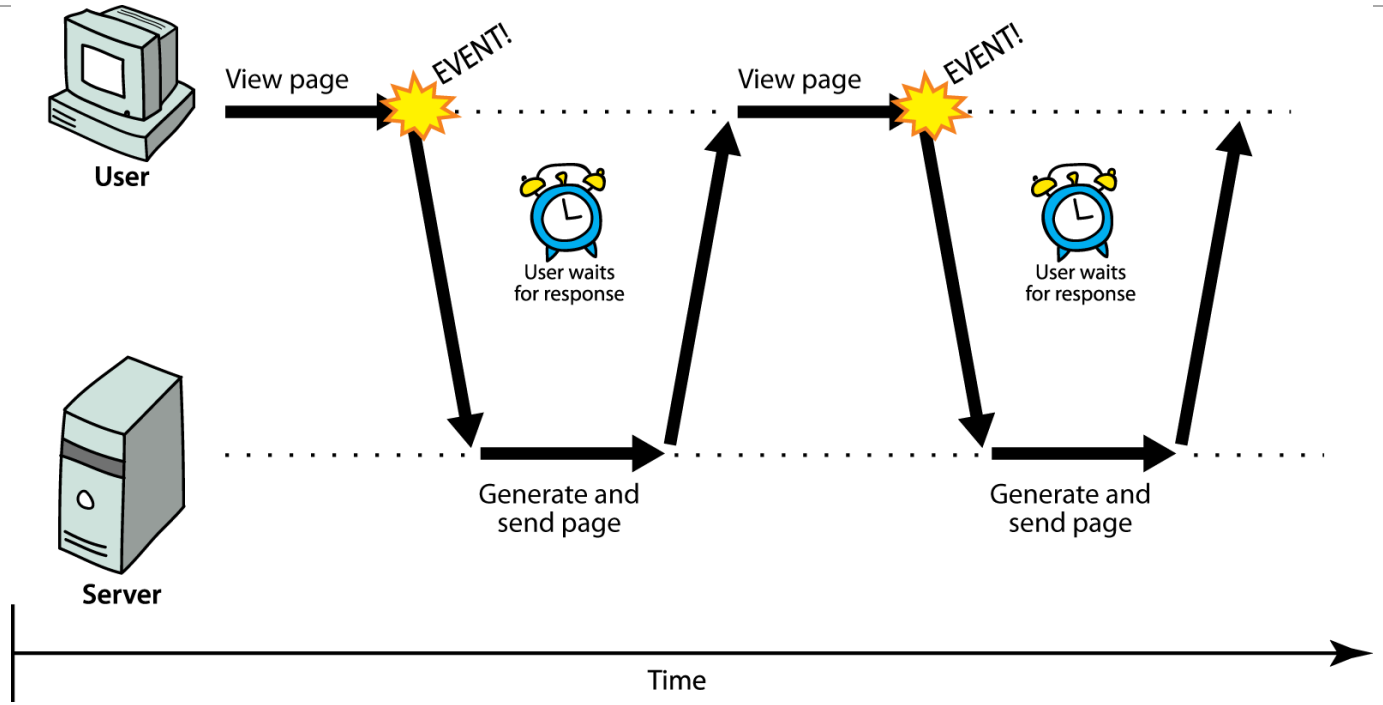

登录 | 帮助
网页 图片 资讯 地图新! 更多 »
Google 地图 BETA
咖啡
搜索地图
搜索地图 搜索周边 行车路线
地图
打印 电子邮件 显示本页链接
福州鼓楼区 新权南路12 号五一广场香格里拉北侧
0591-83365377
福州市鼓楼区古榕渡咖啡屋
福州市鼓楼区仙塔街仙塔新村 5 座 C 1 #店（负责人：房洁）
0591-7679967
古榕渡咖啡屋 - 更多信息 »
福建省福州市鼓楼区仙塔街
木村咖啡 - 更多信息 »
福建省福州市鼓楼区东泰路246号
0591-87670832
上岛咖啡五一店 - 更多信息 »
福州鼓楼区 斗东路12号(近福州大饭店)
0591-83346896
Goooogle
1 2 3 4 下一页
赞助商链接
开家零点咖啡加盟 店
上岛咖啡五一店
更多信息 »
福州鼓楼区 斗东路12号(近福州大饭店)
0591-83346896
在附近搜索
宫巷
津泰路
八一七北路
朱紫坊
法海路
鳌峰坊
福州市第七医院
福建医学高等专科学校
乌塔
于山风景区
乌山路
马道街
乌山支路
八一七路
广达路
斗东路
福州大饭店
福州动物园
200 米
500 英尺
©2007 Google - 地图数据 ©2007 Mapabc.com - 使用条款

浏览器←→服务器交互



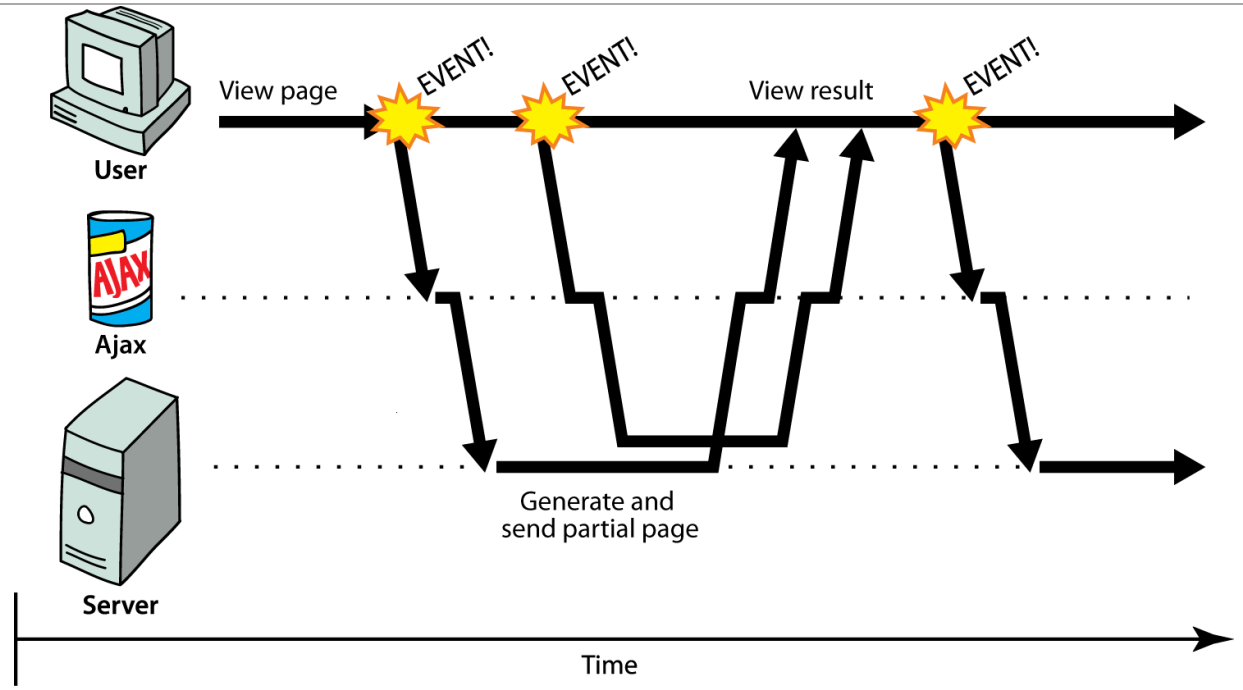
- 浏览器如何与用户交互?
- 什么时候它会触发一个请求?

同步 Web 通信



- 同步（synchronous）：用户必须等待新的页面加载
  - 典型的通信模式，用于 Web 页面（点击，等待，刷新）
- 为了显示新数据需要页面刷新

异步 Web 通信



- 异步（asynchronous）：当数据加载的时候，用户可以保持和页面的交互
  - 采用 AJAX 可以实现的通信模式
- 页面数据更新但不需要页面刷新

AJAX 技术

- AJAX 是一种在无需重新加载整个网页的情况下，能够更新部分网页的技术。
- AJAX 使用 JavaScript 在服务器和浏览器之间发送和接收数据。
- 在浏览器与 Web 服务器之间，它使用异步数据传输 (HTTP requests)，允许网页向服务器索取少量的信息而非整个页面。

AJAX 基于开放的标准

- AJAX基于以下开放的标准：
  - XML
  - XHTML
  - CSS
  - JavaScript

- 因此 AJAX 应用程序能独立于浏览器和平台

- AJAX 不是一门新的编程语言，而仅仅是使用现有标准的新方法

XMLHttpRequest 对象

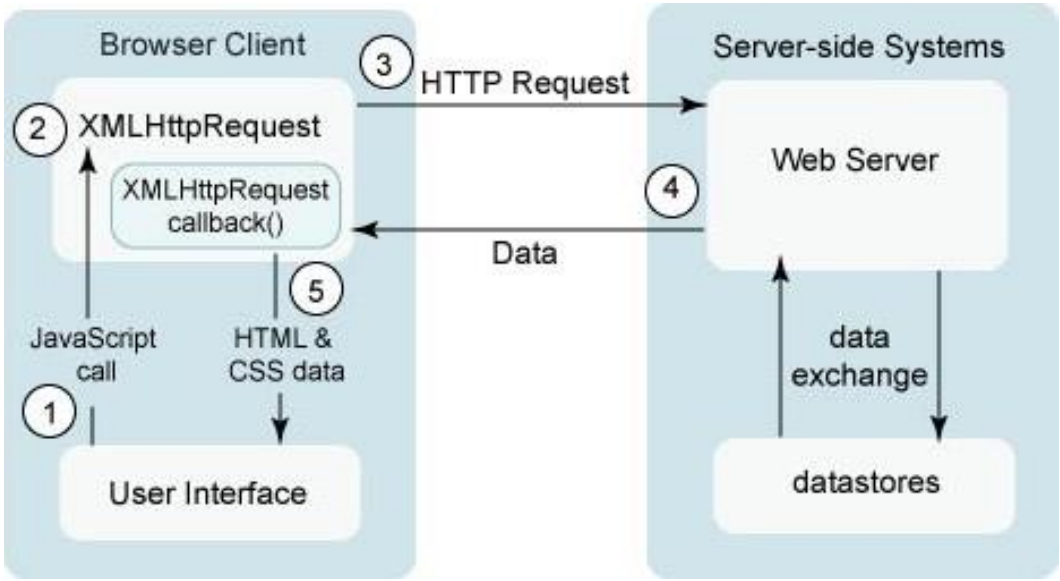
- JavaScript 引入一个 XMLHttpRequest 对象可以从 Web 服务器获取数据
  - IE5+, Safari, Firefox, Opera, Chrome 支持
- 它允许客户端 JavaScript 程序以后台活动的方式（异步的方式）获取数据（对用户透明）
  - 数据格式可以是 XML, JSON, HTML, Image etc.
- 获取的内容可以使用 DOM 更新到页面上

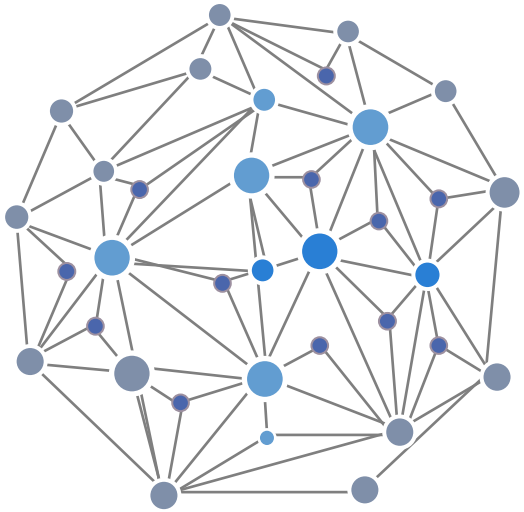
XMLHttpRequest 对象

- 听起来很好很强大~~~
- 但它相当难用，而且有很多浏览器兼容性问题
- 所以我们更喜欢使用 JavaScript 工具库所提供的 Ajax 封装功能
  - jQuery
  - Prototype
  - mootools
  - ……

一个典型的 Ajax 请求过程

- 用户操作，激活一个事件响应函数
- 事件响应函数创建一个 XMLHttpRequest 对象
- XMLHttpRequest 从服务器请求数据
- 服务器生成并发回数据
- 当数据到达的时候，XMLHttpRequest 触发一个事件
- 你的事件回调函数处理数据并显示

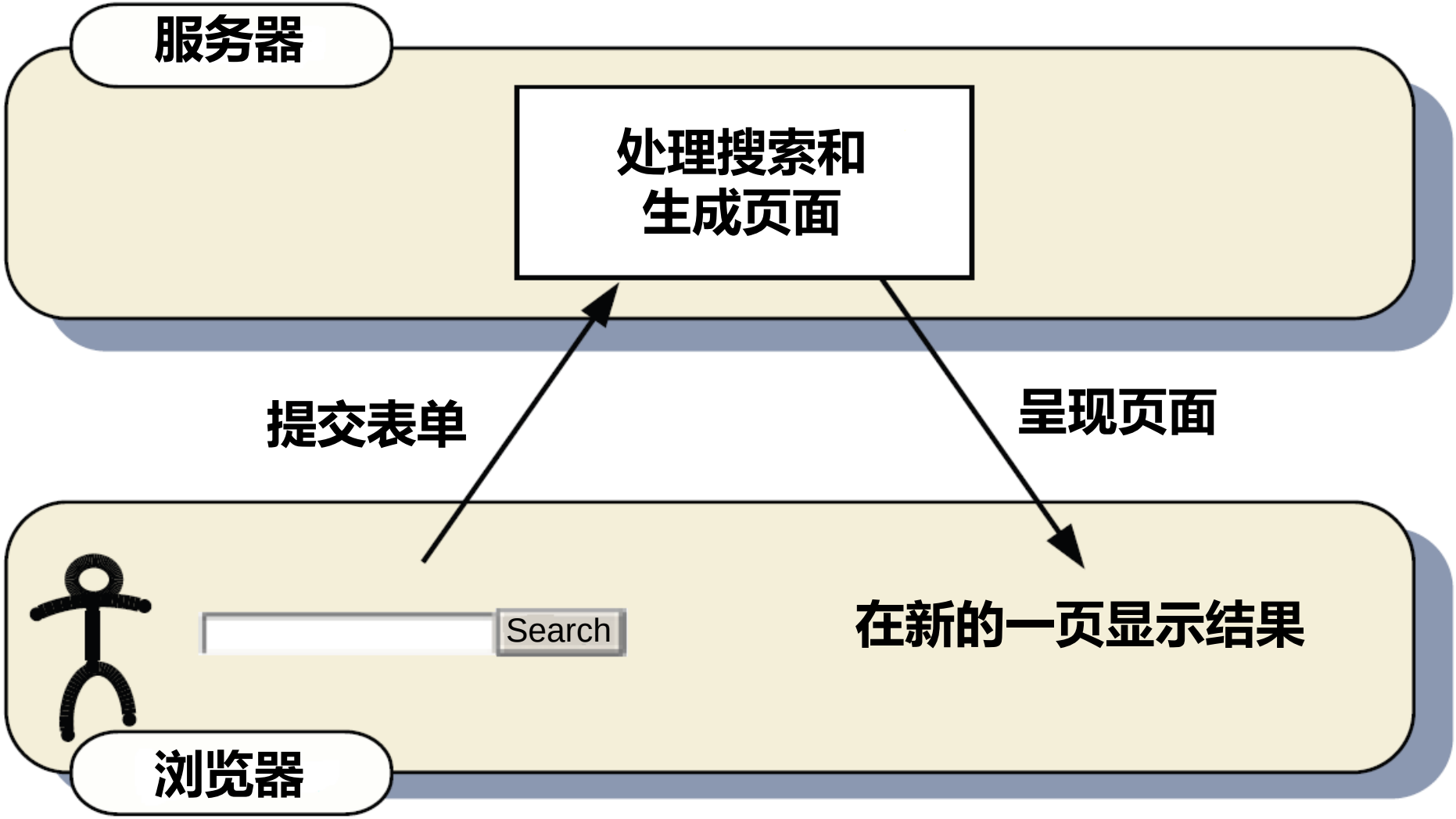




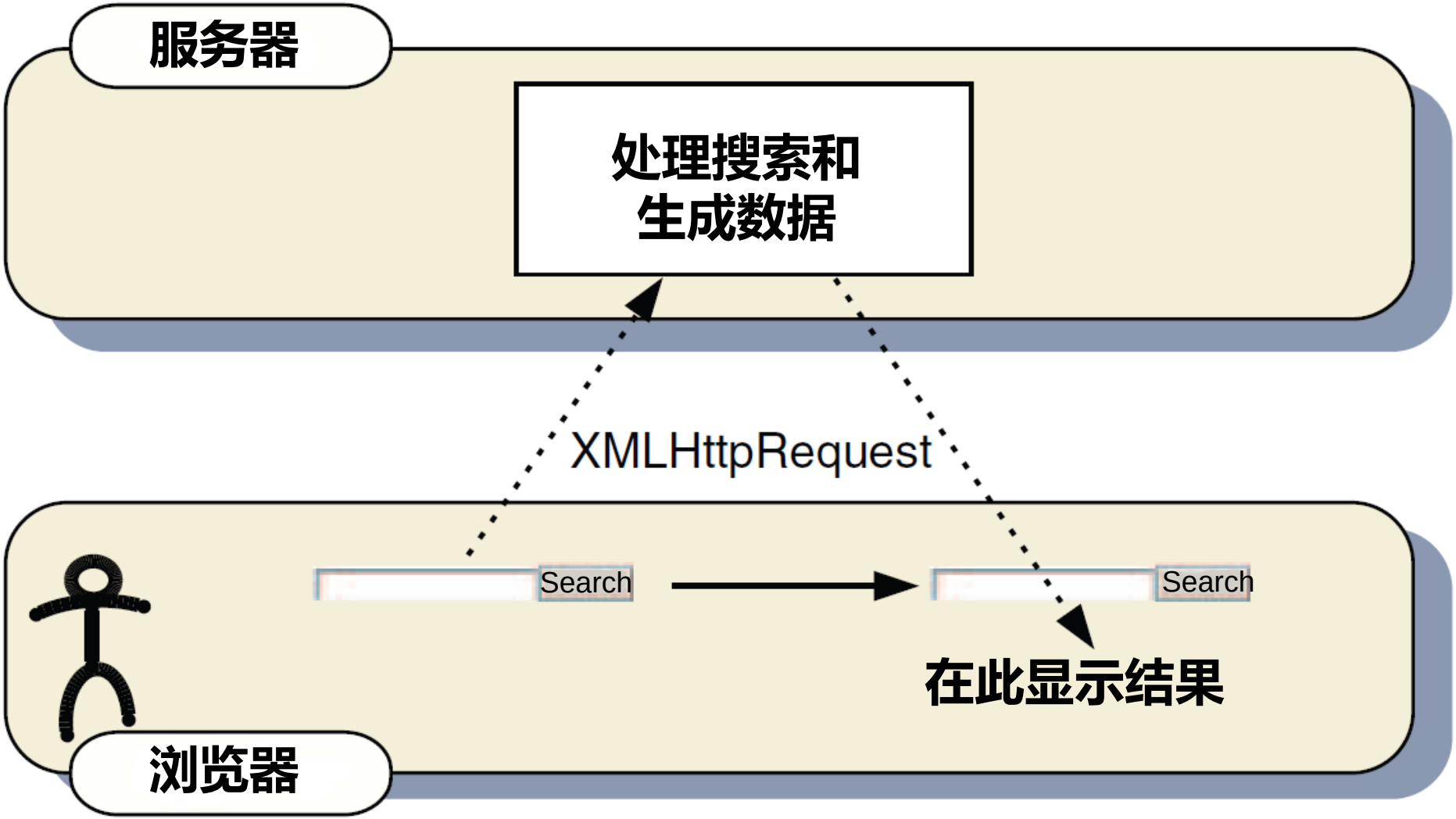
基于 Ajax 的动态搜索

Live Search

传统的搜索模型



基于 Ajax 的搜索模型



Live Search — 客户端页面框架

```
<!DOCTYPE html>
<html>

<head>
    <title>Ajax Demonstration</title>
     <script
    src="http://libs.baidu.com/jquery/1.9.1/jquery.min.js"></
    script>
    这里加载 jquery 工具库（从 Baidu CDN 镜像）
</head>
```

Client

```
<body>
<h1>Ajax Demonstration of Live Search</h1>
<p>
Search for: <input type="text" id="searchstring" />
</p>
<div id="results">
    <p>search result:</p>
    <ul id="list">
    </ul>
</div>
```

Client

```
<script type="text/javascript">
    var t;  //  public variable for the timeout
    $("#searchstring").keydown(function () {
        if (t) window.clearTimeout(t);
        t = window.setTimeout("livesearch()", 200);
    });
    这里将实现 livesearch() 函数
</script>
</body>
</html>
```

服务器端 PHP 程序

```
<?php
header("Content-Type: application/json");

// 定义了一个数组，用来模拟数据库
$people = array( "Abraham Lincoln", "Martin Luther King",
    "Jimi Hendrix", "John Wayne", "John Carpenter",
    "Elizabeth Shue", "Benny Hill",  "Lou Costello", "Bud
    Abbott", "Albert Einstein", "Rich Hall", "Anthony
    Soprano", "Michelle Rodriguez", "Carmen Miranda",
   "Sandra Bullock", "Moe Howard", "Ulysses S. Grant",
    "Stephen Fry", "Kurt Vonnegut", "Yosemite Sam", "Ed
    Norton", "George Armstrong Custer", "Alice Kramden",
    "Evangeline Lilly", "Harlan Ellison");
```

服务器端 PHP 程序

```
if (!isset($query))
        $query = $_GET['query'];
$result = array();
while (list($k, $v) = each($people)) {
        if (stristr($v, $query))
                $result[] = $v;
}
echo json_encode($result);
?>
```

服务器端 PHP 程序

- 比如，当用户输入John时，PHP 输出：

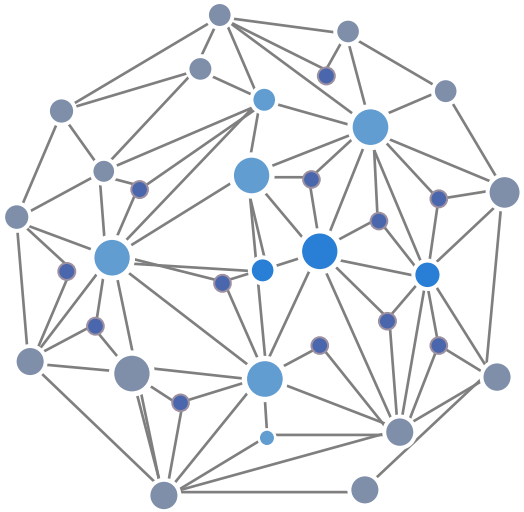
  ["John Wayne","John Carpenter"]
- 这个字符串称为 JSON
  - JavaScript 对象表示法（JavaScript Object Notation）
  - JSON 是存储和交换文本信息的语法, 类似 XML
  - JSON 比 XML 更小、更快，更易解析
- 详见： http://json.org/json-zh.html

Client

```
function livesearch() {
        var myURL = "search.php";
        var query = $("#searchstring").val();
        myURL = myURL + "?query=" + query;
        $.getJSON(myURL, function (result) {
            $("#list").empty();
            for (i = 0; i < result.length; i++) {
                $("#list").append("<li>" + result[i] + "</li>");
            }
            if (result.length == 0) {
                $("#list").append("<li>No results.</li>");
            }
        });
}
```

参考教程

- AJAX 教程
  - http://www.w3school.com.cn/ajax/
- jQuery 教程
  - http://www.w3school.com.cn/jquery/



面向对象的 PHP

为什么使用 OOP?

- PHP 是一门主要面向过程的语言
- 在不用添加任何类和对象的情况下，使得小型程序的编写更加容易
- 相对地，大型程序则会因为大量无组织性的函数而变得混乱
- 通过对象组合相关的数据和行为有助于管理程序大小和复杂度
- 更重要的是，如果你不会 OOP 编程，
- 你可能会像这个可怜的人一样.....

构造和使用对象

```php
# construct an object
$name = new ClassName(parameters);

# access an object's field (if the field is public)
$name->fieldName

# call an object's method
$name->methodName(parameters);
```

```php
$zip = new ZipArchive();
$zip->open("moviefiles.zip");
$zip->extractTo("images/");
$zip->close();
```

- 上面的代码可以用来解压一个文件
- 使用 class_exists 检查一个类是否存在

对象例子：从Web上获取文件

```php
# create an HTTP request to fetch student.php
$req = new HttpRequest("student.php", HttpRequest::METH_GET);
$params = array("first_name" => $fname, "last_name" => $lname);
$req->addPostFields($params);

# send request and examine result
$req->send();
$http_result_code = $req->getResponseCode();    # 200 means OK
print "$http_result_code\n";
print $req->getResponseBody();
```

- PHP 的 HttpRequest 对象能从 Web 上获取一个文件

声明一个类的语法

```php
class ClassName {
  # fields - data inside each object
  public $name;     # public field
  private $name;    # private field

  # constructor - initializes each object's state
  public function __construct(parameters) {
    statement(s);
  }

  # method - behavior of each object
  public function name(parameters) {
    statements;
  }
}
```

- $this: 在构造器和方法里引用当前对象

Class 例子

```php
<?php
class Point {
  public $x;
  public $y;

  # equivalent of a Java constructor
  public function __construct($x, $y) {
    $this->x = $x;
    $this->y = $y;
  }

  public function distance($p) {
    $dx = $this->x - $p->x;
    $dy = $this->y - $p->y;
    return sqrt($dx * $dx + $dy * $dy);
  }

  # equivalent of Java's toString method
  public function __toString() {
    return "(" . $this->x . ", " . $this->y . ")";
  }
}
?>
```

PHP

Class 使用例子

```php
<?php
# this code could go into a file named use_point.php
include("Point.php");

$p1 = new Point(0, 0);
$p2 = new Point(4, 3);
print "Distance between $p1 and $p2 is " . $p1->distance($p2) . "\n\n";

var_dump($p2);   # var_dump prints detailed state of an object
?>
```

PHP

```
Distance between (0, 0) and (4, 3) is 5

object(Point)[2]
  public 'x' => int 4
  public 'y' => int 3
```

PHP

- $p1 和 $p2 是 Point 对象的引用

继承

```
class ClassName extends ClassName {
  ...
}
```
PHP

```
class Point3D extends Point {
  public $z;

  public function __construct($x, $y, $z) {
    parent::__construct($x, $y);
    $this->z = $z;
  }

  ...
}
```
PHP

- 子类会继承所有父类的数据和行为

静态方法，域 和常量

```php
static $name = value;      # declaring a static field
const $name = value;       # declaring a static constant
```

```php
# declaring a static method
public static function name(parameters) {
  statements;
}
```

```php
ClassName::methodName(parameters);   # calling a static method
self::methodName(parameters);        # calling a static method
```

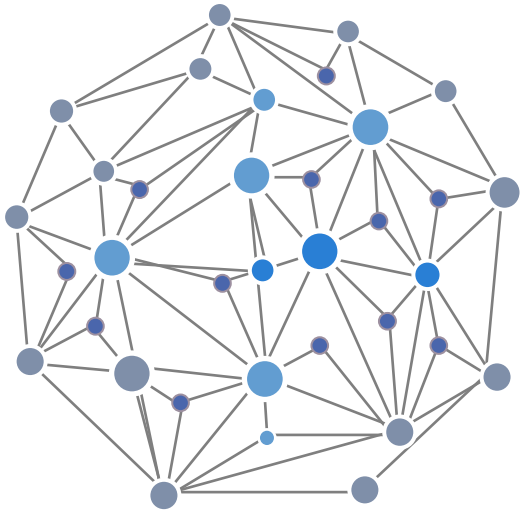
- 整个类的 static 域/方法是共享的，而不是每个对象都拥有该域/方法的副本

抽象类与接口

```php
interface InterfaceName {
  public function name(parameters);
  public function name(parameters);
  ...
}

class ClassName implements InterfaceName { ...
```

```php
abstract class ClassName {
  abstract public function name(parameters);
  ...
}
```

- 接口是超类，它指定了方法头而无需实现
  - 不能被实例化；不能包含函数主体和域
  - 允许在子类中使用多态，而无需共享实现的代码
- 抽象类就像接口，但你能指定域，构造器和方法
  - 也不能实例化
  - 允许使用多态并共享实现代码



Web Service

Web Service

- Web Service 就是一个应用程序，它向外界暴露出一个能够通过 Web 进行调用的API
- 你能够用编程的方法通过 Web 来调用这个应用程序，获得执行结果
- 服务器端和客户端之间的数据通常封装成 XML 的格式传递

Web Service 协议

- 几种常见的 Web Service 通讯协议
  - SOAP (简易对象访问协议)
  - XML-RPC
  - REST

- **Web Service 平台的相关元素**
  - WSDL (Web Service 描述语言)
  - UDDI (通用描述、发现及整合)

SOAP

- SOAP 是一种通信协议
- SOAP 用于应用程序之间的通信
- SOAP 是一种用于发送消息的格式
- SOAP 基于 XML

WSDL

- WSDL 指网络服务描述语言
- WSDL 是基于 XML 的用于描述 Web Service 以及如何访问 Web Service 的语言
- WSDL 使用 XML 编写，是一种 XML 文档
- 它可规定服务的位置，以及此服务提供的操作（或方法）

UDDI

- UDDI 是一种目录服务，通过它，企业可注册并搜索 Web Service

- UDDI
  - Universal Description, Discovery and Integration
  - 指通用描述、发现以及整合

- 是一种由 WSDL 描述的网络服务接口目录

php Soap Extension

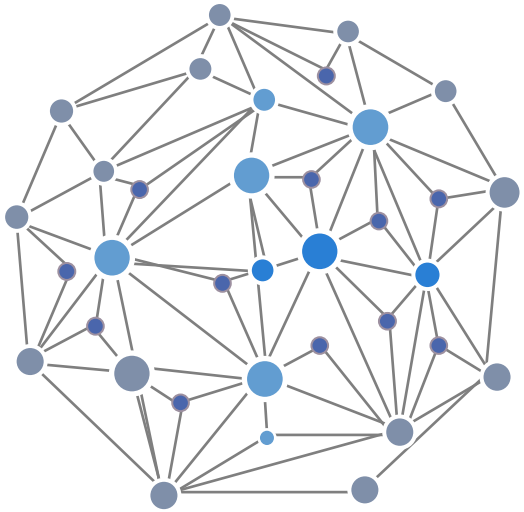
- 在 php.ini 中开启 extension=php_soap.dll 就能实现 Soap 的服务器端和客户端

SOAP 服务器端

```
function add($a, $b) {
    $retval = $a + $b;
    return new SoapParam($retval, 'sum');
}

$server = new SoapServer(null,
                array('uri' => 'TestSoap')); // 命名空间

$server->addFunction('add');
$server->handle();
```

SOAP 客户端

```
$client = new SoapClient(null,
        array(
                'location' => 'http://domain/soap/server.php',
                'uri' => 'TestSoap')
         );

$a = 1; $b = 2;
$result = $client->add(new SoapParam($a, "a"),
                                new SoapParam($b, "b"));
printf("Result = %s", $result);
```



Web 开发框架

框架 vs. 库/工具箱

- 在计算机编程领域中，**软件框架**提供一些通用的功能和函数，用户可以重写和定制这些函数，以达到提供特定功能的目的
- 库是通用资料的集合，它被定义为与软件结构有关的集合
- 例如用于软件开发的类，函数和子例程
- 工具箱是软件工程领域中库的别名
- 框架调用你的代码，而你的代码调用库

Web应用开发中常见的问题

- 数据检测和转换
- URL 映射
- 配置(数据库连接，线程，超时...)
- 会话管理
- 页面模板
- 数据库访问和映射
- 用户认证
- Web 服务
- AJAX
- 缓存

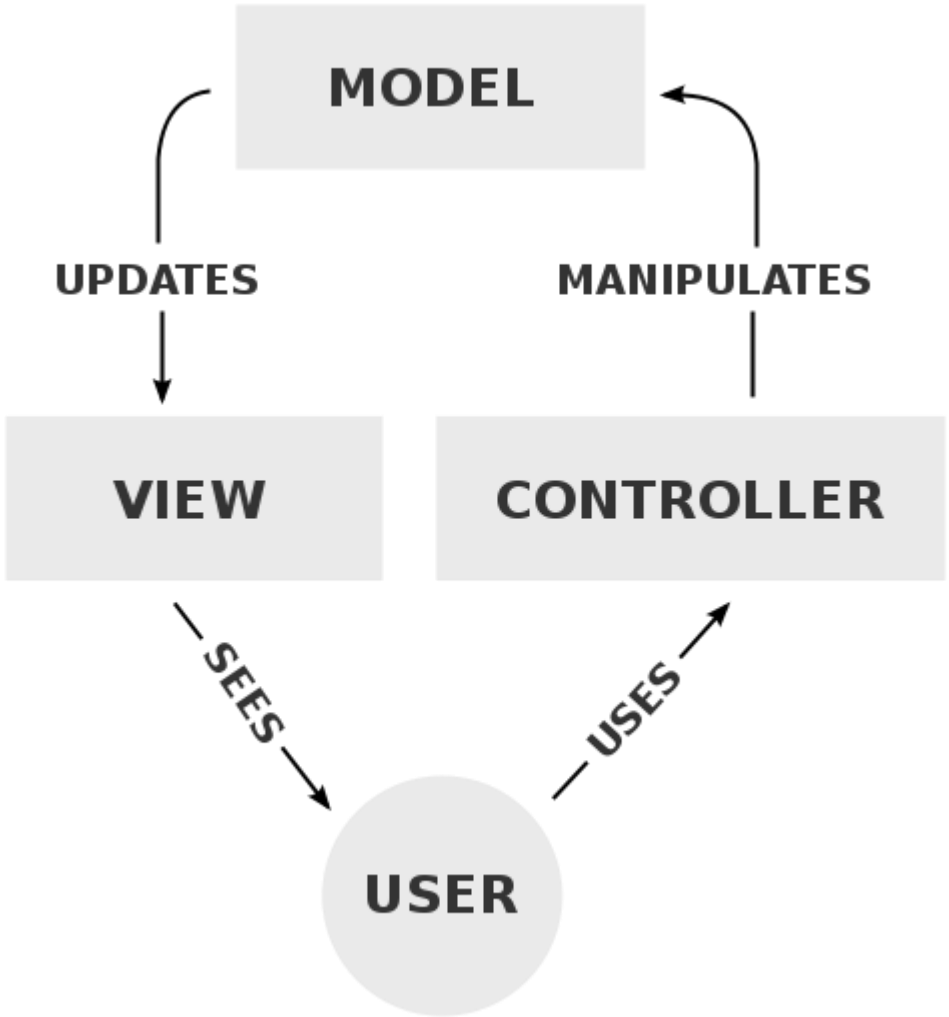
Web 框架

- Web 应用框架是一个软件框架．它用于支持动态网站, Web应用和Web服务的开发．这种框架旨在减轻关联Web开发中常用服务的花费．例如，很多框架提供数据库访问，模板框架和会话管理的库，并且它们提倡代码复用．
- 大部分服务端语言都有它们自己的Web框架
  - Spring Framework，Apache Struts，ASP.NET MVC, Symfony , CakePHP，Zend Framework，Django, Ruby on Rails, ……

MVC

- MVC 是软件工程中的一种软件架构模式，把软件系统分为三个基本部分：模型（Model）、视图（View）和控制器（Controller）
  - **模型 (Model)** 代表你的数据结构。通常来说，你的模型类将包含取出、插入、更新你的数据库资料这些功能。
  - **视图 (View)** 是展示给用户的信息。一个视图通常是一个网页，包含文本、表单等用户界面元素。
  - **控制器 (Controller)** 它负责根据用户从"视图"输入的指令，命令"模型"操作数据，产生最终结果并更新视图。

MVC



Ruby on Rails

Sustainable productivity for web-application development

Rails 的不同之处

- Rails 设计的始终：习惯约定优于配置
- 另外一个设计理念：更少的代码
- 达到了框架易用性上的一次突破
- 得益于完全面向对象和动态特性的 Ruby 语言，实现了一些其他语言无法实现的功能

重新审视我们的生活

- PHP 的创始人 Rasmus Lerdorf 曾经这样表示过，他希望自己能够减少盯着电脑的时间
- 可是这么多年过去了，他发现自己还是要继续盯着该死的电脑

- RoR 让我们重新审视我们之前的开发过程

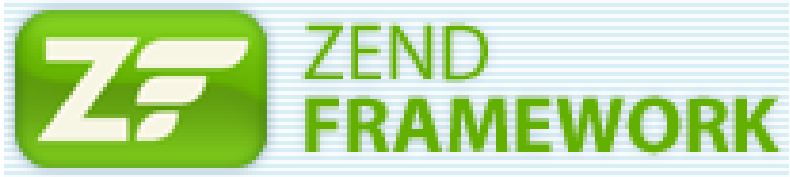
RoR 的影响

- 模仿是最好的恭维
- RoR 的出现，引起其他语言社区的强烈反应

- PHP社区，出现了几十种Web框架

基于 PHP 的 Web 开发框架

- Laravel
- Yii Framework
- Symfony
- CakePHP
- Code Igniter
- Zend Framework
- ……



THANKS

本章结束

陈昱

福州大学 计算机与大数据学院 软件工程系