

讲义 13: Ajax & JSON

本课程研究网页内部与服务器通讯的技术 Ajax。它广泛用于 Gmail, Google Map 等客户端应用（重要!!!!!!）

课程目标：

- 1) Ajax 和 web service 的编程基本技术。
- 2) JSON 在客户端的处理、服务器端的处理技术，在两者之间通讯的问题。

使用工具：

Notepad++ or sublime;
chrome browser;
python2.7; Tornado 3.1;

课程资源与要点：

- 1) W3 教程 JS: **JS**
<http://www.w3schools.com/js/default.asp>
<http://w3school.com.cn/js/index.asp>（中文）
--- HTML FORM: 表单。
- 2) Google Hosted Web Lib: **google web**
<https://developers.google.com/speed/libraries/devguide?hl=zh-CN>
- 3) Prototype 入门教程: **prototype**
<http://prototypejs.org/learn/>
--- Introduction to Ajax
--- Introduction to JSON
- 4) Google Map JavaScript API: **google map**
<https://developers.google.com/maps/documentation/>
- 5) tornado 教程（中文）: **tornado**
<http://demo.pythoner.com/itt2zh/>

任务 1: 准备知识

JavaScript 是博大精深的语言，课上只能讲一些要点

任务 1.1: JavaScript 闭包（closure）初探

<http://www.htmldog.com/guides/javascript/advanced/closures/>

A **closure** is a function that returns a function. The function that is returned (the inner function) is created inside the called function (the outer) so - due to the scoping rules we've seen - the inner has access to the variables and arguments of the outer.

简单来说：JavaScript 函数的子函数可以存取父函数中的参数与变量。这种现象叫“闭包”。这给传统语言（c, java 等）的使用者带来意想不到的 BUG！

- 1) 内部函数引用外部函数参数

打开控制台，将第一案例代码复制到控制台

OK!，你看到了结果。

提示：如果你理解了前面讲的函数即对象，当你引用了这个函数，函数就实例化

了。只要你引用和这个实例，无论 `var p = new Person()` 或 `var f = myFunc()` 这个函数对象将保持在内存中。

这时，子函数和函数共享该对象的内容。

2) 利用这个特点做增量函数的应用

运行第二个案例。

有点意思，但是学语法或 toy 程序有什么意义呢？

语言的学习，一定要放在产业生产环境中去学习！（又要被大家骂死了，这好累人的！）

任务 1.2: Ajax 应用初探

Google Map 利用 Ajax 编程产生的效果。让我们 getting start!

<https://developers.google.com/maps/documentation/javascript/tutorial?hl=zh-cn>

好歹是中文的。不喜欢看中文的就去掉“？”后面的内容

1) 请用 google 浏览器打开附件提供的 googlemap.html

地图有了，你可以放大，缩小，移动地图。

问题：没看见向服务器 GET 或 POST 啊，怎么能交互？神奇的 Ajax！

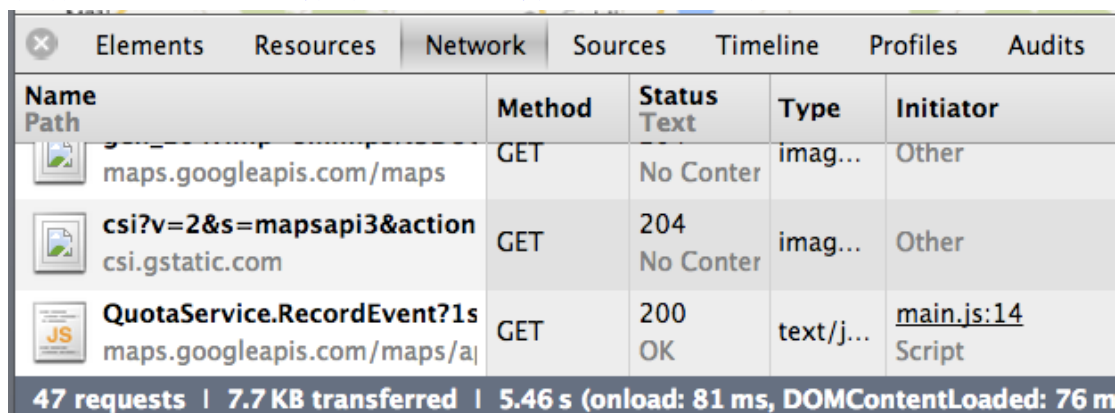
让我们简单阅读这段代码。好的 JavaScript 框架都是易读、易用的。

多读工业级代码！编高质量程序。

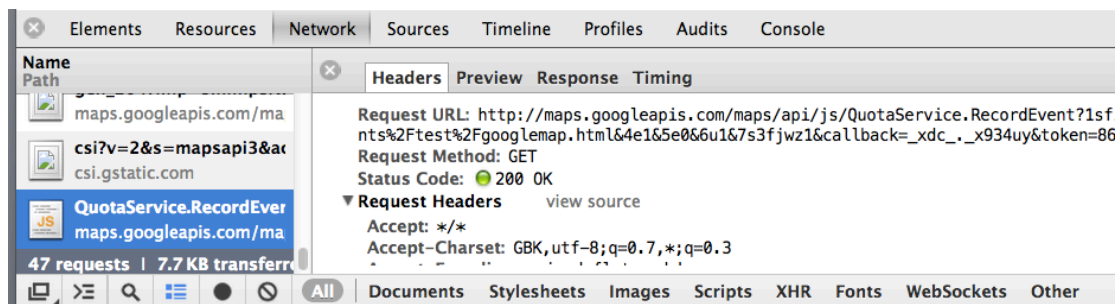
2) 使用编程人员开发工具

打开开发者工具，选择 Network。

刷新页面！我们可以看到该页面与服务器交互的全过程。



点最后一个 type= text/javascript; method=get 的服务器访问的内容。你看到 Google 地图 api 与服务器交互的内容。



点击 main.js 你可以看到程序的代码（由于这段 JavaScript 压缩了，好难读的）。知道工具有这些功能就可以了。

3) 看与服务器的交互

选择开发者工具-->Network

然后，放大或缩小地图，你又看到一堆交互事件产生。

任务 1.3: JavaScript 闭包 (closure) 实战

<https://developers.google.com/maps/documentation/javascript/events?hl=zh-cn>

使用 Google 浏览器，打开附件提供的 closures.html。

1) 点击界面上的地图 mark

2) 阅读 Google map 关于闭包的说明，研究闭包是如何应用的？

好有技术含量啊。多看多做（做中学），你自然就写出这样的代码了!!!

任务 2: Prototype Ajax & JSON 入门

任务 2.1: Introduction to Ajax

参见: <http://prototypejs.org/learn/introduction-to-ajax.html>

1) 使用 python 运行 ajaxhello.py

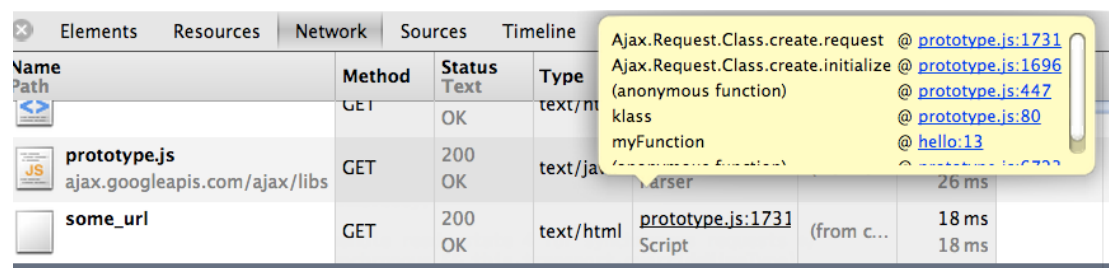
在 Google 浏览器 URL 中输入: <http://localhost:8888/hello>

一个对话框出现了。然后点击“Hello”按钮，看看网页如何变化？

2) 用 notepad++ 打开 ajax.html 和 ajaxhello.py

打开开发者工具，用 Network 监控。刷新程序

然后将光标移到 some_url 的 initiator 列



Name	Path	Method	Status	Text	Type	Initiator	Time
prototype.js	ajax.googleapis.com/ajax/libs	GET	200	OK	text/javascript		26 ms
some_url		GET	200	OK	text/html	prototype.js:1731 Script	18 ms

你可以看到的程序执行序列。那就看 myFunction 写了什么

3) Ajax 对象及其工作过程

先看客户端代码:

```
new Ajax.Request('/some_url', {
  method: 'get',
  onSuccess: function(transport) {
    var response = transport.responseText || "no response text";
    alert("Success! \n\n" + response);
  },
  onFailure: function() { alert('Something went wrong...'); }
});
```

其中 Ajax 叫 namespace，用以防止和其他程序定义的 Request 对象冲突。

高级话题: namespace 请谷歌“JavaScript Namespace”

程序这样理解: 创建 Ajax 内部定义的 Request 对象。该对象将向当前页面的服务器发送 GET /some_url HTTP/1.1。当服务返回后，成功将调用 onSuccess 的函数，失败调用 onFailure。在监控窗口中也看到了这个过程。

4) 阅读官方网站的介绍

<http://prototypejs.org/learn/introduction-to-ajax.html>

本部分要点：

基本形式；

处理服务器返回的 `response`

发送 `QueryString` / 发送表单的内容

检查返回的是否 JSON 格式

如何简单的 `update` 页面

任务 2.2: Web Service 编程

阅读 `ajaxhello.py`

添加了 `“/some_url”` 的处理器，如同响应普通请求一样！

任务 2.3: Introduction to JSON

参见: <http://prototypejs.org/learn/json.html>

1) 阅读第一段英文文字

轻量级数据交换协议；主要用于服务器、客户端交换数据；是替代 XML 交换数据的方法。

2) 什么是 JSON

点击第一段文字的链接，进入 JSON 官方网页。

<http://www.json.org/>

语法简单，文本格式 (`application/json`)，几乎可以和所有语言交换数据（见后面支持列表）

3) 返回 `prototype` 网站，研究学习内容

在 `ajax test` 的 JavaScript 控制台输入案例

4) 请求服务器返回 JSON 格式对象

？服务器该怎么做？

任务 3: Ajax 原理

<https://courses.cs.washington.edu/courses/cse190m/09sp/lectures.shtml>

任务 4: Ajax 的应用案例

这是一个从 GPS 数据中读取出租车汽车轨迹在地图上显示的应用。

`trace.py` 提供了 web 服务，以及数据服务；`sample.py` 是读数据程序；以及一个出租车网页模板 `position_mark_for_trace.html`。

运行 `trace.py` 程序。在浏览器输入: <http://localhost:8888/taxi>

红色表示空车，蓝色表示实载。箭头长度表示速度。

最后我们使用开发者工具 `Source` 调试 JavaScript 代码。同时帮助你理解 Ajax 的异步过程。

作业与练习：

1、无

二、操作与实践

1、将出租车轨迹应用的 jQuery 框架，改成 Prototype 框架。并将出租车的 id（电话号码）从客户端发送到服务器端。

2、根据新浪微博上读取的页面分析新浪微博数据。您（或组团）需要做以下工作：

（1）分析新浪微博网页组织。

a) 使用 google 浏览器打开“炫富大赛的相关微博 - 新浪微博”（感谢吴师兄，他用 python 程序把网页解码还原了）

b) 使用开发者工具，找到 <div class="feed_lists ...> 块，展开它，你会发现一组 <dl class="feed_list" ...> 的块。那些广告或工具条你可以暂时不管他们

c) 分析 <dl class="feed_list" ...> 元素的属性和内容，列出新浪微博的数据。（数据抓取的基础技能）

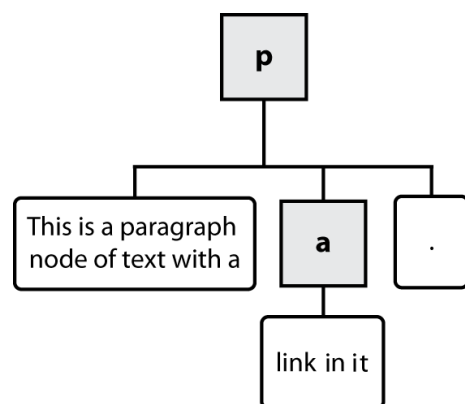
例如：

mid / 16 位数字 / dl.feed_list attribute / 新浪微博 id

isforward / 0,1 / dl.feed_list attribute / 1 表示转发

.....

d) 用树状图（大致如下图），分析 <dl class="feed_list" ...> （数据模板的基础技能）



（2）仿照出租车行驶信息，写一个 web service 能向客户端提供一组微博 JSON 数据。

（3）裁剪该网页，提供 CSS，将它定制成你喜欢的微博显示界面。（可以去除你认为与微博无关的内容）