讲义 8：浏览器与服务器交互技术初探

本课程将在服务器开发技术的基础上，学习浏览器与服务器交互的最基础的技术，迈出 Web 应用程序的第一步。

课程目标：
1）掌握 HTML 表单，与 tornado 建立服务器交互。

使用工具：
Notepad++ or sublime;
chrome browser;
python2.7; Tornado 3.1;

课程资源与要点：
1）实际案例资源：***practical case***
http://s.weibo.com/
2）W3 教程 HTML：***HTML***
http://www.w3schools.com/html/default.asp
--- HTTP Methods
--- HTTP Forms
3）tornado 官方文档（中文）：***tornado doc***
http://www.tornadoweb.cn/documentation
--- Tornado overview
--- Tornado 攻略－请求处理程序和请求参数：第一个案例，理解处理 URL;
--- Tornado 攻略－模板；模板结构；模板的基本使用；模板输出与转义(escape);
--- Tornado 攻略－用户认证

任务 1：实际案例与 HTTP 协议
浏览器与服务器交互是通过 HTTP 协议进行的，采用 Client-Server 模式。你对 HTTP 协议的理解，直接影响了你的编程思考。本任务帮助大家理解，我们在客户端的常见操作，会向服务器发送的内容。包括：URL、表单、文件
任务 1.1：最简单 URL
进入课程案例，微薄搜索：http://s.weibo.com/

1）我们输入这个 URL（http://s.weibo.com/）表示我们用 GET 方法，使用 HTTP 1.1 协议，请求下载了 s.weibo.com 服务器上的 / 文档。

```
GET / HTTP/1.1
Host:s.weibo.com
其他一些关于浏览器信息、编码信息、cookies 等都在 headers 告知服务器
```

Request：由客户端发给服务器的信息，包含 HTTP Method, Headers, Bodies 三个部分。服务器端编程时，通常通过对象 HTTPRequest，获取这些信息。
例如：你需要知道浏览器是 iphone、android 时，一般可以在 header 中找到相关信息。
参考：http://en.wikipedia.org/wiki/List_of_HTTP_header_fields
2）哪些内容是你会做的，哪些是不会做的呢？
置顶的新浪微博的主菜单...进入开发工具，class="WB_global_nav"；
输入框和按钮....不会；流动的文字效果....不会。所以，需要继续学习哦。
任务 1.2：复杂一些的 URL
1）输入“女神”，点搜索。观察 URL，

浏览器发出 GET，下载文档 /weibo/%25E5%25A5..... 这个文档名字好奇怪。
如果，你知道“女”的 utf8 编码“E5 A5 B3”，估计你能猜到一些东西。
后面的&表示什么？Refer=index 表示什么呢？

```
GET /weibo/%25E5%25A5%25B3%25E7%25A5%259E&Refer=index HTTP/1.1
Host:s.weibo.com
其他一些关于浏览器信息、编码信息、cookies 等都在 headers 告知服务器
```

2）点击一个微博用户的 ID
URL 变成了 http://weibo.com/gogoboi，哇，用户 gogoboi 的资料显示了
3）点击高级搜索按钮，选择原创、广东省、广州市。
反正，这个对话框效果肯定不会做了，但这个 URL 你能解释吗？
http://s.weibo.com/weibo/%25E5%25A5%25B3%25E7%25A5%259E&scope=ori&region=custom:44:1&Refer=g
仔细看一下，发现规律。要下载的文档 /weibo/女神；&开始表格项，scope=原

创，region=广东省，广州的编码，Refer=全部 猜的对吗？
任务 1.2：HTTP GET & POST
进入 w3 教程 ***HTML*** -> HTTP Methods
1）什么是 HTTP?
2）The GET Method
服务器上文档是这样表示的。文档名？参数＝值&参数＝值&参数＝值
3）The POST Method
与 GET 比，这些参数写在了 HTTP 协议请求的最后一行。
Query String：发送到服务器的(name/value pairs)。
注意：如果你想与老师互动，请修改 tonado 的 webhello 程序，显示你使用 telnet 发出的 HTTP 请求内容。即 HTTP 请求的文本。
课后，请仔细研究 ***tornado doc*** –>Tornado 攻略－请求处理程序和请求参数
任务 1.3：HTTP 协议与汉字编码
通常，HTTP 协议中是不能出现汉字的，仅能使用 ASCII。
进入 w3 教程 ***HTML*** -> HTML URL Encode
1）URL - Uniform Resource Locator
2）URL Encoding
看来汉字处理真是个麻烦问题。
任务 1.4：HTTP 与文件上传
如果你不是专家，会用就可以了。
如果你喜欢研究它，谷歌“HTTP upload file”或
参考 http://tools.ietf.org/html/rfc1867

任务 2：HTML 表单
HTML 表单，网上的教程都非常简单。可参考：***HTML*** -> HTML Forms
一句话解释表单：将在网页上的用户输入，转变成 Query String 发送给服务器。
今天，我们使用王博士的课件 8
任务 2.1：Form Basics
1）Form & **Query String**
2）HTML Form 元素：action 属性
通常，form 还有 method 属性，确定使用”get”或”post”发送 Query String
任务 2.2：Form Controls
1）Input 元素：name, type, type 属性
Note：In HTML5, like img and br tags, the input tag, which doesn’t surround any content, doesn’t require a closing tag.
2）常用控制属性
disable, maxlength, readonly, size, value.
请完成该页 PPT 的表单，尝试使用这些属性。（使用 w3 教程 Forms 综合案例）
3）多行文本框
4）CheckBox
参考：***HTML*** -> HTML Forms
5）Radio
参考：***HTML*** -> HTML Forms
6）Drop-down list

请参考 w3 school 或 html dog 参考，使用 HTML5 推荐写法
参考：***HTML*** -> HTML Forms
7）文件上传
自己查 HTML Reference <input>

任务 3：使用 tornado 处理 URL 与 Query String
很多同学反映，看不懂 tornado 的程序。对于任何一种编程框架，即使给你原理图，初学者也不可能完全理解。一般性学习过程，首先，把 demo 程序运行起来，形成直观；其次，了解程序的结构，关键的类和模块的功能；进一步，识别关键数据与 API，解释程序运行过程与结果；如果，解释失败，收集官方或可信的文档，修正自己的认识。使用实战程序验证，进一步理解数据与 API。
下面，让我们探讨这个学习过程，从 webhello -> 打印主要的 Request 内容。
任务 3.1：回顾 webhello.py 结构
进入 ***tornado doc*** -> overview
程序的结构，关键类，数据，API.....
问题：哪个类分析 URL
问题：哪个类处理 GET PUT HEAD 等请求
任务 3.2：简介请求与参数处理
进入 ***tornado doc*** ->攻略一请求处理程序和请求参数
1）URL 匹配
2）处理 Query String
问题 1：假设 Handler 的 GET 和 POST 的处理过程没有区别，你会如何编程？
——建议你从 post 方法中调用 get 方法
问题 2：不看源代码，猜测 arguments, flies, path, headers 的数据结构？
——建议你用程序去试验

```
class MainHandler(tornado.web.RequestHandler):
    def get(self):
        self.write('<html><body><form action="/" method="post">'
                   '<input type="text" name="message">'
                   '<input type="submit" value="Submit">'
                   '</form></body></html>')

    def post(self):
                self.set_header("Content-Type", "text/html")
                self.write("Header:" "<br>")
                for key, val in self.request.headers.iteritems():
                        self.write(key + "=" + repr(val) + "<br>")
                self.write("<br>message: " "<br>")
                for key, val in self.request.arguments.iteritems():
                        self.write(key + "=" + repr(val) + "<br>")

class StoryHandler(tornado.web.RequestHandler):
    def get(self, u_id):
        self.write("You requested the user " + u_id)
```

```
application = tornado.web.Application([
    (r"/", MainHandler),
    (r"/u/([0-9]+)", StoryHandler),
],debug=True)
```

如果你打算下载服务器返回的内容，在语句

```
self.set_header("Content-Type", "text/html")
```

的后面添加

```
self.set_header("Content- Disposition", " attachment; filename="fname.ext"")
```

这时，客户端浏览器就产生了下载行为。

如果你打算告诉服务器一定下载 mp3,pdf,doc 等后缀名的文件，你需要定制 class StaticFileHandler 的 get 方法，利用 header 的 Content- Disposition 属性告知浏览器立即启动下载行为。你可以从这个类的源代码中学习到很多东西。 如使用缓存技术。

问题 3：这里没有上传文件的例子，你将如何解决文件上传问题？

——建议你谷歌“tornado upload file”

问题 4：如何实现输出你自己设计的网页给浏览器？

——建议你使用模板；如果是静态 HTML 页面，参考 StaticFileHandler 实现 get 方法。

任务 3.3：用户认证

进入 ***tornado doc*** ->攻略－请求处理程序和请求参数

请自己阅读这段文档。

并运行这个程序（这几乎是每个网站必备哦）

```
class BaseHandler(tornado.web.RequestHandler):
    def get_current_user(self):
        return self.get_secure_cookie("user")

class MainHandler(BaseHandler):
    def get(self):
        if not self.current_user:
            self.redirect("/login")
            return
        name = tornado.escape.xhtml_escape(self.current_user)
        self.write("Hello, " + name)

class LogoutHandler(BaseHandler):
        def get(self):
                self.set_secure_cookie("user", "")
                self.write("logout!!!")

class LoginHandler(BaseHandler):
    def get(self):
        self.write('<html><body><form action="/login" method="post">'
                   'Name: <input type="text" name="name">'
```

```
                '<input type="submit" value="Sign in">'
                '</form></body></html>')

    def post(self):
                usr = self.get_argument("name")
                if usr == "admin" :
                        self.set_secure_cookie("user",
self.get_argument("name"))
                        self.redirect("/")
                self.redirect("/login")

application = tornado.web.Application([
    (r"/", MainHandler),
    (r"/login", LoginHandler),
        (r"/logout", LogoutHandler),
], cookie_secret="61oETzKXQAGaYdkL5gEmGeJJFuYh7EQnp2XdTP1o/Vo=" ,debug=True)
```

假设用户输入 http://locahost:8888/ 后，你能写出浏览器与服务器交互过程吗？

浏览器 ->	<- 服务器
GET / HTTP/1.1 ……(包含 cookies)	
	调用 MainHandler.get cookies 中没有用户名，Response 输出 发送重定位/login (具体 header 参源代码)
GET /login HTTP/1.1 ……(包含 cookies)	
	调用 LoginHandler.get Content-Type: text/html; charset=utf-8 ……(包含 cookies) HTML login 页面文档
浏览器根据服务器 header 指示，展示内容 当你输入 message=123 点 submit 按钮后 POST /login HTTP/1.1 ……(包含 cookies) message =123 &	
	调用 Mainhandler.post 设置 cookies 发送 cookies 与重定位 /
GET / HTTP/1.1 ……(包含 cookies)	
	调用 MainHandler.get cookies 中有用户名，Response 输出 Content-Type: text/html; charset=utf-8

	……(包含 cookies) HTML main 页面文档
浏览器显示主页	

问题：你能写出用户输入 http://localhost:8888/logout 后浏览器与服务器的交互过程？

作业与练习：

1、简答以下问题

1）什么是 URL 的 Query String，并用一个例子说明

2）一个网页中可以有多少 HTML Form

3）HTML Form 提交方法有几种，它们的区别是什么

4）HTML 的输入 Controls 的 tag 有哪些

5）请根据对 Tornado 服务程序的理解，画出服务端处理 HTTP 请求的数据处理流程

6）使用 render 模板和 write 网页到客户端（浏览器）都可以在客户端形成网页。请根据应用场景，分析两种方式的优缺点。

7）如何检测用户是否登陆？

如果每个 Handler 的 GET / POST 方法都独立检测用户登录状态，大一点的程序很难编写与维护。你有解决方案吗，请 Email 给你老师。

二、操作与实践

1、Lab 4 – Forms and PHP （请使用 Tornado 技术代替 PHP）