

搞懂了这几点，你就学会了Web编程

原创：刘欣

做了那么多年Web编程，仔细想想，其实本质上就那点事儿，你抓住了几个重点问题，学起来一点都不难。

1. 理解浏览器/服务器结构 (B/S)

B/S 是从 90年代的客户端/服务器端发展而来，共同点都是由一个（或一组）服务器来服务多个客户端。

差别在于：首先，C/S结构的客户端可能是由不同语言编写的，例如VB,Delphi, PowerBuilder等，B/S结构中浏览器成为了一个通用的客户端，程序以Web的方式呈现，不需要安装，服务器端的升级就意味着所有客户端的升级，这和C/S相比是个翻天覆地的变化。

其次B/S的访问协议也标准化为HTTP(s)，而不是原来各种各样的私有协议。

最后B/S结构中的服务器面向全球用户访问，而不像C/S那样仅仅是局域网，所以压力更大，挑战更大。

2. Web页面是怎么组成的？

简单来说就是HTML + CSS + Javascript，我们看到的Web界面就是由这三者组成。

HTML负责结构，CSS负责展现，而Javascript负责行为。

我们说的前端开发也主要是做这一块，对于前端工程师，需要能理解DOM模型，以及如何通过javascript(例如jQuery等框架)来操作DOM模型。

3. 浏览器和服务是怎么打交道的？

当然是HTTP！HTTP说穿了就是浏览器和服务器聊天是的一种约定，这个约定确保双方互相理解。

完整的HTTP是非常复杂的，《HTTP权威指南》一书厚达700多页。

其实我们最常用，也是最重要的也就那么几点：

1. GET 和 POST。GET从服务器端获取数据，POST向服务器端发送数据(由此引出图片上传问题)
2. HTTP是个没有状态的协议，需要通过额外的机制来维持状态（例如登录状态），常用的方法就是cookie。
3. 理解HTTP 状态码
4. 理解 同步 vs 异步(由此引出AJAX，以及jQuery等框架)

4. URL 和 代码的映射

理解url和代码之间的关联，例如 www.xxx.com?action=login 这样的url是怎么和后端的业务代码关联起来的？

这样的规则是在哪里定义的？用代码、注解还是配置文件？

后端的业务代码该如何组织？相信现在不会有人把业务逻辑都写到Servlet当中了，所以需要很多MVC框架像Struts, SpringMVC 来组织代码，让系统清晰易懂。

5. 数据的验证、转换和绑定

如何保证浏览器发过来的数据是符合要求的？

例如不能为空、不超过8个字符、两个密码必须相等....，出错了得给出错误提示。

浏览器发过来的数据都是形如username=liuxin&password=123456这样简单的文本，但是后台程序却有着丰富的数据类型，什么String, Date, Integer等等。所以要把文本变成指定语言的类型。

类型转换以后，后端的业务代码怎么才能有效的使用呢？

最简单的就是弄一个key : value 这个样的Map 出来，业务代码直接用map.get(key) 即可。

高级一点的可以把页面发来的数据直接绑定到对象的属性上，并且支持数组，嵌套等复杂的结构。

例如user.name=liuxin&user.password=123456 可以绑定到一个叫User的对象，其中有两个属性userName和password。

6. Web安全

如何防止黑客利用SQL 注入，跨站脚本攻击，跨站请求伪造等手段来攻击系统？

7. 数据库访问

这一块是比较麻烦的，毕竟面向对象(OO)世界和关系(Relational)数据库之间存在着天然的鸿沟。

对于简单的应用，直接写点JDBC就够用了，只需要掌握Connection, Statement, Resultset这三个基础。

复杂点的需要用O/R Mapping 框架来搞定，例如 Hibernate, MyBatis，还有RoR的ActiveRecord。

这其中比较棘手的就是表之间的关联，就是所谓的一对多，一对一，多对多这样的关系，如何在面向对象的世界里描述。

扩展开去，还需要处理连接池，事务，锁 等各种烦人问题。

8. 用什么技术来生成Web页面？

这里说的Web页面就是第2点中的页面，包括HTML, CSS, Javascript。

能不能直接用Servlet的PrintWriter 直接输出HTML？当然可以，只是以后就没有人看懂了。

现在用来创建Web页面的技术多如牛毛：例如 JSP, Velocity, Freemaker, Groovy 等等，他们都有一个共同点：**模板技术**。

说白了就是有一个HTML的模板，里边可以嵌入代码，这个模板在运行时（例如在Tomcat当中）就可以根据输入的不同而生成不同内容的Web界面了。

无论哪种模板，都需要面对一个重要问题：如何展示从业务逻辑层发送来的数据？这一步骤其实和第5步中的数据绑定有密切关系。因为这一步需要确定诸如user.name, user.password这样的字段名称。

9. 如何把对象变成XML或者JSON字符串？

由于AJAX以及手机端的存在，对于一个URL的请求，他们要求的返回值通常不是HTML页面，而是XML或者JSON数据，此时需要有框架把对象转化成相应的字符串。

搞定了Web这些基础的东西，在公司里做一个Web程序员应该不在话下了，接下来需要学习的就是像高并发，缓存，搜索，分布式等高级的内容了。