# 第六天上课笔记

https：安全的http协议，目前推荐的协议

比如支付是必须使用https协议的；

HTTP+SSL：

为什么要加密？数据在网络上传输时，存在安全问题，使用https协议对传输的数据进行加密，能够很好的保护数据，防止数据被篡改。

到客户端之后如何正常展示？SSL证书

默认端口：443

### 客户端常见的请求方式：

1. 在浏览器中输入网址发出get请求；
2. css,js,img自动发出get请求；
3. 超连接发出get请求

<a href=‘getUser?name=wgr&pwd=123’>得到用户</a>

1. js发出get请求： window.location.href='url?name=wgr&pwd=123'；
2. 表单提交发出请求(get或post)；
3. ajax请求(get或post)

### 我们做一个功能点：

1. 做一个页面：HTML、CSS、bootstrap，使用调试工具的element进行调试；
2. 服务器端进行数据处理：nodejs、MySQL，在network里面跟踪请求结果；
3. 使用js把前端页面和后台服务器友好的结合起来：JS、jQuery、AJAX，在console里面调试；

### MVC的开发模式：模型-视图-控制器

1. M：model，模型，负责数据处理；
2. V：view，视图，负责页面效果的；
3. C：controller，控制器，负责流程控制；

### AJAX：

具体描述参考jQuery阶段的pdf里面的ajax；

AJAX: Asynchronous JavaScript And XML，不是新技术，是多个技术的组合；

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 是一种用于创建快速动态网页的技术。通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

Async：异步，和异步对应的是 同步；

JavaScript：是一种脚本语言，运行在客户端的js的解释器里面；

XML： 可扩展的 标记 语言，表示一种通用的数据格式；

依赖于JS的一个核心模块：XmlHttpRequest，通常简称为XHR。

XHR的属性及方法：







### 把一般的表单提交改造成ajax处理：

1. submit改成button，此时action和method失效；
2. 给button绑定点击事件；
3. 事件处理函数里面进行ajax操作；

## JSON简介

JSON（JavaScript Object Notation）一种简单的数据格式，比xml更轻巧。JSON是JavaScript原生格式，这意味着在JavaScript中处理JSON数据不需要任何特殊的API或工具包。

JSON的规则很简单：对象是一个无序的“‘名称/值’对”集合。一个对象以“{”（左括号）开始，“}”（右括号）结束。每个“名称”后跟一个“:”（冒号）；“‘名称/值’对”之间使用“,”（逗号）分隔。

处理JSON数据的对象：JSON.parse JSON.stringify

[{"username":"高义昌1", "age":"20"},{"username":"高义昌2", "age":"20"},{"username":"高义昌3", "age":"20"}]

## GET和POST：

是两种数据提交方式；

Get方式： Baidu.com/s?ie=utf-8&f=8&rsv\_bp=0&rsv\_idx=1&tn=baidu

POST方式：数据在后台传递

GET 还是 POST？

* 与 POST 相比，GET 更简单也更快，并且在大部分情况下都能用。Get更多的用于资源定位；
* <https://item.taobao.com/item.htm?id=567878233573>
* <https://item.taobao.com/item.htm?id=567063908899>
* 然而，在以下情况中，请使用 POST 请求：
  + 无法使用缓存文件（更新服务器上的文件或数据库）；
  + 向服务器发送大量数据（POST 没有数据量限制）；

发送包含未知字符的用户输入时，POST 比 GET 更稳定更可靠

### POST数据提交：

Ajax设置请求头：

Content-Type: application/x-www-form-urlencoded

### 服务器端如何接收post数据：

Nodejs是分块接收客户端提交上来的数据；

Nodejs里面使用on绑定事件；

每次接收一块数据，都会触发请求头的data事件，直到所有的数据接收完成，触发end事件。

url.parse(url, true);