# 第七天上课笔记

### 通信基础：

客户端：硬件和软件；

服务器：硬件和软件，MySQL、web服务器（Apache、nodejs里面的http模块创建）；

大体分成五层：

1. 实体层：网卡、网线（双绞线）、WiFi、光纤；
2. 链接层：数据包、头消息（发送的MAC地址和接收的MAC地址）、MAC：物理地址（全球唯一）、广播的方式、局域网；
3. 网络层：IP（根服务器负责IP的分配）、IPv4（32位）、域名（域名解析）、ping（查看域名的IP地址）；
4. 传输层：端口（网卡分配给应用的编号，http：80，https：443，MySQL：3306）；
5. 应用层：应用协议(FTP、UDP、HTTP)

### Nodejs：

是js的一个运行环境：EcmaScript+libuv（作者提供的相关模块）；

事件驱动、非阻塞I/O操作；

Chrome V8负责解析JS代码；

进程（启动一个应用就是一个进程）、线程（每个进程至少一个线程，称为主线程）、

阻塞、非阻塞、异步、同步；

模块化：一个文件看成一个模块；

1. 系统模块：fs、os（EOL）、querystring（parser）、url（parse，第二个参数设置为true）、http（负责提供web服务）、\_\_dirname、\_\_filename、global；
2. 第三方模块：

npm：node package manager，nodejs包管理器；

项目配置文件的生成：npm init -y

npm i~~nstall~~ mysql express jquery --save

npm uninstall express

**指定版本安装：**

在默认安装的时候，安装的是最新版本；

npm i [mysql@2.10.0](mailto:mysql@2.10.0) -g --save

1. 自定义模块：自已创建一个模块；

模块相关的保留字：

1. require：引入模块；
2. exports：导出属性，一个模块里面可以导出多次；
3. module.exports：导出整个模块，一个模块(对应一个js文件)里面只能导出一次；

### HTTP：

超文本传输协议，无状态的，只能客户端发起请求；

Request、response；

目前的版本：HTTP / 1.1:

**Referer:** <http://nodejs.cn/api/http.html> 前一个访问地址

1. 建立连接；
2. 客户端发起请求；
3. 服务器端响应请求；
4. 断开连接；

一个页面里面所有的资源都来自服务器，比如一个图片就是一个请求，一个css文件也是一个请求。

### AJAX：

Async Javascript And Xml:

String、xml、**json**；

核心对象：XMLHttpRequest：XHR

let xhr = new XMLHtppRequest();

xhr.open(‘get|post’, url, true|false);

xhr.send(string);

传值：

Get：直接在url后面跟上要传的值（url.parse(req.url, **true**)）；

Post：使用&把数据连接成字符串，作为send的实参，提交给服务器（会触发req上面的data和end事件，使用on绑定事件）；

接收服务器响应的内容：

xhr.onreadystatechange = function(){

if(xhr.readystate == 4 && xhr.status==200){

xhr.responseText;

xhr.responseXML;

拿到数据之后，进行js操作

}

}

**JSON：**parse、stringify；

### 前后端分离开发：

**同源策略**：为了安全，禁止一个网站通过ajax方式请求另外一个网站

必须是同一个“访问地址”才是同源：

1. 必须是相同的协议：http、 https、 file；
2. 域名必须相同，如果有子域名，子域名也必须相同:passport.jd.com；
3. 端口必须一致： 80、81；

如果是在一个服务器上，不满足上面三个条件一样是跨域；

如果多个域名解析到一个IP服务器上，也是跨域；

跨域：

1. JSONP：只能获取数据；
2. Proxy：代理；
3. CORS：跨域(cross)资源(resource)共享(share)，可以获取和提交数据；

### JSONP:

Script标签属性src支持跨域；

### CORS:跨域资源共享

是在服务器端设置：

// 允许的域名，多个域名之间使用逗号分号，\*代表所有的客户端

res.setHeader('Access-Control-Allow-Origin', 'http://localhost');

// 允许的请求方法

res.setHeader('Access-Control-Allow-Methods','PUT,POST,GET,DELETE,OPTIONS');

// res.setHeader("Content-Type", "application/json;charset=utf-8"); //响应数据类型

// res.setHeader("Access-Control-Allow-Credentials",true); //带cookies

### 把一个项目分成两块：

客户端：

HTML、CSS、JS、BootStrap、jQuery；

服务器端：

nodejs、mysql；

服务器端需要设置允许跨域请求；

Ajax负责客户端和服务器端数据通信；