初识Ajax

ajax最早产生于2005年，Ajax表示Asynchronous JavaScript and XML(异步JavaScript和XML)，但是它不是像HTML、JavaScript或CSS这样的一种“正式的”技术，它是表示一些技术的混合交互的一个术语（JavaScript、Web浏览器和Web服务器），它使我们可以获取和显示新的内容而不必载入一个新的Web页面。增强用户体验，更有桌面程序的感觉。

Ajax可以做什么?

- 显示新的HTML内容而不用载入整个页面
- 提交一个表单并且立即显示结果
- 登录而不用跳转到新的页面
- 星级评定组件
- 遍历数据库信息加载更多而不刷新页面
- ……

Ajax基础知识铺垫

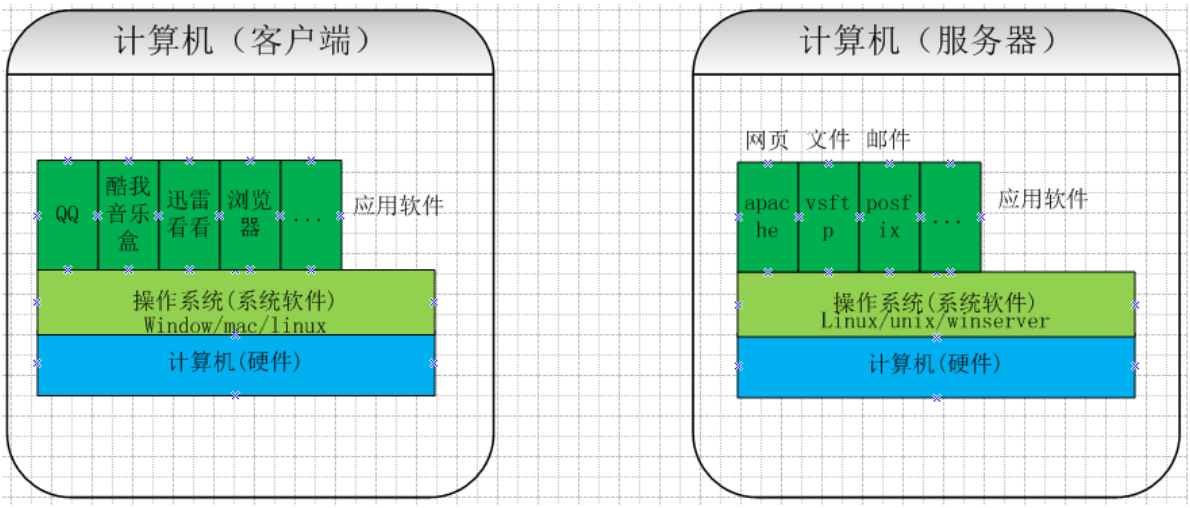
- 客户端与服务器（计算机概述）
- 通信协议（http/ftp/smtp/pop3...）
- 网络、IP地址、端口、域名...
- 静态网站、动态网站
- 同步与异步
- ajax实现异步请求效果

客户端与服务器

- 开发好的网页放在那里?

  通过浏览器查看一个网页，使用调试工具查看页面信息

客户端与服务器对比



客户端

浏览器、app、应用软件

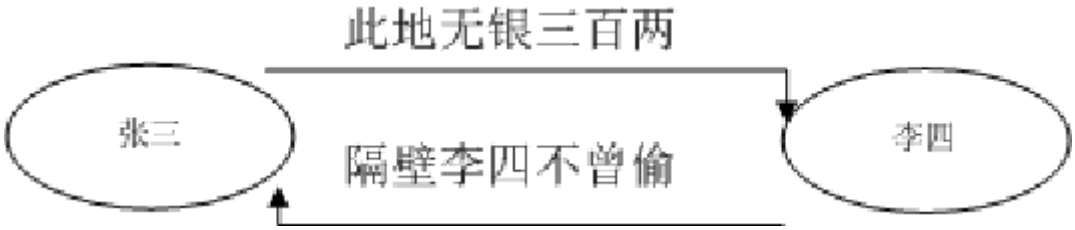
服务器

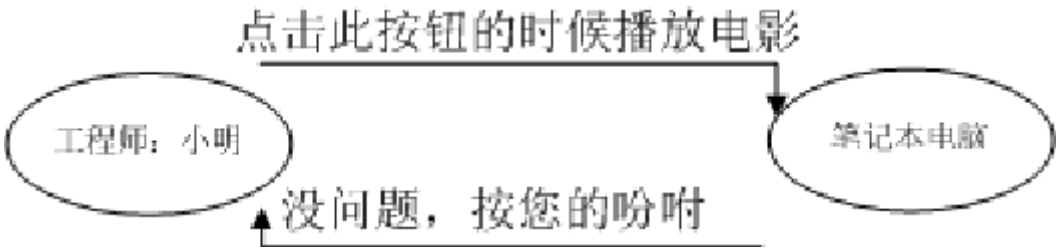
提供网络服务的计算机（网页/下载/邮件...）

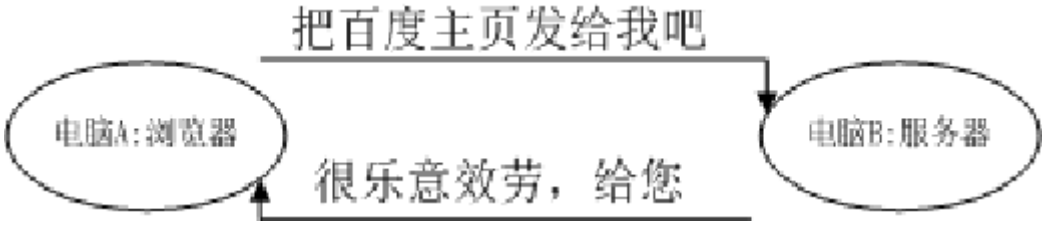
网络相关概念

- IP地址
- 域名
- 端口
- DNS

通信协议

- 张三与李四
- 

- 程序员与计算机
- 

- 客户端与服务器
- 

原生Ajax详解-发送请求步骤

发送ajax请求步骤：
1、创建XMLHttpRequest对象
2、准备发送
3、执行发送动作
4、指定回调函数

原生Ajax详解-请求方法

get

get请求参数在url中传递
需要注意编码问题

post

post请求参数在请求体中传递
需要设置请求头信息

原生Ajax详解-回调函数

onreadysatechange

xhr.readyState

0 xhr对象初始化
1 执行发送动作
2 服务器端数据已经完全返回
3 数据正在解析
4 数据解析完成，可以使用了

xhr.status

200 数据相应正常
404 没有找到资源
500 服务器端错误

同步与异步

页面加载的同步与异步（白屏与不刷新）

普通的页面效果：w3school.org
页面不刷新效果：评论加载

描述两者之间的行为方式

同步 彼此等待 --- 阻塞
异步 各做各的 --- 非阻塞

浏览器Ajax请求服务器

①：浏览器让xhr去跟服务器要点儿数据
②：浏览器接着干别的事情
③：xhr去向服务器请求数据
④：服务器返回数据给xhr
⑤：xhr通知浏览器数据回来了
⑥：浏览器收到xhr返回的数据渲染页面

```
// 使用Ajax发送请求需要如下几步：
            // 1、创建XMLHttpRequest对象
            var xhr = new XMLHttpRequest();
            // 2、准备发送
            // 参数一：请求方式（get获取数据；post提交数据）
```

```
            // 参数二：请求地址
            // 参数三：同步或者异步标志位，默认是true表示异步，false表示同步
            xhr.open('get', 'http://localhost:3000/test?username=' + uname +
'&password=' + pw, true);
            // 3、执行发送动作
            xhr.send(null);
            // 4、指定回调函数
            xhr.onreadystatechange = function () {
                if (xhr.readyState == 4) {
                    // 0  xhr对象初始化
                    // 1  执行发送动作
                    // 2  服务器端数据已经完全返回
                    // 3  数据正在解析
                    // 4  数据解析完成，可以使用了
                    if (xhr.status == 200) {
                        // 200 数据相应正常
                        // 404 没有找到资源
                        // 500 服务器端错误
                        var data = xhr.responseText;
                        console.log(data);
                    }
                }
            }
```

Ajax初步封装

Ajax初步封装

抽取参数
理解回调函数的调用方式

jQuery的Ajax

jQuery的Ajax使用

$.ajax()

Ajax跨域

同源策略

- 同源策略是浏览器的一种安全策略，所谓同源指的是请求URL地址中的协议、域名和端口都相同，只要其中之一不相同就是跨域
- 同源策略主要为了保证浏览器的安全性
- 在同源策略下，浏览器不允许Ajax跨域获取服务器数据

跨域解决方案

- jsonp
- 服务端设置请求头
- ningx

JSONP原理

- 静态script标签的src属性进行跨域请求
- 动态创建script标签，通过标签的src属性发送请求

服务端设置请求头

```
//允许跨域请求
app.all("*", (req, res, next) => {
  res.header("Access-Control-Allow-Origin", "*");
  res.header("Access-Control-Allow-Methods", "POST,GET,DELETE,PUT");
  res.header("Access-Control-Allow-Headers", "Content-Type,X-Requested-
With,token");
  res.header("Content-Type", "application/json;charset=utf-8");
  next();
})
```