Ajax的理解

**目录：**

初始化服务器

Ajax的概念

发送HTTP请求的方式，及优缺点

url

form

a

link,img,script\_src

window.open

Ajax

Ajax发送请求的步骤

new XMLHttpRequest()() / ActiveXObject()

ajax.open(传输方式，地址，同步异步)

ajax.send()

事件监听onload

属性：

ajax.responseText

ajax.responseXML

Ajax的兼容性

构造函数：ActiveXObject() / XMLHttpRequest()

ajax.onload /ajax.onreadystatechange，，属性ajax.readystate

Ajax状态码的理解

客户端向服务端**请求数据**的**方式**

path

queryString

请求头，ajax.setRequestHeader(key,val)

请求主体，send(数据)，修改ajx.setRequestHeader(‘content-style’,val)

客户端接收服务端数据，**数据解析**的方式

普通字符串

HTML

JSON

XML

客户端向服务端发送的数据——**二进制数据**FormData的处理

应用：上传、进度、统计速度

Ajax乱码

问题：ie下.querySting传非ascil会乱码

解决：转urlUnicode——%＋16进制

Ajax缓存

缓存的作用

解决：‘&’+Math.random()

**主体：**

**初始化服务器**：

初始化package.json

>**npm init**

下载express.js中间键

>**npm i express**

搭建服务端serve.js

1. 生成express

2. express()生成serve

3. 引入静态文件

4. get/post获取请求——发送数据

5. 监听



发送HTTP请求的方式，及优缺点

url

form

a

link,img,script\_src

window.open

**Ajax的概念**

Ajax全称“Asynchronous Javascript And XML”（异步 JavaScript 和 XML）。

即用js异步的方式向服务器发送请求，获取数据。

作用：

1.不用开新页面，因为不用借助浏览器就可以像HTTP发请求，和img类似

2.不渲染页面

3.请求的数据保存在对象的属性下，js可以操作，实现无刷新请求

**Ajax发送请求的步骤**

创建对象：

new XMLHttpRequest()() / ActiveXObject()

new一个ajax的构造函数，生成ajax对象，这样就能

配置参数：

open(get/post，地址，true/false)

true/false：

发送请求：

send()

事件监听：

onload

属性：

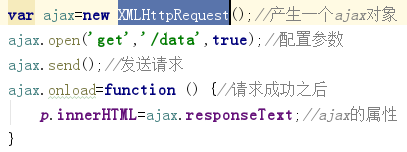
.responseText

.responseXML————是字符串

注意：

onload监听发送请求有没有完成，需要在发送请求之前，因为如果send事件发生的很快，onload根本就监听不到。所以我们一般将监听放在要监听事件的前面。

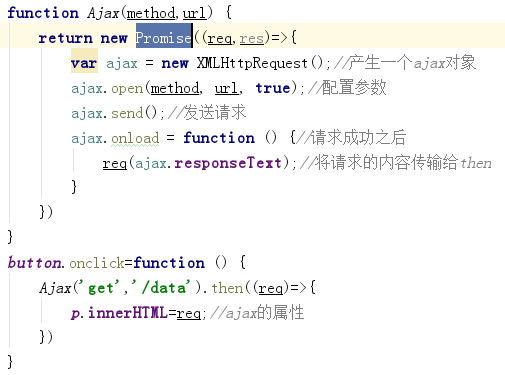
ajax是异步的操作，所以send()发生永远会在onload监听之后，所以在ajax中可以这么放置。



用**promise包装ajax**，这样可以实现做不同异步事件：

等异步事件全部完成之后，通过promise.all()做某个事情；

或只要其中一个异步事件完成，就可以通过promise.race()做某个事情；



**Ajax的兼容性**

构造函数：ActiveXObject() / XMLHttpRequest()

XMLHttpRequest ie9+支持

ajax.onload / ajax.onreadystatechange，，属性ajax.readystate

ie9以下不支持onload事件

Ajax状态码readystate的理解：

ajax对象下的一个属性，该属性表示当前的工作状态，一共有5个值

0：初始化

1：发送请求

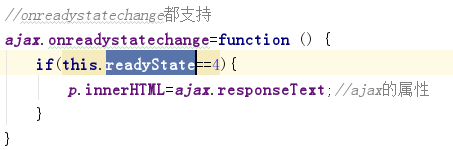
2：请求完成

3：解析数据

4：解析完成

每次 readyState 状态的值发生改变的时候，就会触发onreadystatechange事件





客户端向服务端**请求数据**的方式

path——req.params

queryString——req.query

请求头，ajax.setRequestHeader(key,val)——req.headers

请求主体——req.body

服务端：

服务端要引入文件**body-parser，并且设置接收的格式**

客户端：

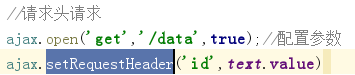
请求方式：post

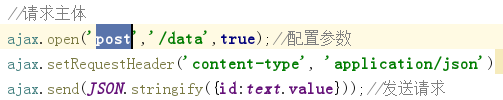
修改文件样式：ajx.setRequestHeader(‘content-style’,val)

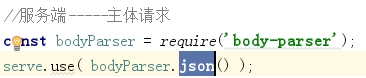
send(传输的数据)

clipboard.png

clipboard.png







客户端向服务端发送的数据——二进制数据FormData的处理

服务端：

安装：>npm i multer

创建multer

处理提交的二进制数据——设置上传数据存储的目录和重命名的文件名称

生成二进制中间键unload

post数据处理unload.sigle('')/.array('')

客户端：

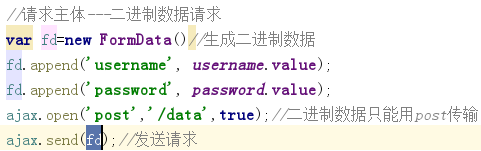
创建二进制数据：new FormData，构造函数

post传输数据

fd.append(key,val)

不用在请求头上加数据类型，系统在请求时会自动加的

send(fd)



客户端接收服务端数据，**数据解析**的方式

普通字符串

HTML

ajax.**responseText**

JSON

***JSON***.parse(ajax.**responseText**)

XML

ajax.**responseXML**

在ajax下还有一个属性，如果返回的字符串是xml的，那么ajax会对这个内容进行解析，首先会把这个内容直接赋值给responseText属性，同时也会对这个内容进行xml解析，解析DOM对象赋值给xhr.responseXML



Ajax乱码

问题：ie下.querySting传非ascil会乱码

解决：将传输的数据转成urlUnicode——%＋16进制

Ajax缓存

缓存的作用

当我们通过HTTP请求向后台请求的时候，电脑会这个url请求的数据保存在电脑里面，如果我们下次再通过这个url请求时，浏览器就会去电脑里面找到这个文件，而不会向服务器发送请求。

作用：提供服务端性能，提高客户端性能，不用请求相同的内容，比如图片

解决：‘&’+Math.random()

二进制数据——应用：Ajax上传文件

流程：

创建对象：new XMLHttpRequest()()

配置参数：open(post，地址，true)

创建二进制数据fd

fd.append(key,value)——key值与后端对应

事件监听onload

子对象

ajax.upload是一个子对象，处理数据传输过程中发生的事情

子对象.onloadstart——数据传输的开始

子对象.oprogress——数据传输过程中

事件对象

e.loaded 已经传输好的数据的大小

e.totle 数据的总大小

子对象.onload——数据传输完成

发送请求：send(fd)

应用：上传文件、进度、统计速度