

1 Ajax

1. Ajax介绍

1.1. Ajax概述

我们前端页面中的数据，如下图所示的表格中的学生信息，应该来自于后台，那么我们的后台和前端是互不影响的2个程序，那么我们前端应该如何从后台获取数据呢？因为是2个程序，所以必须涉及到2个程序的交互，所以这就需要用到我们接下来学习的Ajax技术。

编号	姓名	年龄	性别	成绩	等级
1	Tom	20	男	78	及格
2	Rose	18	女	86	优秀
3	Jerry	26	男	90	优秀
4	Tony	30	男	52	不及格

Ajax: 全称Asynchronous JavaScript And XML，**异步的**JavaScript和XML。其作用有如下2点：

- 与服务器进行数据交换：通过Ajax可以给服务器发送请求，并获取服务器响应的数据。
- 异步交互：可以在**不重新加载整个页面**的情况下，与服务器交换数据并**更新部分网页**的技术，如：搜索联想、用户名是否可用的校验等。

1.2. Ajax作用

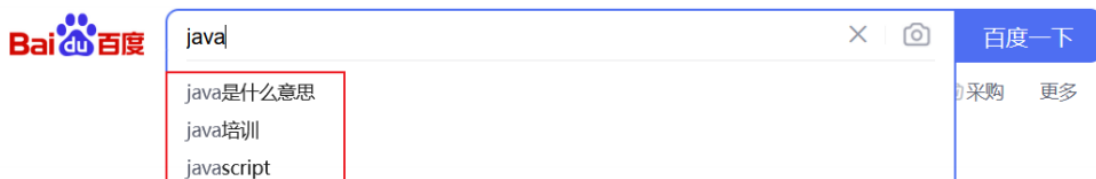
- 与服务器进行数据交互

如下图所示前端资源被浏览器解析，但是前端页面上缺少数据，前端可以通过Ajax技术，向后台服务器发起请求，后台服务器接受到前端的请求，从数据库中获取前端需要的资源，然后响应给前端，前端在通过我们学习的vue技术，可以将数据展示到页面上，这样用户就能看到完整的页面



- 异步交互

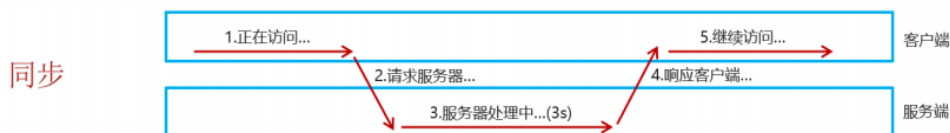
可以在**不重新加载整个页面**的情况下，与服务器交换数据并**更新部分网页**的技术。如下图所示，当我们再百度搜索java时，下面的联想数据是通过Ajax请求从后台服务器得到的，在整个过程中，我们的Ajax请求不会导致整个百度页面的重新加载，并且只针对搜索栏这局部模块的数据进行了数据的更新，不会对整个页面的其他地方进行数据的更新，这样就大大提升了页面的加载速度，用户体验高。



1.3. 同步异步

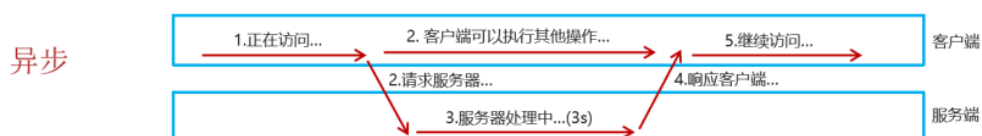
针对于上述Ajax的局部刷新功能是因为Ajax请求是异步的，与之对应的有同步请求。接下来我们介绍一下异步请求和同步请求的区别。

- 同步请求发送过程如下图所示：



浏览器页面在发送请求给服务器，在服务器处理请求的过程中，浏览器页面不能做其他的操作。只能等到服务器响应结束后才能，浏览器页面才能继续做其他的操作。

- 异步请求发送过程如下图所示：



浏览器页面发送请求给服务器，在服务器处理请求的过程中，浏览器页面还可以做其他的操作。

1.4. 原生Ajax

对于Ajax技术有了充分的认知了，我们接下来通过代码来演示Ajax的效果。此处我们先采用原生的Ajax代码来演示。因为Ajax请求是基于客户端发送请求，服务器响应数据的技术。所以为了完成快速入门案例，我们需要提供服务器端和编写客户端。

服务器端接口

随机获取笑话的接口

- 请求地址：<https://autumnfish.cn/api/joke/list>
- 请求方法：get
- 请求参数：num

参数名	参数说明	备注
num	笑话条数	类型为数字

客户端

客户端的Ajax请求代码如下有如下4步，接下来我们跟着步骤一起操作一下。

1. 准备数据地址：<https://autumnfish.cn/api/joke/list>
2. 创建XMLHttpRequest对象：用于和服务器交换数据

```
//1. 创建XMLHttpRequest
var xmlhttpRequest = new XMLHttpRequest();
```

3. 向服务器发送请求

```

//2. 发送异步请求
xmlHttpRequest.open('GET', 'https://autumnfish.cn/api/joke/list');
xmlHttpRequest.send();//发送请求

```

4. 获取服务器响应数据

```

//3. 获取服务响应数据
xmlHttpRequest.onreadystatechange = function(){
    if(xmlHttpRequest.readyState == 4 && xmlHttpRequest.status == 200){
        document.getElementById('div1').innerHTML =
xmlHttpRequest.responseText;
    }
}

```

1.5. JQuery实现方式

1.5.1. \$.ajax()

```

$.ajax({
    type: 'GET',          //请求方式 （GET/POST） 默认GET
    url: 'servlet'        //请求的路径
    data: {'username': 'admin'}, //请求的参数
    dataType: 'json',     //响应的数据格式
    success: function(data){ //data接受服务器响应的值
        //响应成功后的回调函数
    }
})

```

1.5.2. \$.get()

```

//基本语法
$.get(url, data, function(data, status, xhr), dataType)

```

参数	描述
url	必须，规定加载资源的路径
data	可选，发送至服务器的数据
function(data, status, xhr)	可选，请求成功时执行的函数data表示从服务器返回的数据status表示请求的状态值xhr表示当前请求相关的XMLHttpRequest对象
dataType	可选，预期的服务器响应的数据类型（ xml、html、text、script、json、jsonp）

1.5.3. \$.post()

```

$.post()

```

在jQuery中，发送GET请求调用\$.get()方法，发送POST请求调用\$.post()方法，两个方法使用方式完全相同，替换两者的方法名就可以在GET请求和POST请求方式之间切换。

2 Axios

1. Axios介绍

- 介绍: Axios 对原生的Ajax进行了封装, 简化书写, 快速开发。功能强大的网络请求库
- 官网: <https://www.axios-http.cn/>

2. Axios入门

引入Axios的js文件

```
// 离线Axios库
<script src="js/axios-0.18.0.js"></script>

<!-- 官网提供的 axios 在线地址 -->
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
```

使用Axios发送请求, 并获取响应结果

- 发送get请求

```
axios({
  method: "get",
  url: "https://autumnfish.cn/api/joke/list"
}).then((result) => {
  console.log(result.data);
});
```

服务器端接口

用户注册接口

- 请求地址: <https://autumnfish.cn/api/user/reg>
- 请求方法: post
- 请求参数: username

参数名	参数说明	备注
username	用户名	不能为空

发送post请求

```
axios({
  method: "post",
  url: "https://autumnfish.cn/api/user/reg",
  data: "username=熊初墨"
}).then((result) => {
  console.log(result);
});
```

- axios()是用来发送异步请求的，小括号中使用js的JSON对象传递请求相关的参数：
- method属性：用来设置请求方式的。取值为 get 或者 post。
- url属性：用来书写请求的资源路径。如果是 get 请求，需要将请求参数拼接到路径的后面，格式为：url?参数名=参数值&参数名2=参数值2。
- data属性：作为请求体被发送的数据。也就是说如果是 post 请求的话，数据需要作为data 属性的值。
- then() 需要传递一个匿名函数。我们将 then()中传递的匿名函数称为 **回调函数**，意思是该匿名函数在发送请求时不会被调用，而是在成功响应后调用的函数。而该回调函数中的 resp 参数是对响应的数据进行封装的对象，通过 resp.data 可以获取到响应的数据

3. Axios请求方式

- axios.get(url [, config])
- axios.delete(url [, config])
- axios.post(url [, data[, config]])
- axios.put(url [, data[, config]])

3.1. 推荐写法

```
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
axios.get(地址?key=value&key2=values).then(function(response){},function(err){})

axios.post(地址,{key:value,key2:value2}).then(function(response){},function(err){})
```

3.2. 总结

- axios必须先导入才可以使用
- 使用get或post方法即可发送对应的请求
- then方法中的回调函数会在请求成功或失败时触发
- 通过回调函数的形参可以获取响应内容,或错误信息

4. axios+vue

axios如何结合vue开发网络应用

随机获取笑话的接口

- 请求地址：<https://autumnfish.cn/api/joke/list>
- 请求方法：get
- 请求参数：无
- 响应内容：一条随机笑话

实例：

```
/*
  接口：随机获取一条笑话
  请求地址：https://autumnfish.cn/api/joke
  请求方法：get
  请求参数：无
  响应内容：随机笑话
*/
var app = new Vue({
```

```
    el: "#app",
    data: {
      joke: "很好笑的笑话"
    },
    methods: {
      getJoke: function() {
        var that = this;

        axios.get("https://autumnfish.cn/api/joke").then(function(response) {
          console.log(response.data);
          that.joke = response.data;
        }, function(err) {
          //
        })
      }
    }
  },
  })
</script>
```

4.1. 总结

- axios回调函数中的this已经改变,无法访问到data中数据
- 把this保存起来,回调函数中直接使用保存的this即可
- 和本地应用的最大区别就是改变了数据来源

5 Vue结合网络数据开发应用

1. 案例：天气预报案例

主要功能：

- 回车查询
 - 按下回车
 - 查询数据
 - 渲染数据
- 点击查询
 - 点击城市
 - 查询数据
 - 渲染数据

天气接口

请求地址：http://wthrcdn.etouch.cn/weather_mini

请求方法：get

请求参数：city（查询的城市名）

响应内容：天气信息

注意：

应用的逻辑代码建议和页面分离，使用单独的js文件编写

axios回调函数中this指向改变了，需要额外的保存一份

服务器返回的数据比较复杂时，获取的时候需要注意层级结构

2. 综合应用：悦听音乐

主要功能：

- 歌曲搜索
 - 按下回车
 - 查询数据
 - 渲染数据

歌曲搜索接口

- 请求地址：<https://autumnfish.cn/search>
- 请求方法: get
- 请求参数: keywords(查询关键字)
- 响应内容: 歌曲搜索结果

服务器返回的数据比较复杂时，获取的时候需要注意层级结构通过审查元素快速定位到需要操纵的元素

- 歌曲播放
 - 点击播放
 - 歌曲地址获取
 - 歌曲地址设置

歌曲播放接口：

- 请求地址:<https://autumnfish.cn/song/url>
- 请求方法:get
- 请求参数:id(歌曲id)
- 响应内容:歌曲url地址

歌曲id依赖歌曲搜索的结果，对于不用的数据也需要关注

- 歌曲封面
 - 点击播放
 - 歌曲封面获取
 - 歌曲方面设置

歌曲详情获取

- 请求地址:<https://autumnfish.cn/song/detail>
- 请求方法:get
- 请求参数:ids(歌曲id)
- 响应内容:歌曲详情(包括封面信息)

在vue中通过v-bind操纵属性,本地无法获取的数据，基本都会有对应的接口

- 歌曲评论
 - 点击播放
 - 歌曲评论获取
 - 歌曲评论渲染

热门评论获取

- 请求地址:<https://autumnfish.cn/comment/hot?type=0>
- 请求方法:get
- 请求参数:id(歌曲id,地址中的type固定为0)
- 响应内容:歌曲的热门评论

在vue中通过v-for生成列表

- 播放动画
 - 监听音乐播放@play
 - 监听音乐暂停@pause="pause"
 - 操作类名

audio标签的play事件会在音频播放的时候触发

audio标签的pause事件会在音频暂停的时候触发

通过对象的方式设置类名，类名生效与否取决于后面值的真假

- mv播放
 - mv图标显示
 - mv地址获取
 - 遮罩层
 - mv地址设置

mv地址获取

- 请求地址:<https://autumnfish.cn/mv/url>

- 请求方法:get
- 请求参数:id(mvid,为0表示没有mv)
- 响应内容:mv的地址