

AJAX



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



AJAX介绍及axios基本使用

AJAX是什么?

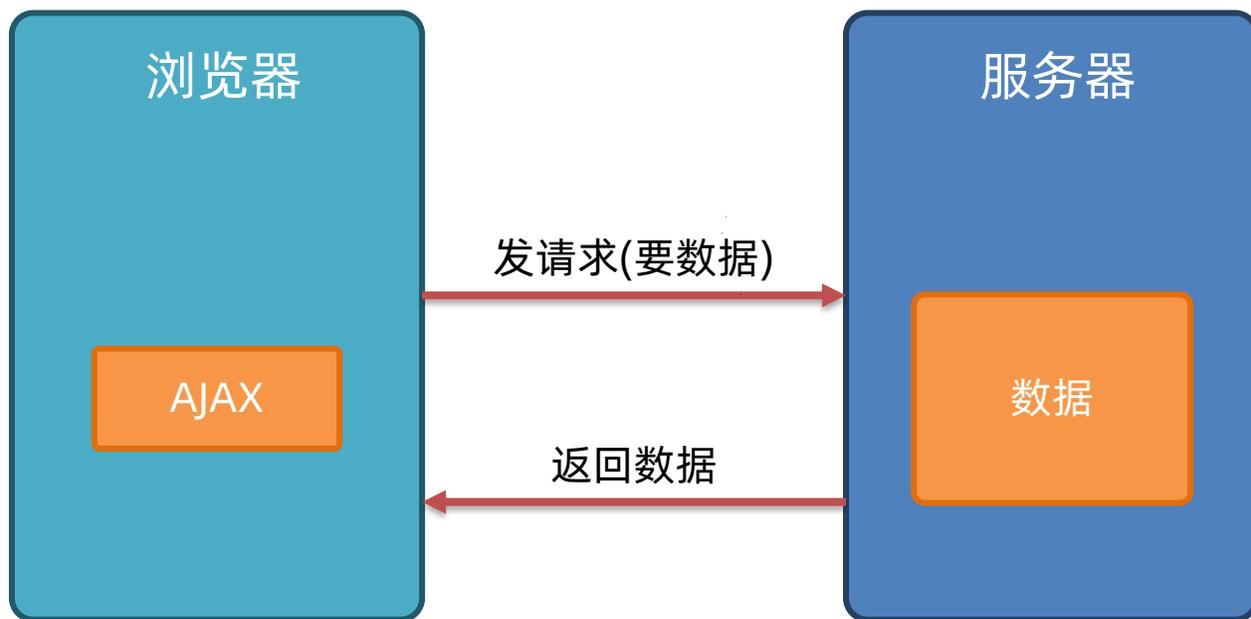
定义: AJAX 是异步的 JavaScript 和 XML (Asynchronous JavaScript And XML)。简单点说，就是使用 XMLHttpRequest 对象与服务器通信。它可以使用 JSON, XML, HTML 和 text 文本等格式发送和接收数据。



AJAX是什么?

定义: AJAX 是异步的 JavaScript 和 XML (Asynchronous JavaScript And XML)。简单点说，就是使用 XMLHttpRequest(XHR)对象与服务器通信。它可以使用 JSON, XML, HTML 和 text 文本等格式发送和接收数据。

概念: AJAX是一种与服务器通信的技术



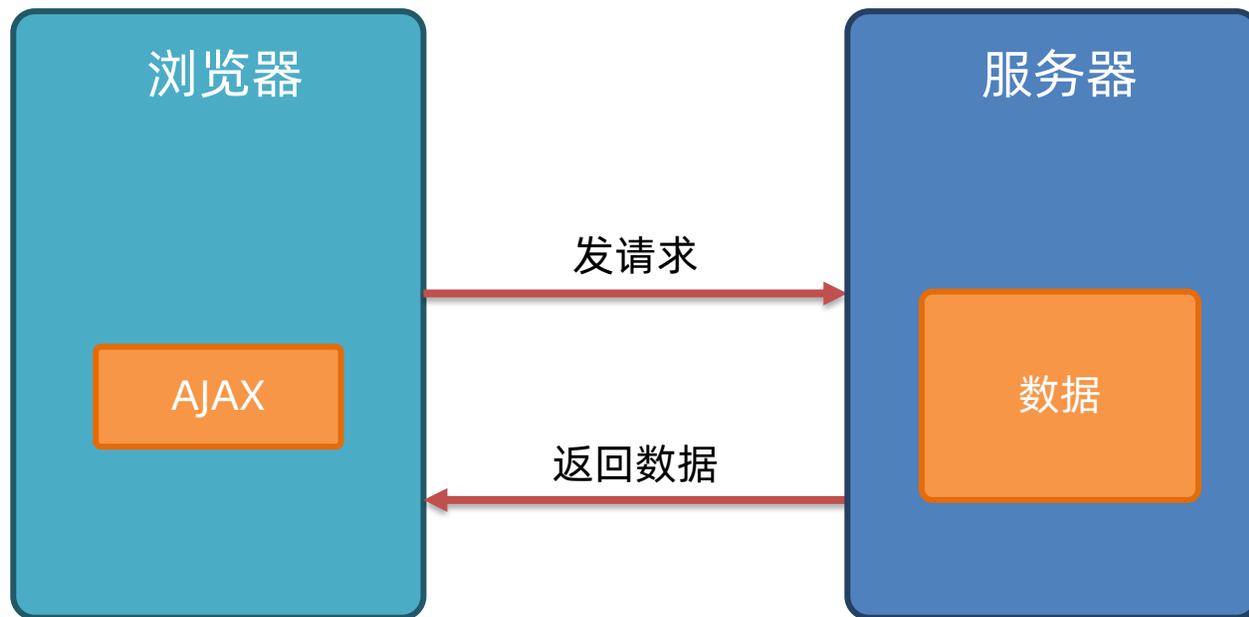
AJAX是一种与服务器通信的技术

开发中的数据来源发生改变

其他的**基本一致**



自己定义数据



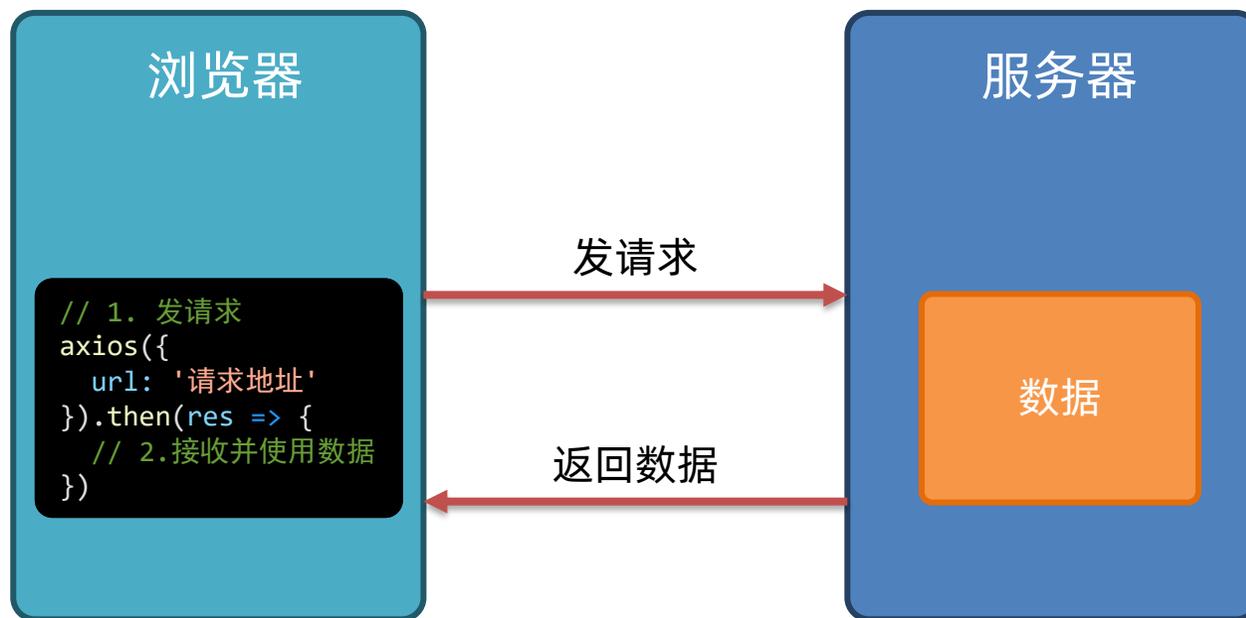
服务器提供数据

AJAX阶段重点学习内容

1. 请求库axios:
 - 基于XHR封装，简单易用，功能强大
 - vue、react中都会用到axios
 - 使用axios和服务器通信
2. XMLHttpRequest(XHR)对象:
 - 使用XHR的原生语法和服务器通信
 - 了解AJAX的原理

请求库axios的基本用法

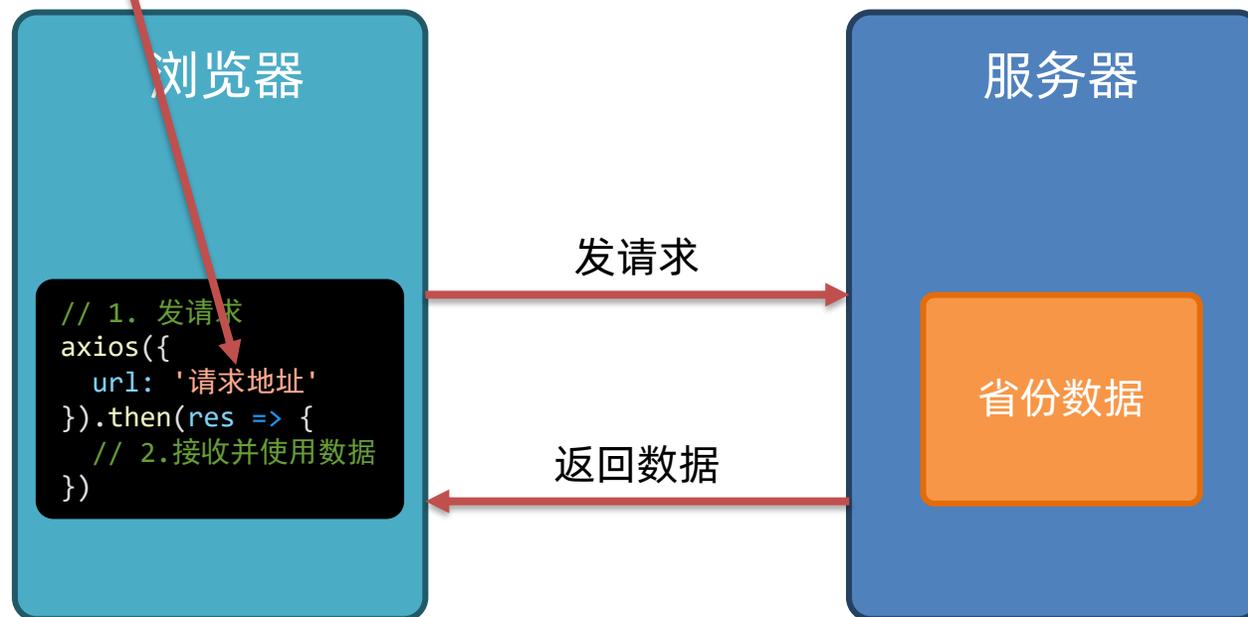
1. 导包: <https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js>
2. 使用
 1. 发请求: 调用axios函数, 传入配置
 2. 接收并使用数据: 通过点语法调用then方法, 传入回调函数并定义形参(比如res)



请求库axios的实际应用

需求：点击按钮通过 `axios` 获取省份数据并渲染到页面

请求地址(url): <http://hmajax.itheima.net/api/province>



总结

1. AJAX是**与服务器通信**的技术
2. AJAX课程的重点学习内容:
 1. 请求库**axios** → **和服务器通信**
 2. 原生XHR → 理解原理
3. **axios的基本使用步骤**为:
 1. 导包
 2. 使用

```
// 1. 发请求
axios({
  url: '请求地址'
}).then(res => {
  // 2. 接收并使用数据
})
```



认识URL

什么是URL

定义：URL 代表着是统一资源定位符（Uniform Resource Locator）。URL 无非就是一个给定的独特资源在 Web 上的地址。理论上说，每个有效的 URL 都指向一个唯一的资源。这个资源可以是一个 HTML 页面，一个 CSS 文档，一幅图像，等等。

测试用URL：

1. <https://www.itheima.com/index.html> ← 网页资源
2. <https://www.itheima.com/images/logo.png> ← 图片资源
3. <http://hmajax.itheima.net/api/province> ← 数据资源

概念：URL是资源在网络上的地址

URL常见组成部分

URL 由不同的部分组成，目前需要关注的是：**协议**、**域名**、**资源路径**

`http://hmajax.itheima.net/api/province`

协议

域名

资源路径

URL常见组成部分-协议

规定了浏览器发送及服务器返回内容的格式，常见的有 http、https

http://hmajax.itheima.net/api/province

协议



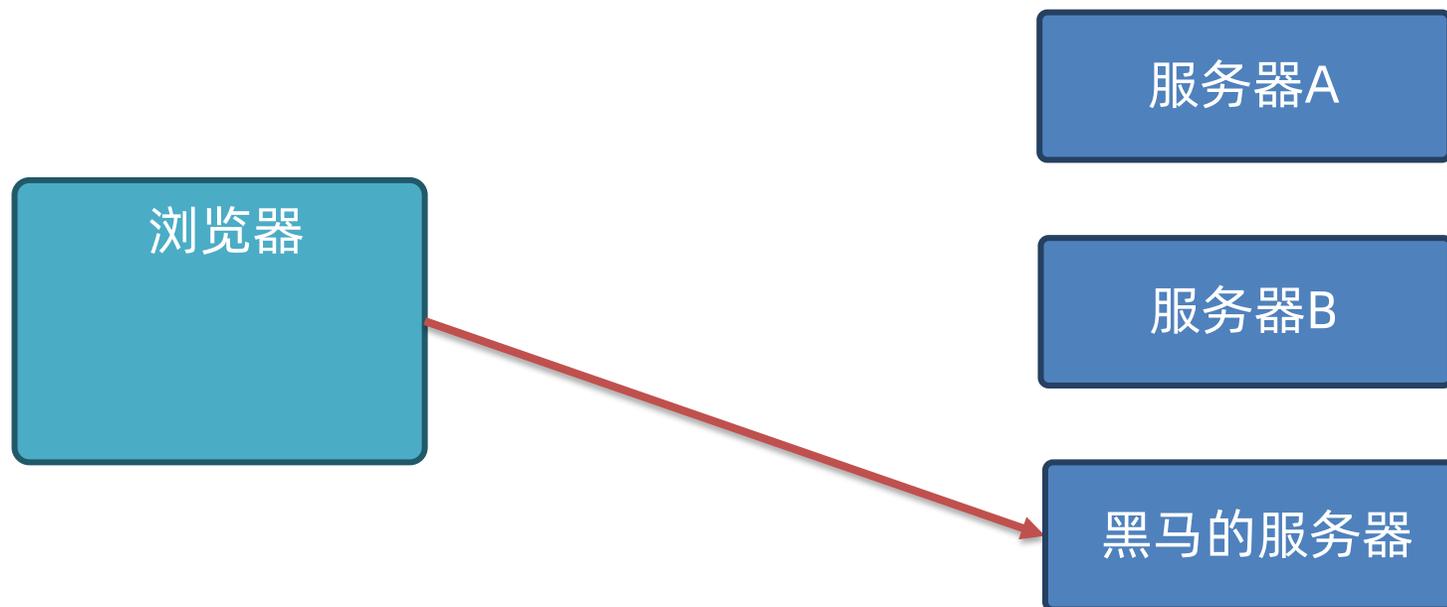
URL常见组成部分-域名

域名表示正在请求网络上的哪个服务器

`http://hmajax.itheima.net/api/province`

协议

域名



URL常见组成部分-资源路径

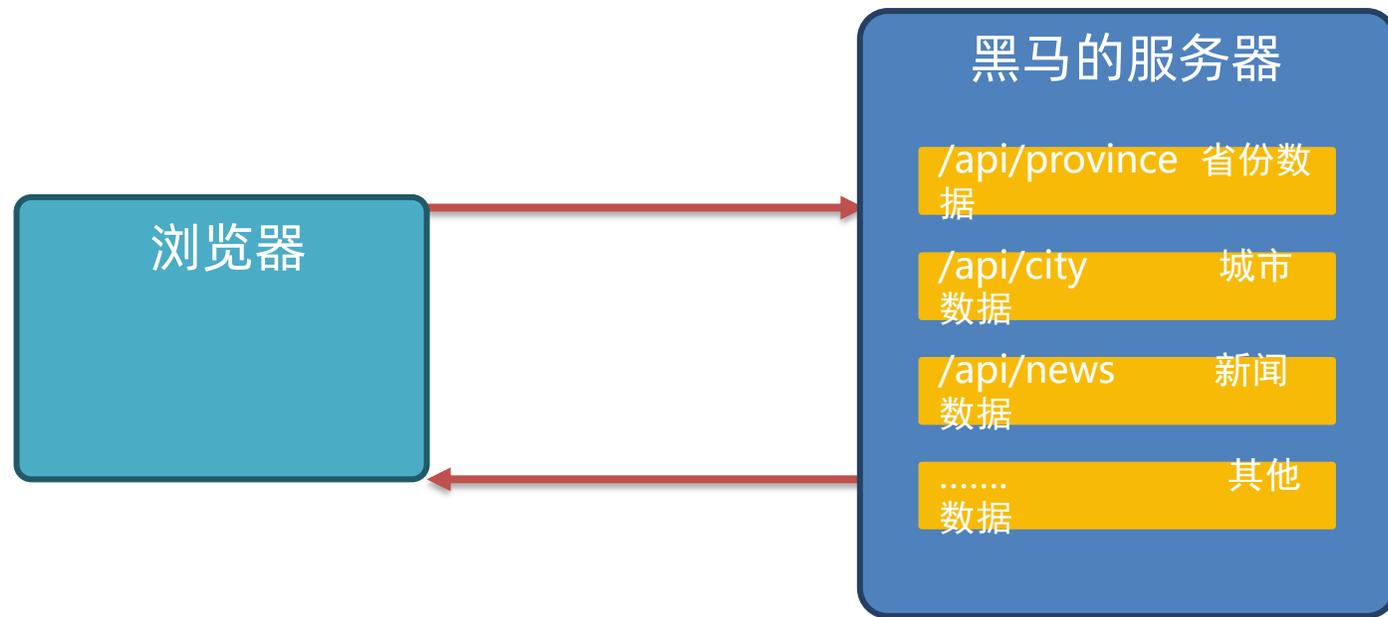
资源在服务器的位置，资源和路径的对应关系由服务器决定

http://hmajax.itheima.net/api/province

协议

域名

资源路径



试一试

需求：点击按钮通过 `axios` 获取新闻数据并通过 `console.log` 输出

请求地址(url): <http://hmajax.itheima.net/api/news>



总结

1. URL是资源在网络上的地址
2. URL常见组成部分:协议、域名、资源地址
3. <http://hmajax.itheima.net/api/city> 各部分分别是?
 1. 协议: http
 2. 域名: hmajax.itheima.net
 3. 资源地址: /api/city



URL查询参数介绍及使用

查询参数示例

定义： URL查询参数是提供给网络服务器的额外参数。这些参数是用 & 符号分隔的键/值对列表。在返回资源之前，Web 服务器可以使用这些参数来执行额外的操作。每个 Web 服务器都有自己关于参数的规则，唯一可靠的方式来知道特定 Web 服务器是否处理参数是通过询问 Web 服务器所有者。

测试用URL:

<https://www.baidu.com/s?wd=黑马程序员>

<https://www.baidu.com/s?wd=传智播客>

查询参数



查询参数的格式

语法：?隔开之后使用&分隔的键值对列表

http://测试域名/资源路径?key=value&key2=value2....

协议

域名

资源路径

查询参数

测试用URL: <https://www.baidu.com/s?wd=黑马程序员>

如何确认查询参数

确认方法：服务器的所有者会提供参数信息

测试用URL：

- <http://hmajax.itheima.net/api/city>
- 获取**某个省**所有的城市
- 查询参数名：**pname**
- 说明：传递省份或直辖市名，比如 **北京、广东省...**

<http://hmajax.itheima.net/api/city?pname=北京>

协议

域名

资源路径

查询参数

使用axios传递查询参数

通过params属性设置对象的方式传递

```
axios({
  url: 'http://测试服务器/资源路径',
  params: {
    key: 'value',
    key2: 'value2'
  }
}).then(res => {
  // 接受并使用数据
})
```

<http://hmajax.itheima.net/api/city?pname=北京>

协议

域名

资源路径

查询参数

试一试

需求： 点击按钮通过 `axios` 获取某个省份的城市，并输出

请求地址(url): <http://hmajax.itheima.net/api/city>

- 获取某个省所有的城市
- 查询参数名: `pname`
- 说明: 传递省份或直辖市名比如 `北京、广东省...`

```
axios({
  url: 'http://测试服务器/资源路径',
  params: {
    key: 'value',
    key2: 'value2'
  }
}).then(res => {
  // 接受并使用数据
})
```

总结

1. URL查询参数:

查询参数是提供给网络服务器的额外参数

2. URL查询参数的格式是:

http://测试服务器/资源路径?key=value&key2=value2...

3. 使用axios传递查询参数

```
axios({
  url: 'http://测试服务器/资源路径',
  params: {
    key: 'value',
    key2: 'value2'
  }
}).then(res=>{
  // 接受并使用数据
})
```

4

案例-地区查询

效果展示

个人设置

 上传/修改头像

昵称

籍贯

- 东城区
- 西城区
- 朝阳区
- 丰台区
- 石景山区
- 海淀区
- 门头沟区
- 房山区
- 通州区
- 顺义区
- 昌平区
- 大兴区
- 怀柔区
- 平谷区

案例效果展示

省份名字 城市名字

地区列表:

江岸区
江汉区
硚口区
汉阳区
武昌区
青山区
洪山区
东西湖区
汉南区
蔡甸区
江夏区
黄陂区
新洲区
其它区

分析及实现

查询地区并渲染列表

1. 如何获取URL地址和参数说明?
2. 如何渲染列表?

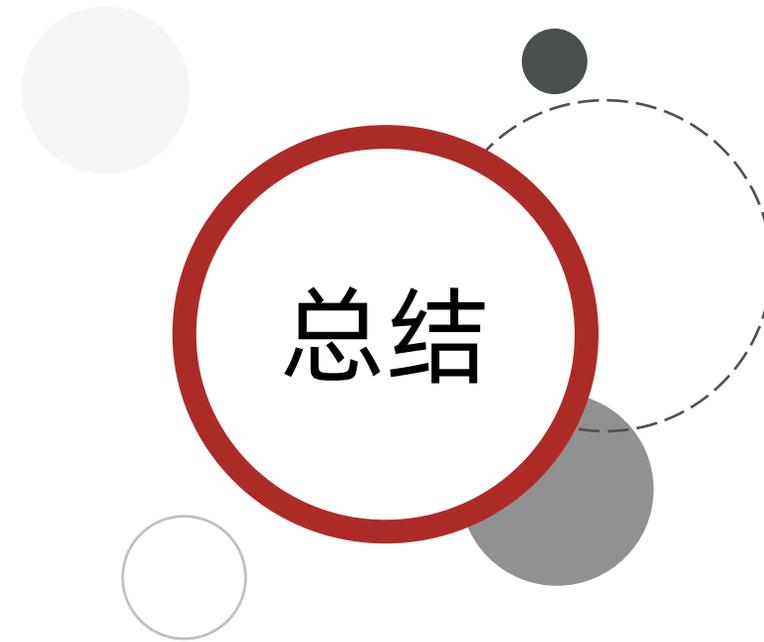
测试用URL:

- <http://hmajax.itheima.net/api/area>
- 查询某个省份内某个城市的所有区县
- 查询参数名: pname、cname
- pname说明: 传递省份或直辖市名, 比如 北京、湖北省...
- cname说明: 省份内的城市, 比如 北京市、武汉市...

<http://hmajax.itheima.net/api/area?pname=北京&cname=北京市>

查询参数

省份/直辖市名字	城市名字
<input type="text" value="湖北省"/>	<input type="text" value="武汉市"/>
<input type="button" value="查询"/>	
地区列表:	
<input type="text" value="江岸区"/>	
<input type="text" value="江汉区"/>	
<input type="text" value="硚口区"/>	
<input type="text" value="汉阳区"/>	
<input type="text" value="武昌区"/>	
<input type="text" value="青山区"/>	
<input type="text" value="洪山区"/>	
<input type="text" value="东西湖区"/>	
<input type="text" value="汉南区"/>	
<input type="text" value="蔡甸区"/>	
<input type="text" value="江夏区"/>	
<input type="text" value="黄陂区"/>	
<input type="text" value="新洲区"/>	
<input type="text" value="其它区"/>	



总结

1. 如何通过axios传递多个查询参数?

```
axios({  
  url: 'http://测试服务器/资源路径',  
  params: {  
    key: 'value',  
    key2: 'value2'  
  }  
})
```

2. ES6对象属性赋值的简写? 属性名和变量名相同

5

常用请求方法和数据提交

常用请求方法

请求方法表明对服务器资源执行的操作。

方法	操作
GET	获取数据
POST	提交数据
PUT	修改数据 (全部)
DELETE	删除数据
PATCH	修改数据 (部分)



提交数据

全部订单	待付款	待发货	待收货	待评价	已完成	已取消
下单时间: 2022-11-01 17:16:40 订单编号: 1587373040889499650						
	女式头层牛皮经典切尔西靴 颜色:黑色牛犛(长绒里) 尺码:36	¥275.00	x1	待发货	¥278.00 (含运费: ¥3.00) 在线支付	查看详情 再次购买
下单时间: 2022-11-01 17:11:50 订单编号: 1587371828051972098						
	新潮运动底, 女式保暖皮毛一体雪地靴 颜色:杏色 尺码:38	¥319.00	x1	已取消	¥319.00 (含运费: ¥0.00) 在线支付	查看详情

欢迎注册

已有帐号? [登录](#)

用户名

手机号

密码 ⓘ

验证码

阅读并接受 [《百度用户协议》](#)、[《儿童个人信息保护声明》](#) 及 [《百度隐私权保护声明》](#)

axios请求配置

url: 请求的URL地址

method: 请求的方法，GET可以省略（不区分大小写）

data: 提交数据

```
axios({
  url: '请求URL',
  method: 'post',
  data: {
    key: 'value'
  },
})
```

试一试

需求： 点击按钮通过 `axios`提交用户数据，完成注册功能

请求地址(url): <http://hmajax.itheima.net/api/register>

- 注册用户
- 请求方法: `POST`
- 参数: `username`, `password`
- 说明: `username` 用户名（中英文和数字组成, 最少8位）, `password` 密码（最少6位）

```
axios({
  url: '请求URL',
  method: '请求方法',
  data: {
    key: 'value'
  },
})
```

总结

1. 请求方法表明对服务器资源的操作，最为常用的是？

POST提交数据、GET查询数据

2. axios的核心配置？

url: 请求URL

method: 请求方法，GET可以省略（不区分大小写）

params: 查询参数

data: 提交数据

```
// POST请求
axios({
  url: '请求URL',
  method: 'post',
  data: {
    key: 'value'
  },
})
```

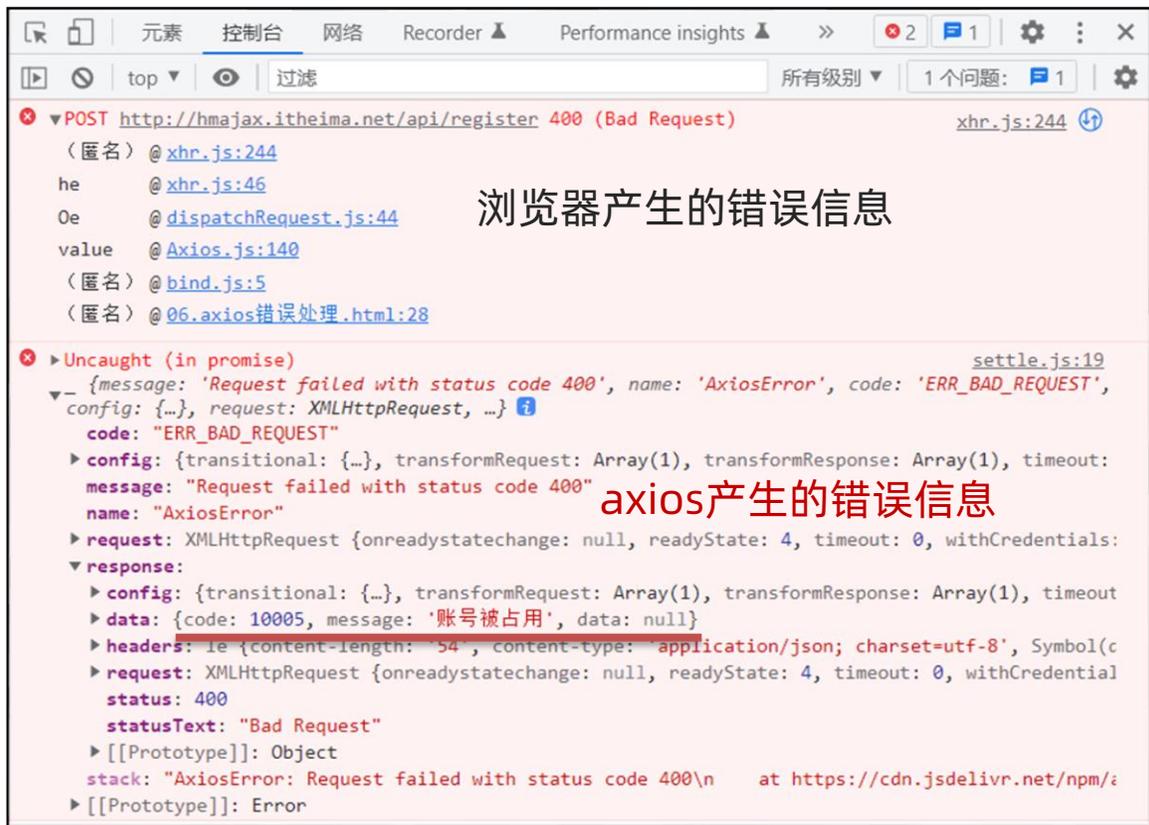
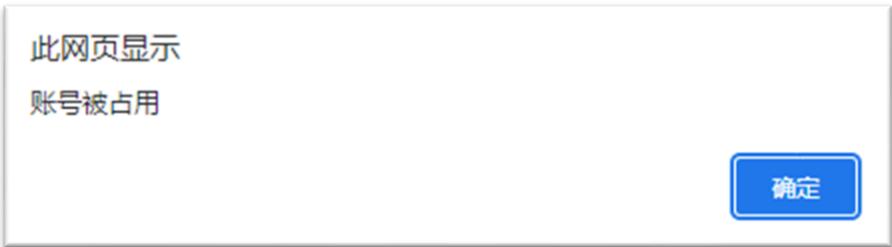
```
// GET请求
axios({
  url: '请求URL',
  method: 'get',
  params: {
    key: 'value'
  },
})
```



axios错误处理

错误原因及处理方法

以普通用户能够看到的方式，来展示错误信息



axios错误处理

- 在then方法的后面，通过点语法调用catch方法，传入回调函数并定义形参

```
axios({  
  // 请求选项  
}).then(res => {  
  // 处理数据  
}).catch(err => {  
  // 处理错误  
})
```

试一试

- 错误处理：上一节的注册案例，重复注册时通过弹框提示用户



HTTP协议-请求报文

HTTP协议

规定了浏览器**发送**及服务器**返回**内容的**格式**

请求报文：浏览器按照HTTP协议要求的**格式**，发送给服务器的内容

```
1 POST http://hmajax.itheima.net/api/register HTTP/1.1
2 Host: hmajax.itheima.net
3 Connection: keep-alive
4 Content-Length: 45
5 Accept: application/json, text/plain, */*
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36
7 Content-Type: application/json
8 Origin: http://127.0.0.1:5500
9 Referer: http://127.0.0.1:5500/
10 Accept-Encoding: gzip, deflate
11 Accept-Language: zh-CN,zh;q=0.9
12
13 {"username":"itheima001","password":"123456"}
```

```
axios({
  url: 'http://hmajax.itheima.net/api/register',
  method: 'post',
  data: {
    username: 'itheima001',
    password: '123456'
  }
})
```

浏览器

请求报文

服务器

请求报文的格式

请求报文的组成部分有：

1. 请求行：请求方法，URL，协议
2. 请求头：以键值对的格式携带的附加信息，比如：Content-Type
3. 空行：分隔请求头，空行之后的是发送给服务器的资源
4. 请求体：发送的资源

```
axios({
  url: 'http://hmajax.itheima.net/api/register',
  method: 'post',
  data: {
    username: 'itheima998',
    password: '123456'
  }
})
```

```
1 POST http://hmajax.itheima.net/api/register HTTP/1.1
2 Host: hmajax.itheima.net
3 Connection: keep-alive
4 Content-Length: 45
5 Accept: application/json, text/plain, */*
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36
7 Content-Type: application/json
8 Origin: http://127.0.0.1:5500
9 Referer: http://127.0.0.1:5500/
10 Accept-Encoding: gzip, deflate
11 Accept-Language: zh-CN,zh;q=0.9
12
13 {"username":"itheima001","password":"123456"}
```

总结

1. 浏览器发送给服务器的内容叫做**请求报文**

2. 请求报文的组成:

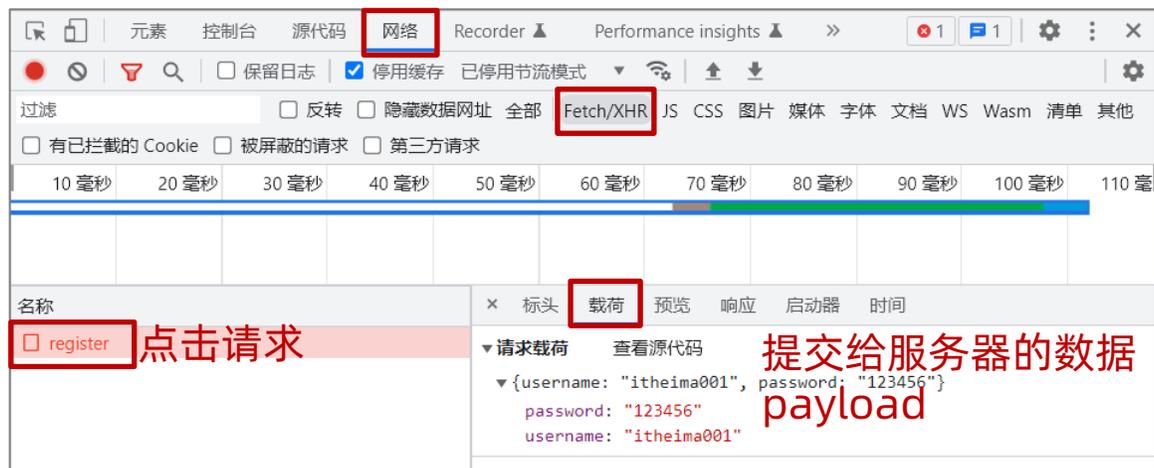
请求行: 请求方法, URL, 协议

请求头: 以键值对的格式携带的附加信息, 比如**Content-Type**

空行: 分隔请求头, 空行之后的是发送给服务器的资源

请求体: 发送的资源

3. 通过Chrome的**网络面板**查看请求报文



8

请求报文-辅助调试

需求

使用刚刚注册的用户登录，一直提示用户名或错误

- 用户名: itheima001
- 密码: 123456

欢迎-登录

用户名或密码错误

用户名

密码

登录



HTTP协议-响应报文

HTTP协议-响应报文

规定了浏览器发送及服务器返回内容的格式

响应报文：服务器按照HTTP协议要求的格式，返回给浏览器的内容

1. 响应行（状态行）：协议、HTTP响应状态码、状态信息
2. 响应头：以键值对的格式携带的附加信息，比如：Content-Type
3. 空行：分隔响应头，空行之后的是服务器返回的资源
4. 响应体：返回的资源

```
axios({
  url: 'http://hmajax.itheima.net/api/register',
  method: 'post',
  data: {
    username: 'itheima001',
    password: '123456'
  }
}).then(res => {
  console.log(res.data)
}).catch(err => {
  // 重复注册，通过弹框提示用户
  alert(err.response.data.message)
})
```

```
1 HTTP/1.1 400 Bad Request
2 Server: nginx
3 Date: Wed, 09 Nov 2022 13:26:06 GMT
4 Content-Type: application/json; charset=utf-8
5 Content-Length: 54
6 Connection: keep-alive
7 Vary: Origin
8 Access-Control-Allow-Origin: http://127.0.0.1:5500
9 Access-Control-Allow-Credentials: true
10 x-frame-options: SAMEORIGIN
11 x-xss-protection: 1; mode=block
12 x-content-type-options: nosniff
13 x-download-options: noopen
14 x-readtime: 16
15
16 {"code":10005,"message":"账号被占用","data":null}
```



HTTP响应状态码

HTTP 响应状态码用来表明请求**是否成功**完成。

比如：404（服务器找不到资源）

状态码	说明
1xx	信息
2xx	成功
3xx	重定向消息
4xx	客户端错误
5xx	服务端错误





接口文档

接口文档

接口文档：描述接口的文章（后端工程师）

接口：使用AJAX和服务器通讯时，使用的 URL，请求方法，以及参数

传送门：[AJAX阶段接口文档](#)

```
axios({  
  url: 'http://hmajax.itheima.net/api/city',  
  method: 'get',  
  params: {  
    pname: '湖北省'  
  }  
})
```

获取-城市列表

GET http://hmajax.itheima.net/api/city

获取-城市列表

请求参数

参数名	位置	类型	必填	说明
pname	query	string	是	示例值: 辽宁省



总结

1. 接口文档:

由后端提供的描述接口的文章

登录

POST http://hmajax.itheima.net/api/login

登录

请求参数

Body 参数(application/json)

数据结构 示例值

object (2)

username	string	✓	用户名	中英文和数字组成 最少8位
password	string	✓	密码	最少6位



案例-用户登录

需求

1. 点击登录时和服务器通信
2. 用户名和密码长度判断
3. 提示信息

欢迎-登录

登录成功

用户名

itheima001

密码

.....

登录



form-serialize插件

form-serialize

快速收集表单元素的值



```
const form = document.querySelector('.example-form')
const data = serialize(form, { hash: true, empty: true })
```



```
{
  username: 'itheima001',
  password: '123456'
}
```

表单元素需要设置name属性

name属性根据接口文档中的参数名设置

hash 设置生成的数据格式

- true: js对象 (常用)

- false: 查询字符串

empty 是否收集空数据

- true: 收集空数据 (推荐)

- false: 空数据不收集 (不推荐)

13

案例-用户登录(整合form-serialize)



传智教育旗下高端IT教育品牌