

AJAX 入门

让数据活起来







- ◆ AJAX 概念和 axios 使用
- ◆ 认识 URL
- ◆ URL 查询参数
- ◆ 常用请求方法和数据提交
- ◆ HTTP协议-报文
- ◆ 接口文档
- ◆ 案例 用户登录
- ◆ form-serialize 插件

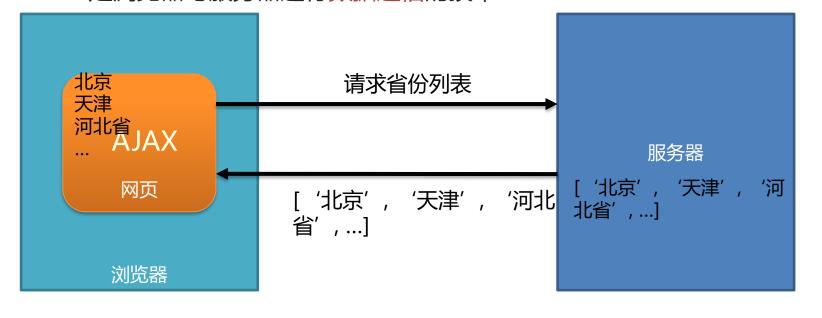


什么是 AJAX [ˈeɪdʒæks]?

<u>定义</u>:

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 最吸引人的就是它的"异步"特性,也就是说它可以在不重新刷新页面的情况下与服务器通信,交换数据,或更新页面。

概念: AJAX 是浏览器与服务器进行数据通信的技术



把数据变活



怎么用 AJAX?

- 1. 先使用 axios [æk 'sioʊs] 库,与服务器进行数据通信
 - 基于 XMLHttpRequest 封装、代码简单、月下载量在 14 亿次
 - Vue、React 项目中都会用到 axios
- 2. 再学习 XMLHttpRequest 对象的使用,了解 AJAX 底层原理



axios 使用

语法:

- 1. 引入 axios.js: https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js
- 2. 使用 axios 函数
 - ✓ 传入配置对象
 - ✓ 再用 .then 回调函数接收结果,并做后续处理

```
axios( {
   url: '目标资源地址'
}) .then( (result) => {
   // 对服务器返回的数据做后续处理
})
```



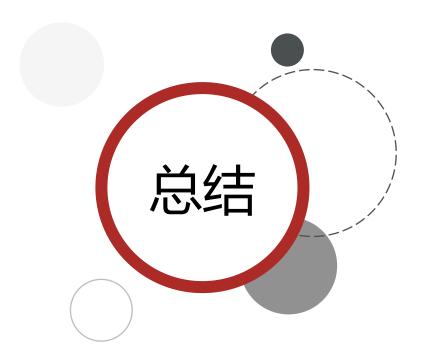
axios 使用

需求:请求目标资源地址,拿到省份列表数据,显示到页面

目标资源地址: http://hmajax.itheima.net/api/province

```
axios( {
  url: 'http://hmajax.itheima.net/api/province'
}) .then( (result) => {
  // 对服务器返回的数据做后续处理
})
```





- 1. AJAX 有什么用?
 - > 浏览器和服务器之间通信, 动态数据交互
- 2. AJAX 如何学:
 - > 先掌握 axios 使用
 - ➤ 再了解 XMLHttpRequest 原理
- 3. 这一节 axios 体验步骤?
 - ▶ 引入 axios 库
 - ➤ 使用 axios 语法

```
axios({
    url: '目标资源地址'
}).then((result) => {
    // 对服务器返回的数据做后续处理
})
```





- ◆ AJAX 概念和 axios 使用
- ◆ 认识 URL
- ◆ URL 查询参数
- ◆ 常用请求方法和数据提交
- ◆ HTTP协议-报文
- ◆ 接口文档
- ◆ 案例 用户登录
- ◆ form-serialize 插件



认识 URL

原因:知道作用和组成,方便与后端人员沟通

```
axios({
  url: 'http://hmajax.itheima.net/api/province'
}).then(result => {
  // 对服务器返回的数据做后续处理
})
```





什么是 URL?

定义: **统一资源定位符**(英语: Uniform Resource Locator,缩写: URL,或称**统一资源定位器、定位地址、URL地址**^[1]) 俗称网页地址,简称**网址**,是因特网上标准的资源的地址(Address),如同在网络上的门牌。它最初是由蒂姆·伯纳斯–李发明用来作为万维网的地址,现在它已经被万维网联盟编制为因特网标准RFC 1738 亿。

例如:

https://www.baidu.com/index.html

网页资源

https://www.itheima.com/images/logo.png

图片资源

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

数据资源

• ..

概念: URL 就是统一资源定位符,简称网址,用于访问网络上的资源



URL 的组成

组成:







协议

http 协议:超文本传输协议,规定浏览器和服务器之间传输数据的格式

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

协议

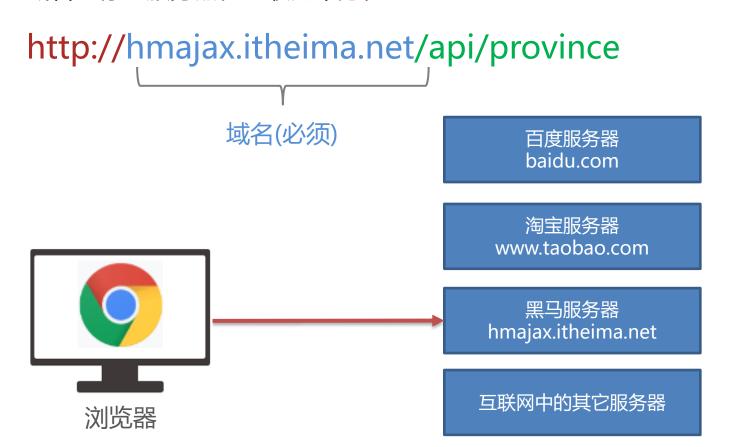


协议范围: http, https, ...



域名

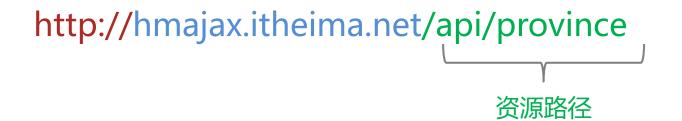
域名: 标记服务器在互联网中方位

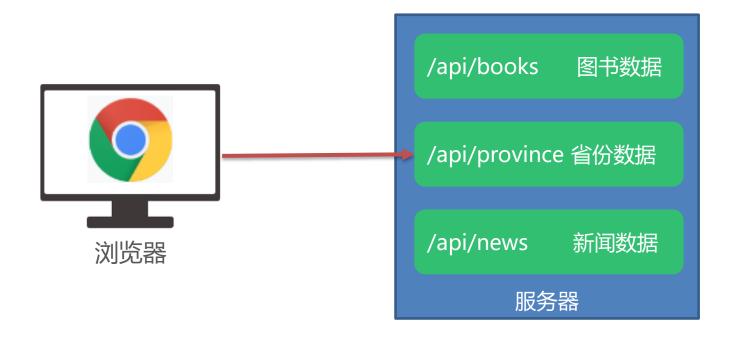




资源路径

资源路径:标记资源在服务器下的具体位置







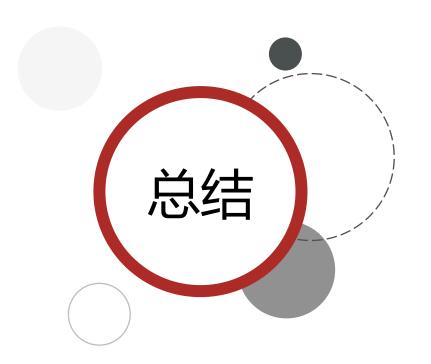
获取-新闻列表

需求: 使用 axios 从服务器拿到新闻列表数据

目标资源地址: http://hmajax.itheima.net/api/news

```
▼ data: Array(9)
 ▶0: {id: 1, title: '5G商用在即,三大运营商营收持续下降', source: '新京报经济新闻', cmtcount: 58, img: 'http://ajax-api.itheima.net/public/images/0.webp', ...}
 ▶1: {id: 2, title: '国际媒体头条速览: 特朗普宣布杀死巴格达迪时,有人想起了奥巴马', source: '环视全球', cmtcount: 56, img: 'http://ajax-api.itheima.net/public/images/1.webp', ...}
 ▶2: {id: 3, title: '华为5G随行WiFi发布, 5G布局全面领先苹果、三星', source: '驱动中国网络传媒', cmtcount: 32, img: 'http://ajax-api.itheima.net/public/images/2.webp', ...}
 ▶3: {id: 4, title: '沈义人: 首批5G手机最好选集成SoC和双模SA+NSA', source: '手机中国', cmtcount: 18, img: 'http://ajax-api.itheima.net/public/images/3.webp', ...}
 ▶4: {id: 5, title: '为什么说中美阶段性协议再近一步,读懂周末的这些关键信息', source: '澎湃新闻', cmtcount: 131, img: 'http://ajax-api.itheima.net/public/images/4.webp', ...}
 ▶5: {id: 6, title: '阿根廷大选结果揭晓: 反对派候选人费尔南德斯当选下任总统', source: '海外网', cmtcount: 99, img: 'http://ajax-api.itheima.net/public/images/5.webp', ...}
 ▶6: {id: 7, title: '赵薇谈演员整容: 不是一件坏事情,但劝大家别整太像', source: '腾讯视频Big磅来了', cmtcount: 154, img: 'http://ajax-api.itheima.net/public/images/6.webp', ...}
 ▶7: {id: 8, title: 'LV母公司史上最大并购: 报价145亿美元购Tiffany', source: '澎湃新闻', cmtcount: 119, img: 'http://ajax-api.itheima.net/public/images/7.webp', ...}
 ▶8: {id: 9, title: '黄峥1350亿蝉联80后白手起家首富: 过去1年财富每天涨1个亿', source: '胡润百富', cmtcount: 676, img: 'http://ajax-api.itheima.net/public/images/8.webp', ...}
  length: 9
 ▶ [[Prototype]]: Array(0)
 message: "获取新闻列表成功"
                                                                                          axios( {
                                                                                            url: 'http://hmajax.itheima.net/api/news'
                                                                                         }) .then( (result) => {
                                                                                             // 对服务器返回的数据做后续处理
                                                                                          })
```





1. URL 是什么:

- ▶ 统一资源定位符,网址,用于访问服务器上资源
- 2. 请解释这个 URL, 每个部分作用?
 - http://hmajax.itheima.net/api/news
 - ▶ 协议://域名/资源路径





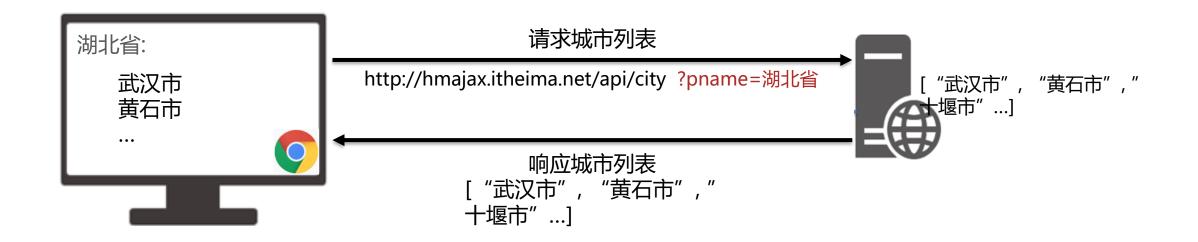
- ◆ AJAX 概念和 axios 使用
- ◆ 认识 URL
- ◆ URL 查询参数
- ◆ 常用请求方法和数据提交
- ◆ HTTP协议-报文
- ◆ 接口文档
- ◆ 案例 用户登录
- ◆ form-serialize 插件



URL 查询参数

定义:浏览器提供给服务器的额外信息,让服务器返回浏览器想要的数据

语法: http://xxxx.com/xxx/xxx?参数名1=值1&参数名2=值2





axios - 查询参数

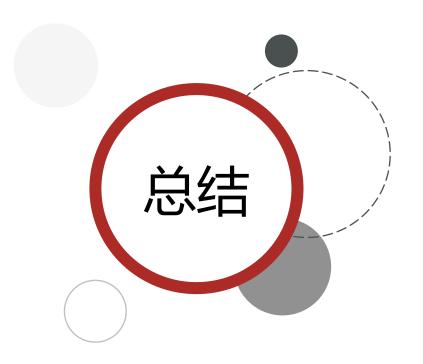
语法: 使用 axios 提供的 params 选项

注意: axios 在运行时把参数名和值,会拼接到 url?参数名=值

城市列表: http://hmajax.itheima.net/api/city?pname=湖北省

```
axios({
  url: 'http://hmajax.itheima.net/api/city
', params: {
  pname: '湖北省'
  }
}).then(result => {
  // 对服务器返回的数据做后续处理
```





- 1. URL 查询参数有什么作用?
 - ▶ 浏览器提供给服务器额外信息,获取对应的数据
- 2. axios 要如何携带查询参数?
 - ➤ 使用 params 选项,携带参数名和值





地区查询

需求:根据输入的省份名字和城市名字,查询地区并渲染列表

省份名字	城市名字
北京	北京市
查询	
地区列表:	
东城区	
西城区	
朝阳区	
丰台区	



1 步骤

地区查询

首先: 确定 URL 网址和参数说明

• 查询某个省内某个城市的所有地区: http://hmajax.itheima.net/api/area

参数名:

pname: 省份名字或直辖市名字,比如北京、福建省、辽宁省...

cname:城市名字,比如北京市、厦门市、大连市...

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

省份名字	城市名字
北京	北京市
查询	
地区列表:	
东城区	
西城区	
朝阳区	
丰台区	





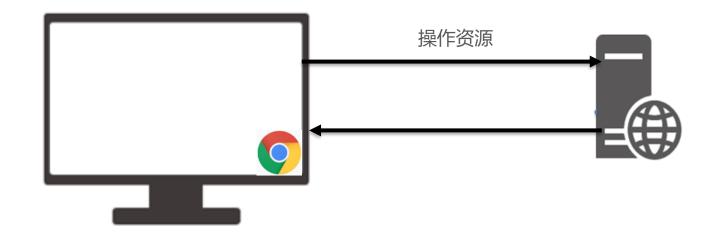
- ◆ AJAX 概念和 axios 使用
- ◆ 认识 URL
- ◆ URL 查询参数
- ◆ 常用请求方法和数据提交
- ◆ HTTP协议-报文
- ◆ 接口文档
- ◆ 案例 用户登录
- ◆ form-serialize 插件



常用请求方法

请求方法:对服务器资源,要执行的操作

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





数据提交

场景: 当数据需要在服务器上保存







axios 请求配置

url:请求的 URL 网址

method:请求的方法,GET可以省略(不区分大小写)

data: 提交数据



数据提交 - 注册账号

需求: 通过 axios 提交用户名和密码, 完成注册功能

注册用户 URL 地址: http://hmajax.itheima.net/api/register

请求方法: POST

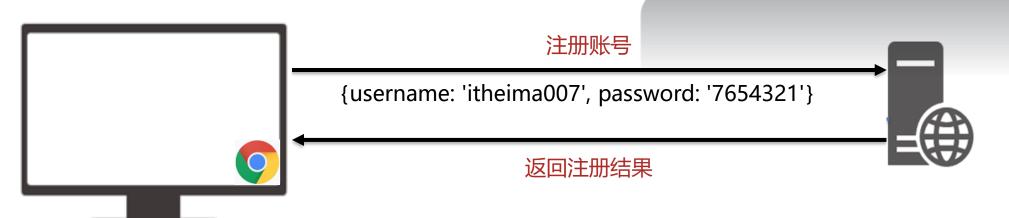
参数名:

username 用户名(中英文和数字组成,最少 8 位)

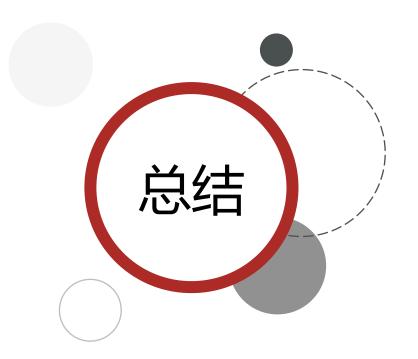
password 密码 (最少 6 位)

```
axios({
  url: 'http://hmajax.itheima.net/api/register',
  method: 'post'

  data: {
    username: 'itheima007',
    password: '7654321'
  }
})
```







- 1. 请求方法表明对服务器资源的操作,最为常用的2个是?
 - ➤ POST 提交数据, GET 查询数据
- 2. axios 的核心配置?
 - ➤ url: 请求 URL 网址
 - 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

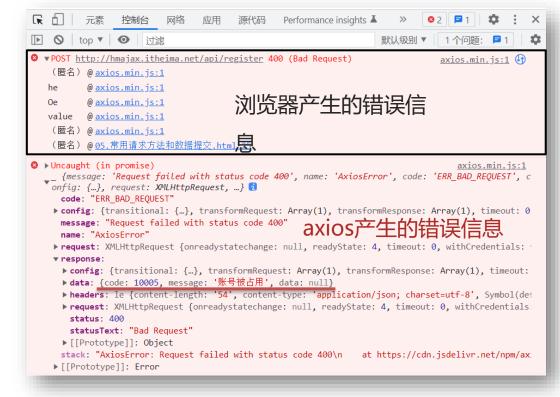


axios 错误处理

场景: 再次注册相同的账号, 会遇到报错信息

处理: 用更直观的方式, 给普通用户展示错误信息









axios 错误处理

语法: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(result => {
  // 处理数据
}).catch(error =>
  // 处理错误
```

处理:注册案例,重复注册时通过弹框提示用户错误原因





- ◆ AJAX 概念和 axios 使用
- ◆ 认识 URL
- ◆ URL 查询参数
- ◆ 常用请求方法和数据提交
- ◆ HTTP协议-报文
- ◆ 接口文档
- ◆ 案例 用户登录
- ◆ form-serialize 插件



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: 46
 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/
                                                                        axios({
10 Accept-Encoding: gzip, deflate
11 Accept-Language: zh-CN, zh; q=0.9
                                                                           url: 'http://hmajax.itheima.net/api/register',
12
                                                                           method: 'post'.
13 {"username": "itheima007", "password": "7654321"}
                                                                           data: {
                                                                              username: 'itheima007',
                                 请求报文
                                                                              password: '7654321'
                                                                        3)
```



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

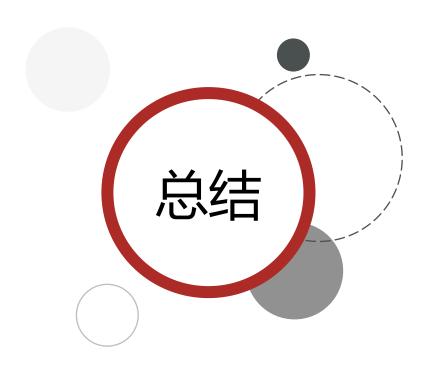
4. 请求体:发送的资源

```
axios({
  url: 'http://hmajax.itheima.net/api/register',
  method: 'post',
  data: {
    username: 'itheima007',
    password: '7654321'
  }
})
```

```
1 POST http://hmajax.itheima.net/api/register HTTP/1.1
2 Host: hmajax.itheima.net
3 Connection: keep-alive
4 Content-Length: 46
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": "itheima007", "password": "7654321"}
```

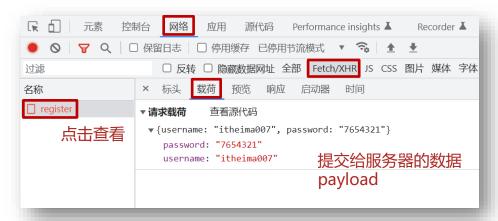




- 1. 浏览器发送给服务器的内容叫做请求报文
- 2. 请求报文的组成:



3. 通过 Chrome 的网络面板查看请求报文





请求报文 - 错误排查

需求:通过请求报文排查错误原因,并修复

输入正确的用户名和密码无法登录

• 用户名: itheima007

• 密码: 7654321





HTTP 协议 - 响应报文

HTTP 协议: 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



```
HTTP/1.1 400 Bad Request

Server: nginx
Date: Wed, 09 Nov 2022 13:26:06 GMT

Content-Type: application/json; charset=utf-8
Content-Length: 54
Connection: keep-alive
Vary: Origin
Access-Control-Allow-Origin: http://127.0.0.1:5500
Access-Control-Allow-Credentials: true
x-frame-options: SAMEORIGIN
x-xss-protection: 1; mode=block
x-content-type-options: nosniff
x-download-options: noopen
x-readtime: 16

["code":10005,"message":"账号被占用","data":null}
```



HTTP 响应状态码

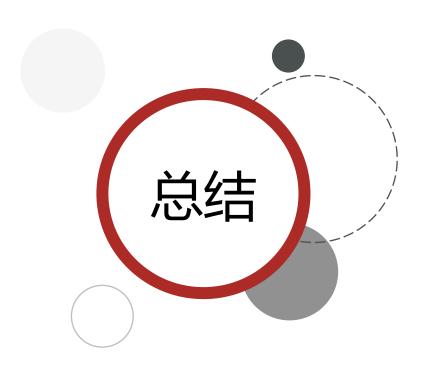
HTTP 响应状态码:用来表明请求是否成功完成

比如: 404 (服务器找不到资源)

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







1. 响应报文的组成:

响应行(状态行):协议、HTTP响应状态码、状态信息

响应头:以键值对的格式携带的附加信息,比如 Content-Type

空行: 分隔响应头, 空行之后的是返回给浏览器的资源

响应体:返回的资源



2. HTTP 响应状态码用来表明请求是否成功完成

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





- ◆ AJAX 概念和 axios 使用
- ◆ 认识 URL
- ◆ URL 查询参数
- ◆ 常用请求方法和数据提交
- ◆ HTTP协议-报文
- ◆ 接口文档
- ◆ 案例 用户登录
- ◆ form-serialize 插件



接口文档

接口文档: 描述接口的文章 (后端工程师)

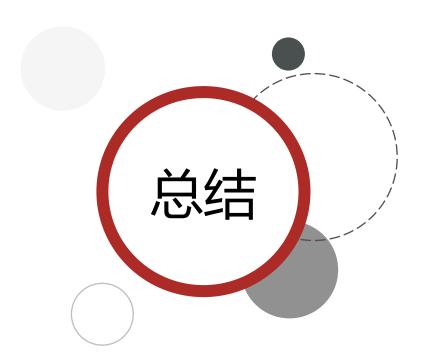
接口:使用 AJAX 和服务器通讯时,使用的 URL,请求方法,以及参数

传送门: AJAX 阶段接口文档









1. 接口文档:

由后端提供的描述接口的文章







- ◆ AJAX 概念和 axios 使用
- ◆ 认识 URL
- ◆ URL 查询参数
- ◆ 常用请求方法和数据提交
- ◆ HTTP协议-报文
- ◆ 接口文档
- ◆ 案例 用户登录
- ◆ form-serialize 插件



案例 - 用户登录

- 1. 点击登录时,判断用户名和密码长度
- 2. 提交数据和服务器通信
- 3. 提示信息







案例 - 用户登录

- 1. 点击登录时,判断用户名和密码长度
- 2. 提交数据和服务器通信
- 3. 提示信息









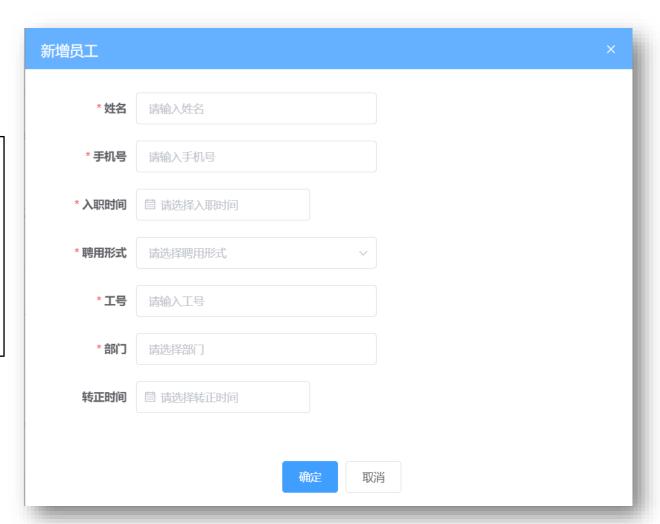
- ◆ AJAX 概念和 axios 使用
- ◆ 认识 URL
- ◆ URL 查询参数
- ◆ 常用请求方法和数据提交
- ◆ HTTP协议-报文
- ◆ 接口文档
- ◆ 案例 用户登录
- ◆ form-serialize 插件



form-serialize 插件

作用: 快速收集表单元素的值

itheima007		
密码		





form-serialize 插件

作用: 快速收集表单元素的值

```
账号名
itheima007

密码
password: '7654321'
}
```

语法:

```
const data = serialize(form, { hash: true, empty: true })

const data = serialize(torm, { hash: true, empty: true })

const data = serialize(torm, { hash: true, empty: true })

const data = serialize(torm, { hash: true, empty: true })
```



案例 - 用户登录

使用 form-serialize 插件,收集用户名和密码





传智教育旗下高端IT教育品牌