

AJAX



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌

14

案例-图书管理

需求

1. 图书列表
2. 新增图书
3. 删除图书
4. 编辑图书

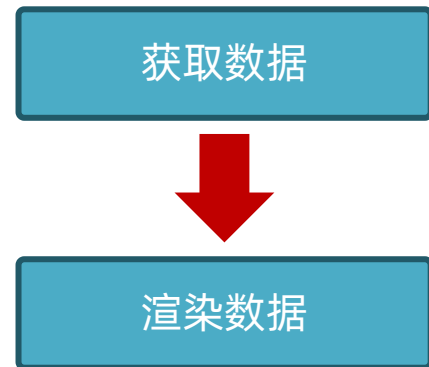
图书管理

[+ 添加](#)

序号	书名	作者	出版社	操作	
1	《西游记》	吴承恩	人民文学出版社	删除	编辑
2	《三国演义》	罗贯中	人民文学出版社	删除	编辑
3	《水浒传》	施耐庵	人民文学出版社	删除	编辑

图书列表

自己的图书数据: 给自己起个**外号**，并告诉服务器，默认会有三本书，基于这三本书做数据的**增删改查**



序号	书名	作者	出版社	操作	
1	《西游记》	吴承恩	人民文学出版社	删除	编辑
2	《三国演义》	罗贯中	人民文学出版社	删除	编辑
3	《水浒传》	施耐庵	人民文学出版社	删除	编辑



需求

1. 图书列表
2. **新增图书**
3. 删除图书
4. 编辑图书

图书管理

[+ 添加](#)

序号	书名	作者	出版社	操作	
1	《西游记》	吴承恩	人民文学出版社	删除	编辑
2	《三国演义》	罗贯中	人民文学出版社	删除	编辑
3	《水浒传》	施耐庵	人民文学出版社	删除	编辑

新增图书

1. 通过属性控制

```
<button data-bs-toggle="modal" data-bs-target="css选择器">  
    弹框  
</button>
```

```
<button data-bs-dismiss="modal">隐藏弹框</button>
```

2. 通过js控制

```
// 创建弹框  
const modalDom = document.querySelector('弹框选择器')  
const modal = new bootstrap.Modal(modalDom)  
// 显示弹框  
modal.show()  
// 隐藏弹框  
modal.hide()
```



添加图书 ×

书名

作者

出版社

取消 增加

需求

1. 图书列表
2. 新增图书
3. 删除图书
4. 编辑图书

图书管理

[+ 添加](#)

序号	书名	作者	出版社	操作	
1	《西游记》	吴承恩	人民文学出版社	删除	编辑
2	《三国演义》	罗贯中	人民文学出版社	删除	编辑
3	《水浒传》	施耐庵	人民文学出版社	删除	编辑

删除图书

绑定点击事件



删除图书

操作	
删除	编辑
删除	编辑
删除	编辑

删除-图书

DELETE <http://hmajax.itheima.net/api/books/{id}>

删除-某本图书

请求参数

参数名	位置	类型	必填	说明
id	path	number	是	示例值: 1 图书id

```
axios({  
  url: 'http://hmajax.itheima.net/api/books/8910',  
  method: 'delete'  
})
```


需求

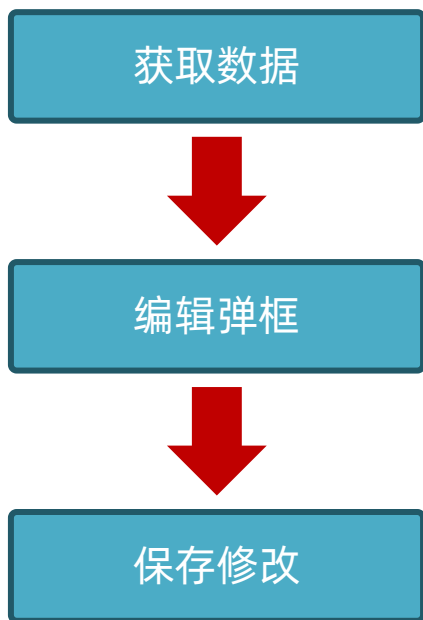
1. 图书列表
2. 新增图书
3. 删除图书
4. **编辑图书**

图书管理

[+ 添加](#)

序号	书名	作者	出版社	操作	
1	《西游记》	吴承恩	人民文学出版社	删除	编辑
2	《三国演义》	罗贯中	人民文学出版社	删除	编辑
3	《水浒传》	施耐庵	人民文学出版社	删除	编辑

编辑图书



1	《西游记》	吴承恩	人民文学出版社	删除	编辑
2	《三国演义》	罗贯中	人民文学出版社	删除	编辑
3	《水浒传》	施耐庵	人民文学出版社	删除	编辑

编辑图书

书名

《三国演义》

作者

罗贯中

出版社

人民文学出版社

取消

修改



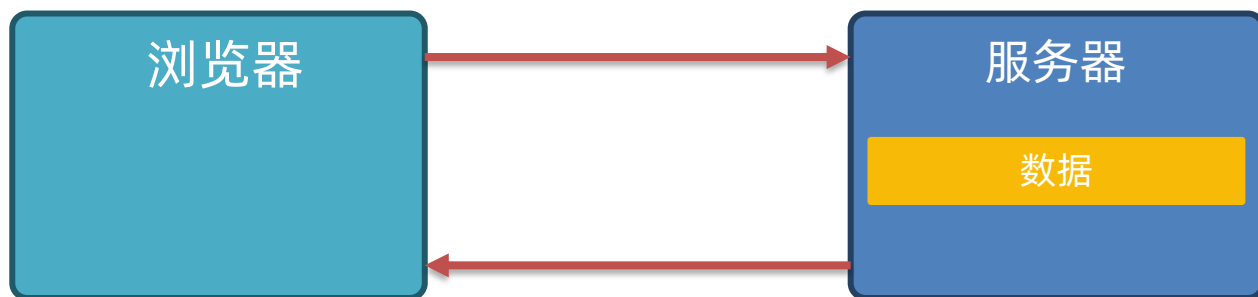
需求

1. 图书列表（渲染数据）
2. 新增图书（新增数据）
3. 删除图书（删除数据）
4. 编辑图书（修改数据）

图书管理

[+ 添加](#)

序号	书名	作者	出版社	操作	
1	《西游记》	吴承恩	人民文学出版社	删除	编辑
2	《三国演义》	罗贯中	人民文学出版社	删除	编辑
3	《水浒传》	施耐庵	人民文学出版社	删除	编辑



核心步骤

渲染数据

获取数据



渲染数据

```
// array->map->htmlArr->join->htmlStr
const htmlArr = list.map((v, index) => {
  return `<tr>
    <td>${index + 1}</td>
    <td>${v.bookname}</td>
    <td>${v.author}</td>
    <td>${v.publisher}</td>
    <td data-id="${v.id}">
      <span class="del">删除</span>
      <span class="edit">编辑</span>
    </td>
  </tr>`
})
// 函数抽取+调用
getBooks()
```

新增数据

删除数据

修改数据

浏览器

服务器

数据

核心步骤

渲染数据

获取数据



渲染数据

新增数据

收集待新增数据



新增数据

```
<button data-bs-toggle="modal" data-bs-target="#css选择器">
  弹框
</button>
```

```
<button data-bs-dismiss="modal">隐藏弹框</button>
```

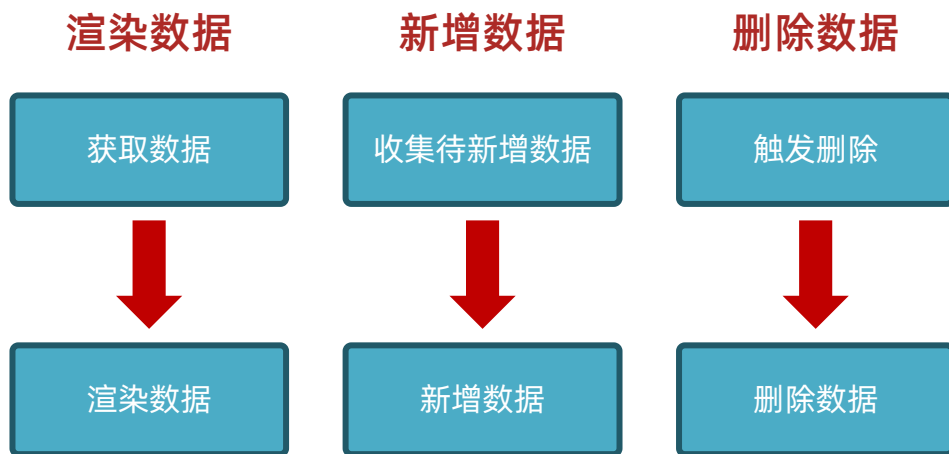
```
// 创建弹框
const modalDom = document.querySelector('弹框选择器')
const modal = new bootstrap.Modal(modalDom)
// 显示弹框
modal.show()
// 隐藏弹框
modal.hide()
```

删除数据

修改数据



核心步骤



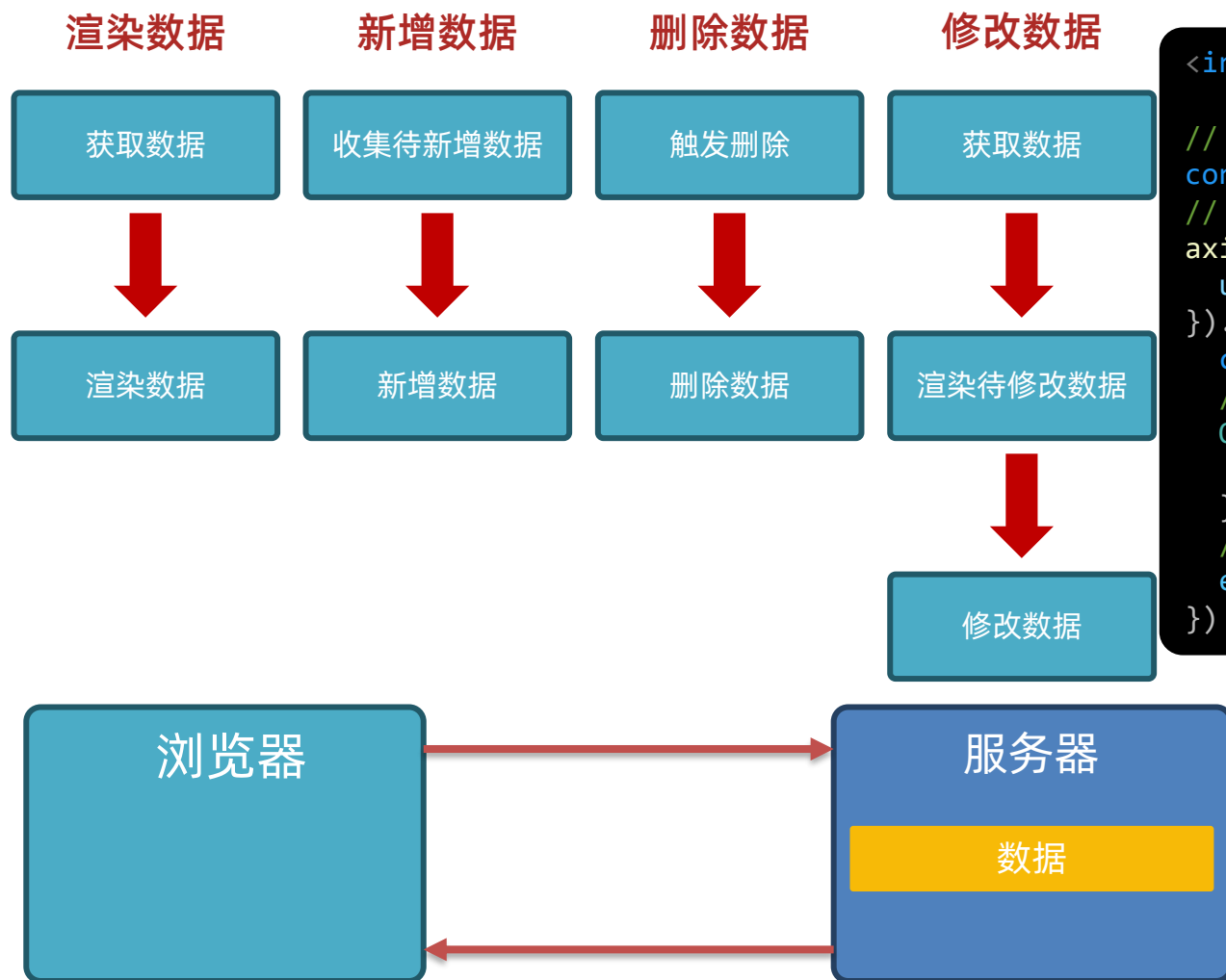
修改数据

```
// 1. 绑定点击事件 - 事件委托
document.querySelector('.list').addEventListener('click', (e) => {
  if (!e.target.classList.contains('del')) {
    return
  }
  // 2. 删除图书
  const { id } = e.target.parentNode.dataset
  axios({
    url: `http://hmajax.itheima.net/api/books/${id}`,
    method: 'DELETE'
  }).then(res => {
    getBooks()
  })
})
```





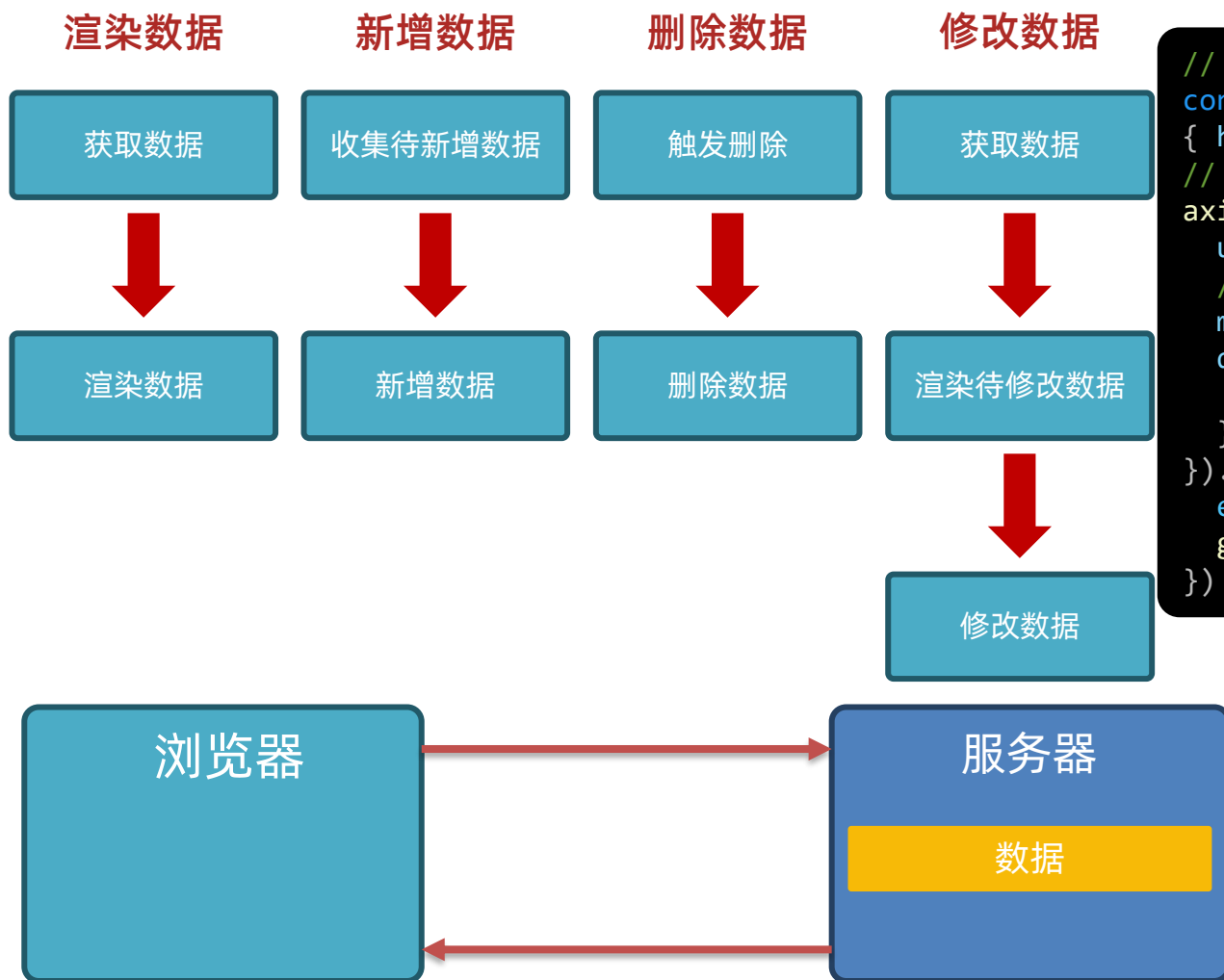
核心步骤



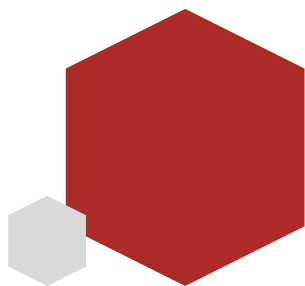
```
<input type="hidden" class="id" name="id">

// 通过自定义属性获取id
const { id } = e.target.parentNode.dataset
// 获取图书数据
axios({
  url: `http://hmajax.itheima.net/api/books/${id}`
}).then(res => {
  const book = res.data.data
  // 设置获取到的图书数据
  Object.keys(book).forEach(key => {
    document.querySelector(`.edit-modal .${key}`).value = book[key]
  })
  // 弹出编辑框
  editModal.show()
})
```

核心步骤



```
// 解构
const { id, bookname, author, publisher } = serialize(editForm,
{ hash: true, empty: true })
// 提交到服务器
axios({
  url: `http://hmajax.itheima.net/api/books/${id}`,
  // put 修改数据的请求方法
  method: 'put',
  data: {
    bookname, author, publisher, creator
  }
}).then(res => {
  editModal.hide()
  getBooks()
})
```

图片上传

图片上传

1. 选择图片

```
<input type="file" class="upload" >
```

```
document.querySelector('.upload').addEventListener('change', function ()  
{  
    console.log(this.files[0])  
})
```

2. FormData对象

内置对象，负责向服务器传递表单数据，常用于文件上传

```
const data = new FormData()  
data.append(key,value)
```

3. 调用上传接口

```
axios({  
    url:'http://hmajax.itheima.net/api/uploading',  
    method:'post',  
    data:formData对象  
})
```

上传-图片

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

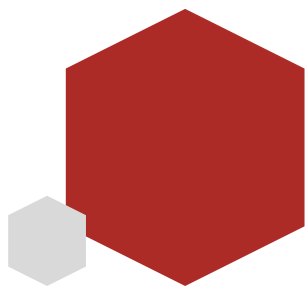
上传-图片

注意: 上传的图片必须在2MB以内

请求参数

Body 参数(multipart/form-data)

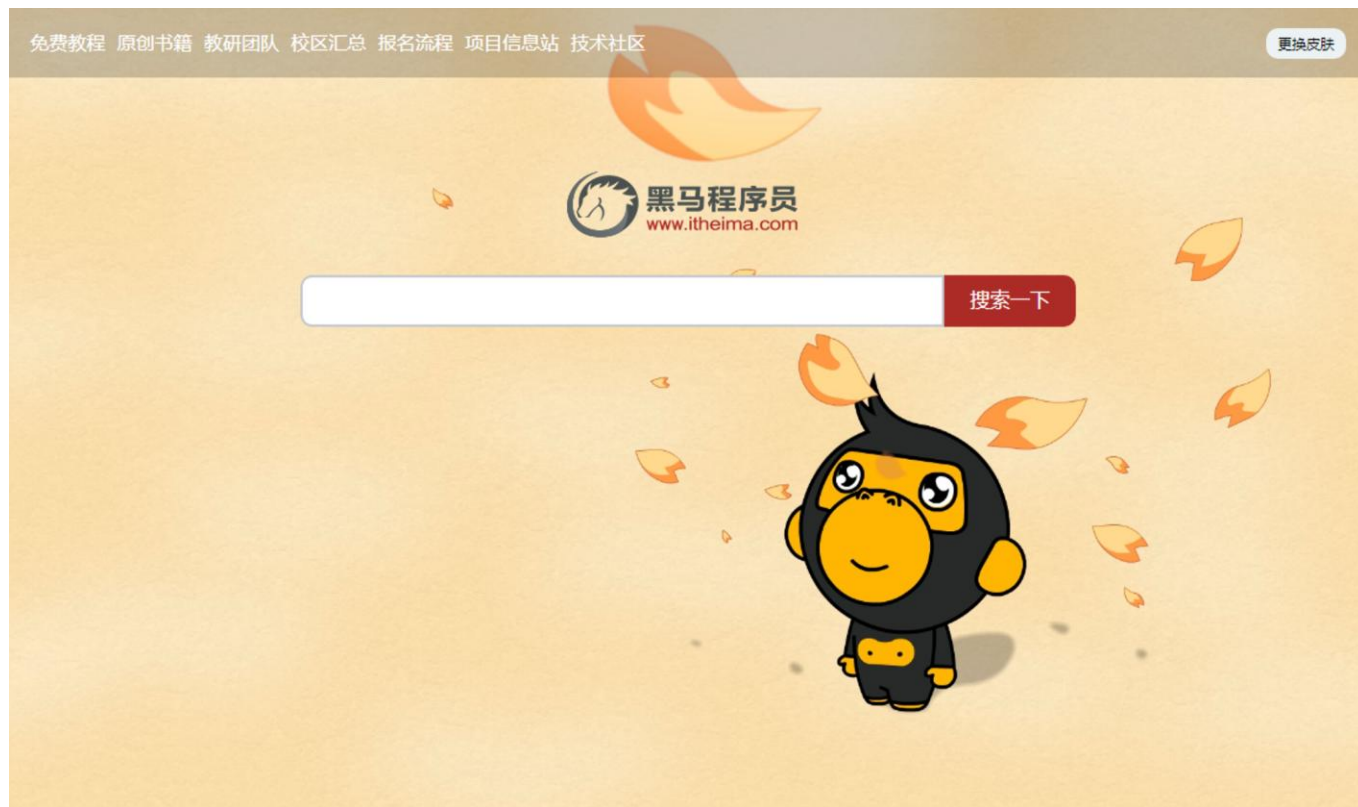
参数名	类型	必填	说明
img	file	是	图片文件(2MB以内)



案例-网站换肤

需求

- 网站换肤（核心功能）
 - 上传图片
 - 设置body的背景图片
- 本地持久化图片URL
 - 保存图片URL
 - 读取图片URL





总结

1. label标签使用技巧

```
<label for="upload">更换皮肤</label>  
<input id="upload" class="skin-ipt" type="file">
```

2. 网站换肤

上传图片

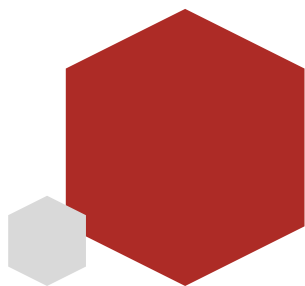
设置body的背景图片

3. 本地持久化图片URL

保存图片URL

读取图片URL

```
localStorage.setItem(key, value)  
localStorage.getItem(key)
```



案例-个人设置

需求

- 信息渲染
- 修改头像
- 修改信息

基本设置

安全设置

账号绑定

新消息通知

基本设置

邮箱

请输入邮箱

昵称

请输入昵称

性别

☐ 男 ☐ 女

个人简介

请输入个人简介

提交

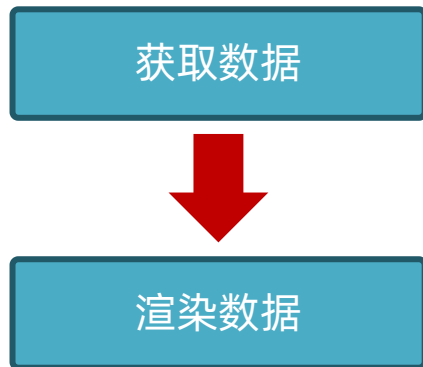
头像



更换头像

信息渲染

自己的用户信息: 给自己起个外号，并告诉服务器，获取对应的用户信息



基本设置

安全设置

账号绑定

新消息通知

基本设置

邮箱

昵称


性别
☒ 男 ☐ 女

个人简介

我是播仔

提交

头像



更换头像

需求

- 信息渲染
- 修改头像
- 修改信息

基本设置

安全设置

账号绑定

新消息通知

基本设置

邮箱

头像

昵称

性别

个人简介

提交

请输入邮箱

请输入昵称

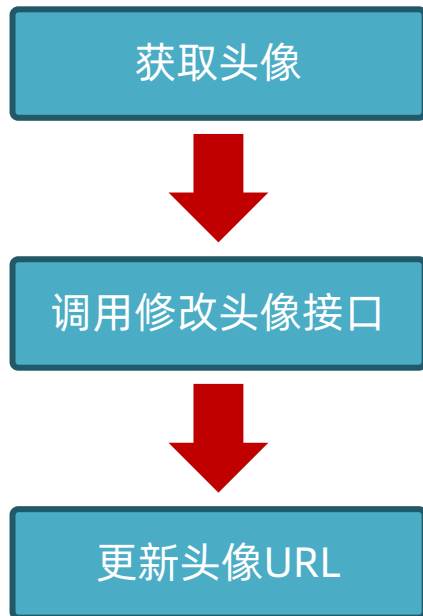
男

女

请输入个人简介

更换头像

修改头像



修改-个人头像

PUT <http://hmajax.itheima.net/api/avatar>

修改-个人头像

请求参数

Body 参数(multipart/form-data)

参数名	类型	必填
avatar	file	是
creator	string	是

需求

- 信息渲染
- 修改头像
- 修改信息

基本设置

安全设置

账号绑定

新消息通知

基本设置

邮箱

头像

昵称

性别

个人简介

提交

请输入邮箱

请输入昵称

男

女

请输入个人简介

更换头像

修改信息

收集表单数据

↓

调用修改信息接口

↓

提示用户

基本

操作成功

邮箱

itheim111@itcast.cn

头像

itheima111

更换头像

性别

☐ 男 ☒ 女

个人简介

我是测试用1111

提交

```
<div class="toast" data-bs-delay="1500" >  
  // 内容略  
</div>
```

```
// 创建弹框  
const toastDom = document.querySelector('.toast')  
const toast = new bootstrap.Toast(toastDom)  
// 显示弹框  
toast.show()
```



总结

1. 信息渲染

获取 -> 渲染

2. 修改头像

调用接口 -> 更新头像URL

3. 修改信息

收集数据 -> 调用接口->提示用户

```
<div class="toast" data-bs-delay="1500" >  
  // 内容略  
</div>
```

```
// 创建弹框  
const toastDom = document.querySelector('提示框选择器')  
const toast = new bootstrap.Toast(toastDom)  
// 显示弹框  
toast.show()
```



传智教育旗下高端IT教育品牌