# QG工作室暑期实习生周记

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名： 林润鑫 | 组别：前端组 | 年级：2024级 | 周次：第四周 |

|  |  |
| --- | --- |
| **第四周周记** | |
| **生活随记** | 我来交周记了  这周有在好好吃饭，好好睡觉，好好干活，好好上班，好好下班，好好聊天    这可怎么办啊 |
| **学习**  **开发**  **比赛**  **情况** | 主要是学习了websocket和socket.io，因为一开始我想着没什么区别直接用考核的时候用的socket.io，害我还得重新写一次  然后主要了解一些关于SDK的内容  然后学习了一下中期考核中发现的遗漏的点 |
| **一周总结** | 保持了稳中有进的工作总基调 |
| **存在问题**  **未来规划** | 缺乏运动，我的运动大计已经十不存一  对于未来要学习的技术栈不清晰，没找到感兴趣的点，导致没有什么动力， |
| **导师评价** |  |

## 一、GET请求 - 使用params传递参数

GET请求的参数应该放在URL的查询字符串中，Axios中通过`params`配置项实现：

```js

// 查询用户列表，带分页参数

axios.get('/api/users', {

params: {

page: 1,

pageSize: 10,

keyword: 'John'

}

});

// 实际请求URL: /api/users?page=1&pageSize=10&keyword=John

```

## 二、POST请求 - 使用data传递请求体

POST请求通常用于创建资源，参数放在请求体(body)中：

```js

// 创建新用户

axios.post('/api/users', {

name: 'John Doe',

email: 'john@example.com',

password: '123456'

});

// 如果需要URL参数和请求体同时存在

axios.post('/api/users?type=admin', {

name: 'John Doe',

email: 'john@example.com'

});

```

## 三、PUT/PATCH请求 - 更新资源

PUT(全量更新)和PATCH(部分更新)也使用请求体传递参数：

```js

// 全量更新用户信息(PUT)

axios.put('/api/users/123', {

name: 'John Smith',

email: 'john.smith@example.com'

});

// 部分更新用户信息(PATCH)

axios.patch('/api/users/123', {

email: 'new.email@example.com'

});

```

## 四、DELETE请求 - 通常使用URL参数

DELETE请求一般通过URL标识要删除的资源，有时也需要请求体：

```js

// 简单删除

axios.delete('/api/users/123');

// 带额外参数的删除

axios.delete('/api/users/123', {

params: {

force: true // 强制删除

}

});

// 需要请求体的删除(较少见)

axios.delete('/api/users/batch', {

data: {

ids: [123, 456, 789]

}

});

```

## 五、特殊参数传递场景

### 1. 数组参数

```js

// GET请求中的数组参数

axios.get('/api/users', {

params: {

ids: [1, 2, 3],

// 实际URL: /api/users?ids=1&ids=2&ids=3

}

});

// POST请求中的数组

axios.post('/api/users/batch', {

ids: [1, 2, 3]

});

```

### 2. 嵌套对象参数

```js

// 嵌套对象在GET请求中会被flatten

axios.get('/api/users', {

params: {

filter: {

status: 'active',

age: { gt: 18 }

}

// 实际URL: /api/users?filter[status]=active&filter[age][gt]=18

}

});

// POST请求中可以直接传递嵌套对象

axios.post('/api/users/search', {

filter: {

status: 'active',

age: { gt: 18 }

}

});

```

## 六、Content-Type注意事项

Axios会根据数据类型自动设置Content-Type：

- `对象` → `application/json`

- `FormData` → `multipart/form-data`

- `URLSearchParams` → `application/x-www-form-urlencoded`

手动设置示例：

```js

// 发送form-data格式

const formData = new FormData();

formData.append('username', 'John');

axios.post('/api/login', formData);

// 发送x-www-form-urlencoded格式

const params = new URLSearchParams();

params.append('username', 'John');

axios.post('/api/login', params);

```