

【前端面试】Ajax、Fetch、Axios三者的区...

前言

做前端开发的小伙伴一定离不开“请求”两个字，这是我们与后端交互的最重要的途径！在前几年面试的时候，面试官很喜欢问的一个问题就是让面试者手撕代码实现 Ajax，我相信很多小伙伴一定遇到过。直至今日，又出现了许多关于请求的新名词，今天就来理一理这些新名词的区别。

这些名词的共同点：都用于发送网络请求。

1.Ajax

它的全称是：Asynchronous JavaScript And XML，翻译过来就是“异步的 Javascript 和 XML”。

很多小伙伴可能会误以为 Ajax 是发请求的一种方式，或者把 XMLHttpRequest 与 Ajax 划等号，其实这是错误和片面的。

正解：

Ajax 是一个技术统称，是一个概念模型，它囊括了很多技术，并不特指某一技术，它很重要的特性之一就是让页面实现局部刷新。

特点：

- 局部刷新页面，无需重载整个页面。

简单来说，Ajax 是一种思想，XMLHttpRequest 只是实现 Ajax 的一种方式。其中 XMLHttpRequest 模块就是实现 Ajax 的一种很好的方式，这也是很多面试官喜欢让面试者手撕的代码之一。

利用 XMLHttpRequest 模块实现 Ajax。

示例代码：

```
1 <body>
2   <script>
3     function ajax(url) {
4       const xhr = new XMLHttpRequest();
5       xhr.open("get", url, false);
6       xhr.onreadystatechange = function () {
7         // 异步回调函数
```

```

8         if (xhr.readyState === 4) {
9             if (xhr.status === 200) {
10                console.info("响应结果", xhr.response)
11            }
12        }
13    }
14    xhr.send(null);
15 }
16 ajax('https://smallpig.site/api/category/getCategory')
17 </script>
18 </body>

```

输出结果：

```

响应结果 {data:{"page":{"currentPage":null,"pageCount":1,"count":4,"isEnd":false},"data":
index.html:20
[{"_id":"616823e2d81ff7037690b5d8","category_id":1,"__v":0,"category_menu_parent_id":0,"category_menu_id":1,"category_menu_name":"资源分享","category_name":"电脑软件"},
{"_id":"617a935fe977d25212d4fe3a","category_id":2,"__v":0,"category_menu_parent_id":0,"category_menu_id":1,"category_menu_name":"资源分享","category_name":"手机软件"},
{"_id":"617a9373e977d25212d4fe3b","category_id":3,"__v":0,"category_menu_parent_id":0,"category_menu_id":1,"category_menu_name":"资源分享","category_name":"精选TV"},
{"_id":"617a93b1e977d25212d4fe3c","category_id":4,"__v":0,"category_menu_parent_id":0,"category_menu_id":1,"category_menu_name":"资源分享","category_name":"教程资源"}]},code:"00000",message:"查询成功"}

```

这里利用 XMLHttpRequest 模块实现了一个最简单的 get 网络请求。

注意：我们使用这种方式实现网络请求时，如果请求内部又包含请求，以此循环，就会出现回调地狱，这也是一个诟病，后来才催生了更加优雅的请求方式。

2.Fetch

Fetch 是在 ES6 出现的，它使用了 ES6 提出的 promise 对象。它是 XMLHttpRequest 的替代品。

很多小伙伴会把它与 Ajax 作比较，其实这是不对的，我们通常所说的 Ajax 是指使用

XMLHttpRequest 实现的 Ajax，所以真正应该和 XMLHttpRequest 作比较。

正解：

Fetch 是一个 API，它是真实存在的，它是基于 promise 的。

特点：

- 使用 promise，不使用回调函数。
- 采用模块化设计，比如 rep、res 等对象分散开来，比较友好。
- 通过数据流对象处理数据，可以提高网站性能。

所以这里就和 Ajax 又很大不同了，一个是思想，一个是真实存在的 API，不过它们都是用来给网络请求服务的，我们一起来看看利用 Fetch 实现网络请求。

示例代码：

```

1 <body>
2   <script>
3     function ajaxFetch(url) {
4       fetch(url).then(res => res.json()).then(data => {

```

```

5     console.info(data)
6   })
7 }
8   ajaxFetch('https://smallpig.site/api/category/getCategory')
9 </script>
10 </body>

```

输出结果:

```

▼ {data: {...}, code: '00000', message: '查询成功'} ⓘ
  code: "00000"
  ▶ data: {page: {...}, data: Array(4)}
  message: "查询成功"
  ▶ [[Prototype]]: Object

```

上段代码利用 Fetch 发送了一个最简单的 get 请求，其中最重要的特点之一就是采用了.then 链式调用的方式处理结果，这样不仅利于代码的可读，而且也解决了回调地狱的问题。

3.Axios

Axios 是随着 Vue 的兴起而被广泛使用的，目前来说，绝大多数的 Vue 项目中的网络请求都是利用 Axios 发起的。当然它并不是一个思想，或者一个原生 API，它是一个封装库。

正解:

Axios 是一个基于 promise 封装的网络请求库，它是基于 XHR 进行二次封装。

特点:

- 从浏览器中创建 XMLHttpRequests
- 从 node.js 创建 http 请求
- 支持 Promise API
- 拦截请求和响应
- 转换请求数据和响应数据
- 取消请求
- 自动转换 JSON 数据
- 客户端支持防御 XSRF

所以说，Axios 可以说是 XHR 的一个子集，而 XHR 又是 Ajax 的一个子集。既然说它是一个库，那么我们在使用的时候就需要引入它。

示例代码:

```

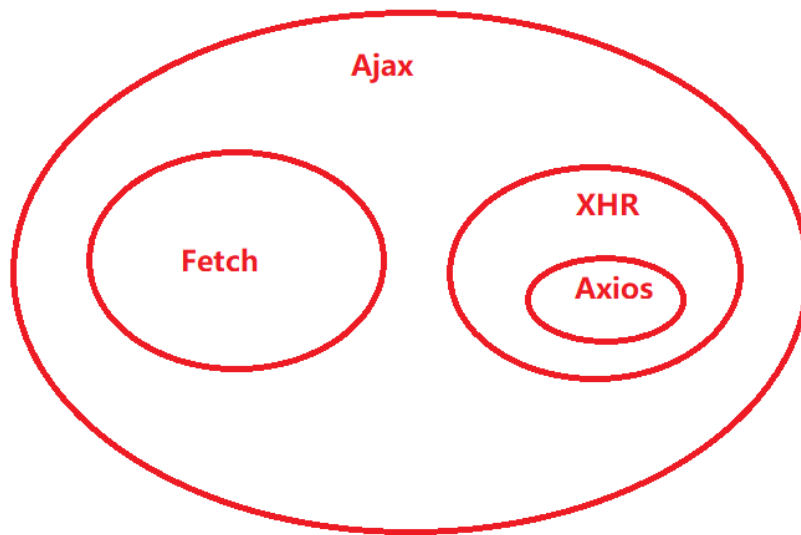
1 // 发送 POST 请求
2 axios({
3   method: 'post',

```

```
4     url: '/user/12345',
5     data: {
6         firstName: 'Fred',
7         lastName: 'Flintstone'
8     }
9 })
```

总结

Ajax、Fetch、axios三者之间的关系可以用一张图来清晰的表示，如图：



三者做个对比：

| | |
|-------|------------------------|
| Ajax | 一种技术统称，主要利用XHR实现网络请求 |
| Fetch | 具体API，基于promise，实现网络请求 |
| Axios | 一个封装库，基于XHR封装，较为推荐使用 |