

# Fetch API 是什么？能代替 AJAX 吗？

短答案：

Fetch 是浏览器提供的原生 AJAX 接口。使用 `window.fetch` 函数可以代替以前的 `$.ajax`、`$.get` 和 `$.post`。

长答案：

前端发展地越来越快，我们用了好几年的 `$.ajax`，居然也渐渐变得过时了。以前我们用 `jQuery.ajax` 发一个请求是这样的：

```
$.ajax('/').then(function(response){
    console.log(response)
})
```

现在我们用 `fetch` 发一个请求是这样的：

```
fetch('/').then(function(response){
    response.text().then(function(text){
        console.log(text)
    })
})
```

是不是感觉很像，像就对了，因为 Fetch API 就是浏览器提供的用来代替 `jQuery.ajax` 的工具。

## AJAX 的原理

我们知道 `jQuery.ajax` 是使用 `XMLHttpRequest` 对象来发送异步请求的。

```
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
    if (this.readyState === 4 && this.status === 200) {
        console.log(this.responseText);
    }
};
xhttp.open("GET", "/", true);
xhttp.send();
```

是不是很麻烦，是不是？

正是由于 `XMLHttpRequest` 使用起来相当麻烦，所以大家才喜欢使用 `jQuery` 提供的 `$.ajax` 方法。

## 用 `fetch` 代替 `$.ajax`

随着 React.js、Angular.js 和 Vue.js 这些前端框架的流行，很多单页面应用已经不再使用 jQuery 了，这意味着你要自己对 XMLHttpRequest 进行封装，而很多人选择封装一个跟 jQuery.ajax 差不多的接口。

Fetch API 的出现，就是为了给类似的操作流程定一个接口规范。

换句话说，就是浏览器帮你把 jQuery.ajax 给实现了，以后大家都是用 fetch 来发送异步请求就好了。

## Fetch API 提供的一组对象

window.fetch 函数只是 Fetch API 提供的众多接口中的一个，还有很多有用的对象：

- window.Headers
- window.Response
- window.Request
- .....

要学的东西又多了起来.....

## Fetch API 的特点

1. 基于 Promise（如果你没有学过 Promise，强烈建议你学一学）
2. 不需要依赖第三方库，就可以优雅地使用 AJAX

## Fetch API 的问题

使用 fetch 无法取消一个请求。这是因为 Fetch API 基于 Promise，而 Promise 无法做到这一点。不过相信很快就会有对策。

## 兼容性

有的浏览器没有 Fetch API，没有关系，只要引入一个 polyfill 就可以了：[GitHub - github/fetch: A window.fetch JavaScript polyfill.](https://github.com/github/fetch)

短答案：