# Axios

## 基础

### 特性

* 从浏览器中创建 [XMLHttpRequests](https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest" \t "_blank)
* 从 node.js 创建 [http](http://nodejs.org/api/http.html) 请求
* 支持 [Promise](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/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](http://en.wikipedia.org/wiki/Cross-site_request_forgery)

### 安装

使用npm

npm install axios

使用 bower:

$ bower install axios

使用 cdn:

<script src="https://unpkg.com/axios/dist/axios.min.js"></script>

### 基础案例

执行GET请求：

// 为给定 ID 的 user 创建请求  
axios.get('/user?ID=12345')  
 .then(function (response) {  
 console.log(response);  
 })  
 .catch(function (error) {  
 console.log(error);  
 });  
  
// 上面的请求也可以这样做  
axios.get('/user', {  
 params: {  
 ID: 12345  
 }  
 })  
 .then(function (response) {  
 console.log(response);  
 })  
 .catch(function (error) {  
 console.log(error);  
 });

执行 POST 请求

axios.post('/user', {  
 firstName: 'Fred',  
 lastName: 'Flintstone'  
 })  
 .then(function (response) {  
 console.log(response);  
 })  
 .catch(function (error) {  
 console.log(error);  
 });

执行多个并发请求

function getUserAccount() {  
 return axios.get('/user/12345');  
}  
  
function getUserPermissions() {  
 return axios.get('/user/12345/permissions');  
}  
  
axios.all([getUserAccount(), getUserPermissions()])  
 .then(axios.spread(function (acct, perms) {  
 // 两个请求现在都执行完成  
 }));

## 进阶

### Axios API

可以通过向axios 传递相关配置来创建请求

#### Axios（config）

// 发送 POST 请求  
axios({  
 method: 'post',  
 url: '/user/12345',  
 data: {  
 firstName: 'Fred',  
 lastName: 'Flintstone'  
 }  
});

// 获取远端图片  
axios({  
 method:'get',  
 url:'http://bit.ly/2mTM3nY',  
 responseType:'stream'  
})  
 .then(function(response) {  
 response.data.pipe(fs.createWriteStream('ada\_lovelace.jpg'))  
});

#### axios(url[, config])

// 发送 GET 请求（默认的方法）  
axios('/user/12345');

#### 请求方法的别名

为方便起见，为所有支持的请求方法提供了别名

axios.request(config)

axios.get(url[, config])

axios.delete(url[, config])

axios.head(url[, config])

axios.options(url[, config])

axios.post(url[, data[, config]])

axios.put(url[, data[, config]])

axios.patch(url[, data[, config]])

**注意:**

在使用别名方法时， url、method、data 这些属性都不必在配置中指定。

#### 并发

处理并发请求的助手函数

axios.all(iterable)

axios.spread(callback)

#### 创建实例

可以使用自定义配置新建一个 axios 实例

const instance = axios.create({  
 baseURL: 'https://some-domain.com/api/',  
 timeout: 1000,  
 headers: {'X-Custom-Header': 'foobar'}  
});

#### 实例方法

以下是可用的实例方法。指定的配置将与示例的配置合并。

axios#request(config)

axios#get(url[, config])

axios#delete(url[, config])

axios#head(url[, config])

axios#options(url[, config])

axios#post(url[, data[, config]])

axios#put(url[, data[, config]])

axios#patch(url[, data[, config]])

#### 请求配置

这些是创建请求时可以用的配置选项。只有 url 是必需的。如果没有指定 method，请求将默认使用 get 方法。