

06.axios 实例化和拦截

学习要点：

- 1.实例化操作
- 2.拦截操作

本节课我们来开始了解 `axios.js` 的实例化和拦截操作

一. 实例化操作

1. 全局属性配置，即将属性中大量重复的内容抽离出来单独设置；

//全局默认值设置，baseUrl 会自动在 url 前面添加

```
axios.defaults.baseURL = 'https://cdn.liyanhui.com';
```

PS：其它更多属性可以参考手册，一般来说 `baseURL`，`timeout` 和 `headers` 比较常用；

2. 实例化，顾名思义就是 `new` 出一个对象来，这样这个对象具有独立性不被干扰；

//实例化

```
//const myAxios = axios.create();
```

//设置默认值

```
//myAxios.defaults.baseURL = 'https://cdn.liyanhui.com';
```

//或

```
const myAxios = axios.create({  
  baseURL : 'https://cdn.liyanhui.com',  
});
```

//request 通用

```
myAxios.request({  
  method : 'get',  
  url : '/data.json'  
}).then(res => {  
  console.log(res.data);  
});
```

二. 拦截操作

1. 所谓拦截操作，就是在 `Ajax` 获取数据之前先拦截处理一些事务的操作；
2. 这些操作包括：修改 `axios` 配置信息，`loading`，判断验证跳转等等；
3. 拦截处理完毕之后，再通过 `return` 返回基础处理即可；

//添加请求拦截

```
myAxios.interceptors.request.use(config => {  
  console.log('loading...');  
  return config;  
});
```

```
//添加响应拦截
myAxios.interceptors.response.use(response => {
    //在输出数据前，对数据进行拦截处理
    return response.data;
});

//request 通用
myAxios.request({
    method : 'get',
    url : '/data.json'
}).then(res => {
    console.log(res);
});
```