

# Ajax和Axios学习

## JSON

### 1. JSON定义

JSON是一串字符串

js中：

```
var 变量名 = '{ "key1": value1, "key2": value2, "key3": value3}'
```

### 2. JSON与JS对象转换

1. JSON字符串转换为JS对象： `JSON.parse(JSON字符串);`

```
var jsObject = JSON.parse(jsonStr);
```

2. JS对象转换为JSON字符串： `JSON.stringify(js对象);`

```
var jsonStr = JSON.stringify(jsObject);
```

## Ajax

### 1. Ajax介绍

Ajax: Asynchronous JavaScript And XML, 异步JavaScript和XML

作用：

- 数据交换：通过Ajax可以给服务器发送请求，并获取服务器响应的数据
- 异步交互：可以在不重新加载整个页面的情况下，与服务器交换数据并更新部分网页的技术

参考手册：[AJAX 简介](#)

### 2. 原生Ajax

1. 准备数据地址 (json格式) : <http://xxx.com/path/json>
2. 创建XMLHttpRequest对象：用与和服务器交换数据
3. 向服务器发送请求
4. 获取服务器响应数据

```

<body>
    <input type="button" value="获取数据" onclick="getData()">
    <div id="div1"></div>
</body>

<script>
    function getData(){
        //1. 创建XMLHttpRequest
        var XMLHttpRequest = new XMLHttpRequest();

        //2. 发送异步请求
        XMLHttpRequest.open('GET','http://yapi.sm.../mock/169327/emp/list');
        XMLHttpRequest.send(); //发送请求

        //3. 获取服务响应数据
        XMLHttpRequest.onreadystatechange = function(){
            if(XMLHttpRequest.readyState == 4 && XMLHttpRequest.status == 200){
                document.getElementById('div1').innerHTML = XMLHttpRequest.responseText;
            }
        }
    }
</script>

```

# Axios

## 1. Axios介绍

Axios：对原生Ajax的封装，简化代码

参考手册：[Axios中文文档](#) | [Axios中文网](#)

## 2. Axios入门

### 2.1 Axios基础使用：

1. 引入Axios的js文件

```
<script src="js/axios-0.18.0.js"></script>
```

2. 使用Axios发送请求，并获取响应结果

**get请求：**

```

axios({           // 传递两个参数的js对象{method: value, url: value}
    method:"get",          // 请求方式
    url:"http://xxx.com/path/json"      // 数据位置
}).then((result) => {           // 成功回调函数，服务器返回result的json对象
    console.log(result.data)       // result.data即服务器响应数据
})

```

**post请求：**

```

axios({           // 传递两个参数的js对象{method: value, url: value}
    method: "post",        // 请求方式
    url: "http://xxx.com/path/json",    // 数据位置
    data: "id=1"        // 请求参数
}).then((result) => {           // 成功回调函数，服务器返回result的json对象
    console.log(result.data)       // result.data即服务器响应数据
})

```

## 2.2 请求方式别名：

- `axios.get(url [, config])`
- `axios.delete(url [, config])`
- `axios.post(url [, data[config]])`
- `axios.put(url [, data[,config]])`

Axios简化使用：

**get请求：**

```
axios.get("http://xxx.com/path/json").then((result) => {
  console.log(result.data)
})
```

**post请求：**

```
axios.post("http://xxx.com/path/json", "id=1").then((result) => {
  console.log(result.data)
})
```

## 2.3 axios的数据封装

axios会自动封装服务器返回的数据，包括`status`、`data`等，比如：

```
{
  "code": 200, // axios封装
  "msg": "Success", // axios封装
  "data": { // 原始数据（服务器响应数据）
    "字段1": "值",
    "data": "值"
  }
}
```

因此，要获取使用axios获取的数据，需要使用`result.data`获得原始数据

# 前后端分离开发

## 1. 业务流程

(原型和需求) -> 需求分析 -> 接口定义 (API接口文档) -> 前后端并行开发 (遵守规范) -> 测试 (前端) -> 前后端联调测试

## 2. 接口文档，YApi

接口文档：产品经理通过页面原型和需求提供的

YApi：api管理平台，[YApi Pro-高效、易用、功能强大的可视化接口管理平台](#)

- API接口管理
- Mock服务