

《Vue进阶》第四单元

后端 API 请求及 axios



API

- Application Programming Interface
- API = 接口
- 前端和后端的数据通信方式

AJAX

- 用 JS 脚本操作 HTTP 请求
- XMLHttpRequest 对象，异步
- 支持多种数据类型传输，JSON

JSON

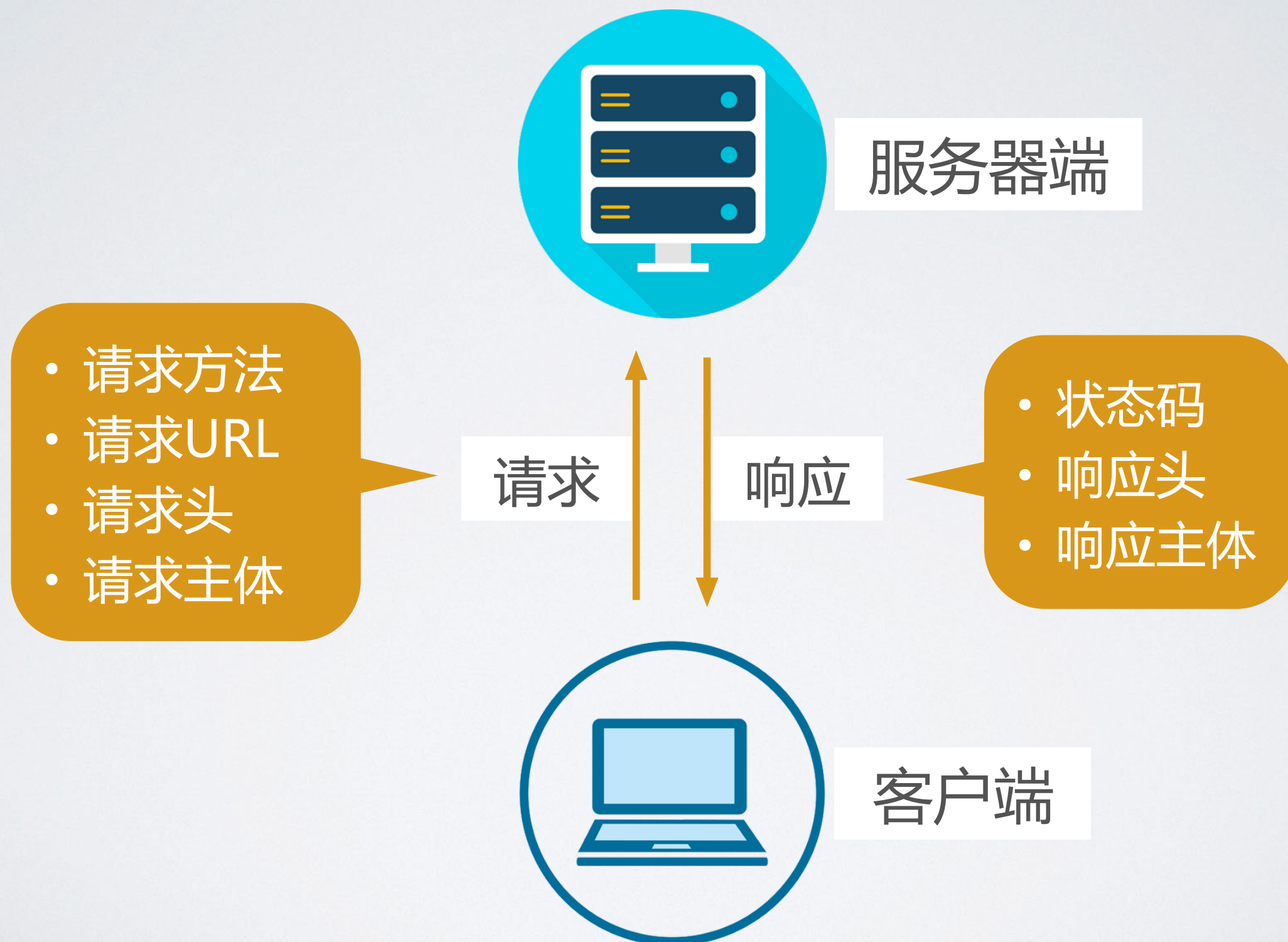
- 轻量级的数据交换格式，易读易写易解析
- 双括号表示对象 { }
- 中括号表示数组 []
- 双引号内是属性或值 " "
- 冒号表示后者是前者的值：
- 值可以是双引号括起来的字符串、数字、true、false、null、对象或者数组

JSON

```
{  
  "people": [{  
    "name": "Zhang San",  
    "age": 29,  
    "isAdult": true  
  }, {  
    "name": "Li Si",  
    "age": 15,  
    "isAdult": false  
  }]  
}
```

JSON是Web开发中，前后端公认的标准传输格式

HTTP 请求



XMLHttpRequest

课堂案例

```
const xhr = new XMLHttpRequest();
xhr.open('POST', 'http://jxsjs.com/song-search');
xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');
xhr.onreadystatechange = () => {
  if (xhr.readyState === 4 && xhr.status === 200) {
    const type = xhr.getResponseHeader('Content-Type');
    if (type.indexOf('/application/json') > -1) {
      console.log(JSON.parse(xhr.responseText));
    }
  }
};
xhr.send(JSON.stringify({
  "keyword": "海阔天空"
}));
```

一次 JSON 数据格式的 AJAX 请求

Content-Type

Content-Type	格式
text/html	HTML 文件
text/css	CSS 文件
text/javascript	JS 文件
text/plain	纯文本
image/jpeg	jpg 图片
application/json	JSON 数据
application/pdf	pdf 文件
application/octet-stream	二进制流数据
application/x-www-form-urlencoded	表单默认提交数据
multipart/form-data	在表单中文件上传

- HTTP 请求头中传输类型的定义

axios

- 基于 Promise 和 XMLHttpRequest 的请求封装库
- 支持请求和响应拦截 Interceptors
- 自动转换 JSON 数据

安装 axios

- `npm install axios`
- `Vue.prototype.$http = axios`

将 axios 挂在 Vue 构造函数原型上，
每个 Vue 实例通过 this 方便使用

axios 请求

```
import axios from 'axios';
```

```
axios.get(url).then(response => {  
  console.log(response);  
}).catch(error => {  
  console.log(error);  
});
```

通过实例 this 调用

```
this.$http.get(url).then(response => {  
  console.log(response);  
}).catch(error => {  
  console.log(error);  
});
```



回忆：Promise 的用法

axios 请求

```
· axios.get('singer-detail?id=003CoxJh1zFPpx');  
· axios.get('singer-detail', {  
·   ·   · params: {  
·   ·   ·   · id: '003CoxJh1zFPpx'  
·   ·   · }  
· });
```

- get 请求在 url 中拼参数，也可用 params 定义参数

```
· axios.post('song-search', {  
·   ·   · keyword: '海阔天空'  
· });
```

axios 自动将数据请求
使用 JSON 格式传输

- post 请求

axios 配置

- timeout 请求超时设置
- baseURL 公共路径前缀
- headers 请求头设置
- responseType 响应类型设置

```
axios.defaults.timeout = 30000;  
axios.defaults.baseURL = '/api';  
axios.defaults.responseType = 'json';  
axios.headers.post['Content-Type'] = 'application/json';
```

axios 数据格式

- axios 封装了服务器端的数据响应

```
{  
  data: {},  
  status: 200,  
  statusText: 'OK',  
  headers: {},  
  config: {},  
  request: {}  
}
```

服务器端返回的数据

HTTP请求状态码

HTTP请求状态文本

请求响应头

请求的配置

XMLHttpRequest 实例

axios 拦截 Interceptors

- request 请求拦截器，可以用来在请求发送之前统一定义请求头等
- response 响应拦截器，可以用来在收到服务器响应后，统一做权限跳转或者错误提示等

```
axios.interceptors.response.use(({ data }) => {  
  if (data.code > 0) {  
    toast(data.msg || 'Sorry, 暂不可以');  
  }  
  return data;  
}, error => {  
  return Promise.reject(error);  
});
```

音乐播放器接口

功能	地址	方法	参数
首页轮播图	slider-list	GET	-
热门单曲推荐	hit-single-list	GET	-
歌手列表	singer-list	GET	-
歌手详情	singer-detail	GET	id
歌词	lrc-detail	GET	id
热门搜索	hot-search	GET	-
搜索歌曲	song-search	POST	keyword

- 公共前缀 <http://jxsjs.com/api/>

接口代理

- 本地服务器代理 api 接口请求，绕开浏览器跨域限制

```
module.exports = {  
  devServer: {  
    proxy: {  
      '/api': {  
        target: 'http://jxsjs.com/',  
        changeOrigin: true,  
        pathRewrite: {  
          '/api': ''  
        }  
      }  
    }  
  }  
};
```

vue.config.js