

请求后端

axios

文档: <https://www.axios-http.cn/docs/intro>

安装

```
npm install axios
```

请求

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

封装请求

utils/request.js

```
import axios from "axios";
import { MessageBox, Message } from "element-ui";
const request = axios.create({
  baseURL: "/api", //请求的url
  timeout: 10000, // 请求超时
  withCredentials: false, //跨域请求是否需要携带 cookie
});
function getToken() {
  //获取token, token可以放在store, 也可以放在sessionStorage或localStorage
  const token = sessionStorage.getItem("token") || "";
  return token;
}
request.interceptors.request.use(
  function (config) {
    // 在发送请求之前做些什么
    // 请求接口统一添加token
    config.headers["Authorization"] = "Bearer " + getToken();
    return config;
  },
  function (error) {
    // 对请求错误做些什么
    return Promise.reject(error);
  }
);
```

```

// 添加响应拦截器
request.interceptors.response.use(
  function (response) {
    // 2xx 范围内的状态码都会触发该函数。
    // 对响应数据做点什么
    // 后端返回格式 定义:
    // {
    //   data,
    //   message,
    //   code
    // }
    const code = response.data.code;
    const msg = response.data.message;
    // 当状态码不等于200时, 请求数据有问题
    if (code === 401) {
      // 比如设置401: 登录过期
      MessageBox.confirm(
        "登录状态过期, 您可以继续留在该界面, 或重新登录",
        "系统提示",
        {
          confirmButtonText: "重新登录",
          cancelButtonText: "取消",
          type: "warning",
        }
      ).then(() => {
        //跳转至登录页面
      });
    } else if (code === 500) {
      Message({ message: msg, type: "error" });
      return Promise.reject(new Error(msg));
    } else {
      // 当状态码等于200时, 说明成功
      return response.data.data;
    }
  },
  function (error) {
    // 超出 2xx 范围的状态码都会触发该函数。
    // 对响应错误做点什么
    Message({ message: error, type: "error", duration: 5 * 1000 });
    return Promise.reject(error);
  }
);

export default request;

```

api/user.js

```

import request from "@/utils/request";
export function getUserInfo() {
  return request({
    url: "/getInfo",
    method: "get",
    params: {
      id: "1111"
    }
  });
}

```

```

    }
  });
}
export function changePwd() {
  return request({
    url: "/changePwd",
    method: "post",
    data: {
      id: "1111"
    }
  });
}
}

```

Vue页面

```

import { getUserInfo } from "@api/user";
export default {
  data() {
    return {
      name: 11,
    };
  },
  mounted() {
    getUserInfo().then((res) => {
      console.log(res);
    });
  },
};

```

跨域问题

vue.config.js

```

const { defineConfig } = require("@vue/cli-service");
module.exports = defineConfig({
  transpileDependencies: true,
  devServer: {
    port: 9091,
    open: true,
    proxy: {
      "/api": {
        target: `http://localhost:9090`, //实际请求的后端地址
        changeOrigin: true,
        pathRewrite: {
          "/api": "",
        },
      },
    },
  },
});

```