mail:caofuqiangmail@163.com

axios

一、接口文档

接口名	请求方式	请求参数	请求地址
获取所有学生	get	无参	href
根据id获取学生	get	id string	href
根据名字获取学生	post	name string	href
获取所有学生	post	无参	href

一、axios

```
axios是基于promise对ajax的一种封装
ajax mvc
axios mvvm
```

二、axiso的基本使用

```
//使用默认方式发送无参请求
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
    axios({
        url:'http://localhost:9999/student/student/getAllStudent'
    }).then(res=>{
        console.log(res);
    })
</script>
```

默认使用 get方式进行请求

```
//指定请求方式为get的无参请求
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
    axios({
        url:'http://localhost:9999/student/student/getAllStudent',
        method:'get'
    }).then(res=>{
        console.log(res);
    })
</script>
```

```
//axios发送get方式的有参请求
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
   axios({
       url:'http://localhost:9999/student/student/findStudentById?id=1'
   }).then(res=>{
       console.log(res);
   });
</script>
```

```
//axios发送get方式请求其他形式
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
   axios({
       url:'http://localhost:9999/student/student/findStudentById',
       params:{
           id:'1'
       }
   }).then(res=>{
       console.log(res);
   });
</script>
```

> axiso发送post请求

```
//使用post方式发送无参请求
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
    axios({
        url:'http://localhost:9999/student/student/pGet',
        method:'post'
    }).then(res=>{
        console.log(res);
    })
</script>
```

```
//使用axios发送带有参数的post请求
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
   axios({
       url:'http://localhost:9999/student/student/findStudentByName',
       method:'post',
       params:{
           name:'张三' //url上拼接上name参数 ? name=张三
       }
   }).then(res=>{
       console.log(res);
   });
</script>

//使用axios发送带有参数的post请求 使用data传递
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
   axios({
       url:'http://localhost:9999/student/student/findStudentByName',
       method:'post',
       data:{
           name:'张三' //后台接收为null
       }
   }).then(res=>{
       console.log(res);
   });
</script>
```

后台控制器接收到的name null axios使用post携带参数请求默认使用application/json
解决方式一： params属性进行数据的传递
解决方式二： "name=张三"
解决方式三： 服务器端给接收的参数加上@requestBody

三、axios请求方式

```
//使用axios.get方式发送无参请求
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
    axios.get('http://localhost:9999/student/student/getAllStudent').then(res=>{
        console.log(res);
    }).catch(err=>{
        console.log(err);
    })
</script>
```

```
//使用axios.get发送有参请求
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
    axios.get('http://localhost:9999/student/student/findStudentById',{params:
{id:1}}).then(res=>{
        console.log(res);
    }).catch(err=>{
        console.log(err);
    })
</script>
```

```
//使用axios.post无参请求
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
   axios.post('http://localhost:9999/student/student/pGet').then(res=>{
       console.log(res);
   }).catch(err=>{
       console.log(err);
   })
</script>
```

```
//使用axios.post发送有参请求 只修改前端代码
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
    axios.post('http://localhost:9999/student/student/findStudentByName',"name=张三
&age=10").then(res=>{
        console.log(res);
    });
</script>
//发送post请求携带参数 直接试用"name=张三&age=10"

//使用data传递数据 后台需要将axiso自动装换的json数据装换为java对象 //修改后台代码
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
    axios.post('http://localhost:9999/student/student/findStudentByName',{name:'张
三'}).then(res=>{
        console.log(res);
    });
</script>
//接收参数添加@requestBody注解
```

四、axios的并发请求

```
//axios的并发请求

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

```
<script>
   axios.all([
       axios.get('http://localhost:9999/student/student/getAllStudent'),
       axios.get('http://localhost:9999/student/student/findStudentById',{params:{id:1}})
   ]).then(res=>{//请求成功响应的是一个数组
       console.log(res[0]);
       console.log("----------");
       console.log(res[1]);
   }).catch(err=>{
       console.log(err);
   })
</script>
```

```
//使用spread方法处理响应数组结果
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
   axios.all([
       axios.get('http://localhost:9999/student/student/getAllStudent'),
       axios.get('http://localhost:9999/student/student/findStudentById',{params:{id:1}})
   ]).then(
       axios.spread((res1,res2)=>{
           console.log(res1);
           console.log(res2);
       })
   ).catch(err=>{
       console.log(err);
   })
</script>
```

五、axiso的全局配置

```
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
    axios.defaults.baseURL='http://localhost:9999/student/student';//配置全局属性
    axios.defaults.timeout=5;

    axios.get('getAllStudent').then(res=>{ //在全局配置的基础上去网络请求
        console.log(res);
    });
    axios.post('pGet').then(res=>{
        console.log(res);
    }).catch(err=>{
        console.log(err);
    })
</script>
```

六、axiso的实例

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

```
<script>
    let newVar = axios.create({
        baseURL:'http://localhost:9999/student/student',
        timeout:5000
    });//创建axios的实例

    let newVar1 = axios.create({
        baseURL:'http://localhost:9999/student',
        timeout:5000
    });//创建axios的实例

    newVar({
        url:'getAllStudent'
    }).then(res=>{
        console.log(res);
    })

    newVar1({
        url:'student/getAllStudent'
    }).then(res=>{
        console.log(res);
    }).catch(err=>{
        console.log(err);
    })

</script>
```

七、axios的拦截器

axios给我们提供了两大类拦截器 一种是请求方向的拦截（成功请求，失败的）
另一种是响应方向的（成功的 失败的）

拦截器的作用 用于我们在网络请求的时候在发起请求或者响应时对操作进行响应的处理
发起请求时可以添加网页加载的动画 强制登录
响应的时候可以进行相应的数据处理

```
//请求方向拦截器
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
    axios.interceptors.request.use(config=>{
        console.log("进入请求拦截器");
        console.log(config);
        return config;//放行请求
    },err=>{
        console.log("请求方向失败");
        console.log(err);
    });
```

```
    axios.get("http://localhost:9999/student/student/getAllStudent").then(res=>{
        console.log(res);
    })

</script>
```

```
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
    axios.interceptors.response.use(config=>{
        console.log("进入响应拦截器");

        return config.data;
    },err=>{
        console.log("响应方向失败");
        console.log(err);
    });

    axios.get("http://localhost:9999/student/student/getAllStudent").then(res=>{
        console.log(res);
    })

</script>
```

八、axios在vue中的模块封装

```
//封装位置 文件绝对位置./network/request/request.js
import axios from 'axios'
export function request(config,success,fail) {
    axios({
        url:config
    }).then(res=>{
        success(res);
    }).catch(err=>{
        fail(err);
    })

}
//调用者位置
import {request} from './network/request/request'
request('http://localhost:9999/student/student/getAllStudent',res=>{
  console.log(res);
},err=>{
  console.log(err);
})
```

```
 //封装位置
export function request(config){

axios.defaults.baseURL="http://localhost:9999/student/student";
```

```
    axios(config.url
    ).then(
        res=>{
            config.success(res);
        }
    ).catch(err=>{
        config.fail(err);
    })
}
//调用者位置
import {request} from './network/request/request'
request({
  url:'getAllStudent',
  success:res=>{
    console.log(res);
  },
  fail:err=>{
    console.log(err);
  }
})
```

```
//封装的位置
export function request(config){
return new Promise((resolve, reject) => {
        let newVar = axios.create({
            baseURL:"http://localhost:9999/student/student",
            timeout:5000
        });
        newVar(config).then(res=>{
            resolve(res);
        }).catch(err=>{
            reject(err);
        })
    })
}

//调用位置
import {request} from './network/request/request'
request({
  url:'getAllStudent'
}).then(res=>{
  console.log(res);
}).catch(err=>{
  console.log(err);
})
```

```
//封装位置
export function request(config){
let newVar = axios.create({
        baseURL:"http://localhost:9999/student/student",

        timeout:5000
```

```
    });
    return newVar(config);
//调用者
    import {request} from './network/request/request'
    request({
        url:'getAllStudent'
            }).then(res=>{
            console.log(res);
        })
```