

# axios的四种请求方式

## 1. get

```
axios.get('/data.json',{
  params: {
    name: '汤姆',
    username: 'tom',
    phone: '17398891277',
    currentPage: 1,
    limit: 10
  }
}).then((res)=>{
  console.log(res)
})
```

后台接收方式:

```
/**
 * 分页查询User
 * @param userVo
 * @return
 */
@GetMapping("user")
@ApiOperation("分页获取User")
public Result getUser(UserVo userVo) {
    Page<User> userPage = userService.getUserByPage(userVo);
    if (userPage != null) {
        return Result.ok().data("userPage", userPage).message("用户查询成功");
    }
    return Result.error().message("用户查询失败");
}
```

## 2. delete

```
axios.delete("user", {
  params: {
    uid: row.uid,
  }
}).then((response) => {

}).catch(() => {
  this.$router.push("error");
})
```

后台接收方式:

```

@DeleteMapping("user")
@ApiOperation("删除用户")
public Result deleteUser(UserVo userVo) {
    User user = userService.getOne(new QueryWrapper<User>().eq("uid",
userVo.getId()));
    if (user != null) {
        userService.remove(new QueryWrapper<User>().eq("uid",
user.getId()));
        return Result.ok().message("删除成功!");
    }
    return Result.error().message("删除失败!");
}

```

### 3. put

```

axios.put("/user", {
    uid: row.uid,
    password: 123
}).then((response) => {

}).catch(() => {
    this.$router.push("/error");
})

```

后台接收方式:

```

@PutMapping("user")
@ApiOperation("更新用户")
public Result updateUser(@RequestBody User user) {
    User one = userService.getOne(new QueryWrapper<User>().eq("uid",
user.getId()));
    one.setPassword("123");
    if (one != null) {
        userService.updateById(one);
        return Result.ok().message("更新成功!");
    }
    return Result.error().message("更新失败");
}

```

### 4. post

```

axios.post("user", {
    name: this.editForm.name,
    username: this.editForm.username,
    password: this.editForm.password,
    phone: this.editForm.phone,
    address: this.editForm.address,
    sex: this.editForm.sex == "男"? '1': '0'
}).then((response) => {
    if (response.data.code != 200) {
        this.$message.error("添加失败! " + response.data.message);
        return;
    }
    this.$message.success("用户添加成功!");
})

```

```
}).catch(() => {  
    this.$router.push("error");  
})
```

后台接收方式:

```
@PostMapping("user")  
@ApiOperation("保存用户")  
public Result saveUser(@RequestBody User user) {  
    User one = userService.getOne(new QueryWrapper<User>  
().eq("username", user.getUsername()));  
    if (one != null) {  
        return Result.error().message("用户已存在!");  
    }  
    if (userService.save(user)) {  
        return Result.ok().message("保存成功!");  
    }  
    return Result.error().message("保存失败!");  
}
```