文档地址：<https://www.kancloud.cn/yunye/axios/234845>

补充说明：

属性headers：可以理解为ajax中的contentType

属性params：必须是一个无格式对象（即“简单对象”通过通过 "{}" 或者 "new Object" 创建的对象）或者[URLSearchParams](https://blog.csdn.net/TNTnine/article/details/82713484) 对象

属性paramsSerializer是一个负责 `params` 序列化的函数，说白了就是处理提交的参数，可以使用node的qs模块。

属性transformRequest

// `transformRequest` 允许在向服务器发送前，修改请求数据

// 只能用在 'PUT', 'POST' 和 'PATCH' 这几个请求方法

// 后面数组中的函数必须返回一个字符串，或 ArrayBuffer，或 Stream

属性data：

// `data` 是作为请求主体被发送的数据

// 只适用于这些请求方法 'PUT', 'POST', 和 'PATCH'

// 在没有设置 `transformRequest` 时，必须是以下类型之一：

// - string, plain object, ArrayBuffer, ArrayBufferView, URLSearchParams

// - 浏览器专属：[FormData](https://mp.csdn.net/mdeditor/82714332), File, Blob

// - Node 专属： Stream

data: {

firstName: 'Fred'

}

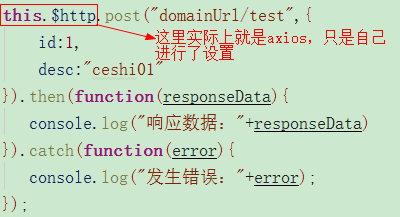
关于qs模块的介绍：

<https://blog.csdn.net/suwu150/article/details/78333452>

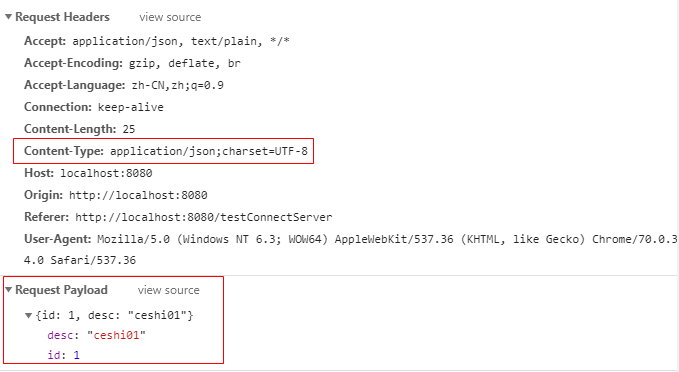
# Axios的post方法（post方式也是一样的）：

axios.post()方法不设置headers会默认Content-type为application/json；charset=UTF-8。

案例1：

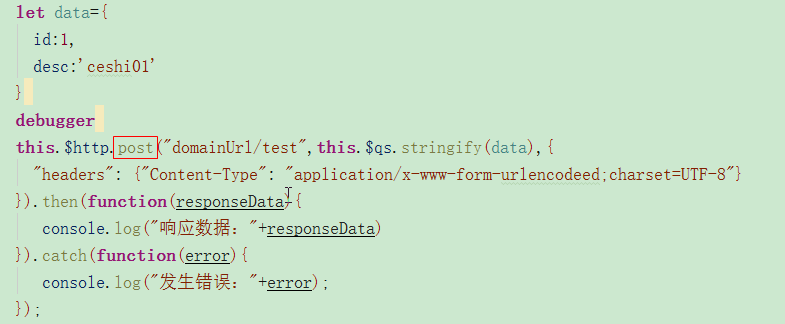


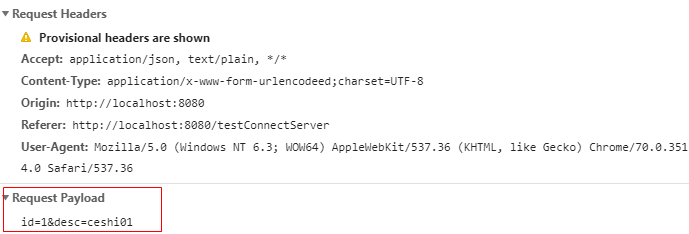
请求体：



**注意：此种方式在SpringMVC中后台不能使用@RequestParam注解来获取参数；同时定义参数名和请求参数一样的名字也不能获取到数据（就是在此处把方法中参数名定义为id和desc也是不能获取到前端传递的数据的）。要接收此数据可以使用@RequestBody注解，这这里可写成这种形式--@RequestBody String str，这里的str的内容就是**

案例2：



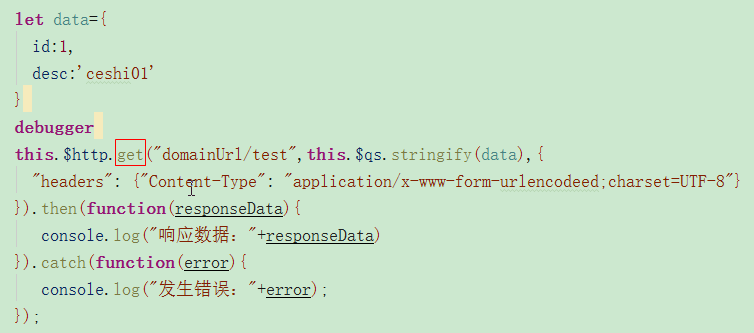


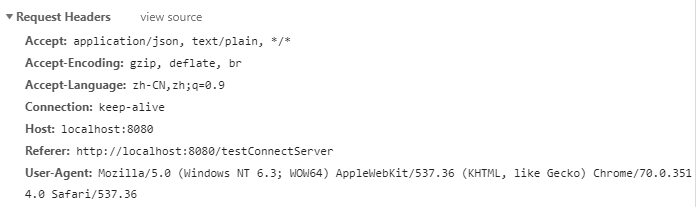
**注意，post方式传递的数据，在springmvc中不能使用@RequestParam注解来获取，http请求会报错误代码400，可以使用参数来接收（参数名要和传递的参数名一样）**

# Axios的get方法（get方式也是一样的）

Axios的get方法（**注意：get请求是没有Content-Type这个字段的）**

案例1





得到结果，不能传递参数。如果想要传递参数可改为如下两种方式的写法：

1. 在url后加参数：这里可写为
2. 定义params属性的值：