6、跨域：

同源策略：浏览器的一种安全策略，当前网页的url和ajax请求目标资源的url两者之间必须协议，域名，端口保持一致。满足同源策略时，发送请求的url可以省略协议/域名/端口

ajax默认遵循同源策略，无法跨域。

跨域就是违背同源策略，即协议，域名，端口任意一个不同就是跨域。（http和https是不同的协议）

跨域的解决方案：

分为前端解决和后端解决

（1）JSONP

非官方的跨域解决方案，前端解决，只支持GET请求，实现原理是借助script标签实现跨域（img，a，link，script标签本身就具备跨域特性）

原生实现：

服务端：

let data=….

data=JSON.stringify(data)

response.send(`func(${data})`)

客户端：

func(data){//处理}

需要发送请求的内部，如点击事件；

let s=document.createElement(‘script’)

s.src=’url’

document.body.append(s)

通过script标签发送请求，服务端返回一个函数的调用，从而执行func()

jQuery实现；

服务端：

let data=….

data=JSON.stringify(data)

let cb=request.query.callback

response.send(`cb(${data})`)

客户端：

$.getJSON(‘url?callback=?’,function(data){})

callback的值?其实是该回调函数，服务端获取该参数后，将其作为函数调用返回，然后在这个回调函数中对数据进行处理。

（2）CORS

中文为跨域资源共享，后端解决，官方的跨域解决方案，客户端不需要额外的操作，只需要在服务端处理就能完成跨域。支持GET，POST等请求。

服务端只需设置相应头：

response.setHeader('Access-Control-Allow-Origin','\*')

第一个参数是固定写法，第二个参数是允许发送请求的url，\*表示所有url都允许

Vue跨域：

在config文件夹的index.js中的proxyTable中设置