```

      __       _______..______      .______     ______ 
     |  |     /       ||   _  \     |   _  \   /      |
     |  |    |   (----`|  |_)  |    |  |_)  | |  ,----'
.--.  |  |     \   \    |      /     |   ___/  |  |     
|  `--'  | .----)   |   |  |\  \----.|  |      |  `----.
 \______/  |_______/    | _| `._____|| _|       \______|

```

-- js逆向之远程调用(rpc)免去抠代码补环境

> 黑脸怪
>

目录结构

> main.go （服务器的主代码）
> resouces/JsEnv_De.js （客户端注入js环境）
> config.yaml （可选配置文件）

基本介绍

运行服务器程序和js脚本 即可让它们通信，实现调用接口执行js获取想要的值(加解密)

实现

原理：在网站的控制台新建一个WebScoket客户端链接到服务器通信，调用服务器的接口 服务器会发送信息给客户端 客户端接收到要执行的方法执行完js代码后把获得想要的内容发回给服务器 服务器接收到后再显示出来

> 说明：本方法可以https证书且支持wss

食用方法

打开编译好的文件，开启服务(releases下载)

如图所示

如需更改部分配置，请查看 "其他说明"

api 简介

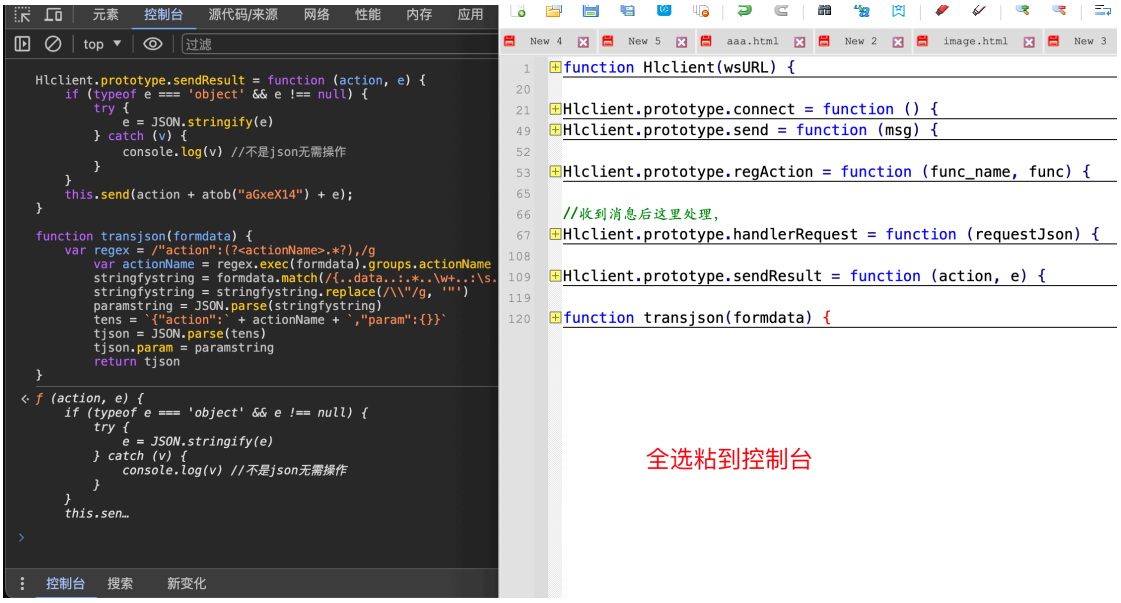
- `/list` :查看当前连接的ws服务 (get)
- `/ws` :浏览器注入ws连接的接口 (ws | wss)
- `/wst` :ws测试使用-发啥回啥 (ws | wss)
- `/go` :获取数据的接口 (get | post)
- `/execjs` :传递jscode给浏览器执行 (get | post)
- `/page/cookie` :直接获取当前页面的cookie (get)
- `/page/html` :获取当前页面的html (get)

说明：接口用?group分组 如 "ws://127.0.0.1:12080/ws?group={}"
以及可选参数 clientId
clientId说明：以group分组后，如果有注册相同group的 可以传入这个id来区分客户端，如果不传 服务程序会自动生成一个。当访问调用接口时，服务程序随机发送请求到相同group的客户端里。

//注入例子 group可以随便起名(必填)
http://127.0.0.1:12080/go?group={}&action={}&param={} //这是调用的接口
group填写上面注入时候的，action是注册的方法名,param是可选的参数 param可以传string类型或者object类型(会尝试用JSON.parse)

注入JS，构建通信环境（/resouces/JsEnv_De.js）

打开JsEnv 复制粘贴到网站控制台(注意：可以在浏览器开启的时候就先注入环境，不要在调试断点时候注入)



连接通信

```
// 注入环境后连接通信
var demo = new Hlclient("ws://127.0.0.1:12080/ws?group=zzz");
// 可选
//var demo = new Hlclient("ws://127.0.0.1:12080/ws?
group=zzz&clientId=hliang/"+new Date().getTime())
```

I 远程调用0:

接口传js代码让浏览器执行

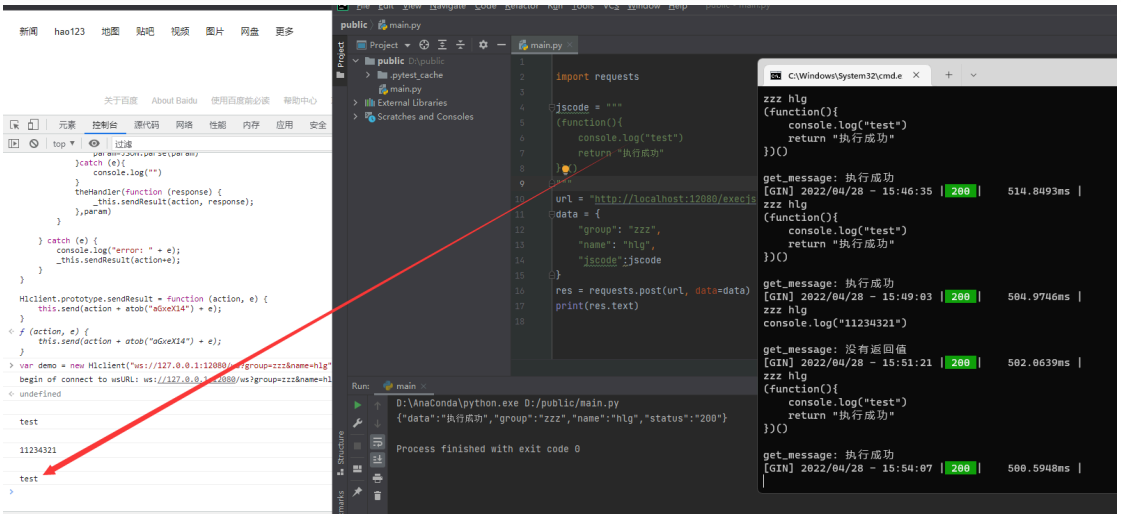
浏览器已经连接上通信后 调用execjs接口就行

```
import requests

js_code = """
(function(){
    console.log("test")
    return "执行成功"
```

```
})()
"""

url = "http://localhost:12080/execjs"
data = {
    "group": "zzz",
    "code": js_code
}
res = requests.post(url, data=data)
print(res.text)
```



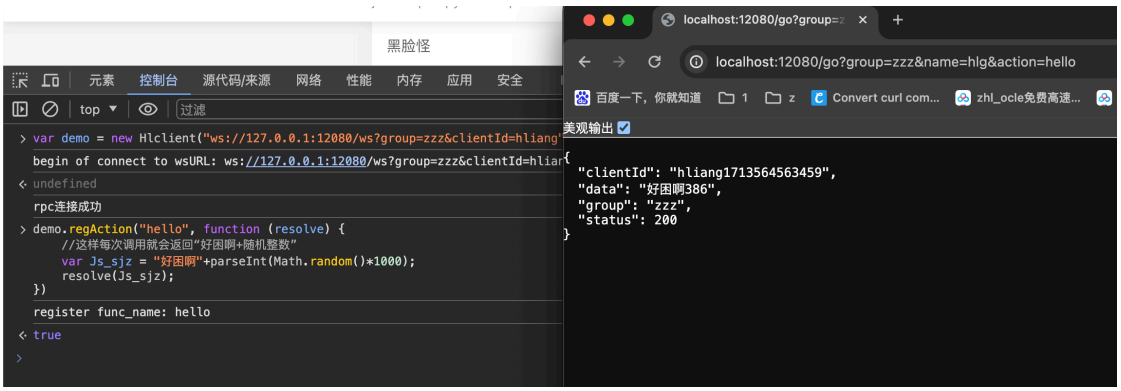
II 远程调用1： 浏览器预先注册js方法 传递函数名调用

远程调用1：无参获取值

```
// 注册一个方法 第一个参数hello为方法名，
// 第二个参数为函数，resolve里面的值是想要的值(发送到服务器的)
demo.regAction("hello", function (resolve) {
    //这样每次调用就会返回“好困啊+随机整数”
    var Js_sjz = "好困啊"+parseInt(Math.random()*1000);
    resolve(Js_sjz);
})
```

访问接口，获得js端的返回值

http://127.0.0.1:12080/go?group=zzz&action=hello

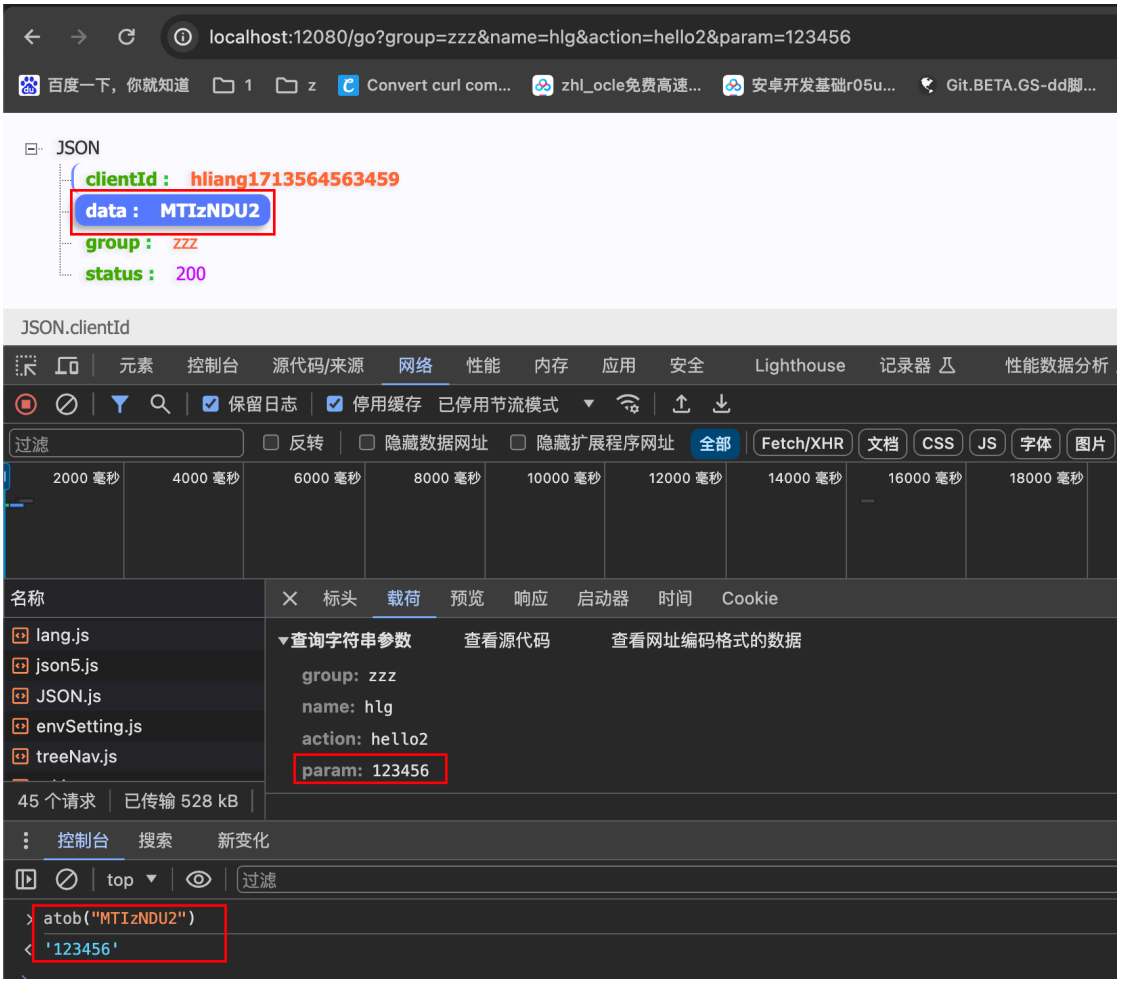


远程调用2：带参获取值

```
//写一个传入字符串，返回base64值的接口(调用内置函数btoa)
demo.regAction("hello2", function (resolve,param) {
    //这样添加了一个param参数，http接口带上它，这里就能获得
    var base666 = btoa(param)
    resolve(base666);
})
```

访问接口，获得js端的返回值

http://127.0.0.1:12080/go?group=zzz&action=hello2¶m=123456



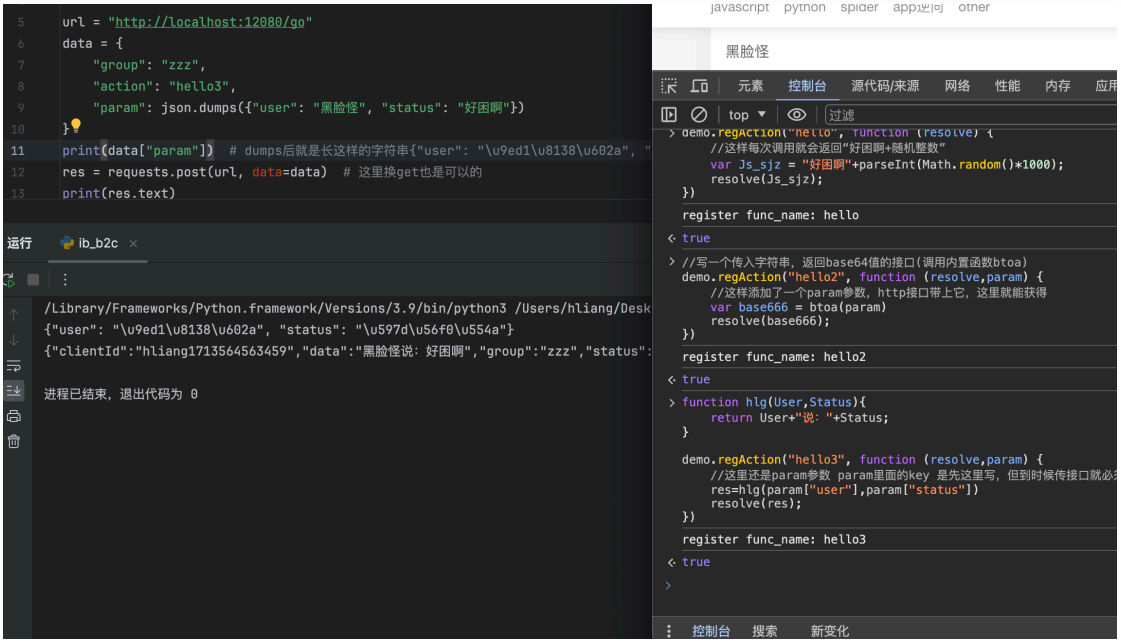
远程调用3：带多个参获 并且使用post方式 取值

```
//假设有一个函数 需要传递两个参数
function hlg(User,Status){
    return User+"说: "+Status;
}

demo.regAction("hello3", function (resolve,param) {
    //这里还是param参数 param里面的key 是先这里写，但到时候传接口就必须对应的上
    res=hlg(param["user"],param["status"])
    resolve(res);
})
```

访问接口，获得js端的返回值

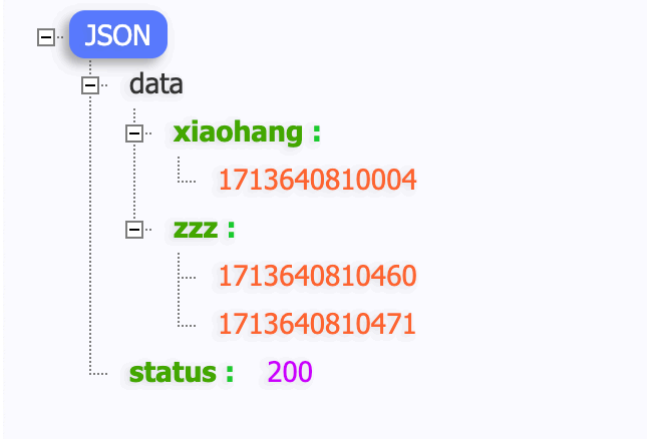
```python
url = "http://127.0.0.1:12080/go"
data = {
    "group": "zzz",
    "action": "hello3",
    "param": json.dumps({"user":"黑脸怪","status":"好困啊"})
}
print(data["param"]) #dumps后就是长这样的字符串{"user": "\u9ed1\u8138\u602a", "status": "\u597d\u56f0\u554a"}
res=requests.post(url, data=data) #这里换get也是可以的
print(res.text)
```



远程调用4：获取页面基础信息

```python
resp = requests.get("http://127.0.0.1:12080/page/html?group=zzz")      # 直接获取当前页面的html
resp = requests.get("http://127.0.0.1:12080/page/cookie?group=zzz")    # 直接获取当前页面的cookie
```

list接口可查看当前注入的客户端信息

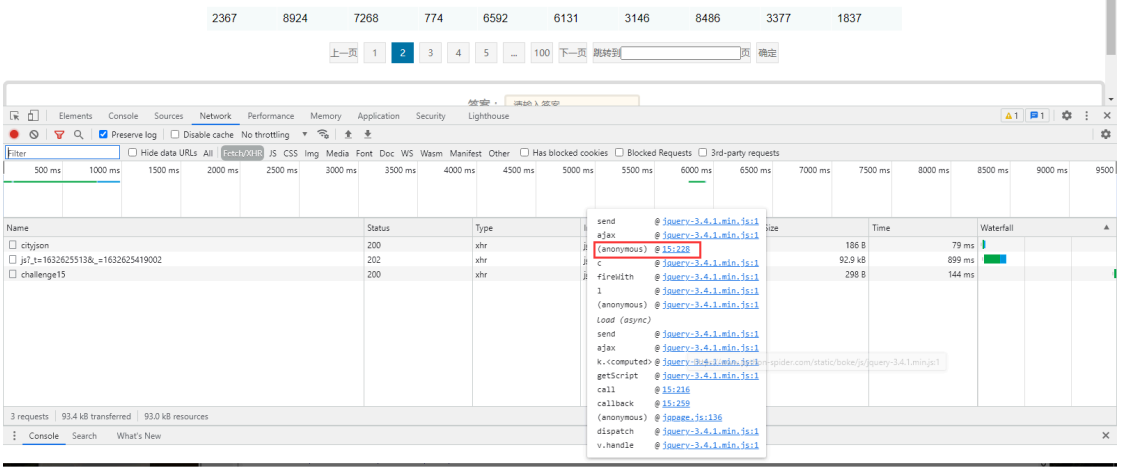


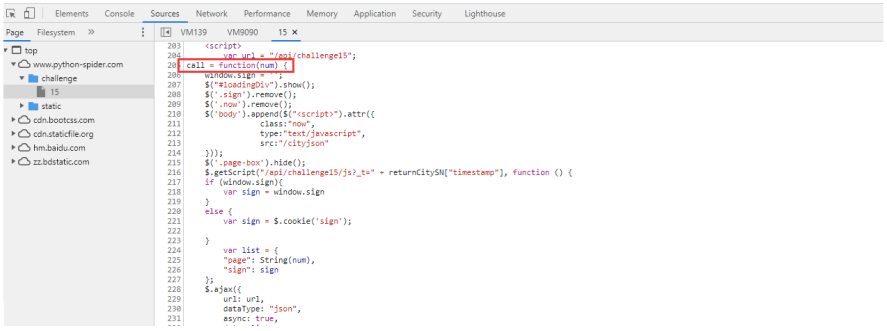
食用案例-爬虫练手-xx网第15题

本题解是把它ajax获取数据那一个函数都复制下来，然后控制台调用这样子~

1.f12查看请求，跟进去 找到ajax那块，可以看到call函数就是主要的ajax发包 输入页数就可以，那我们复制这个函数里面的代码备用

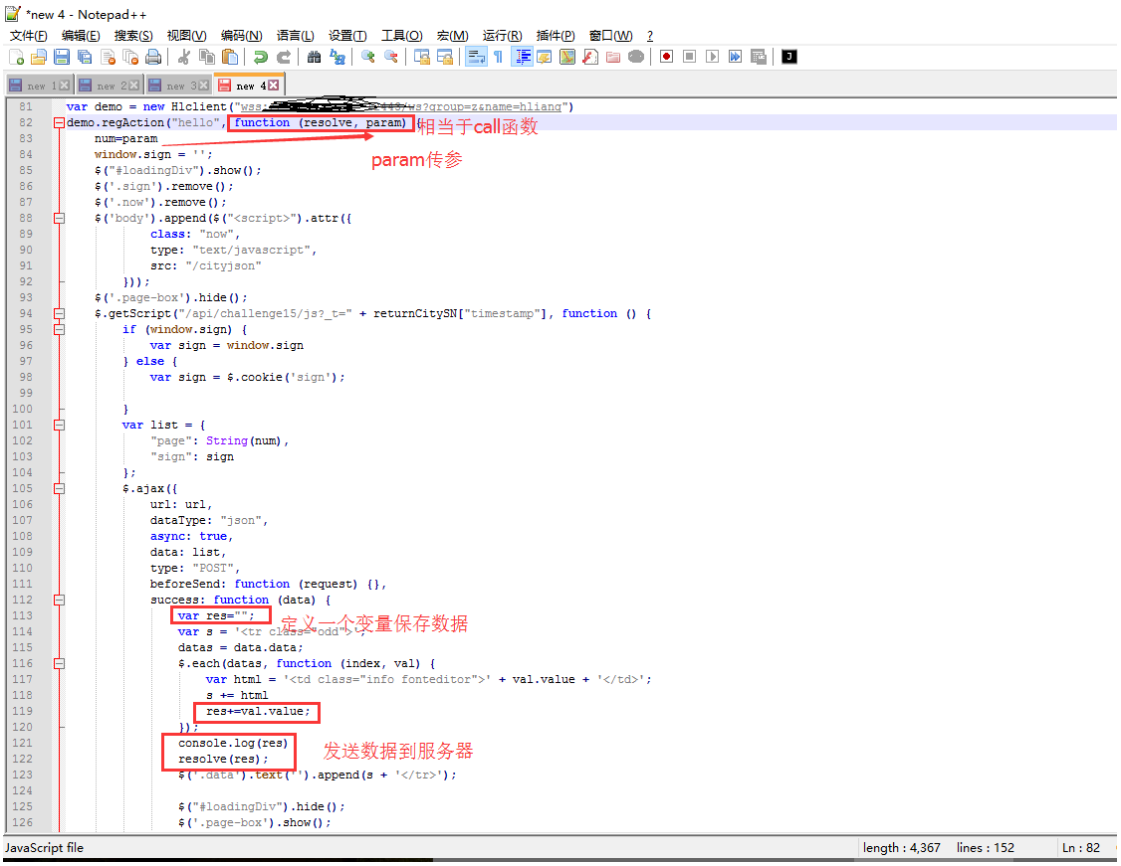
目标：采集100页的全部数字，并计算所有数据加和。
本题为JS课程中 jsRPC例题，如果感兴趣，请另外尝试用AST/解JS方式解题，提升会更大~





2.先在控制台粘贴我的js环境，再注入一个rpc链接 注册一个call方法，名字自定义 第二个参数粘贴上面call的代码，小小修改一下

先定义num=param 这样就传参进来了，再定义一个变量来保存获取到的数据，resolve(变量) 就是发送。完了就注入好了，可以把f12关掉了



1 3.调用接口就完事了，param就是传参页数

1 控制台可以关，但是注入的网页不要关哦

其他说明

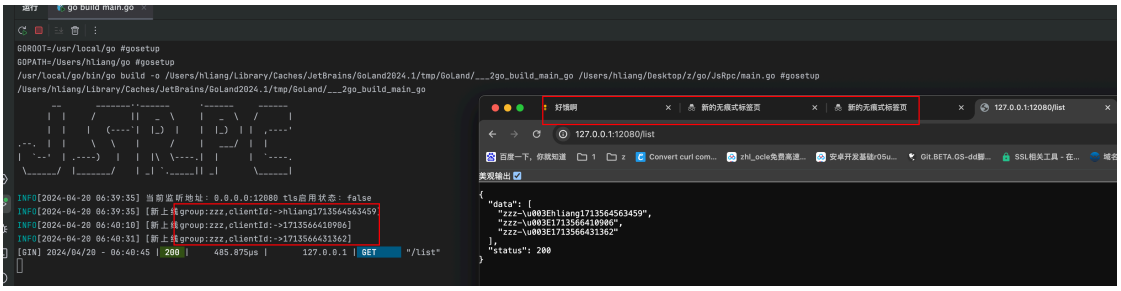
如果需要更改rpc服务的一些配置 比如端口号啊，https/wss服务，打印日志等
可以在执行文件的同路径 下载 config.yaml 文件配置
或使用-c参数指定配置文件路径

```
./JsRpc.exe -c config1.yaml
```

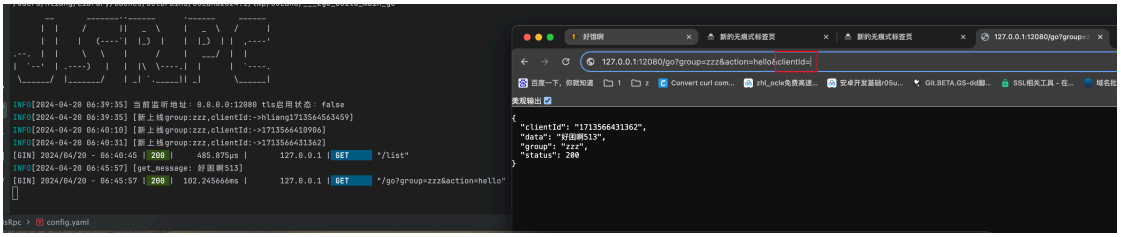
group说明

一般配置group名字不一样分开调用就行

特别情况，可以一样的group名，比如3个客户端(标签演示)执行加密，程序会随机一个客户端来执行并返回。



请确保action也都是一样



多个group除了随机 还可以根据clientId指定客户端执行

http://127.0.0.1:12080/go?group=zzz&action=hello

http://127.0.0.1:12080/go?group=zzz&action=hello&clientId=hliang171356456 3459

可选

BUG修复

1.修复ResultSet函数，在并发处理环境下存在数据丢失，响应延迟等问题。

其他案例

```
1. JsRpc实战-猿人学-反混淆刷题平台第20题（wasm）
    https://mp.weixin.qq.com/s/DemSz2NRkYt9YL5fSUiMDQ
2. 网洛者-反反爬练习平台第七题（JSVMPZL - 初体验）
    https://mp.weixin.qq.com/s/nvQNV33QkzFQtFscDqnXWw
```

常见问题

```
1. websocket连接失败
  内容安全策略 (Content Security Policy)
  Refused to connect to 'xx.xx' because it violates the following Content Security Policy directive: "connect-src 'self'
  这个网站不让连接websocket，可以用油猴注入使用，或者更改网页响应头
2. 异步操作获取值
  [参考](https://github.com/jxhczhl/JsRpc/issues/12)
```

TODO

- [ ] 异步方法调用

```
demo.regAction('token', async (resolve) => {
    let token = await grecaptcha.execute(0, { action: '' }).then(function (token) {
        return token
    });
    resolve(token);
})
```

- [ ] ssl Docker Deploy
- [ ] K8s Deploy