Websocket使用指南

1. 后台推送

异步操作的最终结果或其他需要从后台推送给前台的消息通过调用此接口向前台推送。

url：<http://api.devops.inspur.com/console/websocket/v1/messages> （生产环境要改成生产环境的域名，并把调用方式改成https）

type：post

body：

{

"userName": "jindengke",

"handlerName": "operateEcsHandler",

"instanceId": "ecs-0001",

"instanceStatus": "active",

"operateType": "single-reboot",

"messageType": “success”,

"message": "云服务器ecs-0001开机成功"

}

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 必须 | 说明 |
| userName | String | 是 | 用户名，从token中获取，取的是token中的preferred\_username字段 |
| handlerName | String | 是 | operate+产品缩写+Handler |
| instanceId | String | 否 | 实例id |
| instanceStatus | String | 否 | 实例的当前状态 |
| operateType | String | 否 | 操作类型，各产品根据自己的业务去定义，前后端约定好，保证一致 |
| messageType | String | 否 | 消息提示类型，'success' | 'info' | 'warning' | 'error' | 'blank'，message不为空时该字段才起作用。默认为blank。 |
| message | String | 否 | 提示消息，为空时前台不进行提示。 |
| tags | Array<String> | 否 | 扩展字段 |

此接口会验证token，建议调用这个方法来获取最新的token，避免token失效导致的消息丢失。

1. 前台接收

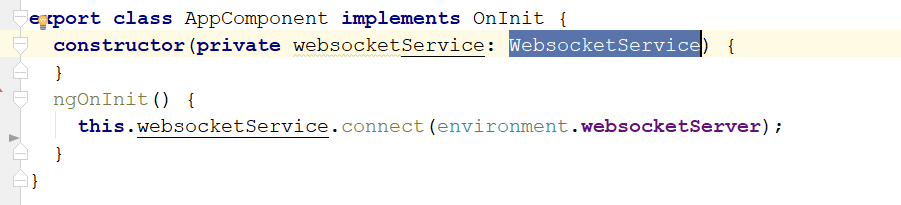
前台接收到消息以后，跟据接收的消息体做具体的处理。

前台监听地址ws://api.devops.inspur.com/console/websocket/v1/servers（生产环境要改成生产环境的域名，并把调用方式改成wss）

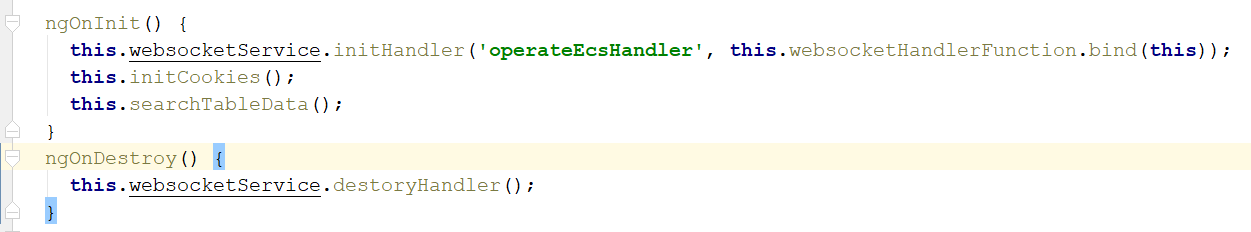
1．package.json引入依赖

**"@global/shared"**: **"^0.0.11-SNAPSHOT.11"**,

2．app.component.ts的ngOnInit中调用WebsocketService的connect方法建立websocket连接（传参为websocket服务器地址ws://tenantconservice.devops.inspur.com/console/webServer）



3. 每个页面组件的ts文件的ngOnInit中调用WebsocketService的initHandler。该方法用于向WebsocketService注册currentHandlerName与回调方法。ngOnDestroy中调用destoryHandler，该方法用于页面跳转时销毁当前页面注册的handler。



4. 注册完currentHandlerName和回调方法以后，如果接受到消息并且消息体内的handlerName字段与当前页面注册的currentHandlerName一致，则会执行回调方法。如果handlerName不等于当前页面注册的currentHandlerName，则不会执行回调方法。回调方法用于处理前端接受到对应消息后的逻辑。

1. 简单的示例
2. 云服务器重启成功，后台发送消息体

{

"userName": "jindengke",

"handlerName": "operateEcsHandler",

"instanceId": "024cc936-f594-43d5-858d-72aac32e02c3",

"instanceStatus": "active",

"operateType": "single-reboot",

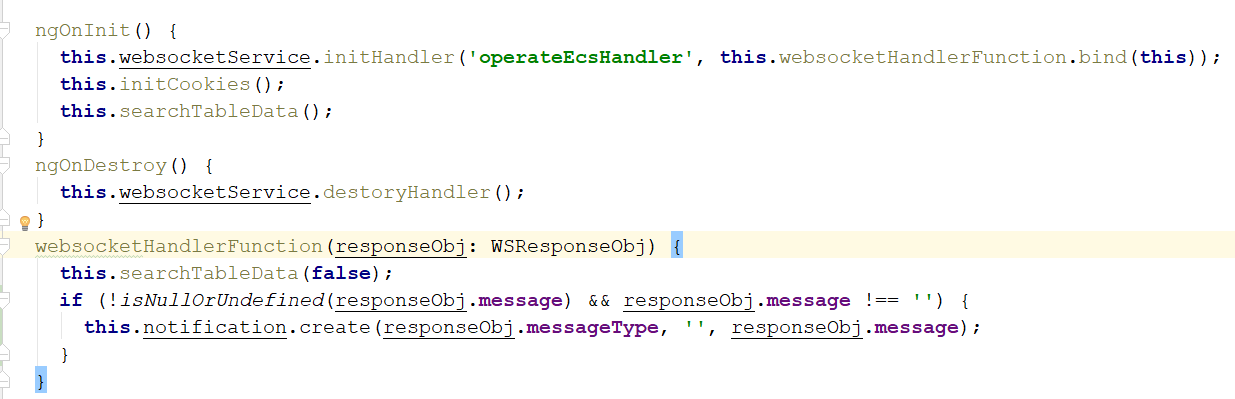
"messageType": "",

"message": "",

"tags": [],

}

前台处理：



假如要监听多个handlerName的消息的话。InitHandler的第一个参数改为字符串数组。

