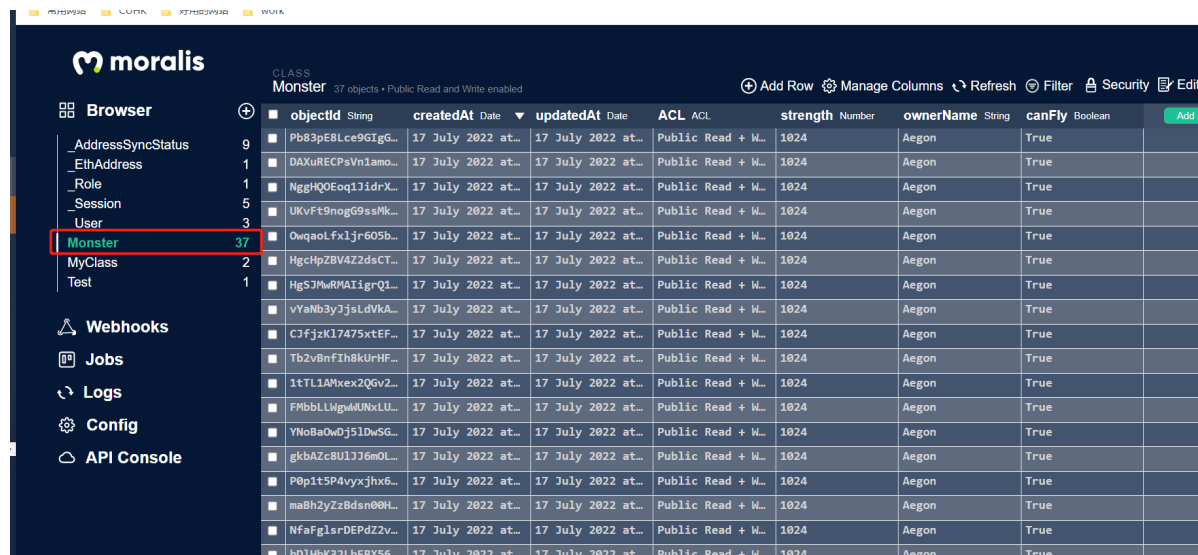


监控特定事件并回调

一、监控Moralis中的Dapp里面保存的对象

具体实现：



objectId	String	createdAt	Date	updatedAt	Date	ACL	ACL	strength	Number	ownerName	String	canFly	Boolean
PB83pE8Lce9GI6...		17 July 2022 at...		17 July 2022 at...		Public Read + W...		1024		Aegon		True	
DAXURECPaVn1amo...		17 July 2022 at...		17 July 2022 at...		Public Read + W...		1024		Aegon		True	
NggHQ0Eeq1JidrX...		17 July 2022 at...		17 July 2022 at...		Public Read + W...		1024		Aegon		True	
UKvFt9nogG9ssMk...		17 July 2022 at...		17 July 2022 at...		Public Read + W...		1024		Aegon		True	
OwqaoLfx1jr605b...		17 July 2022 at...		17 July 2022 at...		Public Read + W...		1024		Aegon		True	
HgChpZBV4Z2dsCT...		17 July 2022 at...		17 July 2022 at...		Public Read + W...		1024		Aegon		True	
Hg5JMwRMAIigrQ1...		17 July 2022 at...		17 July 2022 at...		Public Read + W...		1024		Aegon		True	
vYaNb3y1jsldVKA...		17 July 2022 at...		17 July 2022 at...		Public Read + W...		1024		Aegon		True	
C3fjzK17475xtEF...		17 July 2022 at...		17 July 2022 at...		Public Read + W...		1024		Aegon		True	
Tb2vBnFi8kUrHF...		17 July 2022 at...		17 July 2022 at...		Public Read + W...		1024		Aegon		True	
1tTL1AMxex2QGV2...		17 July 2022 at...		17 July 2022 at...		Public Read + W...		1024		Aegon		True	
FHbbLLHgwhUNxLU...		17 July 2022 at...		17 July 2022 at...		Public Read + W...		1024		Aegon		True	
YNoBaOwDj51DwSG...		17 July 2022 at...		17 July 2022 at...		Public Read + W...		1024		Aegon		True	
gkbAZc8U1J76mOL...		17 July 2022 at...		17 July 2022 at...		Public Read + W...		1024		Aegon		True	
P0p1t5P4vyxjhx6...		17 July 2022 at...		17 July 2022 at...		Public Read + W...		1024		Aegon		True	
maBh2yZzBdsn00H...		17 July 2022 at...		17 July 2022 at...		Public Read + W...		1024		Aegon		True	
NfAfglSrDEPdZ2v...		17 July 2022 at...		17 July 2022 at...		Public Read + W...		1024		Aegon		True	
bd1HbK32LbFBX56...		17 July 2022 at...		17 July 2022 at...		Public Read + W...		1024		Aegon		True	

moralis里存在一系列object,这里以Monster对象为例子，创建一个trigger,(aftersave事件)等Monster对象存储以后回调一个http地址

Save一个Monster对象的逻辑：

```
import {init} from "./index"

/* import moralis */
const Moralis = require("moralis/node");

/* Moralis init code */

const SaveData = async () => {

  init();

  const Monster = Moralis.Object.extend("Monster");
  const monster = new Monster();

  monster.set("strength", 1024);
  monster.set("ownerName", "Aegon");
  monster.set("canFly", true);

  await monster.save();

  console.log("The object has been saved")
};

SaveData();
```

定义一个云函数：

```

Parse.Cloud.afterSave("Monster", async (request) => {
  //const confirmed = request.object.get("confirmed")
  const logger = Moralis.Cloud.getLogger();
  logger.info("begin")
  let data = '{"msg_type":"text","content":{"text":"request example"}}'
  await Moralis.Cloud.httpRequest({
    method:"POST",
    url:"https://function-2-skfpnmzm6q-uc.a.run.app",
    headers: {
      'Content-Type':'application/json',
    }
  }).then(function(httpResponse) {
    logger.info("The response is loading ");
  }, function(httpResponse) {
    logger.error('Request failed with response code ' +
    httpResponse.status);
  });

  return "The monster has been sent";
});

```

这个云函数的执行逻辑是等到Monster对象存储以后，向 (<https://function-2-skfpnmzm6q-uc.a.run.app>) 网站发一个post请求,并且打印日志 (begin),回调成功或者失败发送相应的日志

<https://function-2-skfpnmzm6q-uc.a.run.app>是我部署在谷歌云上的一个webhook地址

具体函数方法如下:

```

import functions_framework

@functions_framework.http
def hello_http(request):
    """HTTP Cloud Function.
    Args:
        request (flask.Request): The request object.
        <https://flask.palletsprojects.com/en/1.1.x/api/#incoming-request-data>
    Returns:
        The response text, or any set of values that can be turned into a
        Response object using `make_response`
        <https://flask.palletsprojects.com/en/1.1.x/api/#flask.make_response>.
    """
    request_json = request.get_json(silent=True)
    request_args = request.args

    if request_json and 'name' in request_json:
        name = request_json['name']
    elif request_args and 'name' in request_args:
        name = request_args['name']
    else:
        name = 'world'
    return 'Hello {}'.format(name)

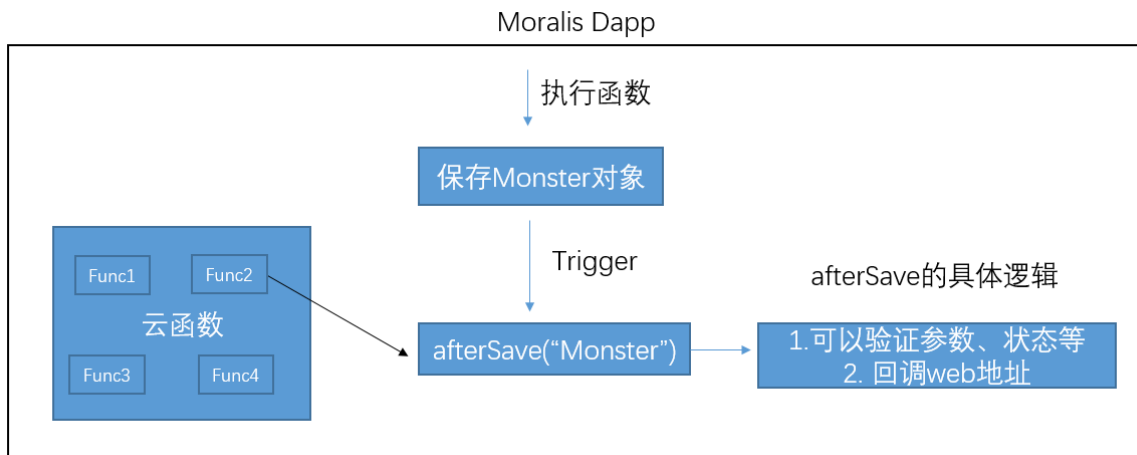
```

执行一个Post请求以后

```
curl -X POST https://function-2-skfpnmzm6q-uc.a.run.app -H "Content-Type:application/json" -d '{}'
```

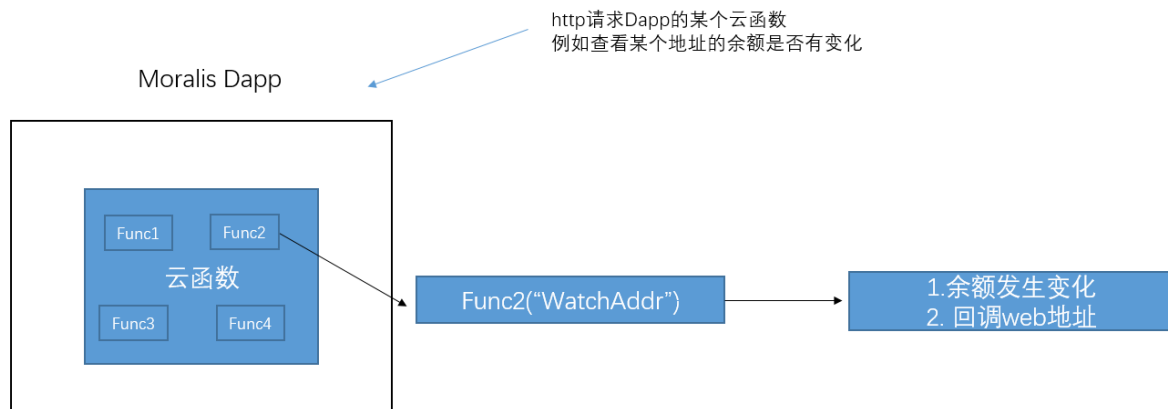
返回 "Hello word"

流程图：



- 最终效果：1. 执行完保存Monster对象以后
2. 触发云函数的afterSave("Monster")的逻辑
 3. 输出相应日志，向回调网址发送post请求

二、监控链上的对象，非Moralis中数据库保存的对象



流程“

1. 用http请求调用相应的Dapp的相应云函数，<https://docs.moralis.io/moralis-dapp/cloud-code/cloud-functions#call-via-rest-api>
2. 云函数处理传入的参数，执行相应的逻辑，如果满足一些条件（交易被确认，余额发生变化...），就给回调地址发Post请求
3. 回调地址收到相应的参数。