

# Flutter 异步编程二 (isolate、compute)

面朝对象\_春暖花开

## 一、书接上文：

- 上篇讲到我们如何利用 Future、async、await、completer 实现 flutter 异步编程，它的原理其实就是线程上依据中断信号和中断指针来实现的，虽然实现了异步，不过这不是真正意义上的多线程编程，反而更像是并发的概念。
- isolate 实现了真正意义上的多线程编程，每一个 isolate 对应一个自己的线程，并且拥有自己的内存空间。
- isolate 对比线程来说，拥有了自己的内存空间，所以更像是 iOS 进程的概念，而且 isolate 之前的通讯和 iOS 进程相同，都是使用 port 传输数据。

## 二、isolate 的用法

代码说明：

```
void main() {  
    print(Isolate.current.debugName);  
    loadDataOprition('https://jsonplaceholder.typicode.com/  
posts');  
}  
loadDataOprition(String urlString) async{
```

```

//创建一个ReceivePort
ReceivePort receivePort = ReceivePort();

//创建Isolate 绑定一个方法和向外通讯的port，创建完之后会立马执行
requestData方法，而且是在新的Isolate中，（可以
print(Isolate.current.debugName);来看当前的Isolate)

await Isolate.spawn(requestData, receivePort.sendPort);

//接收消息
await for(var msg in receivePort){
    if(msg is SendPort){
        //这里表示收到requestData来索要请求用的url了
        SendPort sendPort = msg;
        sendPort.send(urlString);
    }else{
        //这里表示收到requestData已经请求完数据并传回来的数
        据。

        print(msg);
        receivePort.close();
    }
}

static requestData(SendPort sendport) async{

    ReceivePort receivePort = ReceivePort();

    //这里是去要请求用的urlstr
    sendport.send(receivePort.sendPort);

    //等待urlstr返回
    String urlString = await receivePort.first;

```

```

        //urlstr拿到以后receivePort就没用了
receivePort.close();

//根据urlstr发起请求
var client = new HttpClient();
var request = await
client.getUrl(Uri.parse(urlString));
    var response = await request.close();
    var responseBody = await
response.transform(utf8.decoder).join();
    var dataList = jsonDecode(responseBody);

        //拿到请求回来的数据，发出去
sendport.send(dataList);
}

```

### 三、compute的使用：

体验了 isolate，脑袋嗡嗡地吧！看一下 compute 的使用，效果相同，但更容易理解和书写。

代码说明：

```

void main() {
    print(Isolate.current.debugName);
    loadDataWithCompute('https://
jsonplaceholder.typicode.com/posts');
}

loadDataWithCompute(String urlString) async{

    var dataList = await
compute(requestDataWithCompute,urlString);

```

```

        print('zzzzzzzz');
        print(dataList);
    }
    static dynamic requestDataWithCompute(String urlString)
    async{
        print(Isolate.current.debugName);
        var client = new HttpClient();
        var request = await
client.getUrl(Uri.parse(urlString));
        var response = await request.close();
        var responseBody = await
response.transform(utf8.decoder).join();
        var dataList = jsonDecode(responseBody);
        return dataList;
    }

```

flutter的广大爱好者们，我在微信创建了一个公众号，搜索 **flutter 干货**、或微信扫一扫下面的二维码关注一下吧，平时会发一些flutter源码解读、技术点使用、平台化落地相关的干货，也是希望大家能把问题抛出来，让我也能研究一下，也希望能互相帮助。