

flutter开发实战-Completer实现将回调Callback转换成Future返回结果

BruceGwo

flutter 开发实战-Completer实现将回调 Callback 转换成 Future 返回结果

一、Future

Dart 是单线程模型语言，如果需要执行一些延迟的操作或者 IO 操作，默认采用单线程同步的方式就有可能造成主 isolate 阻塞，从而导致渲染卡顿。基于这一点我们需要实现单线程的异步方式，基于 Dart 内置的非阻塞 API 实现，那么其中「Future 就是异步中非常重要的角色。Future 表示异步返回的结果，当执行一个异步延迟的计算时候，首先会返回一个 Future 结果，后续代码可以继续执行不会阻塞主 isolate，当 Future 中的计算结果到达时，如果注册了 then 函数回调，对于返回成功的回调就会拿到最终计算的值，对于返回失败的回调就会拿到一个异常信息」。

这部分参考 <https://zhuanlan.zhihu.com/p/353578352>

当然这里并不是介绍 Future，我们下面使用 Completer 实现将回调 Callback 转换成 Future 返回结果

二、Completer

Completer 是一种生成 Future 对象并稍后返回一个 value 或者 error 来完成他们的方法。

大多数情况下，创建一个 Future 最简单的方式，就是使用

Future 构造方法，来捕获单个异步计算。

例如

```
Future doStuff(){
    return someAsyncOperation().then((result) {
        return someOtherAsyncOperation(result);
    });
}
```

但是，当你需要将基于回调的 API 转换为基于 Future 的 API 时，这时候就需要使用 Completer

例如

```
class AsyncOperation {
    final Completer _completer = new Completer();

    Future<T> doOperation() {
        _startOperation();
        return _completer.future; // Send future
object back to client.
    }

    // Something calls this when the value is ready.
    void _finishOperation(T result) {
        _completer.complete(result);
    }

    // If something goes wrong, call this.
    void _errorHappened(error) {
        _completer.completeError(error);
    }
}
```

```
}  
}
```

三、Completer实现Callback转换Future

在上面中的 AsyncOperation，可以使用 Completer 来将回调的 API 转换成 Future 的 API，下面列出一个具体的实例。

代码中有一个获取用户签名的 Api，通过封装的 Callback 来获得具体的 userSign，在需要使用的方法中需要返回一个 Future。

这里就用到了 Completer

具体代码如下

```
// userSign过期获取接口  
class UserSignFetcherImpl implements  
UserSignFetcher {  
    @override  
    Future<String> getUserSign(String userId) {  
        // TODO: implement getUserSign  
  
        Completer<String> completer = Completer();  
        UserSignReq req = UserSignReq();  
        req.userId = userId;  
  
        ApiRepository.getUserSign(DFaceApiConstant.kGetByUs  
erId, req: req)  
            .then((value) {
```

```
        print("getUserSign value:${value}");
        completer.complete(value);
    }).whenComplete(() {
        print("whenComplete");
    }).catchError((e) {
        print("catchError e:${e}");
        completer.completeError(e);
    });
    return completer.future;
}
}
```

这里就可以将接口 Callback 的结果，通过 `completer.complete` 与 `completer.completeError` 将结果转换为基于 Future 的返回的方法了。

四、小结

flutter 开发实战-Completer 实现将回调 Callback 转换成 Future 返回结果