



链滴

如何在微信小程序中使用 Hprose (三)

作者: [andot](#)

原文链接: <https://ld246.com/article/1479659508678>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

Hprose 技术交流群: 48855729

- [如何在微信小程序中使用 Hprose](#)
- [如何在微信小程序中使用 Hprose \(二\)](#)

书接上回，上一回中我们讲到 Hprose 提供的协程可以让 Hprose 的异步调用同步化。但是在最后的子中，有一个小的细节，不知道你有没有注意到，就是程序的最后，我们使用了这样的代码：

```
this.setData({
  userInfo: yield app.getUserInfo
});
```

将原本复杂的通过异步调用来设置 `setData` 的过程同步化了。这是因为 hprose 提供了的协程，可以 `yield` 单参数回调方法的函数，并将回调方法的参数作为该 `yield` 调用的返回值。从而避免了 `that` 和 `this` 之间的转换，也避免了写回调函数。

但是 hprose 的所提供对协程的支持不止于此，hprose 还将 `wx` 的所有 API 都封装到了 `hprose.wx` 面，将需要使用回调方式的 API 封装成了返回 promise 对象的 API，通过这种封装，所有这些 API 就都可以使用 `co/yield` 方式同步调用了。

例如微信官方例子中，上传文件的例子是这样的：

```
wx.chooseImage({
  success: function(res) {
    var tempFilePaths = res.tempFilePaths
    wx.uploadFile({
      url: 'http://example.weixin.qq.com/upload', //仅为示例，非真实的接口地址
      filePath: tempFilePaths[0],
      name: 'file',
      formData: {
        'user': 'test'
      },
    },
    success: function(res){
      var data = res.data
      //do something
    }
  )
})
})
```

现在换成 hprose 所提供的 API，就可以变成这样：

```
var hprose = require('../utils/hprose.js');
var wx = hprose.wx;
var co = hprose.co;

co(function*() {
  var tempFilePaths = (yield wx.chooseImage()).tempFilePaths;
  var data = (yield wx.uploadFile({
    url: 'http://example.weixin.qq.com/upload', //仅为示例，非真实的接口地址
    filePath: tempFilePaths[0],
    name: 'file',
    formData: {
      'user': 'test'
    }
  }));
});
```

```
    }  
  })).data;  
  //do something  
})
```

除去引入 hprose 的代码以外，你会发现 `co` 程序段中的代码，已经完全变成了同步调用的写法，避免了回调，让程序变得更加简单直观和清晰了。

再比如操作文件的例子：

```
wx.startRecord({  
  success: function(res) {  
    var tempFilePath = res.tempFilePath  
    wx.saveFile({  
      tempFilePath: tempFilePath,  
      success: function(res) {  
        var savedFilePath = res.savedFilePath  
      }  
    })  
  }  
})
```

转化成同步代码是这样的：

```
var res = yield wx.startRecord();  
var tempFilePath = res.tempFilePath;  
res = yield wx.saveFile({ tempFilePath: tempFilePath });  
var saveFilePath = res.savedFilePath;
```

没有回调，没有嵌套。整个程序就这么清爽！

好了，我就不多举例了，剩下的，我想你已经知道该怎么做了。