



黑客派

如何在微信小程序中使用 Hprose

作者: [andot](#)

原文链接: <https://hacpai.com/article/1479198647080>

来源网站: [黑客派](#)

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

首先下载 [微信小程序开发工具](#)，之后安装。

然后按照网上那一堆微信小程序 Hello World 教程创建一个小程序。

接下来我们下载 [hprose-wx](#)，或者从[开源中国的 Git 服务器镜像](#)下载。

可以用 git clone 下载，也可以只下载 dist 目录下的文件，dist 目录下面的这两个版本都可以用：

- hprose.src.js 是源码版本
- hprose.js 是压缩版本

调试阶段建议用源码版本。

之后，你可以把它们复制到你创建的那个微信小程序的 utils 目录下（复制其中一个就可以），然后它改名为 hprose.js（这一步可选，只为后面引用的时候，名称统一）。

接下来，打开 pages/index/index.js 文件。

在开头加上：

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

然后在 onLoad 事件中加入以下代码：

```
var client = hprose.Client.create("http://www.hprose.com/example/", ["hello"]);
client.hello("world", function(result) {
  console.log(result);
});
```

总体看上去是这样的：

```
//index.js
```

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

```
//获取应用实例
```

```
var app = getApp()
```

```
Page({
```

```
  data: {
```

```
    motto: 'Hello World',
```

```
    userInfo: {}
```

```
  },
```

```
  //事件处理函数
```

```
  bindViewTap: function() {
```

```
    wx.navigateTo({
```

```
      url: '../logs/logs'
```

```
    })
```

```
  },
```

```
  onLoad: function () {
```

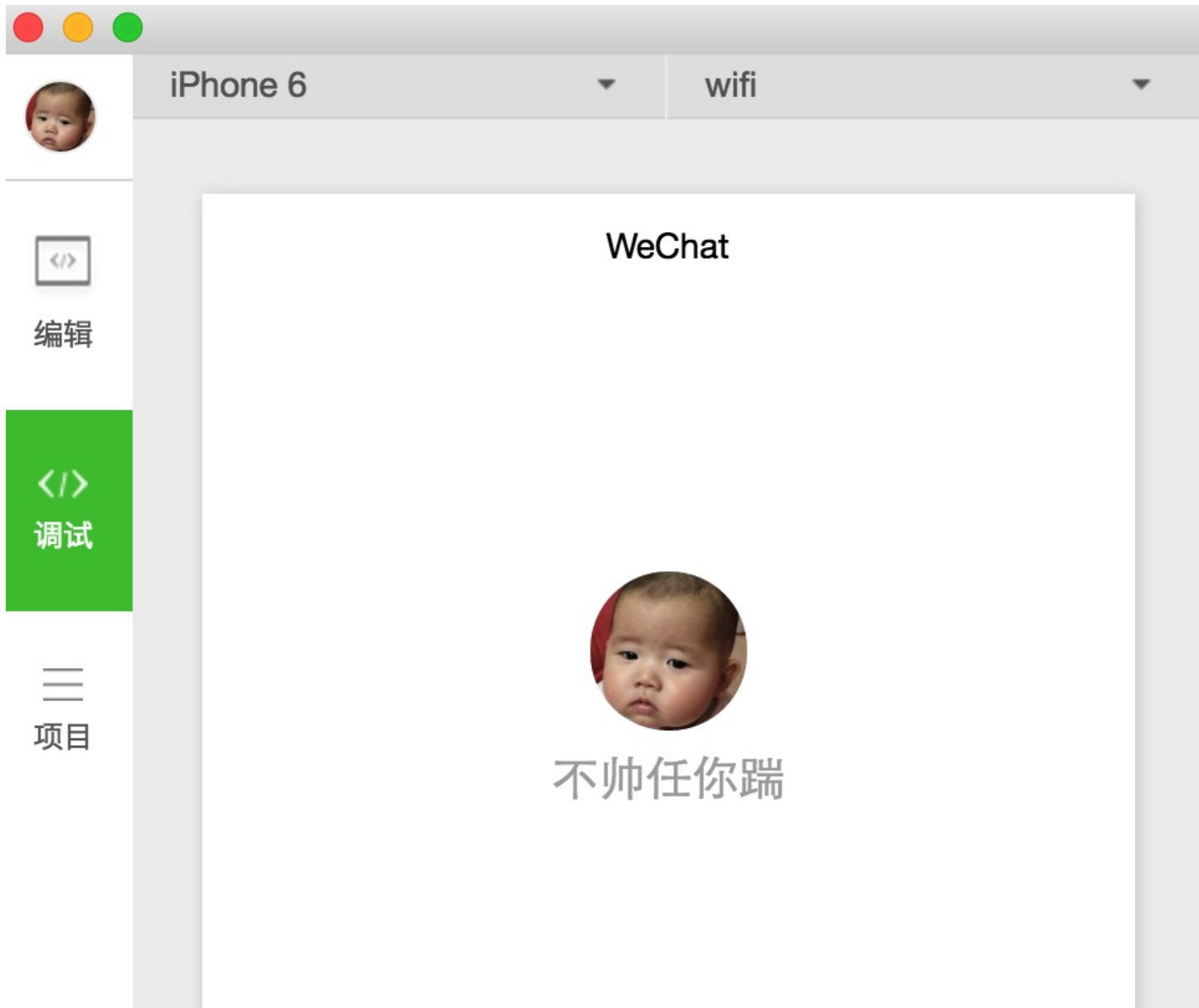
```
    console.log('onLoad')
```

```
    var client = hprose.Client.create("http://www.hprose.com/example/", ["hello"]);
```

```
    client.hello("world", function(result) {
```

```
    console.log(result);
  });
  var that = this
  //调用应用实例的方法获取全局数据
  app.getUserInfo(function(userInfo){
    //更新数据
    that.setData({
      userInfo:userInfo
    })
  })
}
```

然后点编译，运行，如果你的网络没有问题的，你会在调试控制台中看到：



好了，就这么简单，接下来，你就可以用 hprose 来做微信小程序开发了。

虽然上面仅仅列出了简单调用，但实际上 Hprose 的所有功能在微信小程序中全部可以使用，包括负载均衡，故障切换，幂等性调用失败自动重试，订阅推送等等。而且不仅仅支持 https 服务，也支持 wss

服务，而且使用方式也是一样的。当然，如果你需要推送数据，或者需要传输二进制数据，使用 ss 服务是最好的选择。

对于 hprose 服务器来说，编写 wss 服务和编写 https 服务器也是一样的，而且发布了 wss 服的同时，也会发布相同的 https 服务，因此，你两种服务可以同时使用。

如果你用的是微信小程序的正式开发账号，那么连接的服务器地址应该是 https 的而不能是 http 的而且应该在开发账号里设置要连接的服务器地址。

- [如何在微信小程序中使用 Hprose \(二\)](#)
- [如何在微信小程序中使用 Hprose \(三\)](#)