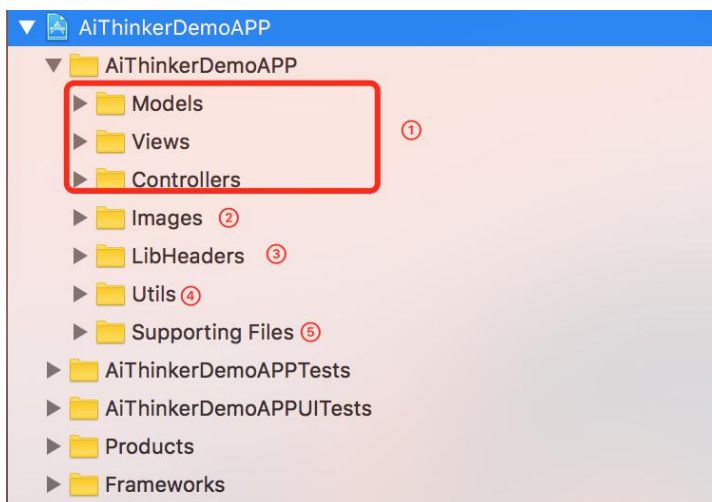


AiCloud2.0 iOS Demo App 快速使用指南

一：SDK 环境说明

1. 用户登录返回值只保留 15 分钟，在 APP 开发过程中需要做好判断以免产生不必要的 bug 。
2. 使用 SDK 前配置：“Build Settings”中的“Other Linker Flags”选项添加“-all_load”。
3. 此 SDK 引用的第三方框架有 MQTTClient, MKNetwork, AsyncSocket 开发者要避免重复引用。
4. Uuid: uuid 应当保证在应用安装后保持不变，且尽可能唯一，以便跟踪回溯。建议在 APP 第一次启动时把 unix 时间戳*1000+毫秒值将其转换成十六进制字符串，然后截取前 12 个字节，如果不足 12 个字节，用随机字符串填充。(加毫秒值考虑到了同一个用户有可能在一秒内同时打开多于一个 APP，使用毫秒的话就很难做到在一毫秒内同时打开多个 APP，这样对于同一个用户而言就很难不唯一了)生成完毕后保存，并在 APP 卸载之前保证不变。

二：项目目录结构：

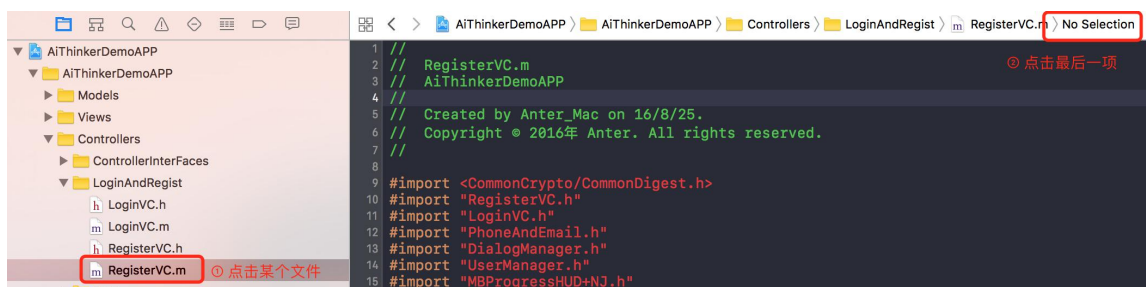


详细说明:

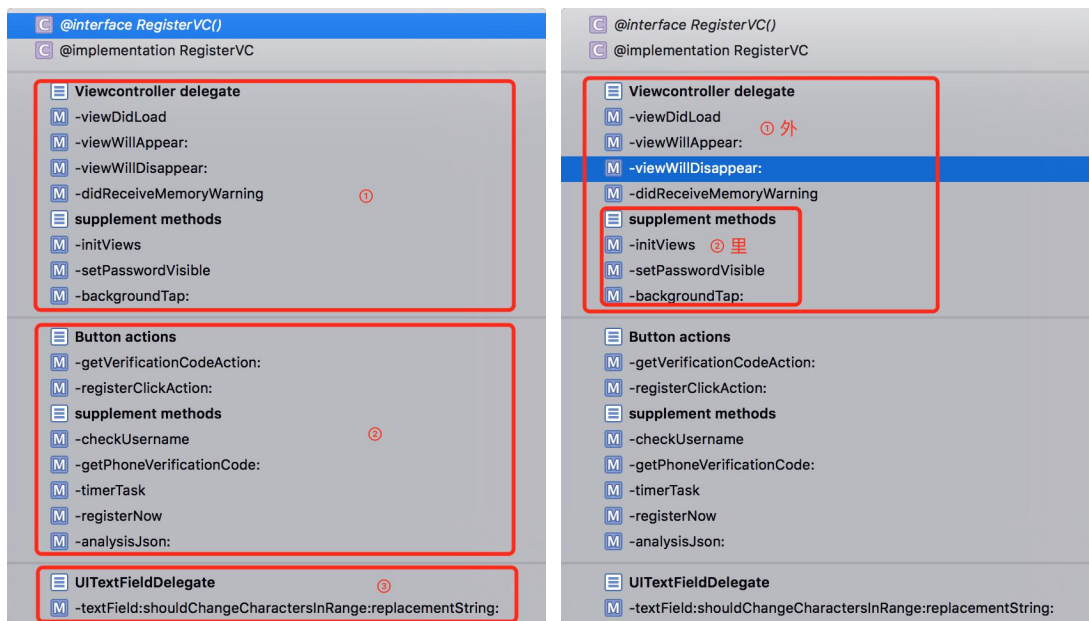
- 1: 图中序号①包含的是 MVC (模型, 视图, 控制器)。
- 2: 图中序号②包含的是图片资源
- 3: 图中序号③包含的是 'AiThinker.a' 静态库头文件
- 4: 图中序号④包含的是第三方组件和工具类
- 5: 图中序号⑤包含的是库, pch 文件, info.list 等等

三: 操作说明

1: 查看某个文件的所有方法



2: 下拉得到的方法说明



整个文件拆分为几个部分（左图为三个部分）；外面的部分将调用里面的方法（右图）

三：设备列表界面说明

1：刷新动作

(1) 位置：

```
- (void)setRefresh { //... }
```

(2) 方法简单说明：

```
[operationQueue addOperation:op]; //开启部署线程
```

```
[self startrequestTimeoverTimer]; //开启请求超时定时器
```

```
[self getDeviceListFromServer]; //获取服务器列表
```

```
refreshTimer = [NSTimer scheduledTimerWithTimeInterval:8.0 repeats:NO
```

```
block:^(NSTimer * _Nonnull timer) { // ... } //设置刷新超时
```

2：点击列表项

(1) 位置：

```
- (void)tableView:(UITableView *)tableView
```

```
didSelectRowAtIndexPath:(NSIndexPath *)indexPath { // ... } UITableView 代理方法
```

(2) 方法简单说明：

点击的动作只发生在未绑定和在线的 section 内，点击未绑定 section 内的列表项时将开启请求超时定时器并调用绑定方法；点击在线 section 内的列表项时将当前设备的信息传递给将要跳转的界面，跳转界面根据设备的 PRODEUCTID 进行区分。

3：添加设备

(1) 位置：

初始化 view 的 UIBarButtonItem 时添加的事件

(2) 简单说明:

点击跳转到智能配网界面

4: 绑定

(1) 位置:

```
-(void)bindDeviceForUser:(NSDictionary *)deviceInfo  
indexRow:(NSInteger)indexRow {///  
}
```

(2) 方法简单说明:

绑定方法调用 SDK 中的绑定方法

5: 解绑

(1) 位置:

```
-(void)unbindDeviceForUser:(NSDictionary *)devInfo  
deviceToUnbind:(NSIndexPath *)indexPath remove:(NSInteger)section  
{///  
}
```

(2) 方法简单说明:

绑定方法调用 SDK 中的绑定方法，解绑成功时在列表中对应项。

6: 离线设备的收起和展开

(1) 位置:

```
-(void)headerViewClickAction:(UIButton *)button {///  
}
```

(2) 方法简单说明:

在 tabelview 点击 headerview

7: TableView 背景

(1) 位置:

- (void)setBackgroundViewImage {}

(2) 方法简单说明:

[self.view addSubview:imageView1]; 添加背景图片, 当列表为空时, 列表将隐藏并显示背景图片

四: 透传界面说明

1: 输入限定

(1) 位置:

- (BOOL)textField:(UITextField *)textField
shouldChangeCharactersInRange:(NSRange)range
replacementString:(NSString *)string {///
}

(2) 方法简单说明:

检查是否为粘贴的情况, 粘贴情况下, 粘贴的内容严格控制为十六进制字符串并且形成配对, 而且在输入法中选择长串时也将判断为粘贴; 键盘输入限制为十六进制字符, 不限制大小写, 输入两个字符将自动添加一个空格。

2: 数据接收显示

(1) 位置:

- (void)tcpReceiveData:(NSData *)data {///
} 和 -
(void)tcpReceiveDictionary:(NSDictionary *)receiveDict {///
}

(2) 方法简单说明:

tcpReceiveData, 将接收到的 NSData 转成字节, 字节再转成十六进制字符串并显示到接收框中; tcpReceiveDictionary 的数据解析根据具体的协议进行解析并显示。

3: 数据发送显示

(1) 位置:

- (IBAction)sendButtonAction:(id)sender {///
...} 按钮点击事件

(2) 方法简单说明:

将输入框的内容去掉空格, 转成大写, 再重新两两配对并以空格为间隔, 最后的结果显示到发送框中

4: 发送按钮事件

(1) 位置:

- (IBAction)sendButtonAction:(id)sender {///
...}

(2) 方法简单说明:

格式化处理输入框中的字符为两两大写十六进制以空格为间隔的字符串并显示到发送框中(配对剩余, 则在最后一个前面补0), 接着去空格, 转成 NSData, 调用 SDK 中的 tcpSendData(NSData) 或者 publishMessage(NSDictionary) 发送到设备。