# 使用Olam SDK语音控制一个支持HomeKit的智能家居的iOS程序

**原文网址:** [**http://blog.csdn.net/dfman1978/article/details/72317312**](http://blog.csdn.net/dfman1978/article/details/72317312)

**前言**

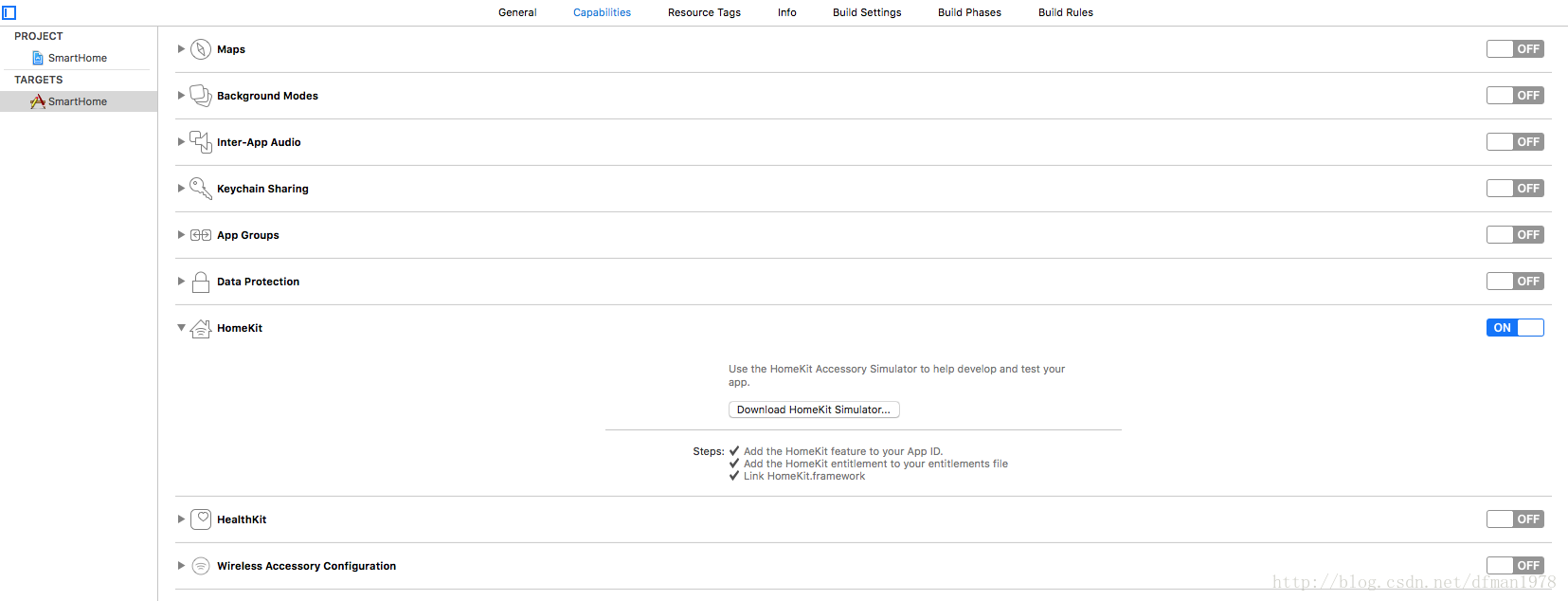
HomeKit是苹果发布的智能家居平台。通过HomeKit组件，用户可以通过iphone、iPad和ipod Touch来控制智能灯泡，风扇、空调等支持HomeKit的智能家居,尤其是可以通过Siri进行语音控制。   
但是通过Siri进行语音控制有个很大的问题，就是Siri支持的语料无法进行自由的扩展，没办法添加更多的说法。而Olami SDK则可以通过OSL(OLAMI 语法描述语言 OLAMI Syntax Language，简称：OSL)自由的进行扩展，对智能对话的能力扩展变得非常容易通过。Olami的服务器对输入的语料进行语义的解析，获得想要的控制结果。这样就可以自己对语料和说法进行扩展，丰富了操作的功能。

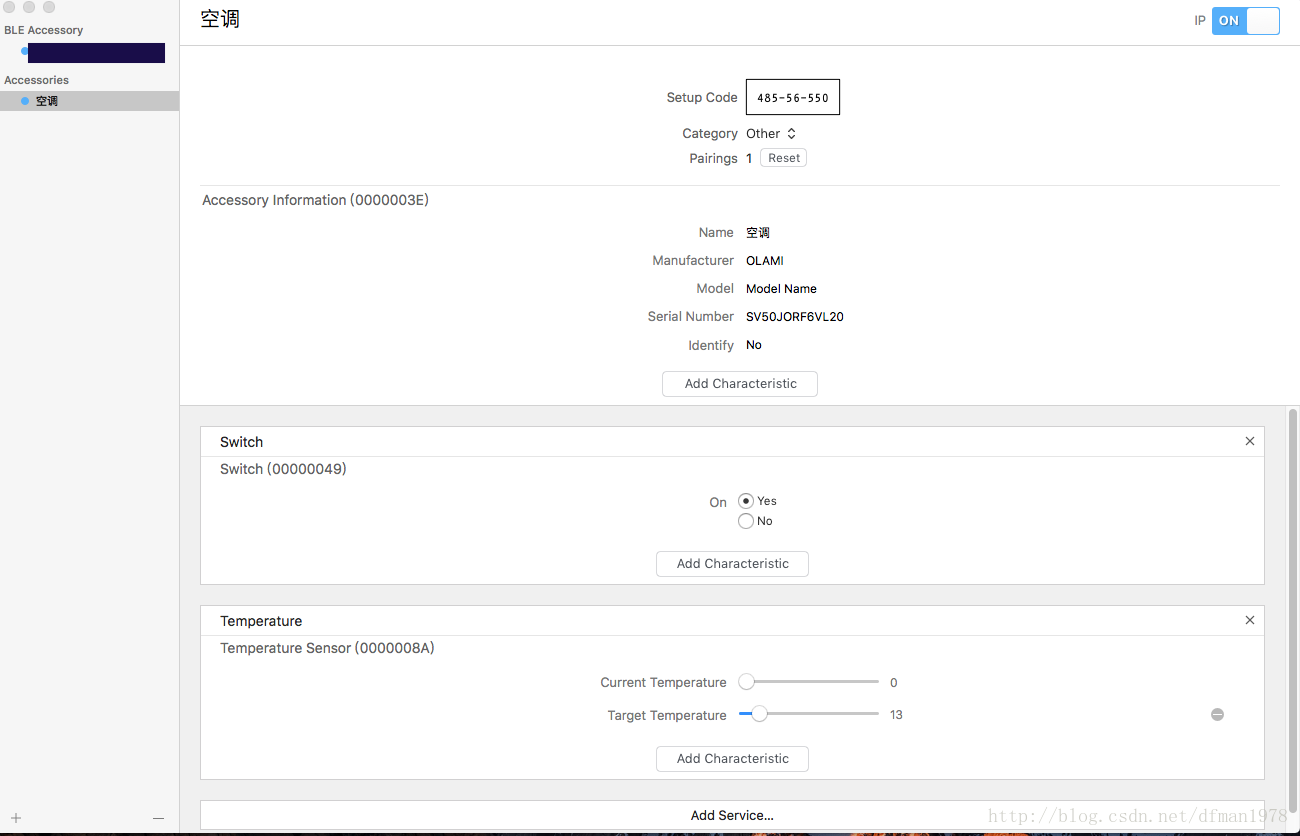
**准备工作**

* HomeKit关于智能设备的一些基本概念和模拟器的安装

由于市面上支持HomeKit的设备不多也比较昂贵，苹果贴心的提供了一个HomeKit模拟器来帮助程序员进行HomeKit设备的开发。这个Demo是模拟一个智能空调，功能是通过语音控制空调的开关和调节温度。

HomeKit模拟器默认不是安装的，要去下载安装   
在Taget->Capabilities->HomeKit一项中可以下载，见下图



安装好以后，就可以打开使用了。见下图   


在Homekit中有几个概念，在这里简单的介绍一下APP里用到的，网上有比较详细的介绍，也可以参考我转载的这篇博文：   
<http://blog.csdn.net/dfman1978/article/details/72179458>   
HomeKit允许用户添加多个home，但是使用的时候只能有一个home，称为primaryHome。每个Home可以有多个room.智能设备称为Accessory属于home，但是每个Accessory可以配个一个room。如果没有指定room，则默认分配给roomForEntireHome返回的默认room.每个Accessory可以有多个Service，每个Service可以有多个特性(HMCharacteristic)。有的特性是只读的，例如空调当前的温度；有的特性是可读可写的，例如空调的开关。APP就是通过控制这些特性的值来控制智能设备的。

在程序中添加了一个Accessory，名称起为空调。然后给它添加了两个Service：一个是Switch 开关，一个是Temperature 控制空调的温度。通过APP，可是实现空调的开关和温度的调节。

* Olami 智能设备的语法规则的编写和配置   
  用户的说的话， APP怎么知道说的是什么意思，怎么去理解呢？这就涉及到了OSL语法描述语言。想要使用Olami平台提供的语音和语义理解API，首先要根据OLAMI 语法描述语言（OLAMI Syntax Language，简称：OSL）的规则编写一套语法。OSL的简介如下网址有详细的介绍<https://cn.olami.ai/wiki/?mp=osl&content=osl1.html>   
  通过Olami平台提供的NLI 自然语言语义互动系统,可以学习到如何为自己应用的业务编写一套语法规则。

自然语言语义互动（Natural Language Interaction, 简称：NLI）管理系统是一套在线语义解析管理工具，NLI 系统采用 OLAMI 语法描述语言（OLAMI Syntax Language，简称：OSL）取代复杂的编码编程，让即便没有软件研发背景的使用者也能轻松快速的维护包含语义扩展及答案的智能对话流。

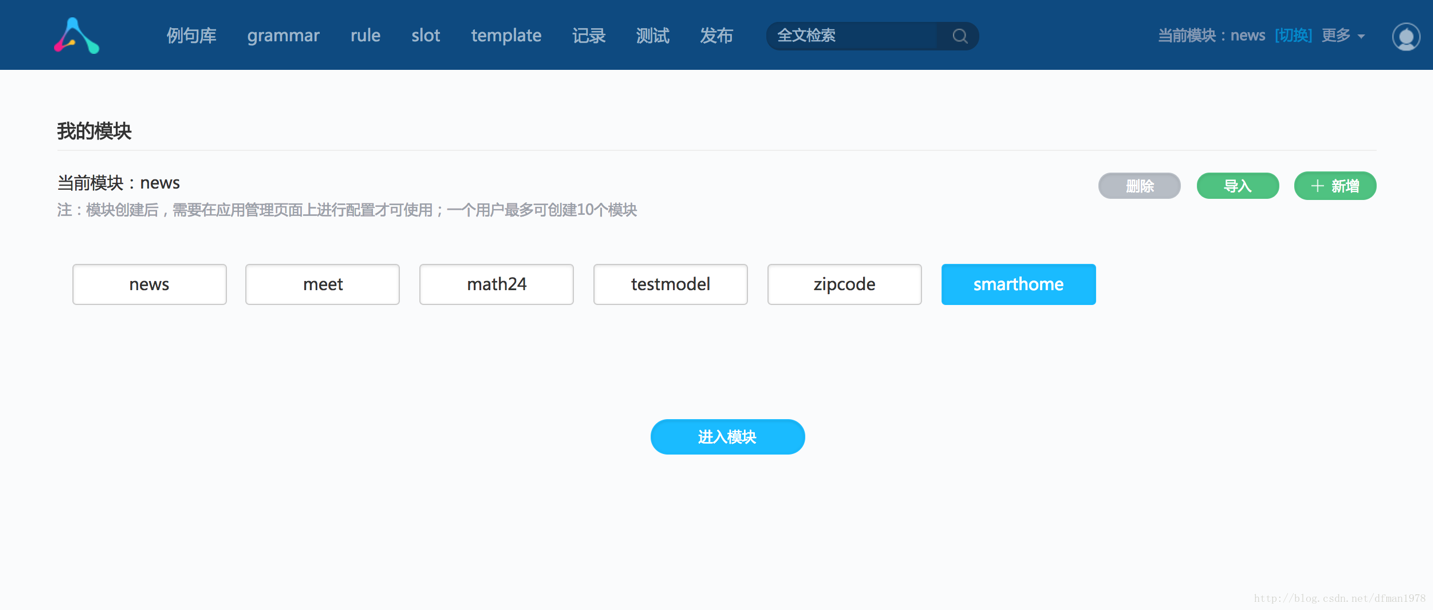
在如下网址可以了解到更加详细的内容   
<https://cn.olami.ai/wiki/?mp=nli&content=nli1.html>   
方便的是Olami平台已经对很多领域方面的提供了一些写好的语法规则，这些在Olami中称为模块。其中关于智能设备已经写好，下面就一步一步配置一下。   
首先要去Olami的平台注册一下，注册后进入到这个界面

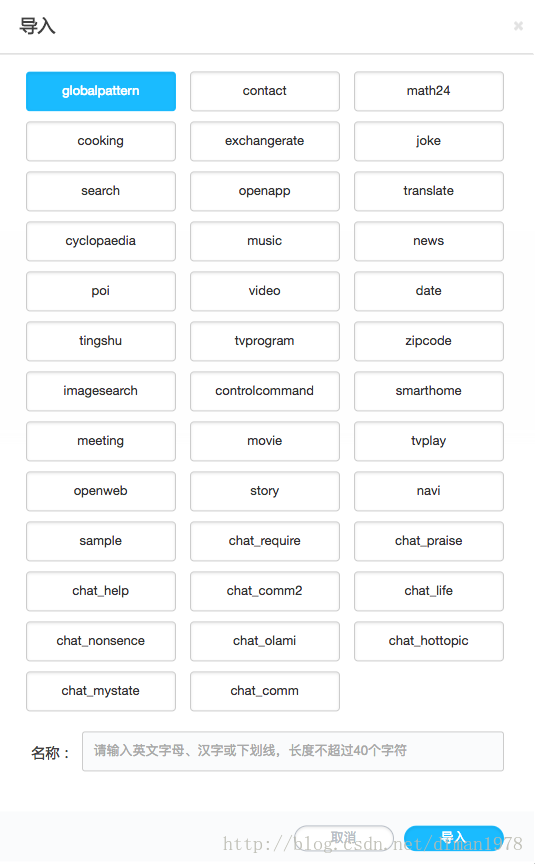


点击“创建应用”转到下面这个页面

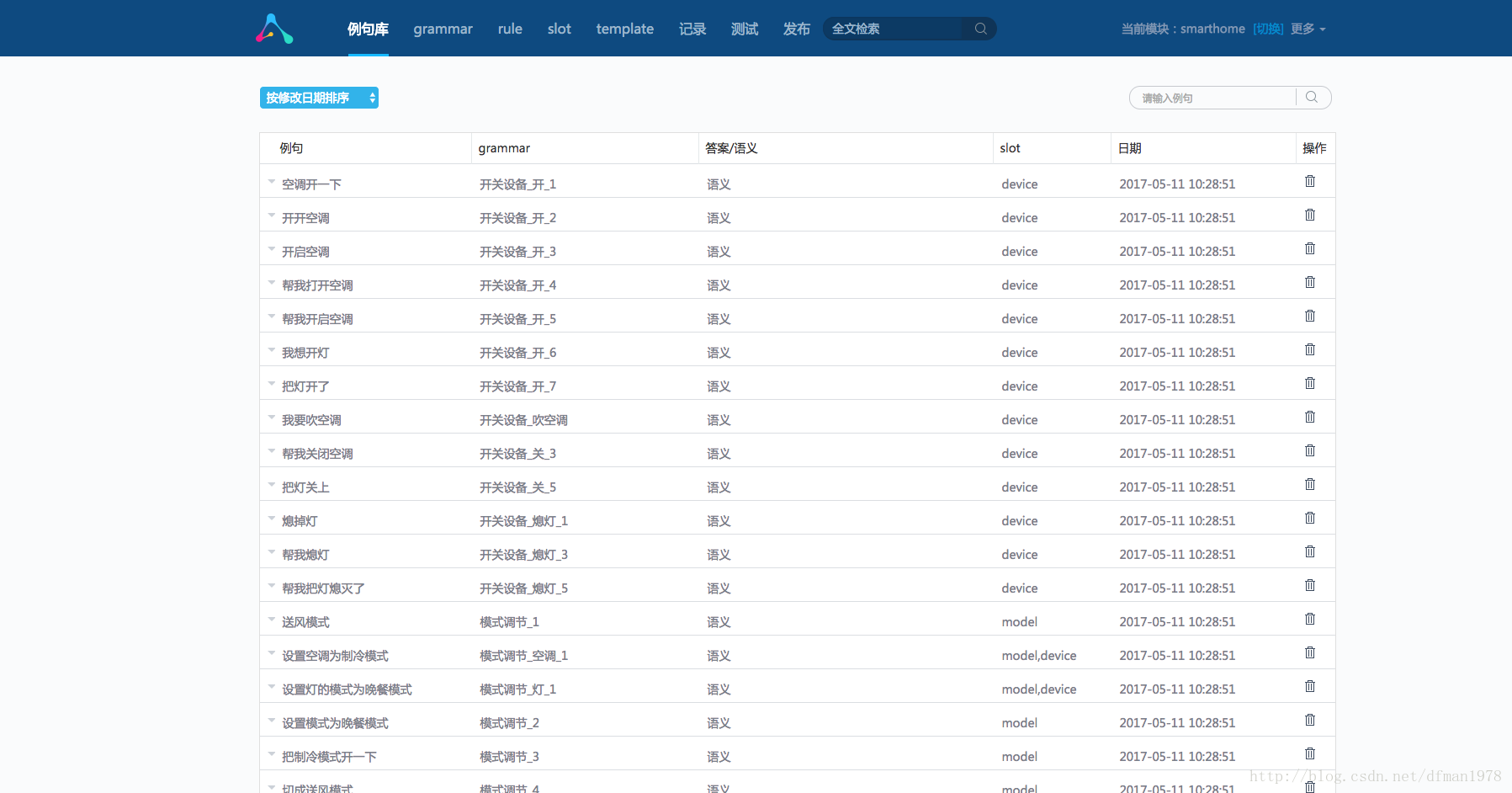


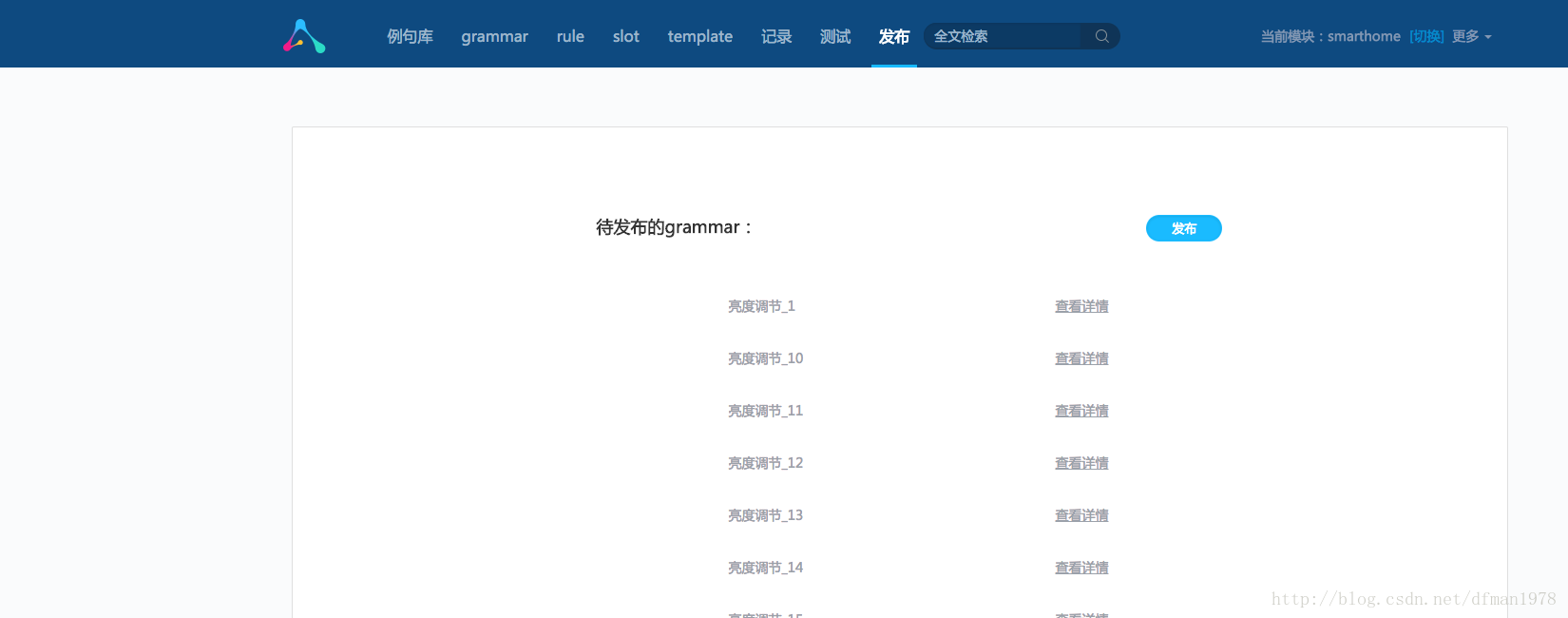
填写 应用名称，应用描述，应用介绍以后，就可以创建了。回到上一个页面，就可以看到创建的应用了。

点击”进入NLI系统”就可以进入模块页面   


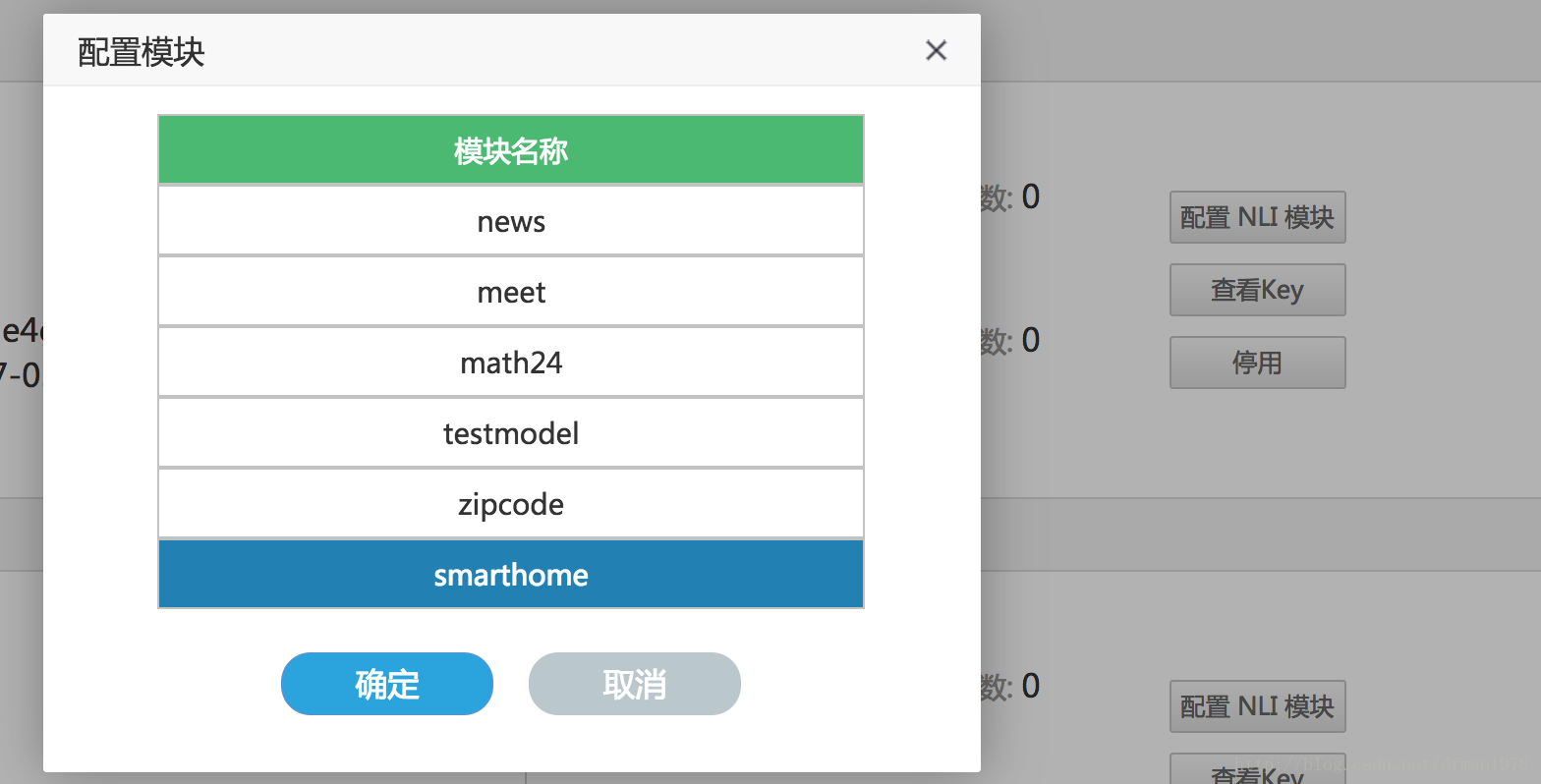
在官网已经内置了很多领域的grammar.在模块页面大家点击“导入”按钮，查看已有领域的模块   


选择一个要使用的，例如我要导入”smarthome”这个模块，先选择它，点击“导入” 按钮   


然后进入 smarthome模块，就可以看到例句了   


但是这个时候还是不能使用，需要先进行发布。点击页面上方的”发布”按钮，进入发布页面   


点击“发布”按钮，发布成功   


最后还要回到“我的应用”界面，点击”配置NLI模块”按钮，让自己创建的应用和模块关联起来   


这样smarthome的语法文件就配置好了，就可以使用了。

**代码的实现**

1.获取home和Accessory的对象

- (void)viewWillAppear:(BOOL)animated {

[super viewWillAppear:animated];

self.accessories = [NSMutableArray array];

if (self.homeManager && self.homeManager.primaryHome) {

for (HMAccessory \*accessory in self.homeManager.primaryHome.accessories) {

[self.accessories insertObject:accessory atIndex:0];

accessory.delegate = self;

[self.tableView reloadData];

}

}

if (\_currentHome) {

\_currentHomeLabel.text = [NSString stringWithFormat:@"当前Home：%@", \_currentHome.name];

}

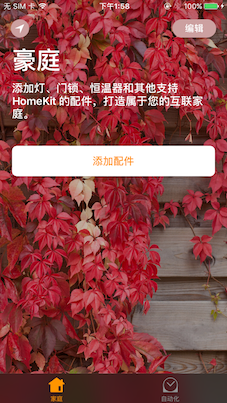
}

在ViewWillAppear的时候来获取当前Home的对象和名字，还有Accessory的对象，并保存起来，用一个tableView来显示下图是程序刚启动的时候的界面，这个时候因为没有添加Home，所以什么都没有   


点击“添加Home”按钮弹出一个对话框，可以填入一个Home的名称。对应的代码

- (IBAction)addHomeBtnClicked:(id)sender

下一步就是添加Accessory，在代码里没有做这一步。是通过iOS提供的“家庭”APP来添加智能空调的。   
点击“家庭”App,弹出页面   


点击“开始使用”，弹出   


点击“添加配件”,App会自动搜索刚才在模拟器添加的那个空调，按照提示一步一步添加，最后显示空调的两个Service：Switch和Temperature   


在回到APP，这个时候APP，就会显示出搜索到的Accessory：空调   


点击“空调”,进入语音控制页面   
   
在这个页面中，上面的TableView显示了空调的一些属性。   
下面的TextView用来显示ASR的一些结果   
圆形按钮用来录音

2..去网址<https://cn.olami.ai/wiki/?mp=sdk&content=sdk_and_sample.html>下载Olami SDK.包括两个文件，其中的一个是Olami的静态函数库，一个是其头文件   
第一步是初始化Olami的语音识别对象，并设置代理

olamiRecognizer= [[OlamiRecognizer alloc] init];

olamiRecognizer.delegate = self;

3.调用setAuthorization函数进行授权

[olamiRecognizer setAuthorization:@"d13bbcbef2a4460dbf19ced850eb5d83"

api:@"asr" appSecret:@"3b08b349c0924a79869153bea334dd86" cusid:OLACUSID];

这个函数的参数的说明在OlamiRecognizer中有说明，也可以去在线API说明去查看   
<https://cn.olami.ai/wiki/?mp=sdk&content=sdk/ios/reference.html>

参数就是刚才创建应用的时候获得的。

4.设置语系

[olamiRecognizer setLocalization:LANGUAGE\_SIMPLIFIED\_CHINESE];

在进行录音之前必须要先进行设置，否则会得不到结果。目前只支持简体中文(LANGUAGE\_SIMPLIFIED\_CHINESE)

4.开始录音   
调用 start()接口开始进行录音

[olamiRecognizer start];

5.得到录音的文字和语义，并对其进行处理   
通过调用stop()函数或者自动停止，都会获得录音的文字和对其进行的语义分析的结果   
实现OlamiRecognizerDelegate onResult函数可以获得结果，其结果以一个json字符串的形式回调过来，对这个字符串进行解析，就可以获得想要的数字。例如对着话筒说”打开空调”，得到的结果如下

{

"data": {

"asr": {

"result": "打开空调",

"speech\_status": 0,

"final": true,

"status": 0

},

"nli": [

{

"desc\_obj": {

"status": 0

},

"semantic": [

{

"app": "smarthome",

"input": "打开空调",

"slots": [

{

"name": "device",

"value": "空调"

}

],

"modifier": [

"open"

],

"customer": "58df685e84ae11f0bb7b4893"

}

],

"type": "smarthome"

}

]

},

"status": "ok"

}

这个是根据OSL语法描述语言定义的一套规则，返回的结果。这个结果的说明在 <https://cn.olami.ai/wiki/?mp=api_nlu&content=api_nlu3.html> 这个网址上有说明。   
6.获得所有Accessory的对象

for(HMService \*service in \_accessory.services) {

[\_serviceDic setObject:service forKey:service.name];

NSLog(@"service.name is %@",service.name);

}

保存在一个Dictionary中，服务的名称是key，对象的指针是Value

7.onResult 函数的说明   
在整个程序中，最主要的一个函数就是onResult函数，这个函数就是对传过来的结果进行处理。在这个函数中，调用了三个函数，分别来处理josn格式中的三个比较重要的节点

- (void)processASR:(NSDictionary\*)asrDic

这个用来处理ASR节点，获得语音识别的结果，如果没有结果，则弹出一个对话框进行提示

- (void)processSemantic:(NSDictionary\*)semanticDic

这个用来处理Semantic节点，这个节点中包含了slot的值和modifier的值。OSL 语法描述语言中的 slot 可理解为语义中的变量，用于传递、提取信息，是代码处理的数据的来源。对于本程序来说，就是进行控制空调的各种动作，例如开，关。还有就是调节的温度值。关于slot的值可以参考 <https://cn.olami.ai/wiki/?mp=osl&content=osl_slot.html>，这里有详细说明

- (void)processModify:(NSString\*) str

这个用来处理语音和语义的结果。这个函数主要是处理json字符串中的modifier节点。modifier 语法描述规则是 OSL 语法描述语言中，除了 slot 以外的另一种内置的信息传递机制，一般用来表示语义目的，也可以理解为对于语义的一种注释方式，以便让应用程序的开发者得知 grammar 所代表的相应意图。详细说明参考   
<https://cn.olami.ai/wiki/?mp=osl&content=osl_regex.html#11>，通过modifier，我们才能知道程序的意图是什么？例如是想打开、关闭空调。还是调节温度

在代码中我们处理了三个modifier:”open”、”close” 和”control\_temperature”.然后根据slot的值进行进行进一步处理。   
拿其中处理“关闭”空调的动作来做说明

if ([str isEqualToString:@"close"]){//关闭空调

HMService \*tmpService = \_serviceDic[@"Switch"];

HMCharacteristic \*characteristic = tmpService.characteristics[1];

if ([characteristic.characteristicType isEqualToString:HMCharacteristicTypeTargetLockMechanismState] ||

[characteristic.characteristicType isEqualToString:HMCharacteristicTypePowerState] ||

[characteristic.characteristicType isEqualToString:HMCharacteristicTypeObstructionDetected]) {

[characteristic writeValue:@NO completionHandler:^(NSError \*error){

if(error == nil) {

dispatch\_async(dispatch\_get\_main\_queue(), ^ {

\_asrTextView.text = @"空调已关闭";

});

} else {

NSLog(@"error in writing characterstic: %@", error);

\_asrTextView.text = @"空调关闭失败，请重试!";

}

}];

}

}

如果modifier等于”close”,那么说明这是一个Switch服务的属性。通过“Switch”获得Service的指针，然后获得Switch的属性。然后通过

- (void)writeValue:(nullable id)value completionHandler:(void (^)(NSError \* \_\_nullable error))completion;

函数来修改这个属性的值。因为是关闭空调，所以直接写入@NO就可以了   
因为这个函数是异步返回的，返回的时候不一定操作成功，所以要对返回结果进行一下处理。

**代码下载**

代码可以到GitHub上下载   
<https://github.com/lym-ay/SmartHome>

**Olami SDK 相关开发资源**

另外这里还有两篇anroid上使用Olami SDK开发程序的文章

这个是一个听书的程序   
<http://blog.csdn.net/ls0609/article/details/71519203>

这个是一个关于天气的程序   
<http://blog.csdn.net/zhangxy0605/article/details/71601604>