

简介

芯联米家插件的脚手架

相关记录

版本	修改人	相关信息
1.0.2	kkopite	首个米家插件生成脚手架

使用

```
npm install -g corelink-cli

cd xx/米家插件项目/projects
cr1 init <项目名称>
```

模板的目录结构

```
| index.ios.js(ios入口)
| index.js(android入口)
| package.json
| project.json
| ratio.js(尺寸适配相关)
| README.md
| router.js(路由声明)
| style.js(通用样式文件)
|
├─component(通用组件文件夹)
|   AppText.js(自带传入默认的字体样式的Text组件)
|
├─constant(常量文件夹)
|   index.js(设备的相关属性，action等相关信息)
|
├─i18n(语言适配，如果要加其他语言，自行添加并在index.js引入)
|   en.js
|   index.js(导出所有语言)
|   ko.js
|   zh-hk.js
|   zh-tw.js
|   zh.js
|
├─mock(mock属性读写，订阅等)
|
├─page(页面文件夹，需要的可自行再添加)
|   CommonSetting.js(通用设置页面，已处理好米家规范内所需添加
    的内容，以及固件升级的提示等)
|   MainPage.js(首页)
```

```

|
├─resources(资源目录, 存放图片等)
|   └─raw(存放一些中英文的用户协议, 隐私协议, 新建项目后, 需要替换好
这些内容)
|       agreement-zh.html
|       agreement.html
|       privacy-zh.html
|       privacy.html
|
└─util
    LocalizedStrings.js(语言适配功能模块, 直接从米家demo项目
中抽出来的, 无需修改)
    privacy.js(处理隐私等相关内容首次进入弹窗的模块)
    device.js(封装对硬件的操作:属性读写, 订阅等, 便于在真实以
及模拟调用的切换)

```

注意

关于隐私协议相关的

- 模板项目中, 默认引入了中英文的隐私, 用户协议, 可见 `i18n/zh.js`, `i18n/en.js`, 新项目只需替换好 `resource/raw` 下的相关文件, 如若需要其他语言的隐私协议, 可按照 `zh.js` 的写法引入相应文件
- 已经处理好撤销授权后, 自动退出插件
- 已经处理好首次进入插件弹出授权提示及通用界面内引入隐私政策(参加 `i18n/zh.js`, `i18n/en.js` 的内容)

```

// zh.js
export default {
  agreement: require('../resources/raw/agreement-
zh.html'), // 用户协议
  privacy: require('../resources/raw/privacy-zh.html'), //
隐私政策
  hello: '你好'
}

```

这里要注意的是, 项目组引用隐私的地方都是按 `agreement`, `privacy` 这两个key来引用的, 所有不要去删除或修改

关于语言适配

```

// 引入
import i18n from 'xxx/xx/i18n'

// 使用
function Hello() {
  return (<Text>{i18n.hello}</Text>)
}

```

[更多的语言模板用法点击查看](#)

关于设置中固件升级

- 已处理好设置界面中固件升级的小红点显示/隐藏, 具体代码见 `./page/CommonSetting.js`

关于mock

与做普通的前端项目时后端接口尚未实现, 插件开发会碰到硬件尚未完成亦或是设备刚好被借走, 这时候就可以通过mock对设备属性的读写, 订阅等功能, 来快速验证前端的业务逻辑. 默认是不开启的

```
// mock/index.js

// 启用mock
const USE MOCK = true
// /关闭mock
const USE MOCK = false
```

切记打包前请关闭mock, 该功能仅限于硬件未完成时使用, 硬件功能完成后要使用真实设备来开发

mock属性的默认值需要手动添加

```
// mock/index.js

obj['prop.2.1'] = true
```

mock订阅采用的是在mock内部建一个定时器, 定时去通知订阅者, 一开始是去代理mockObj, 监听其属性变化后再去通知, 这样实时性好, 但是貌似react native有限制, 在console里面调用setState是不会去更新的, 而这种做法也是相当于在console调用更新

因此当前的做法便是不断去轮询属性, 而后判断是否需要通知(会记录上传通知的值), 避免更多的通知

当然其实是无所谓的, 因为mock模块只会在开发时使用

米家后台的功能定义中, 各个客户的属性填写的值不规范, 没办法直接编写脚本来获取生成, 所以当前只能通过手动添加

关于Text组件

使用默认的Text组件会导致在部分小米手机上文字显示不全, 需要添加字体样式或者设置文字宽度

因此该模板提供一个带有默认字体的Text组件, 省去传入字体样式的麻烦

```
import Text from 'root/component/AppText'

function Hello() {
  // 使用方法的React native提供的Text组件一样
  return (<Text>hello world</Text>)
  ...
}
```

关于标题栏再切换页面时, 会出现变化的问题

首页的状态栏是白字的, 进入设置界面或其他界面, 状态栏是黑字的, 此时首页收到订阅通知, 没做变化直接更新, 引起render后, 会重新渲染出白字的状态栏, 导致再当前界面状态栏出现错乱, 之前的很多bug都是这个引起的

同时, 没法返回首页后, 需要重新设置下状态栏的颜色, 字体(如果首页和其他页面的状态栏字体是一样的, 就不用)

todo

✓ 改用yo来作为脚手架搭建工具