app开发主要工具库使用介绍

*——目的是为了让app开发更简单，后期维护起来不用那么费劲。*

平台集成：

1. http.js

此模块导出与ajax请求相关的方法，使用vue推荐ajax库axios再次封装。



**formateUrl (controllerName)**：

-- 传入controllerName，控制器加方法名（如“productPadCommController!getAppFirstPageInfo.m”）；

-- 返回一个格式化后带自动登录的url；

**post (params)**：post请求方法

-- 传入一个对象，默认值为：

params = {

url: '', // 非全部长度的url，应该是控制器加方法名，参考http.url 这个产量对象

data: {}, // post请求需要传入的对象，默认空对象

autoLogin:false // 是否自动登录，默认false，大部分情况都要带上，设置为true

}

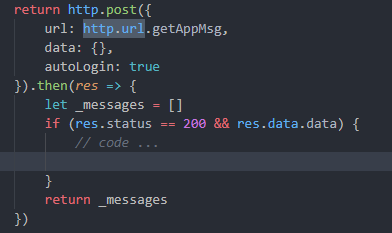
-- 返回一个promise对象，详情查阅js的es6规范。

**get (params)**：同post请求方法。。

**url** 一个保存控制器加方法名的产量

-- 目的为了方便统一管理项目当中使用到的url，可以一次修改全局应用，推荐使用。

http.js 工具使用实例：



2. business\_product.js

此模块导出与ajax请求后，需要进行处理的、或者直接要对数据进行处理，不涉及独立业务逻辑的方法，类似一个服务提供者（provider）。



举例：

**login (params)** ：登录方法

与普通post请求相似，传入参数不同

-- 传入一个对象，默认值为：

params = {

username: 'admin', // 用户名

password: '123456', // 密码

}

-- 返回一个promise对象，默认回调参数是 登录返回结果

**getMenulist (config)** ：获取菜单数据方法

-- 传入一个对象，默认值为：

config = {

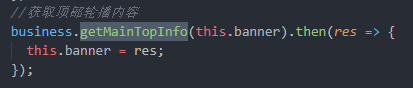
filter: true, // 布尔值，是否过滤菜单编辑中未选中的菜单，默认是

fold:true, // 布尔值，是否按16个菜单为一组来返回，默认是

}

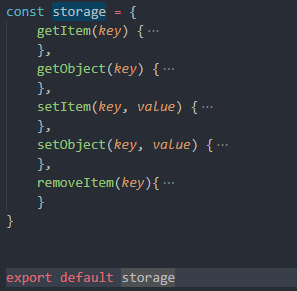
-- 返回一个promise对象，默认回调参数是 已处理的菜单，数组。

business\_product.js 工具使用实例：



3. storage.js

此模块导出与前台缓存localStorage相关的方法。



**getItem (key)** ：通过key值获取一个localStorage缓存

-- 返回与key值对应的缓存**字符串（默认）**

**getObject (key)** ：通过key值获取一个localStorage缓存对象

-- 返回与key值对应的缓存**对象**

**setObject (key,value)** ：通过key/value键值对的形式，存入一个localStorage缓存对象

-- 返回与key值对应的缓存**对象**

使用实例：

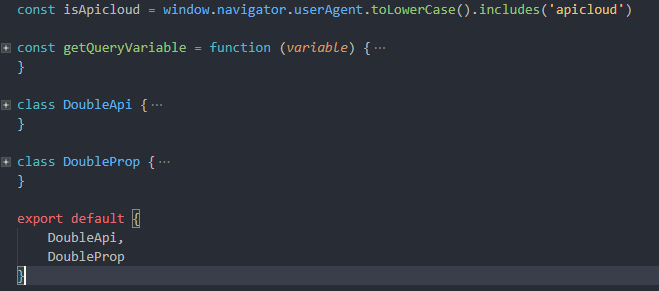
clipboard.png

4. static.js

此模块导出应用会用到的一些常量，便于后期维护、修改。

5. doubleApi.js

此模块导出两个js类（class），用于结合apicloud来进行h5开发。



**DoubleApi**

通过巧妙的方式，使得只能在apicloud上真机调试的应用可以在浏览器上进行开发和调试。

使用方式：

一般是main.js，在页面加载完成时，初始化doubleApi，传入apicloud原生接口对象——window.api，然后将其作为一个data对象的属性。



在用到window.api调用原生接口方法时，使用vue实例上的根对象的api代替：

clipboard.png

实际上，web端是没有api的原生接口的，我们调用的是使用js重写类似原生接口的实现方法。



所以当想要在web端用一个我们从来没有重写的api原生方法，会报错，这时就要考虑能不能通过js去重写它，如果不能，就使用一个空方法来代替，避免报错。

目前已经重写的方法大概包括常用的不到十个，后续有需要再继续添加。

**DoubleProp**

包含doubleApi用到的一些属性，或者常量，一般用来设置app整体配置，开发时能用到的地方不多。

