一．接口

1. 打开对话框统一采用View.open方法，方法和mini.open兼容；
2. 关闭对话框统一采用View.close方法。
3. 接口描述
4. View.open

* 通过链式调用的方式来注册相应的回调函数。回调函数包括：

load: 同mini.open的onload。

上下文（this）为子窗口window对象；

参数-无；

viewReady: 子窗口的View对象就绪时回调。

上下文（this）为子窗口window对象；

参数-子窗口View对象

close: 子窗口关闭时回调。

上下文（this）为子窗口window对象；

参数-result（参数说明见View.close）

1. View.close

* 关闭当前窗口；
* 可选参数-result：用于返回给父窗口的数据（内部会进行深拷贝，防止内存泄漏，所以不建议传递function及dom等会导致内存泄漏的对象）；
* 父窗口可在close回调函数中（第一个参数）拿到子窗口返回的数据（见View.open 的 close回调）；

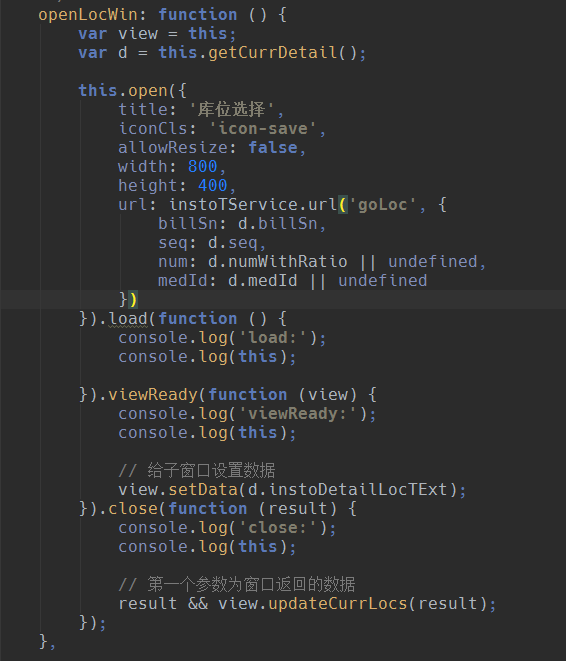
1. 注意：

* View.open接口兼容mini.oepn，也就是说支持原先的onload和ondestroy回调，但为了代码统一，不建议使用；
* 子窗口点击右上角的关闭按钮：相当于以无参方式调用View.close方法；

三．交互

1. 父窗口向子窗口传递数据：

* 简单类型可通过url查询参数方式：建议使用Service的url方法，将参数以名值对方式传入方法第二个参数；子窗口可通过View.getSearchParameter方法获取参数；
* 复杂类型：在View.open的load回调和viewReady回调里将数据主动设置到子窗口（子窗口的window或View对象需提供相应的接口）。load回调只能拿到子窗口的window对象（this），viewReady回调可拿到子窗口的window对象（this）和View对象（第一个参数）；
* 代码示例：



1. 子窗口向父窗口返回数据：

* 参见“接口描述”

1. 说明

* load 和 viewReady 的执行顺序：

load 在 viewReady之前执行，多个load和viewReady 按照各自的注册顺序执行；

* 关于load回调函数，目前有做特殊处理：子窗口的加载完成事件被延迟到了view对象准备好之后，所以在load里目前可以获取到准备好的view对象。**但是，不应该依赖这个事实（因为后续可能去除这个特殊处理）**，load应该被理解成类似DOMContentLoaded事件；