uexWindow

窗口对象。窗口对象是实现多窗口机制的基本单元。使用窗口对象构建 Hybrid App，在性能方面更加接近 Native App 的体验。

AppCan 平台中，维护了一个窗口堆栈，每个窗口以唯一的窗口名字来区别。窗口名字是通过 uexWindow.open 来赋值的。然而有一个窗口是比较特殊的，即加载起始页的窗口，因为起始页是由 config.xml 中配置的，无法进行窗口名的赋值，因此，AppCan 对加载起始页的窗口统一命名为'root'，也就是说，其它通过 uexWindow.open 开启的窗口，不能再命名为'root'。

window.uexOnload 方法

窗口加载完毕后平台将触发此方法。类比 window.onload 方法，都是 html 页面加载完成之后触发的方法。区别是，window.uexOnload 方法会晚于 window.onload 方法，原因是 window.uexOnload 需要等待 AppCan 扩展对象，即'uex'前缀的对象初始化完毕。事件加载完成之后，可以安全的使用 uex 扩展对象。

浮动窗口

主窗口之上可以有多个浮动窗口，即浮动窗口是附属于某个主窗口的。主窗口关闭后，其上所有浮动窗口也都会关闭。所有的窗口都有唯一的名字，通过 uexWindow.open 打开的是一个主窗口，浮动窗口则通过 uexWindow.openPopover 创建。一个主窗口上的多个浮动窗口名字是唯一的，但不同主窗口上的浮动窗口名字可以相同。浮动窗口可以有弹动效果，可以有数学变化：放大，旋转，移动等。浮动窗口能够解决的事情很多，比如解决手机浏览器不支持局部 DIV 滚动, 上下拉刷新特效，抽屉特效等问题。

多窗口之间的通讯

窗口之间的通讯，比如从网络获取一个数据，根据返回的数据，让其它窗口执行相应的变化，这就需要用到窗口间通讯机制：

主窗口之间通讯:

uexWindow.evaluateScript(winName, type, script)

主窗口与浮动窗口之间通讯:

uexWindow.evaluateScript(winName, type, script)

uexWindow.evaluatePopoverScript(winName, type, script)

最后一个参数 script，是目标窗口的执行脚本。脚本形参限定为数字和字符串。如果是特殊字符和汉字，则无法传递，可以通过 window.localStorage 暂存数据，在执行脚本的目标窗口中获取 localStorage。

方法

open 打开窗口

close 关闭窗口

forward 前进到下一个页面

back 返回上一个页面

pageForward 前进到下一个页面

pageBack 返回上一个页面

windowForward 前进到下一个窗口

windowBack 返回上一个窗口

showSoftKeyboard 弹出软键盘

alert 弹出 alert 对话框

confirm 弹出 confirm 对话框

prompt 弹出 prompt 对话框
actionSheet 弹出菜单列表
setReportKey 设置当前页面是否拦截某个按键
setWindowFrame 移动当前窗口位置
openSlibing 打开一个兄弟窗口
closeSlibing 关闭一个兄弟窗口
showSlibing 显示兄弟窗口
evaluateScript 执行 js 脚本
evaluatePopoverScript 在浮动窗口中执行 js 脚本
setSwipeRate 设置左右手势的灵敏度
loadObfuscationData 加载加密网页
toast 弹出消息提示框
closeToast 关闭消息提示框
openPopover 打开浮动窗口
closePopover 关闭浮动窗口
preOpenStart 开始浮动窗口的预加载
preOpenFinish 结束浮动窗口的预加载
setPopoverFrame 更改浮动窗口的位置和大小
openMultiPopover 打开多页面浮动窗口
closeMultiPopover 关闭多页面浮动窗口
setSelectedPopOverInMultiWindow 设置多页面浮动窗口跳转到的子页面窗口的索引
bringToFront 置顶当前浮动窗口
sendToBack 置底当前浮动窗口
insertAbove 将当前浮动窗口插入到指定浮动窗口之上
insertBelow 将当前浮动窗口插入到指定浮动窗口之下
bringPopoverToFront 置顶指定浮动窗口
sendPopoverToBack 置底指定浮动窗口
insertPopoverAbovePopover 将指定浮动窗口插入到另一浮动窗口之上
insertPopoverBelowPopover 将指定浮动窗口插入到另一浮动窗口之下
insertWindowAboveWindow 将指定窗口插入到另一窗口之上
insertWindowBelowWindow 将指定窗口插入到另一窗口之下
setWindowHidden 设置当前窗口显示和隐藏
setOrientation 设置屏幕方向
setWindowScrollbarVisible 设置滚动条的显示和隐藏
createProgressDialog 创建全局对话框
destroyProgressDialog 销毁全局对话框
postGlobalNotification 发送全局消息
subscribeChannelNotification 注册接收全局消息的监听器
publishChannelNotification 发送全局消息
openAd 打开广告窗口
getState 获取当前窗口处于前台还是后台
statusBarNotification 发送消息到状态栏
beginAnimition 开始设置动画的相关参数
setAnimitionDelay 设置动画延迟执行时间

setAnimitionDuration 设置动画持续时间
setAnimitionCurve 设置动画曲线类型
setAnimitionRepeatCount 设置动画重复次数
setAnimitionAutoReverse 设置动画结束后自动恢复位置和状态
makeTranslation 设置移动动画
makeScale 设置伸缩动画
makeRotate 设置旋转动画
makeAlpha 设置透明度动画
commitAnimition 提交动画设置并开始执行动画
setBounce 设置是否支持网页弹动
notifyBounceEvent 注册接收弹动事件
showBounceView 显示弹动效果
resetBounceView 设置弹动效果结束后显示的网页
setBounceParams 设置弹动参数
hiddenBounceView 隐藏弹动效果
getUrlQuery 获取加载页面时传入的参数

回调方法

cbConfirm 弹出 confirm 对话框的回调方法
cbPrompt 弹出 prompt 对话框的回调方法
cbActionSheet 弹出菜单列表的回调方法
cbGetState 获取窗口是否处于前台的回调方法
cbPageBack 返回到上一个页面的回调方法
cbPageForward 前进到下一个页面的回调方法
cbOpenMultiPopover 打开多页面浮动窗口的回调方法
cbGetUrlQuery 获取参数时的回调方法
uexOnload 网页加载完成时的回调方法

监听方法

onSlipedUpward 上滑的监听方法
onSlipedDownward 下滑的监听方法
onSlipedUpEdge 滑到顶部的监听方法
onSlipedDownEdge 滑到底部的监听方法
onAnimationFinish 动画执行完成的监听方法
onSetWindowFrameFinish 窗口位置移动完成的监听方法
onSwipeRight 向右滑动的监听方法
onSwipeLeft 向左滑动的监听方法
onBounceStateChange 弹动状态改变的监听方法
onGlobalNotification 全局消息的监听方法
onKeyPressed 按键事件的监听方法
onOAuthInfo 授权验证窗口 url 变化的监听方法
onStateChange 窗口前后台状态变化的监听方法

open

打开窗口
uexWindow.open(windName, dataType, data, animID, w, h, flag, animDuration)
说明：
打开一个新窗口，如果窗口名字相同，则会覆盖相同窗口名字的页面内容。
参数：
windName: (String 类型)必选 窗口名字，可为空，不能为"root"，若已经打开过该名字的窗口，则直接跳转至该窗口。
dataType: (Number 类型)必选 窗口载入的数据的类型，0：url 方式载入；1：html 内容方式载入；2：既有 url 方式，又有 html 内容方式
data: (String 类型)必选 url 或 html 数据，支持“wgtroot://”协议头，此协议头用于某些将项目部署在服务器上的 appcan 应用，在应用执行过程中加载本地网页用。当 dataType 为 0 时，url 支持相对路径、绝对路径。其中，当 url 以“wgtroot://”协议开头时，支持从服务器网页中打开本地应用沙箱中相应 widget 目录下的网页文件。例如：当前窗口加载的是服务器上的 http://www.xxx.com/xxx.html 网页，如果在 xxx.html 页面中 open 一个窗口时，传入的 data 为“wgtroot://index.html”，那么本次 open 执行时，引擎将会到本应用沙箱目录的 widget 路径下去寻找此页面，例如 Android 上找到的路径会是：file:///android_assert/widget/index.html。
animID: (Number 类型)必选 动画 ID，详见 CONSTANT 中 Window Animi ID
w: (Number 类型)必选 窗口宽度，支持百分数，默认为屏幕宽度
h: (Number 类型)必选 窗口高度，支持百分数，默认为屏幕高度
flag: (Number 类型)必选 窗口标记，详见 CONSTANT 中 Window Flags
animDuration: (Number 类型)可选 动画持续时长，单位为毫秒，默认为 260 毫秒
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+
示例：

```
<!DOCTYPE html>
<html>
  <head>
    <title>
    </title>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script type="text/javascript">
    function openn(){
        uexWindow.open('dd', '0', 'index.html', 1, '', '', 0, 500);
```

```
    }
    function opennn(){
        uexWindow.open('ddd', '0', 'xx.html', 1, '', '', 2);
    }
    function openh(arg){
        uexWindow.open('exc', '0', 'hidden.html', '5', '', '', arg);
    }
    function closew(){
        uexWindow.close(-1, 1000);
    }
</script>
  </head>
<body class="um-vp" ontouchstart>
    <div class="conbor">
        <div class="consj">
            <span>打开一个新窗口</span>
            <input class="btn"  type="button" value="打开新窗口" onclick=openn()>
            <input class="btn"  type="button" value="打开新窗口-加密的网页" onclick=opennn()>
            <input class="btn"  type="button" value="打开隐藏窗口" onclick=openh(32)><br>
            <input class="btn"  type="button" value="将隐藏窗口打开到屏幕" onclick=openh(0)>
            <input class="btn"  type="button" value="关闭" onclick=closew()>
        </div>
    </div>
</body>
</html>
```

close

关闭窗口
uexWindow.close(animID, animDuration)
说明：
　　关闭当前窗口，若为浮动窗口直接关闭，若为主窗口，则同时会关闭在其上打开的所有浮动窗口
参数：
　　animID: (Number 类型)可选 动画 ID，为空时无动画，-1 时代表 Open 时指定动画的方向动画。
　　animDuration: (Number 类型)可选 动画持续时长，单位为毫秒，默认为 260 毫秒
平台支持：
　　Android 2.2+

iOS 4.3+
版本支持：
3.0.0+
示例：
见 open 示例

forward

前进到下一个页面
uexWindow.forward()
说明：
仅在主窗口起作用，针对通过 a 标签跳转的网页，支持加密网页。
参数：
无
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+

back

返回上一个页面
uexWindow.back()
说明：
仅在主窗口起作用，针对通过 a 标签跳转的网页，支持加密网页。
参数：
无
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+

pageForward

前进到下一个页面
uexWindow.pageForward()
说明：
在主窗口和浮动窗口中起作用，针对通过 a 标签跳转的网页，不支持加密网页。

参数：
无
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+

pageBack

返回上一个页面
uexWindow.pageBack()
说明：
在主窗口和浮动窗口中起作用，针对通过 a 标签跳转的网页，不支持加密网页。
参数：
无
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+

windowForward

前进到下一个窗口
uexWindow.windowForward(animID, animDuration)
说明：
前进到下一个窗口
参数：
animID: (Number 类型)可选 动画 ID，详见 CONSTANT 中 Window Animi ID
animDuration: (Number 类型)可选 动画持续时长，单位为毫秒，默认为 260 毫秒
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+

windowBack

返回上一个窗口

uexWindow.windowBack(animID, animDuration)
说明：
返回上一个窗口
参数：
animID: (Number 类型)可选 动画 ID，详见 CONSTANT 中 Window Animi ID
animDuration: (Number 类型)可选 动画持续时长，单位为毫秒，默认为 260 毫秒
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+

showSoftKeyboard

弹出软键盘
uexWindow.showSoftKeyboard()
说明：
弹出 Android 设备软键盘
参数：
无
平台支持：
Android 2.2+
版本支持：
3.0.0+

alert

弹出 alert 对话框
uexWindow.alert(title, message, buttonLable)
说明：
弹出只有一个确定按钮的对话框
参数：
title: (String 类型)必选 标题
message: (String 类型)必选 内容
buttonLable: (String 类型)必选 显示在按钮上的文字
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+

示例：

```
<!DOCTYPE html>
<html>
  <head>
    <title>
    </title>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width,
initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script type="text/javascript">
    function alertW(){
        uexWindow.alert("提示","alert 框测试","OK");
    }
</script>
  </head>
<body class="um-vp" ontouchstart>
    <div class="conbor">
        <div class="consj">
            <input class="btn"  type="button" value="alert 框" onclick=alertW()>
        </div>
    </div>
</body>
</html>
```

confirm

弹出 confirm 对话框

uexWindow.confirm(title, message, buttonLable)

说明：

弹出至少包含一个至多包含 3 个按钮的对话框

参数：

title: (String 类型)必选 标题

message: (String 类型)必选 内容

buttonLable: (Array 类型)必选 显示在按钮上的文字的集合

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

```
<!DOCTYPE html>
```

```
<html>
  <head>
    <title>
    </title>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width,
initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script type="text/javascript">
        function confirmW(){
            uexWindow.confirm("警告","确定退出么？ ","OK,Cancel");
        }
        function cbConfirm(opId, dataType, data){
            alert('选择了' + data);
        }
        window.uexOnload = function(type){
            uexWindow.cbConfirm = cbConfirm;
        }
    </script>
  </head>
<body class="um-vp" ontouchstart>
    <div class="conbor">
        <div class="consj">
            <input class="btn"  type="button" value="confirm 框" onclick=confirmW()>
        </div>
    </div>
</body>
</html>
```

prompt

弹出 prompt 对话框

uexWindow.prompt(title, message, defaultValue, buttonLable)

说明：

弹出包含两个按钮且带输入框的对话框

参数：

title: (String 类型)必选 标题

message: (String 类型)必选 内容

defaultValue: (String 类型)必选 输入框默认文字

buttonLable: (Array 类型)必选 显示在按钮上的文字的集合

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

```
<!DOCTYPE html>
<html>
  <head>
    <title>
    </title>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width,
initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script type="text/javascript">
        function promptW(){
            uexWindow.prompt("提示","请输入内容：","","OK,Cancel");
        }
        function cbPrompt(opId, dataType, data){
            alert(data);
        }
        window.uexOnload = function(type){
            uexWindow.cbPrompt = cbPrompt;
        }
    </script>
  </head>
<body class="um-vp" ontouchstart>
    <div class="conbor">
        <div class="consj">
            <input class="btn"  type="button" value="prompt 框" onclick=promptW()>
        </div>
    </div>
</body>
</html>
```

actionSheet

弹出菜单列表

uexWindow.actionSheet(title, cancel, buttonLables)

说明：

从界面底部弹出按钮列表

参数：

title: (String 类型)必选 标题

cancel: (String 类型)必选 显示在取消按钮上的文本

buttonLables: (Array 类型)必选 按钮列表文字

平台支持：

Android 2.2+
iOS 4.3+

版本支持：

3.0.0+

示例：

```
<!DOCTYPE html>
<html>
  <head>
    <title>
    </title>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script type="text/javascript">
        function actionSheet(){
            uexWindow.actionSheet("菜单","Cancel","Opt1,Opt2,Opt3,Opt4,Opt5,Opt6");
        }
        function cbActionSheet(opId, dataType, data){
            alert(data);
        }
        window.uexOnload = function(type){
            uexWindow.cbActionSheet = cbActionSheet;
        }
    </script>
  </head>
<body class="um-vp" ontouchstart>
    <div class="conbor">
        <div class="consj">
            <input class="btn"  type="button" value="actionSheet" onclick=actionSheet()>
        </div>
    </div>
</body>
</html>
```

setReportKey

设置当前页面是否拦截某个按键

uexWindow.setReportKey(keyCode, enable)

说明：

设置当前页面是否拦截某个按键

参数：

keyCode: (Number 类型)必选 要拦截的键值,0-返回键，1-菜单键

enable: (Number 类型)必选 是否拦截,0-不拦截，1-拦截

平台支持：

Android 2.2+

版本支持：

3.0.0+

示例：

```
<!DOCTYPE html>
<html>
  <head>
    <title>
    </title>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width,
initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script type="text/javascript">
        window.uexOnload = function(type){
            uexWindow.onKeyPressed = onKeyPressed;
        }
        function onKeyPressed(keyCode){
            if(keyCode == 0){
                alert('点击了返回键');
            }else if(keyCode == 1){
                alert('点击了菜单键');
            }
        }
    </script>
  </head>
<body class="um-vp" ontouchstart>
    <div class="conbor">
        <div class="consj">
            <input class="btn"   type="button" value="拦截返回键"
onclick=uexWindow.setReportKey(0,1)>
            <input class="btn"   type="button" value="拦截菜单键"
onclick=uexWindow.setReportKey(1,1)>
        </div>
    </div>
</body>
</html>
```

setWindowFrame

移动当前窗口位置

uexWindow.setWindowFrame(x, y, animDuration)
说明：
移动当前 Window 相对屏幕的位置
参数：
x: (Number 类型)必选 x 坐标
y: (String 类型)必选 y 坐标
animDuration: (String 类型)必选 动画持续时长，单位为毫秒，默认为 260 毫秒
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+
示例：

```
<!DOCTYPE html>
<html>
  <head>
    <title>
    </title>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width,
initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script type="text/javascript">
        window.uexOnload = function(type){
            uexWindow.onSetWindowFrameFinish = onSetWindowFrameFinish;
        }
        function onSetWindowFrameFinish(){
            alert('移动完成！');
        }
    </script>
  </head>
<body class="um-vp" ontouchstart>
    <div class="conbor">
        <div class="consj">
            <input class="btn"  type="button" value="移动窗口"
onclick=uexWindow.setWindowFrame(200,200,1000)>
        </div>
    </div>
</body>
</html>
```

openSlibing

打开一个兄弟窗口
uexWindow.openSlibing(type, dataType, url, data, w, h)

说明：

打开一个兄弟窗口

参数：

type: (Number 类型)必选 窗口的类型，1-头部；2-底部

dataType: (String 类型)必选 窗口载入的数据的类型，0：url 方式载入；1：html 内容方式载入；2：既有 url 方式，又有 html 内容方式

url: (Number 类型)必选 窗口路径

data: (String 类型)必选 数据，可为空

w: (Number 类型)必选 窗口宽度，支持百分数，默认为屏幕宽度

h: (Number 类型)必选 窗口高度，支持百分数，默认为屏幕高度

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

```
<!DOCTYPE HTML>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1.0, user-scalable=no" />
<script type="text/javascript">
    window.uexOnload = function(slibing_type){
        switch (slibing_type) {
        case 0://main
            uexWindow.openSlibing("1", "0", "nav.html", "", "", "75");
            uexWindow.openSlibing("2", "0", "bar.html", "", "", "75");
            break;
        case 1://top
            uexWindow.showSlibing("1");
            break;
        case 2://bottom
            uexWindow.showSlibing("2");
            break;
        }
    }
</script>
<style>
    body{
        background:#27A73F;
    }
    a, .btc{
        font-size:1.5em;
```

```
            color:#fff;
            background:-webkit-gradient(linear, left top, left bottom, from(#a2a2a2), to(#010101), color-stop(0.5, #585858), color-stop(0.5, #2c2c2c));
             border:1px solid #000;
             -webkit-border-radius:5px;
             text-align:center;
        }
        div{
            text-align: center;
        }

        </style>
</head>

<body>
<br><br>
   <a href="javascript:void(0);" onclick=uexWindow.closeSlibing("1")> 关 闭 top</a><br><br><br>
   <a href="javascript:void(0);" onclick=uexWindow.showSlibing("1")> 显 示 top</a><br><br><br>
   <a href="javascript:void(0);" onclick=uexWindow.closeSlibing("2")> 关 闭 bottom</a><br><br><br>
   <a href="javascript:void(0);" onclick=uexWindow.showSlibing("2")> 显 示 bottom</a><br><br><br>
</body>
</html>
```

closeSlibing

关闭一个兄弟窗口
uexWindow.closeSlibing(type)
说明：
　　关闭一个兄弟窗口
参数：
　　type : (Number 类型)必选 窗口的类型，1-头部；2-底部
平台支持：
　　Android 2.2+
　　iOS 4.3+
版本支持：
　　3.0.0+
示例：
　　见 openSlibing 示例

showSlibing

显示兄弟窗口
uexWindow.showSlibing(type)
说明：
显示 open 过的兄弟窗口
参数：
type : (Number 类型)必选 窗口的类型，1-头部；2-底部
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+
示例：
见 openSlibing 示例

evaluateScript

执行 js 脚本
uexWindow.evaluateScript(windName, type, js)
说明：
执行 js 脚本
参数：
windName: (String 类型)必选 窗口名称，默认为当前窗口
type: (Number 类型)必选 窗口的类型，uex.cWindowTypeNormal，uex.cWindowTypeTop 或 uex.cWindowTypeBottom，详见 CONSTANT 中 Window Types
js:(String 类型)必选 js 脚本内容
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+
示例：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script type="text/javascript">
```

```
            function evaluateScript(){
                uexWindow.evaluateScript("", 0, "alert('执行去吧！！');");
            }
        </script>
    </head>
<body class="um-vp" ontouchstart>
        <div class="conbor">
            <div id="slibing">
            </div>
            <div class="consj">
                <input class="btn"    type="button" value="执行脚本" onclick=evaluateScript()>
            </div>
        </div>
</body>
</html>
```

evaluatePopoverScript

在浮动窗口中执行 js 脚本
uexWindow.evaluatePopoverScript(windName, popName, js)
说明：
在浮动窗口中执行 js 脚本
参数：
windName: (String 类型)必选 主窗口名称，默认为当前窗口
popName: (String 类型)必选 浮动窗口名称
js:(String 类型)必选 js 脚本内容
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+
示例：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <meta    name="viewport"    content="target-densitydpi=device-dpi,    width=device-width,
initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
        <script type="text/javascript">
            function evaluatePopoverScript(){
                uexWindow.evaluatePopoverScript("", "sss", "alert('执行去吧！！');");
```

```
            }
        </script>
    </head>
<body class="um-vp" ontouchstart>
        <div class="conbor">
            <div id="slibing">
            </div>
            <div class="consj">
                <input class="btn" type="button" value=" 执 行 脚 本 "
onclick=evaluatePopoverScript()>
            </div>
        </div>
</body>
</html>
```

setSwipeRate

设置左右手势的灵敏度
uexWindow.setSwipeRate(rate)
说明：
设置左右手势的灵敏度
参数：
rate: (Number 类型)必选 灵敏度，大于等于 1
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+

loadObfuscationData

加载加密网页
uexWindow.loadObfuscationData(url)
说明：
加载加密过的网页。网页如何加密，请下载【AppCan HTML 加密工具】。
参数：
url: (Number 类型)必选 网页路径
平台支持：
Android 2.2+
iOS 4.3+
版本支持：

3.0.0+

示例：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script type="text/javascript">
    </script>
  </head>
<body class="um-vp" ontouchstart>
    <div class="conbor">
        <div id="slibing">
        </div>
        <div class="consj">
            <input class="btn" type="button" value="打开加密网页" onclick="uexWindow.loadObfuscationData('xx.html')" /><br>
        </div>
    </div>
</body>
</html>
```

toast

弹出消息提示框

uexWindow.toast(type, location, msg, duration)

说明：

弹出消息提示框，常见的用于获取网络数据，在请求过程中给个加载提示，数据加载完成时关闭提示。

参数：

type: (Number 类型)必选 类型，0-没有进度条；1-有进度条

location: (Number 类型)必选 显示位置，详见 CONSTANT 中 Window Toast Location

msg: (Number 类型)必选 消息

duration: (Number 类型)必选 显示时间，非正整数时，提示框一直存在，不会自动关闭

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script type="text/javascript">
        window.uexOnload = function(slibing_type){
            uexWindow.toast(1,5,"正在加载...",0);
        }
    </script>
  </head>
<body class="um-vp" ontouchstart>
    <div class="conbor">
        <div class="consj">
            <input class="btn" type="button" value="close" onclick="uexWindow.closeToast()"/>
        </div>
    </div>
</body>
</html>
```

closeToast

关闭消息提示框
uexWindow.closeToast()
说明：
　　关闭消息提示框
参数：
　　无
平台支持：
　　Android 2.2+
　　iOS 4.3+
版本支持：
　　3.0.0+
示例：
　　见 toast 示例

openPopover

打开浮动窗口
uexWindow.openPopover(popName, dataType, url, data, x, y, w, h, fontSize, flag, bottomMargin)

说明：

打开浮动窗口，如果浮动窗口名字相同，则会覆盖相同浮动窗口名字的页面内容。

参数：

popName: (String 类型)必选 名称

dataType: (Number 类型)必选 窗口载入的数据的类型，0：url 方式载入；1：html 内容方式载入；2：既有 url 方式，又有 html 内容方式

url: (String 类型)必选 url 类型数据，支持“wgtroot://”协议头，此协议头用于某些将项目部署在服务器上的 appcan 应用，在应用执行过程中加载本地网页用。当 dataType 为 0 时，url 支持相对路径、绝对路径。其中，当 url 以“wgtroot://”协议开头时，支持从服务器网页中打开本地应用沙箱中相应 widget 目录下的网页文件。例如：当前窗口加载的是服务器上的 http://www.xxx.com/xxx.html 网页，如果在 xxx.html 页面中 open 一个窗口时，传入的 data 为“wgtroot://index.html”，那么本次 open 执行时，引擎将会到本应用沙箱目录的 widget 路径下去寻找此页面，例如 Android 上找到的路径会是：file:///android_assert/widget/index.html。

data: (String 类型)必选 data 类型数据

x: (Number 类型)必选 x 坐标

y: (Number 类型)必选 y 坐标

w: (Number 类型)必选 宽度，为空或为 0 时默认为 window 的宽度

h: (Number 类型)必选 高度，为空或为 0 时默认为 window 的高度

fontSize: (Number 类型)必选 字体大小

flag: (Number 类型)必选 浮动窗口标记，详见 CONSTANT 中 Window Flags

bottomMargin: (Number 类型)可选 浮动窗口相对父窗口底部的距离。为空或 0 时，默认为 0。当值不等于 0 时，inHeight 参数无效。

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script type="text/javascript">
        window.uexOnload = function(slibing_type){
        }
        function openPop(){
            uexWindow.openPopover("sss",0,"hidden.html","",400,0,",",",0,100);
        }
        function setPopoverFrame(){
            uexWindow.setPopoverFrame('sss',500,200,400,400);
        }
```

```
    </script>
  </head>
<body class="um-vp" ontouchstart>
    <div class="conbor">
        <div class="consj">
            <input class="btn" type="button" value="open" onclick="openPop()"/>
            <input class="btn" type="button" value="setPopFrame" onclick="setPopoverFrame()"/>
            <input class="btn" type="button" value="close" onclick="uexWindow.closePopover('sss')"/>
        </div>
    </div>
</body>
</html>
```

closePopover

关闭浮动窗口
uexWindow.closePopover(popName)
说明：
　　关闭浮动窗口
参数：
　　popName: (String 类型)必选 名称
平台支持：
　　Android 2.2+
　　iOS 4.3+
版本支持：
　　3.0.0+
示例：
　　见 openPopover 示例

preOpenStart

开始浮动窗口的预加载
uexWindow.preOpenStart()
说明：
　　popOver 的预加载必须要与 uexWindow.open 中的 flag：uex.cWiondowFlagPreOpen = 64 配合使用，即 open 时有此 flag 方可对应使用预加载。 开始 popOver(浮动窗口)的预加载。即一个窗口中需要有多个浮动窗口，可以让这些浮动窗口预先加载出来。其执行过程：A 窗

口打开 B 窗口，B 窗口中需要预加载多个浮动窗口。那么 A 窗口中执行 uexWindow.open 时，其 flag 参数需要：uex.cWiondowFlagPreOpen = 64 配合使用，即 open 时有此 flag，B 窗口方可使用预加载。 此时在 B 窗口中，会等所有预加载的浮动窗口都加载完毕（不包括异步获取网络数据），方才显示 B 窗口。预加载的浮动窗口的开启函数，即 uexWindow.openPopover，需要放置于 uexWindow.preOpenStart 和 uexWindow.preOpenFinish 之间。

参数：

无

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

在 A 窗口中，首先 open 窗口 B，在 B 窗口中，预加载多个浮动窗口：

A.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>AppCan API uexWindow A</title>
    <meta charset="utf-8">
    <script>
      window.uexOnload = function(type){
        if(!type){
          uexWindow.open("B",0,"B.html",0,"","",64);
        }
      }
    </script>
  </head>
  <body>
  </body>
</html>
```

B.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>AppCan API uexWindow B</title>
    <meta charset="utf-8">
  </head>
  <body>
  </body>
  <script>
    window.uexOnload = function(type){
      if(!type){
        var s = window.getComputedStyle($$("content"), null);
```

```
            uexWindow.preOpenStart();
uexWindow.openPopover("B1","0","B1.html","",0,int($$("header").offsetHeight),int(s.width),int(
s.height),int(s.fontSize),"0");
uexWindow.openPopover("B2","0","B2.html","",0,int($$("header").offsetHeight),int(s.width),int(
s.height),int(s.fontSize),"0");
            uexWindow.preOpenFinish();
        }
    }
  </script>
</html>
```

preOpenFinish

结束浮动窗口的预加载
uexWindow.preOpenFinish()
说明：
结束浮动窗口的预加载
参数：
无
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+
示例：
见 preOpenStart 示例

setPopoverFrame

更改浮动窗口的位置和大小
uexWindow.setPopoverFrame(popName, x, y, w, h)
说明：
更改浮动窗口的位置和大小
参数：
popName: (Number 类型)必选 名称
x: (Number 类型)必选 x 坐标
y: (Number 类型)必选 y 坐标
w: (Number 类型)必选 宽度，为空或为 0 时默认为 window 的宽度
h: (Number 类型)必选 高度，为空或为 0 时默认为 window 的高度
平台支持：
Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见 openPopover 示例

openMultiPopover

打开多页面浮动窗口

uexWindow.openMultiPopover(content, popName, dataType, x, y, w, h, fontSize, flag, indexSelected)

说明：

在当前 window 打开一个多页面浮动窗口, 页面之间滑动切换

参数：

content: (String 类型)必选 多页面窗口数据格式为 json。不可为空。json 格式如下：

'{"content":[{"inPageName":"p1", "inUrl":"xxx1.html","inData":""},

{"inPageName":"p2", "inUrl":"xxx2.html","inData":""}]}'

各字段含义如下:

参数	是否必须	说明
inPageName	是	所包含的单页面窗口的名字
inUrl	是	url 类型数据
inData	是	窗口的内容的二进制数据，可为空

popName: (String 类型)必选 浮动窗口名称

dataType: (Number 类型)必选 窗口载入的数据的类型，0：url 方式载入；1：html 内容方式载入；2：既有 url 方式，又有 html 内容方式

x: (Number 类型)必选 x 坐标

y: (Number 类型)必选 y 坐标

w: (Number 类型)必选 宽度，为空或为 0 时默认为 window 的宽度

h: (Number 类型)必选 高度，为空或为 0 时默认为 window 的高度

fontSize: (Number 类型)必选 字体大小

flag: (Number 类型)必选 浮动窗口标记，详见 CONSTANT 中 Window Flags

indexSelected: (Number 类型)必选 默认打开的页面索引，默认为 0

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
```

```
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script type="text/javascript">
        window.uexOnload = function(slibing_type){
        }
        function openMultiPop(){
            uexWindow.openMultiPopover('{"content":[{"inPageName":"p1", "inUrl":"hidden.html","inData":""},{"inPageName":"p2", "inUrl":"hidden1.html","inData":""}]}',"sss",0,400,0,",",",0,1);
        }
        function setIndex(){
            uexWindow.setSelectedPopOverInMultiWindow('sss',0);
        }
    </script>
  </head>
<body class="um-vp" ontouchstart>
    <div class="conbor">
        <div class="consj">
            <input class="btn" type="button" value="open" onclick="openMultiPop()"/>
            <input class="btn" type="button" value="setIndex" onclick="setIndex()"/>
            <input class="btn" type="button" value="close" onclick="uexWindow.closeMultiPopover('sss')"/>
        </div>
    </div>
</body>
</html>
```

closeMultiPopover

关闭多页面浮动窗口

uexWindow.closeMultiPopover(popName)

说明：

关闭多页面浮动窗口

参数：

popName: (Number 类型)必选 名称

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见 openMultiPopover 示例

setSelectedPopOverInMultiWindow

设置多页面浮动窗口跳转到的子页面窗口的索引
uexWindow.setSelectedPopOverInMultiWindow(popName, indexPage)
说明：
设置多页面浮动窗口跳转到的子页面窗口的索引
参数：
popName: (String 类型)必选 浮动窗口名称
indexPage: (Number 类型)必选 索引
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+
示例：
见 openMultiPopover 示例

bringToFront

置顶当前浮动窗口
uexWindow.bringToFront ()
说明：
置顶当前浮动窗口
参数：
无
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+
示例：
主窗口 A 中打开浮动窗口 B1,B2,B3，代码如下：
A.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script type="text/javascript">
```

```
            window.uexOnload = function(type){
                uexWindow.openPopover("pop1",0,"B1.html","",400,0,",",",0,0);
                uexWindow.openPopover("pop2",0,"B2.html","",400,100,",",",0,0);
                uexWindow.openPopover("pop3",0,"B3.html","",400,200,",",",0,0);
            }
            function bringPopoverToFront(){
                uexWindow.bringPopoverToFront("pop1");
            }
            function sendPopoverToBack(){
                uexWindow.sendPopoverToBack("pop1");
            }
            function insertPopoverAbovePopover(){
                uexWindow.bringPopoverToFront("pop1","pop3");
            }
            function insertPopoverBelowPopover(){
                uexWindow.insertPopoverBelowPopover("pop1","pop3");
            }
        </script>
    </head>
<body class="um-vp" ontouchstart>
        <div class="conbor">
            <div class="consj">
                <input class="btn" type="button" value="置顶窗口"
onclick="bringPopoverToFront()"/>
                <input class="btn" type="button" value="置底窗口"
onclick="sendPopoverToBack()"/>
                <input class="btn" type="button" value="插入之上"
onclick="insertPopoverAbovePopover()"/>
                <input class="btn" type="button" value="插入之下"
onclick="insertPopoverBelowPopover()"/>
            </div>
        </div>
</body>
</html>
```

B1.html

```
<!DOCTYPE HTML>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="viewport" content="target-densitydpi=device-dpi, width=device-width,
initial-scale=1.0, user-scalable=no" />
<link rel="stylesheet" type="text/css" href="css/index.css">
        <script type="text/javascript">
```

```
            window.uexOnload = function(type){
                uexWindow.bringToFront();
            }
        </script>
</head>
<body style="background:#077333;">
        <div class="tit">B1</div>
                <div class="consj">
                <span class="tit"></span>
                    <input class="btn" type="button" value="置底当前浮动窗口"
onclick="uexWindow.sendToBack();">
                     <input class="btn" type="button" value="插入之上"
onclick="uexWindow.insertAbove('pop2');">
                    <input class="btn" type="button" value="插入之下"
onclick="uexWindow.insertBelow('pop2');">
            </div>
</body>
</html>
```

sendToBack

置底当前浮动窗口
uexWindow.sendToBack()
说明：
　　置底当前浮动窗口
参数：
　　无
平台支持：
　　Android 2.2+
　　iOS 4.3+
版本支持：
　　3.0.0+
示例：
　　见 bringToFront 示例

insertAbove

将当前浮动窗口插入到指定浮动窗口之上
uexWindow.insertAbove(name)
说明：
　　将当前浮动窗口插入到指定浮动窗口之上
参数：
　　name: (String 类型)必选 目标浮动窗口的名称

平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+
示例：
见 bringToFront 示例

insertBelow

将当前浮动窗口插入到指定浮动窗口之下
uexWindow.insertBelow(name)
说明：
将当前浮动窗口插入到指定浮动窗口之下
参数：
name: (String 类型)必选 目标浮动窗口的名称
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+
示例：
见 bringToFront 示例

bringPopoverToFront

置顶指定浮动窗口
uexWindow.bringPopoverToFront(name)
说明：
置顶指定浮动窗口，只在主窗口中有效
参数：
name: (String 类型)必选 指定浮动窗口的名称
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+
示例：
见 bringToFront 示例

sendPopoverToBack

置底指定浮动窗口
uexWindow.sendPopoverToBack(name)
说明：
　　置底指定浮动窗口，只在主窗口中有效
参数：
　　name: (String 类型)必选 指定浮动窗口的名称
平台支持：
　　Android 2.2+
　　iOS 4.3+
版本支持：
　　3.0.0+
示例：
　　见 bringToFront 示例

insertPopoverAbovePopover

将指定浮动窗口插入到另一浮动窗口之上
uexWindow.insertPopoverAbovePopover(nameA, nameB)
说明：
　　将浮动窗口 A 插入到浮动窗口 B 之上，只在主窗口中有效
参数：
　　nameA: (String 类型)必选 指定浮动窗口 A 的名称
　　nameB: (String 类型)必选 指定浮动窗口 B 的名称
平台支持：
　　Android 2.2+
　　iOS 4.3+
版本支持：
　　3.0.0+
示例：
　　见 bringToFront 示例

insertPopoverBelowPopover

将指定浮动窗口插入到另一浮动窗口之下
uexWindow.insertPopoverBelowPopover(nameA, nameB)
说明：
　　将浮动窗口 A 插入到浮动窗口 B 之下，只在主窗口中有效
参数：
　　nameA: (String 类型)必选 指定浮动窗口 A 的名称

nameB: (String 类型)必选 指定浮动窗口 B 的名称

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见 bringToFront 示例

insertWindowAboveWindow

将指定窗口插入到另一窗口之上

uexWindow.insertWindowAboveWindow(nameA, nameB)

说明：

将指定窗口 A 插入到另一窗口 B 之上，该接口仅对显示在屏幕上且不被隐藏的 window 起作用。（即 open 该 window 时，flag 传入的是 256）

参数：

nameA: (String 类型)必选 指定窗口 A 的名称

nameB: (String 类型)必选 指定窗口 B 的名称

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

insertWindowBelowWindow

将指定窗口插入到另一窗口之下

uexWindow.insertWindowBelowWindow(nameA, nameB)

说明：

将指定窗口 A 插入到另一窗口 B 之下，该接口仅对显示在屏幕上且不被隐藏的 window 起作用。（即 open 该 window 时，flag 传入的是 256）

参数：

nameA: (String 类型)必选 指定窗口 A 的名称

nameB: (String 类型)必选 指定窗口 B 的名称

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

setWindowHidden

设置当前窗口显示和隐藏
uexWindow.setWindowHidden(visible)
说明：
　　设置当前窗口显示和隐藏，该接口仅对显示在屏幕上且不被隐藏的 window 起作用。（即 open 该 window 时，flag 传入的是 256）
参数：
　　visible: (Number 类型)必选　显示或隐藏，0-显示；1-隐藏
平台支持：
　　Android 2.2+
　　iOS 4.3+
版本支持：
　　3.0.0+

setOrientation(待定)

设置屏幕方向
uexWindow.setOrientation(orientation)
说明：
　　设置屏幕方向
参数：
　　orientation: (Number 类型)必选　屏幕方向
　　　　1：竖屏，home 键在屏幕下方；
　　　　2：横屏，home 键在屏幕右边；
　　　　4：竖屏，home 键在屏幕上方；
　　　　8：横屏，home 键在屏幕左边；
　　　　3：既支持 1 又支持 2；
　　　　15：随系统设置自动转屏。
平台支持：
　　Android 2.2+
　　iOS 4.3+
版本支持：
　　3.0.0+

setWindowScrollbarVisible

设置滚动条的显示和隐藏
uexWindow.setWindowScrollbarVisible(visible)

说明：

设置滚动条的显示和隐藏

参数：

visible: (Bool 类型)必选 显示或隐藏，true-显示；false-隐藏

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script type="text/javascript">
        window.uexOnload = function(type){
        }
        function setDis(vis){
            uexWindow.setWindowScrollbarVisible(vis);
        }
    </script>
  </head>
<body class="um-vp" ontouchstart>
    <div class="conbor">
        <div class="consj">
            <input class="btn" type="button" value="显示" onclick="setDis('true')"/>
            <input class="btn" type="button" value="隐藏" onclick="setDis('false')"/>
        </div>
    </div>
</body>
</html>
```

createProgressDialog

创建全局对话框

uexWindow.createProgressDialog(title, msg, canCancel)

说明：

创建一个全局对话框，屏蔽用户对界面的一切操作。常见的用于获取网络数据，在请求过程中给个加载提示，数据加载完成时关闭提示。

参数：

title: (String 类型)必选 标题

msg: (String 类型)必选 内容

canCancel: (Number 类型)可选 是否可以取消，即点击屏幕上除对话框以外的任何地方，或者点击返回键，对话框是否消失。0-可以取消，1-不能取消。设置为 1 时，该对话框只能在通过调用 destroyProgressDialog 取消，否则会一直显示。默认可以取消

平台支持：

Android 2.2+

版本支持：

3.0.0+

示例：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script type="text/javascript">
        window.uexOnload = function(type){
        }
        function createD(){
            uexWindow.createProgressDialog('提示','正在加载...',0);
        }
    </script>
  </head>
<body class="um-vp" ontouchstart>
    <div class="conbor">
        <div class="consj">
            <input class="btn" type="button" value="创建" onclick="createD()"/>
        </div>
    </div>
</body>
</html>
```

destroyProgressDialog

销毁全局对话框

uexWindow.destroyProgressDialog()

说明：

销毁全局对话框

参数：

无

平台支持：

Android 2.2+

版本支持：

3.0.0+

postGlobalNotification

发送全局消息

uexWindow.postGlobalNotification(content)

说明：

发送全局消息，用于窗口之间的通信，调用该方法时，所有打开（通过调用 uexWindow 的 open 和 openPopover 方法）的窗口只要注册过 onGlobalNotification，都会被调用。

参数：

content : (String 类型)必选 发送的内容

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

这里一共涉及到 3 个网页，index.html，index1.html，index2.html，其中在 index2.html 中发送全局消息。代码如下：

index.html

```
<!DOCTYPE html>
<html class="um landscape min-width-240px min-width-320px min-width-480px min-width-768px min-width-1024px">
<head>
    <title></title>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script>
        function onGlobalNotification(ret){
            console.log("index:"+ret);
        }
        function openWindow(){
            uexWindow.open('index','0','index1.html','2','',',0);
            //uexWindow.openPopover('index',"0",'index1.html',"","","","","","","","0");
        }
    </script>
</head>
<body class="um-vp c-wh" ontouchstart>
```

```
    <div id="page_0" class="up ub ub-ver" tabindex="0">
        <!--header 开始-->
        <div id="header" class="uh c-org c-m1 t-wh ub">
            <h1 class="ut ub-f1 ulev0 ut-s tx-c">index.html</h1>
        </div>
        <!--header 结束-->
        <div>
            <input type="button" value="index1" onClick="openWindow()" style="line-height:2em;font-size:14px"/>
        </div>
    </div>
</body>
<script>
window.uexOnload = function(type){
    uexWindow.onGlobalNotification = onGlobalNotification;
    uexWindow.subscribeChannelNotification("1", "onGlobalNotification");
}
</script>
</html>
```

index1.html

```
<!DOCTYPE html>
<html class="um landscape min-width-240px min-width-320px min-width-480px min-width-768px min-width-1024px">
<head>
    <title></title>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script>
        function onGlobalNotification(ret){
            console.log("index1:"+ret);
        }
        function openWindow(){
            uexWindow.open('index1','0','index2.html','2','',0);
            //uexWindow.openPopover('index1',"0",'index2.html',"","","","","","","","0");
        }
    </script>
</head>
<body class="um-vp c-wh" ontouchstart>
    <div id="page_0" class="up ub ub-ver" tabindex="0">
        <!--header 开始-->
        <div id="header" class="uh c-org c-m1 t-wh ub">
            <h1 class="ut ub-f1 ulev0 ut-s tx-c">index1.html</h1>
```

```
            </div>
            <!--header 结束-->
            <div>
                <input type="button" value="index2" onClick="openWindow()" style="line-height:2em;font-size:14px"/>
            </div>
        </div>
</body>
<script>
window.uexOnload = function(type){
    uexWindow.onGlobalNotification = onGlobalNotification;
    uexWindow.subscribeChannelNotification("2", "onGlobalNotification");
}
</script>
</html>
```

index2.html

```
<!DOCTYPE html>
<html class="um landscape min-width-240px min-width-320px min-width-480px min-width-768px min-width-1024px">
<head>
    <title></title>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script>
        function onGlobalNotification(ret){
            console.log("index2:"+ret);
        }
    </script>
</head>
<body class="um-vp c-wh" ontouchstart>
    <div id="page_0" class="up ub ub-ver" tabindex="0">
        <!--header 开始-->
        <div id="header" class="uh c-org c-m1 t-wh ub">
            <h1 class="ut ub-f1 ulev0 ut-s tx-c">index2.html</h1>
        </div>
        <!--header 结束-->
    </div>
</body>
<script>
    window.uexOnload = function(type){
        uexWindow.onGlobalNotification = onGlobalNotification;
```

```
        uexWindow.subscribeChannelNotification("1", "onGlobalNotification");
        uexWindow.subscribeChannelNotification("2", "onGlobalNotification");
        uexWindow.postGlobalNotification("test just!");
        uexWindow.publishChannelNotification("1","channel 1 test just!");
        uexWindow.publishChannelNotification("2","channel 2 test just!");
    }
</script>
</html>
```

subscribeChannelNotification

注册接收全局消息的监听器
uexWindow.subscribeChannelNotification(channelId, functionName)
说明：
窗口之间的通信，可以通过发布/订阅模式来实现。 窗口调用此接口订阅频道监听， 当在另一窗口调用 publishChannelNotification 时，对应此频道的回调方法将被调用，并传入相应的参数。
参数：
channelId: (Number 类型)必选 频道唯一标识符
functionName: (String 类型)必选 回调方法名称
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+
示例：
见 postGlobalNotification 示例

publishChannelNotification

发送全局消息
uexWindow.publishChannelNotification(channelId, content)
说明：
发布消息通知，此频道的所有订阅者，将收到消息，回调函数将被调用，并传入相应的参数。
参数：
channelId: (Number 类型)必选 频道唯一标识符
content: (String 类型)必选 发送的内容
平台支持：
Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见 postGlobalNotification 示例

openAd

打开广告窗口

uexWindow.openAd(type, dTime, interval, flag)

说明：

打开广告窗口，一个 Window 只能有一个 Ad 窗口。

参数：

type: (Number 类型)必选 广告窗口的位置，0-上；1-中；2-下

dTime: (String 类型)必选 广告窗口的显示时长（单位为：秒），0：一直显示。

interval: (String 类型)必选 广告窗口的显示间隔（单位为：秒）

flag: (Number 类型)可选 窗口标记，详见 CONSTANT 中 Window Flags

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script type="text/javascript">
        window.uexOnload = function(type){
        }
        function openAd(des){
            uexWindow.openAd(des,5,5,0);
        }
    </script>
  </head>
<body class="um-vp" ontouchstart>
    <div class="conbor">
        <div class="consj">
            <input class="btn" type="button" value="创建（上）" onclick="openAd(0)"/>
            <input class="btn" type="button" value="创建（中）" onclick="openAd(1)"/>
```

```
            <input class="btn" type="button" value="创建（下）" onclick="openAd(2)"/>
        </div>
    </div>
</body>
</html>
```

getState

获取当前窗口处于前台还是后台
uexWindow.getState()
说明：
　　获取当前窗口处于前台还是后台
参数：
　　无
平台支持：
　　Android 2.2+
　　iOS 4.3+
版本支持：
　　3.0.0+
示例：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script type="text/javascript">
        window.uexOnload = function(type){
            uexWindow.onStateChange=function(state){
                console.log(state);
            }
            uexWindow.cbGetState=function(opId,datatype,data){
                console.log(data);
            }
            uexWindow.onSlipedUpward = function(){
                console.log('onSlipedUpward');
            }
            uexWindow.onSlipedDownward = function(){
                console.log('onSlipedDownward');
            }
            uexWindow.onSlipedUpEdge = function(){
                console.log('onSlipedUpEdge');
```

```
			}
			uexWindow.onSlipedDownEdge = function(){
				console.log('onSlipedDownEdge');
			}
			uexWindow.onSwipeRight = function(){
				console.log('onSwipeRight');
			}
			uexWindow.onSwipeLeft = function(){
				console.log('onSwipeLeft');
			}
		}
		function openAd(){
			uexWindow.open('sss',0,'hidden.html',0,'','',0);
		}
	</script>
  </head>
<body class="um-vp" ontouchstart>
	<div class="conbor">
		<div class="consj">
			<input class="btn" type="button" value="获取" onclick="uexWindow.getState()"/><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
			<input class="btn" type="button" value="打开" onclick="openAd()"/>

		</div>
	</div>
</body>
</html>
```

statusBarNotification

发送消息到状态栏

uexWindow.statusBarNotification(title, msg)

说明：

发送消息到状态栏

参数：

title: (String 类型)必选 标题

msg:(String 类型)必选 消息

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script type="text/javascript">
        window.uexOnload = function(type){
        }
        function send(){
            uexWindow.statusBarNotification('title','msg');
        }
    </script>
  </head>
<body class="um-vp" ontouchstart>
    <div class="conbor">
        <div class="consj">
            <input class="btn" type="button" value="发送" onclick="send()"/>
        </div>
    </div>
</body>
</html>
```

beginAnimition

开始设置动画的相关参数

uexWindow.beginAnimition()

说明：

开始设置动画的相关参数，仅对浮动窗口有效

参数：

无

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见 commitAnimition 示例

setAnimitionDelay

设置动画延迟执行时间
uexWindow.setAnimitionDelay(delay)
说明：
　　设置动画延迟执行时间，仅对浮动窗口有效
参数：
　　delay: (Number 类型)可选 延迟执行的时间(单位：毫秒)，默认为 0。
平台支持：
　　Android 2.2+
　　iOS 4.3+
版本支持：
　　3.0.0+
示例：
　　见 commitAnimition 示例

setAnimitionDuration

设置动画持续时间
uexWindow.setAnimitionDuration(duration)
说明：
　　设置动画持续时间，仅对浮动窗口有效
参数：
　　duration: (Number 类型)可选 持续时间(单位：毫秒)，默认为 260
平台支持：
　　Android 2.2+
　　iOS 4.3+
版本支持：
　　3.0.0+
示例：
　　见 commitAnimition 示例

setAnimitionCurve

设置动画曲线类型
uexWindow.setAnimitionCurve(curve)

说明：
设置动画曲线类型，仅对浮动窗口有效
参数：
curve: (Number 类型)可选 动画曲线类型，默认为 0。详见 CONSTANT 中 Window AnimCurveType
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+
示例：
见 commitAnimition 示例

setAnimitionRepeatCount

设置动画重复次数
uexWindow.setAnimitionRepeatCount(count)
说明：
设置动画重复次数，仅对浮动窗口有效
参数：
count: (Number 类型)可选 重复次数，默认为 0
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+
示例：
见 commitAnimition 示例

setAnimitionAutoReverse

设置动画结束后自动恢复位置和状态
uexWindow.setAnimitionAutoReverse(isReverse)
说明：
设置动画结束后自动恢复位置和状态，仅对浮动窗口有效
参数：
isReverse: (Number 类型)可选 是否恢复。0-不恢复；1-恢复。默认为 0
平台支持：
Android 2.2+
iOS 4.3+

版本支持：

3.0.0+

示例：

见 commitAnimition 示例

makeTranslation

设置移动动画

uexWindow.makeTranslation(toX, toY, toZ)

说明：

设置移动动画，仅对浮动窗口有效

参数：

toX: (Number 类型)必选 相对于当前位置的 x 轴方向上的平移距离，int 型整数，负数或正数。

toY: (Number 类型)必选 相对于当前位置的 y 轴方向上的平移距离，int 型整数，负数或正数。

toZ ：(Number 类型)必选 相对于当前位置的 z 轴方向上的平移距离，int 型整数，负数或正数。

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见 commitAnimition 示例

makeScale

设置伸缩动画

uexWindow.makeScale(toX, toY, toZ)

说明：

设置伸缩动画，仅对浮动窗口有效

参数：

toX: (Number 类型)必选 相对于当前大小的 x 轴方向上的放大倍率，大于 0 的 float 型数据。

toY: (Number 类型)必选 相对于当前大小的 y 轴方向上的放大倍率，大于 0 的 float 型数据。

toZ ：(Number 类型)必选 相对于当前大小的 z 轴方向上的放大倍率，大于 0 的 float 型数据。

平台支持：

Android 2.2+
iOS 4.3+
版本支持：
3.0.0+
示例：
见 commitAnimition 示例

makeRotate

设置旋转动画
uexWindow.makeRotate(degrees, toX, toY, toZ)
说明：
设置旋转动画，仅对浮动窗口有效
参数：
degrees: (Number 类型)必选 相对于当前角度的旋转度数
toX: (Number 类型)必选 是否绕 X 轴旋转。0 为 false，1 为 true。
toY: (Number 类型)必选 是否绕 Y 轴旋转。0 为 false，1 为 true。
toZ : (Number 类型)必选 是否绕 Z 轴旋转。0 为 false，1 为 true。
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+
示例：
见 commitAnimition 示例

makeAlpha

设置透明度动画
uexWindow.makeAlpha(alpha)
说明：
设置透明度动画，仅对浮动窗口有效
参数：
alpha: (Number 类型)必选 相对于当前 alpha 的值，0.0 到 1.0 的 float 型数据。
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+

示例：

见 commitAnimition 示例

commitAnimition

提交动画设置并开始执行动画

uexWindow.commitAnimition()

说明：

提交动画设置并开始执行动画，仅对浮动窗口有效，所有参数的设置仅一次有效，动画完了后将清除。

参数：

无

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

```
<!DOCTYPE HTML>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1.0, user-scalable=no" /><link rel="stylesheet" type="text/css" href="css/index.css">
<script type="text/javascript">
function onAnimationFinish() {
    uexWindow.alert("应用名称","动画完毕","ok");
}
function leftAnimition(){
    uexWindow.beginAnimition();
    uexWindow.setAnimitionDelay(200);
    uexWindow.setAnimitionDuration(4000);
    uexWindow.setAnimitionAutoReverse(1);
    uexWindow.setAnimitionRepeatCount(0)
    uexWindow.setAnimitionCurve(1);
    //uexWindow.makeScale(2,1,1);
    //uexWindow.makeTranslation(100,0,0);
    uexWindow.makeRotate(90, 1, 0, 1);
    //uexWindow.makeAlpha(0.5);
    uexWindow.commitAnimition();
    uexWindow.onAnimationFinish = onAnimationFinish;
}
</script>
```

```
</head>
<body style="background-color:#0F0">
    <input class="btn" type="button" value="动画" onclick="leftAnimition();">
</body>
</html>
```

setBounce

设置是否支持网页弹动
uexWindow.setBounce(flag)
说明：
    设置是否支持网页弹动
参数：
    flag: (Number 类型)必选 1：支持；0：不支持
平台支持：
    Android 2.2+
    iOS 4.3+
版本支持：
    3.0.0+
示例：
    见 showBounceView 示例

notifyBounceEvent

注册接收弹动事件
uexWindow.notifyBounceEvent(type, status)
说明：
    注册接收弹动事件
参数：
    type: (Number 类型)必选 弹动的位置，0：顶端弹动；1：底部弹动
    status: (Number 类型)必选 是否调用 onBounceStateChange 方法，0：不调用；1-调用
平台支持：
    Android 2.2+
    iOS 4.3+
版本支持：
    3.0.0+
示例：
    见 showBounceView 示例

showBounceView

显示弹动效果
uexWindow.showBounceView(type, color, flag)
说明：
显示弹动效果
参数：
type: (Number 类型)必选 弹动显示的部位，0：顶端；1：底部
color: (String 类型)必选 弹动显示部位的颜色值
flag: (String 类型)必选 是否显示内容，1：显示；0：不显示
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+
示例：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script type="text/javascript">
        window.uexOnload = function(type){
            uexWindow.setBounce(1);
            uexWindow.notifyBounceEvent(0,1);
            uexWindow.notifyBounceEvent(1,1);
            var json ='{"textColor":"#000","imagePath":"res://refesh_icon.png","levelText":"更新日期","pullToReloadText":"拖动到底部","releaseToReloadText":"释放回原处","loadingText":"更新中..."}';
            uexWindow.setBounceParams(0, json);
            uexWindow.showBounceView("0","#00F", 1);
            var json2 ='{"textColor":"#fff","imagePath":"res://refesh_icon.png","levelText":"修改日期","pullToReloadText":"上拉到底部","releaseToReloadText":"松手回原处","loadingText":"下载中..."}';
            uexWindow.setBounceParams(1, json2);
            uexWindow.showBounceView("1","rgba(15, 155, 155, 100)", 1);
            uexWindow.onBounceStateChange = onBounceStateChange;
        }
        function onBounceStateChange(type, state){
            switch(type) {
                case 0:
                    if (status == 2) {
```

```
                        uexWindow.resetBounceView("0");
                    }
                    break;
                case 1:
                    if (status == 2) {
                        uexWindow.resetBounceView("1");
                    }
                    break;
            }
        }
    </script>
  </head>
<body class="um-vp" ontouchstart>
    <div class="conbor">
        <div class="consj">
            <input class="btn" type="button" value="隐藏(上)"
onclick="uexWindow.hiddenBounceView(0)"/>
            <input class="btn" type="button" value="隐藏(下)"
onclick="uexWindow.hiddenBounceView(1)"/>
        </div>
    </div>
</body>
</html>
```

resetBounceView

设置弹动效果结束后显示的网页
uexWindow.resetBounceView(type)
说明：
　　设置弹动效果结束后显示的网页，一般在 onBounceStateChange 监听方法中调用该方法
参数：
　　type: (Number 类型)必选 弹动显示的部位，0：顶端；1：底部
平台支持：
　　Android 2.2+
　　iOS 4.3+
版本支持：
　　3.0.0+
示例：
　　见 showBounceView 示例

setBounceParams

设置弹动参数
uexWindow.setBounceParams(type, json)
说明：
设置弹动参数，用于自定义 view 样式
参数：
type: (Number 类型)必选 弹动的位置，0：顶端弹动；1：底部弹动
status: (String 类型)必选 参数，json 格式，json 各字段格式如下：

参数	是否必须	说明
imagePath	是	下拉状态小图标的路径，只支持 res://格式。路径协议详见 CONSTANT 中 Path types
textColor	是	展示下拉状态文字的颜色,如：″#ffffff″
levelText	是	显示的二级文字,如：“上次更新时间：xxxxx”。
pullToReloadText	是	开始拖动直到超过刷新临界线之前显示的文字,如：“拖动刷新”
releaseToReloadText	是	拖动超过刷新临界线后显示的文字,如：“释放刷新”
loadingText	是	拖动超过刷新临界线并且释放拖动，进入刷新状态时显示的文字,如：“加载中，请稍等”
loadingImagePath	否	等待状态 loading 小图标的路径，只支持 res://格式。（该字段为定制需求，默认无效）

平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+
示例：
见 showBounceView 示例

hiddenBounceView

隐藏弹动效果
uexWindow.hiddenBounceView(type)
说明：
隐藏弹动效果
参数：
type: (Number 类型)必选 弹动显示的部位，0：顶端；1：底部
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+
示例：

见 showBounceView 示例

getUrlQuery

获取加载页面时传入的参数
uexWindow.getUrlQuery()
说明：
获取加载页面时传入的参数
参数：
无
平台支持：
Android 2.2+
iOS 4.3+
版本支持：
3.0.0+
示例：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <script type="text/javascript">
        window.uexOnload = function(type){
            uexWindow.cbGetUrlQuery = cbGetUrlQuery;
        }
        function cbGetUrlQuery(opId, dataType, data){
            alert(data);
        }
    </script>
  </head>
<body class="um-vp" ontouchstart>
    <div class="conbor">
        <div class="consj">
            <input class="btn" type="button" value="获取" onclick="uexWindow.getUrlQuery()"/>
        </div>
    </div>
</body>
</html>
```

cbConfirm

弹出 confirm 对话框的回调方法

uexWindow.cbConfirm(opId, dataType, data)

参数：

opId:　(Number 类型)必选 操作 ID，在此函数中不起作用，可忽略

dataType:　(Number 类型)必选 参数类型 详见 CONSTANT 中 Callback 方法数据类型

data: (Number 类型)必选　返回的数据，用户点击的按钮索引。

版本支持：

3.0.0+

cbPrompt

弹出 prompt 对话框的回调方法

uexWindow.cbPrompt(opId, dataType, data)

参数：

opId:　(Number 类型)必选 操作 ID，在此函数中不起作用，可忽略

dataType:　(Number 类型)必选 参数类型 详见 CONSTANT 中 Callback 方法数据类型

data: (String 类型)必选　返回用户点击对话框上的按钮索引及输入框中的值,json 格式为 {"num":"0","value":"xxx"}

各字段含义如下:

参数	是否必须	说明
num	是	索引
value	是	输入框中的值

版本支持：

3.0.0+

cbActionSheet

弹出菜单列表的回调方法

uexWindow.cbActionSheet(opId, dataType, data)

参数：

opId:　(Number 类型)必选 操作 ID，在此函数中不起作用，可忽略

dataType:　(Number 类型)必选 参数类型 详见 CONSTANT 中 Callback 方法数据类型

data: (Number 类型)必选　返回的数据，用户点击的按钮索引。

版本支持：

3.0.0+

cbGetState

获取窗口是否处于前台的回调方法

uexWindow.cbGetState(opId, dataType, data)

参数：

opId: (Number 类型)必选 操作 ID，在此函数中不起作用，可忽略

dataType: (Number 类型)必选 参数类型 详见 CONSTANT 中 Callback 方法数据类型

data: (Number 类型)必选 返回的数据，0：前台；1：后台

版本支持：

3.0.0+

cbPageBack

返回到上一个页面的回调方法

uexWindow.cbPageBack(opId, dataType, data)

参数：

opId: (Number 类型)必选 操作 ID，在此函数中不起作用，可忽略

dataType: (Number 类型)必选 参数类型 详见 CONSTANT 中 Callback 方法数据类型

data: (Number 类型)必选 返回结果，0：成功；1：失败

版本支持：

3.0.0+

cbPageForward

前进到下一个页面的回调方法

uexWindow.cbPageForward(opId, dataType, data)

参数：

opId: (Number 类型)必选 操作 ID，在此函数中不起作用，可忽略

dataType: (Number 类型)必选 参数类型 详见 CONSTANT 中 Callback 方法数据类型

data: (Number 类型)必选 返回结果，0：成功；1：失败

版本支持：

3.0.0+

cbOpenMultiPopover

打开多页面浮动窗口的回调方法

uexWindow.cbOpenMultiPopover(opId, dataType, data)

参数：

opId: (Number 类型)必选 操作 ID，在此函数中不起作用，可忽略

dataType: (Number 类型)必选 参数类型 详见 CONSTANT 中 Callback 方法数据类型

data: (String 类型)必选 返回当前选择的浮动窗口页面的数据，json 格式如下：

{"multiPopName":"name","multiPopSelectedIndex":"index"}

各字段含义如下:

参数	是否必须	说明
multiPopName	是	多页面浮动窗口的名字
multiPopSelectedIndex	是	子页面窗口索引

版本支持：

3.0.0+

cbGetUrlQuery

获取参数时的回调方法

uexWindow.cbGetUrlQuery(opId, dataType, data)

参数：

opId:　(Number 类型)必选 操作 ID，在此函数中不起作用，可忽略

dataType:　(Number 类型)必选 参数类型 详见 CONSTANT 中 Callback 方法数据类型

data: (String 类型)必选　返回的数据

版本支持：

3.0.0+

uexOnload

网页加载完成时的回调方法

window.uexOnload(type)

参数：

type: (Number 类型)必选 当前加载完毕 View 的类型。0:主窗口或者浮动窗口(即代表自己); 1:上 slibing; 2:下 slibing。

版本支持：

3.0.0+

onSlipedUpward

上滑的监听方法，内容超过一屏时有效

uexWindow.onSlipedUpward()

参数：

无

版本支持：

3.0.0+

onSlipedDownward

下滑的监听方法，内容超过一屏时有效

uexWindow.onSlipedDownward()

参数：

无

版本支持：

3.0.0+

onSlipedUpEdge

滑到顶部的监听方法，内容超过一屏时有效

uexWindow.onSlipedUpEdge()

参数：

无

版本支持：

3.0.0+

onSlipedDownEdge

滑到底部的监听方法，内容超过一屏时有效

uexWindow.onSlipedDownEdge()

参数：

无

版本支持：

3.0.0+

onAnimationFinish

动画执行完成的监听方法，只对浮动窗口有效

uexWindow.onAnimationFinish()

参数：

无

版本支持：

3.0.0+

onSetWindowFrameFinish

当前窗口位置移动完成的监听方法

uexWindow.onSetWindowFrameFinish()

参数：

无

版本支持：

3.0.0+

onSwipeRight

向右滑动的监听方法

uexWindow.onSwipeRight()

参数：

无

版本支持：

3.0.0+

onSwipeLeft

向左滑动的监听方法

uexWindow.onSwipeLeft()

参数：

无

版本支持：

3.0.0+

onBounceStateChange

弹动状态改变的监听方法

uexWindow.onBounceStateChange(type, state)

参数：

type: (Number 类型)必选 对应的部位值，0：网页顶端；1：网页底部

state: (Number 类型)必选 状态值，0：向下拉；1：超越边界；2：向上返回到最初状态

版本支持：

3.0.0+

onGlobalNotification

全局消息的监听方法

uexWindow.onGlobalNotification(data)

参数：

data: (String 类型)必选 消息，postGlobalNotification 发送的消息数据

版本支持：

3.0.0+

onKeyPressed

按键事件的监听方法

uexWindow.onKeyPressed(keyCode)

参数：

keyCode: (Number 类型)必选 按键的值，0:返回键; 1:菜单键

平台支持：

Android2.2+

版本支持：

3.0.0+

onOAuthInfo

授权验证窗口 url 变化的监听方法

uexWindow.onOAuthInfo(windowName, url)

参数：

windowName: (Number 类型)必选 窗口名称，调用 open 函数时传入的 name

url: (Number 类型)必选 相应的 url 值

版本支持：

3.0.0+

说明：

通过调用 uexWindow 对象下的 open 函数指定 flag 为 0x1（见 CONSTANT 中 Window Flags 下的 uex.cWindowFlagOAuth），只要 window 一直存在，那么对名为 windowName 的窗口 url 的变化监听会一直存在，并且每当这个名为 windowName 的 window 的 url 发生变化时，一直会有消息通过此监听方法发给 open 他的 window。此方法多用于第三方开放平台 Oauth 验证，用于监听授权界面的 url 变化，从而获取用户授权后第三方平台返回的 access_token.比如以下示例，是用于新浪微博 Oauth2.0 验证，index_content.html 加载新浪微博授权页面；在 index.html 主窗口中监听浮动窗口 index_content.html 的 url 变化 index.html

```
<!DOCTYPE html>
<html class="um landscape min-width-240px min-width-320px min-width-480px min-width-768px min-width-1024px">
 <head>
  <meta charset="utf-8">
  <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
  <link rel="stylesheet" href="css/ui-btn.css">
  <link rel="stylesheet" href="css/ui-base.css">
  <link rel="stylesheet" href="css/ui-box.css">
  <link rel="stylesheet" href="css/ui-color.css">
  <link rel="stylesheet" href="css/ui-res.css">
  <script src="js/zy_control.js">
  </script>
  <script src="js/zy_click.js">
  </script>
  <script>
  </script>
 </head>
 <body class="um-vp c-wh" ontouchstart>
  <div id="page_0" class="up ub ub-ver" tabindex="0">
```

```
<!--header 开始-->
<div id="header" class="uh c-org c-m1 t-wh">
<!--按钮开始-->
<div class="btn btn-l btn-a ub ub-ac " ontouchstart="goback();">
 <div class="ulim">返回</div>
</div>
<!--按钮结束-->
<h1 class="ut ulev0 ut-s tx-c" tabindex="0">AppCan</h1>
</div>
<!--header 结束--><!--content 开始-->
<div id="content" class="ub-f1 tx-l t-bla ub-img6 res10">
</div>
<!--content 结束--><!--footer 开始-->
<div id="footer" class="uf c-m2 c-bla t-wh">
<h1 class="ut ulev-2 tx-c" tabindex="0">(c) Copyright 3G2WIN and others 2011.
<br>
 All rights reserved.
</h1>
</div>
<!--footer 结束-->
</div>
</body>
<script>
zy_init();
function onOAuth(winNam, url){
  if(winNam=="LoginWin"){
    if(url.indexOf("error_code")>0){
    uexWindow.evaluateScript("LoginWin","0","uexWindow.close('-1')");
    uexWindow.close("-1");
    return;
  }
  if(url.indexOf("expires_in")>0){
    var pars = urlParse(url);
    if(window.localStorage){
      localStorage.sinaacctoken = pars['access_token'];
      localStorage.sinatokenexp = pars['expires_in'];
      localStorage.sinacurrentime = parseInt((new Date().getTime())/1000);
    }
    else
      uexWindow.toast("0","5","No support localStorage!",1000);
  uexWindow.close("-1");
  }
  }
}
```

```
    window.uexOnload = function(type){
      if (!type) {
        uexWindow.onOAuthInfo = onOAuth;
        var s = window.getComputedStyle($$("content"), null);
        uexWindow.openPopover("LoginWin", "0", "index_content.html key=xxxxx&redirurl=xxxx
xxx","",0,int($$("header").offsetHeight),int(s.width),int(s.height),int(s.fontSize),"1");
      }
    }
    function urlParse(url){
      var params = {};
      var loc = String(url);
      var pieces = loc.substr(loc.indexOf('#') + 1).split('&');
      params.keys = [];
      for (var i = 0; i < pieces.length; i += 1) {
        var keyVal = pieces[i].split('=');
        params[keyVal[0]] = decodeURIComponent(keyVal[1]);
        params.keys.push(keyVal[0]);
      }
      return params;
    }
    function goback(){
      uexWindow.close("-1");
    }
  </script></html>
index_content.html<!DOCTYPE html><html class="um landscape min-width-240px min-width-3
20px min-width-480px min-width-768px min-width-1024px">
  <head>
    <title>
    </title>
    <meta charset="utf-8">
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-sc
ale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <link rel="stylesheet" href="css/ui-base.css">
    <link rel="stylesheet" href="css/ui-box.css">
    <link rel="stylesheet" href="css/ui-color.css">
    <script src="js/zy_control.js">
    </script>
    <script src="js/zy_click.js">
    </script>
    <script>
    </script>
  </head>
  <body class="um-vp" ontouchstart>
      正在加载新浪微博...
```

```
 </body>
 <script>
 zy_init();
 var params =null;
 window.uexOnload = function(type){
  params = zy_parse();
  if (!type) {
   uexWindow.setBounce("1");
   uexWindow.showBounceView("0", "#FFF", "0");
   uexWindow.showBounceView("1", "#FFF", "0");
  }
  var url = "https://api.weibo.com/oauth2/authorize client_id="+params["key"]+"&response_type
=token&redirect_uri="+params["redirurl"]+"&display=mobile";
  location.href = url;
 }
 </script>
</html>
```

onStateChange

窗口前后台状态变化的监听方法

uexWindow.onStateChange(state)

参数：

state: (Number 类型)必选 状态值，0:回到前台; 1:压入后台

版本支持：

3.0.0+