**基于 HTML5 的 UI 系统**

**接口详细设计文档**

**(内部资料 请勿外传)**

**修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **时间** | **修订内容** | **修订人** | **审批人** |
| 1 | 2017年07月17日 | 创建文档 | **王龙** |  |
|  |  |  |  |  |
|  |  |  |  |  |

目 录

[目 录 3](#_Toc488084928)

[1 编写目的 4](#_Toc488084929)

[2 组件分布图 4](#_Toc488084930)

[3 程序结构 4](#_Toc488084931)

[3.1 UMD 4](#_Toc488084932)

[4 JS SDK类说明 5](#_Toc488084933)

[4.1 方法isReady() 5](#_Toc488084934)

[4.2 注册启动函数 ready() 5](#_Toc488084935)

[4.3 设置设备状态 setDeviceStatus() 5](#_Toc488084936)

[4.4 获取设备状态 getDeviceStatus() 6](#_Toc488084937)

[4.5 profile获取 getProfile() 7](#_Toc488084938)

[4.6 获取当前运行平台currentPlatform() 7](#_Toc488084939)

[4.7 获取平台SDK对象 8](#_Toc488084940)

# 编写目的

预期读者：

产品 UI 的开发者。（不包括平台 Adapter 开发者）

背景说明

* + 软件系统名称：基于 HTML5 的 UI 系统 的 JS SDK 部分。
  + 描述 <接口> 接收来自平台 Adapter 的输出，转发给上层 UI，并将用户的操作转化为指令控制设备的行为

# 组件分布图

# 程序结构

## UMD

以umd形式提供给开发者使用

# JS SDK类说明

## ~~方法isReady()~~

~~isReady()~~

~~返回 JS SDK 状态，true 或者 false。~~

~~一般不需要调用，只需要在 JSDK.ready() 中注册启动函数即可。~~

## 注册启动函数 ready()

ready()

功能: 启动函数，当 JSDK 就绪之后，调用启动函数。

输人项: 无

返回：Promise，如jsdk初始化成功则resolved，否则rejected。Resolved会传入设备状态

//只有DNA平台会返回下面字段

'deviceID': '00000000',

'subDeviceID':'00000000000000'

## 设置设备状态 setDeviceStatus()

两种形式

setDeviceStatus(cmd)

~~setDeviceStatus(cmd，subDeviceID)~~

功能: 给设备发送命令

输人项:

* ~~subDeviceID： 子设备ID，为后期支持子设备预留，当前保持为 null 即可。~~
* cmd：命令对象。

|  |
| --- |
| {  'mark': 1,  'temp': 25  } |

返回：

一个 Promise 对象。调用者可以这样使用：

|  |
| --- |
| var prms = JSDK.setDeviceStatus(…);  // 显示等待沙漏  prms.then(function(data) {  //受到平台的限制，不一定会传入data参数，如国美  //data是设备最新状态，数据结构为：  {  ~~'status':0,~~  ~~'msg':'success'~~  'status': {  'mark': 1,  'temp': 25  }，  //出于平台能力的现在，一下2个字段不一定会存在，开发者需要注意判断  'online': '0/1/2/3',  //设备状态 未知/本地/远程/离线  'name':'refrigerator'  }  }).catch(function(error) {  ~~// 命令发送失败，显示 errMsg~~  error为一个Error对象  error ={  message:’错误描述’，  code:11 //错误码  msg：平台返回的错误信息  }    }); |

## 获取设备状态 getDeviceStatus()

getDeviceStatus() //查询所有状态

getDeviceStatus(params) //通过参数与值查询状态，多用于虚拟设备，如{ page:1}

getDeviceStatus(array) //查询指定状态，如[‘pwr’,’temp’]

~~getDeviceStatus(params, subDeviceID)~~

~~getDeviceStatus(array, subDeviceID)~~

功能: 尝试更新设备状态

输人项:

* ~~subDeviceID： 子设备，可为 null~~
* params：参数与值，可选。若不选，则获取所有接口状态。
* array参数数组. 若不选，则获取所有接口状态。

返回：

个 Promise 对象。调用者可以这样使用：

|  |
| --- |
| var prms = JSDK.getDeviceStatus(…);  // 显示等待沙漏  prms.then(function(data) {  //data是设备最新状态，数据结构为：  {  ~~'status':0,~~  ~~'msg':'success'~~  'status': {  'mark': 1,  'temp': 25  },  'online': '0/1/2/3',  //设备状态 未知/本地/远程/离线  'name':'refrigerator'  }  }).catch(function(error) {  ~~// 命令发送失败，显示 errMsg~~  error为一个Error对象  error ={  message:’错误描述’，  code:11 //错误码  msg：平台返回的错误信息  }  }); |

## onStatusChanged

getDeviceStatus(callback) //查询所有状态

功能: 尝试更新设备状态。该方法在没有提供原生监听方法的平台上被调用后会开启一个间隔为3秒的interval查询，如果callback为空会停止该查询任务。

输人项:

callback： 状态变化后的回调函数，其参数为status：  
  
{

'status': {

'mark': 1,

'temp': 25

},

'online': '0/1/2/3',

}

返回：

## profile获取 getProfile()

getProfile()

功能: 获取设备profile

输人项:

返回：profile

//列举profile格式，附上说明文档

## 获取当前运行平台currentPlatform()

currentPlatform()

功能: 获取当前运行在哪个平台

输人项: 无

返回： 枚举值：

jd、ali、dna、gome

## 获取平台SDK对象

platformSDK()

功能: 获取当前平台提供的sdk对象

输人项: 无

返回：平台SDK对象，如果是空则方法已暴露在window全局对象中

|  |
| --- |
| 京东为: JDSMART对象  国美为：new GS.Device()实例化之后的对象  dna:null // Cordova已在window对象中，可直接调用《APP平台化框架\_v1.7》中提供的接口 |