

Host Javascript API

3.1.6.0

Generated by Doxygen 1.7.4

Fri Nov 4 2011 15:35:42

Contents

1	Host Javascript API	1
1.1	Overview	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	App Class Reference	7
4.1.1	Detailed Description	8
4.1.2	Member Function Documentation	8
4.1.2.1	attachBundle	8
4.1.2.2	authorizeUrl	8
4.1.2.3	error	8
4.1.2.4	expandString	9
4.1.2.5	getCommandLine	9
4.1.2.6	getConfig	9
4.1.2.7	getElevated	9
4.1.2.8	getImplode	9
4.1.2.9	getRemoveLocalStorage	10
4.1.2.10	getRestart	10
4.1.2.11	getRestartElevated	10
4.1.2.12	hasBundle	10
4.1.2.13	info	10

4.1.2.14	loadInterop	11
4.1.2.15	release	11
4.1.2.16	setCommandLine	11
4.1.2.17	setConfig	11
4.1.2.18	setImplode	11
4.1.2.19	setRemoveLocalStorage	11
4.1.2.20	setRestart	12
4.1.2.21	setRestartElevated	12
4.1.2.22	warn	12
4.1.3	Member Data Documentation	12
4.1.3.1	instanceId	12
4.1.3.2	onBundleAttached	12
4.1.3.3	onInteropLoaded	12
4.1.3.4	onWillConstructWebGetTarget	13
4.1.3.5	onWillRestart	13
4.2	Array Class Reference	13
4.2.1	Detailed Description	13
4.2.2	Member Function Documentation	13
4.2.2.1	indexOf	13
4.2.2.2	removeElement	14
4.2.2.3	removeIndex	14
4.3	BundleLoader Class Reference	14
4.3.1	Detailed Description	15
4.3.2	Member Function Documentation	15
4.3.2.1	start	15
4.3.3	Member Data Documentation	15
4.3.3.1	name	15
4.3.3.2	onComplete	15
4.3.3.3	onProgress	15
4.3.3.4	url	15
4.4	Host Class Reference	15
4.4.1	Detailed Description	16
4.4.2	Member Function Documentation	16
4.4.2.1	addBundleAttachedEventHandler	16

4.4.2.2	addInteropLoadedEventHandler	16
4.4.2.3	elevate	17
4.4.2.4	getJSON	17
4.4.2.5	idFromString	17
4.4.2.6	init	17
4.4.2.7	loadBundles	17
4.4.2.8	loadJS	18
4.4.2.9	loadJSONArray	18
4.4.2.10	release	18
4.4.2.11	removeBundleAttachedEventHandler	18
4.4.2.12	removeInteropLoadedEventHandler	18
4.4.2.13	urlToFileName	18
4.4.3	Member Data Documentation	19
4.4.3.1	app	19
4.4.3.2	skinWindow	19
4.5	HttpDownload Class Reference	19
4.5.1	Detailed Description	20
4.5.2	Member Function Documentation	20
4.5.2.1	getExitCode	20
4.5.2.2	getFileName	20
4.5.2.3	getUrl	20
4.5.2.4	getUseCache	20
4.5.2.5	getUseWriteCallback	21
4.5.2.6	release	21
4.5.2.7	setFileName	21
4.5.2.8	setUrl	21
4.5.2.9	setUseCache	21
4.5.2.10	setUseWriteCallback	21
4.5.2.11	start	22
4.5.3	Member Data Documentation	22
4.5.3.1	instanceId	22
4.5.3.2	onComplete	22
4.5.3.3	onDownloadRequired	22
4.5.3.4	onProgress	22

4.5.3.5	onStart	22
4.5.3.6	onWrite	22
4.6	SkinWindow Class Reference	23
4.6.1	Detailed Description	23
4.6.2	Member Function Documentation	23
4.6.2.1	beginMove	23
4.6.2.2	beginSize	24
4.6.2.3	close	24
4.6.2.4	getHeight	24
4.6.2.5	getWidth	24
4.6.2.6	getX	24
4.6.2.7	getY	25
4.6.2.8	hide	25
4.6.2.9	load	25
4.6.2.10	minimize	25
4.6.2.11	release	25
4.6.2.12	restore	25
4.6.2.13	setOrigin	26
4.6.2.14	setSize	26
4.6.2.15	setTitle	26
4.6.2.16	show	26
4.6.3	Member Data Documentation	26
4.6.3.1	instanceId	26
4.6.3.2	onDidClose	27
4.6.3.3	onWillClose	27
5	File Documentation	29
5.1	app.js File Reference	29
5.1.1	Detailed Description	29
5.1.2	Function Documentation	29
5.1.2.1	createApp	29
5.2	bundleloader.js File Reference	29
5.2.1	Detailed Description	30
5.2.2	Function Documentation	30

5.2.2.1	createBundleLoader	30
5.3	host.js File Reference	30
5.3.1	Detailed Description	30
5.3.2	Variable Documentation	30
5.3.2.1	host	30
5.4	httpdownload.js File Reference	31
5.4.1	Detailed Description	31
5.4.2	Function Documentation	31
5.4.2.1	createHttpDownload	31
5.5	skinwindow.js File Reference	31
5.5.1	Detailed Description	31
5.5.2	Function Documentation	32
5.5.2.1	createSkinWindow	32

Chapter 1

Host Javascript API

1.1 Overview

The host javascript API allows users to integrate with the Solid State Networks host platform.

The usage of this API is restricted to entities which have signed a license agreement with Solid State Networks, Inc.

Licenses are valid for only one PRODUCT usage. Please see the license agreement for further details.

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

App (Solid State's Application Interface)	7
Array (Array extension)	13
BundleLoader (Helper class to load bundles from JSON file)	14
Host (Host Top level container for the host)	15
HttpDownload (HTTP downloader)	19
SkinWindow (Solid State's HTML Skinned Window)	23

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

app.js (File containing App class and creation function)	29
bundleloader.js (File containing BundleLoader class and creation function) . .	29
host.js (File containing Host class)	30
httpdownload.js (File containing HttpDownload class and creation function) . .	31
skinwindow.js (File containing SkinWindow class and creation function)	31

Chapter 4

Class Documentation

4.1 App Class Reference

Solid State's Application Interface.

Public Member Functions

- void [attachBundle](#) (string bundleName, string path)
- bool [authorizeUrl](#) (string url)
- void [error](#) (string message)
- string [expandString](#) (string expand)
- string [getCommandLine](#) ()
- string [getConfig](#) (string key, string defaultValue)
- bool [getElevated](#) ()
- bool [getImplode](#) ()
- bool [getRemoveLocalStorage](#) ()
- bool [getRestart](#) ()
- bool [getRestartElevated](#) ()
- bool [hasBundle](#) (string bundleName)
- void [info](#) (string message)
- void [loadInterop](#) (string fileName)
- void [release](#) ()
- void [setCommandLine](#) (string value)
- void [setConfig](#) (string key, string value)
- void [setImplode](#) (bool value)
- void [setRemoveLocalStorage](#) (bool value)
- void [setRestart](#) (bool value)
- void [setRestartElevated](#) (bool value)
- void [warn](#) (string message)

Public Attributes

- string [instanceId](#)
- event [onBundleAttached](#)
- event [onInteropLoaded](#)
- event [onWillConstructWebGetTarget](#)
- event [onWillRestart](#)

4.1.1 Detailed Description

Solid State's Application Interface.

4.1.2 Member Function Documentation

4.1.2.1 void App.attachBundle (string *bundleName*, string *path*)

This method attaches a bundle.

Parameters

<i>bundleName</i>	Name of bundle to be checked.
<i>path</i>	Local location of the bundle.

4.1.2.2 bool App.authorizeUrl (string *url*)

This method checks if a url is in the app config whitelist.

Parameters

<i>url</i>	Url to be checked.
------------	--------------------

Returns

true if authorized, false otherwise.

4.1.2.3 void App.error (string *message*)

Adds an error line to the host log file.

Parameters

<i>message</i>	string to add to the log file.
----------------	--------------------------------

4.1.2.4 string App.expandString (string *expand*)

This method expands the string with the current macros.

Parameters

<i>expand</i>	Name of string to be expanded.
---------------	--------------------------------

Returns

expanded string.

4.1.2.5 string App.getCommandLine ()

gets the application command line

Returns

application command line

4.1.2.6 string App.getConfig (string *key*, string *defaultValue*)

This method retrieves a config setting.

Parameters

<i>key</i>	Name of config setting to be retrieved.
<i>defaultValue</i>	Default value of key if nothing is found.

Returns

value of key's setting.

4.1.2.7 bool App.getElevated ()

gets the host's current elevation status

Returns

host's current elevation status

4.1.2.8 bool App.getImplode ()

gets the self delete on exit

Returns

self delete on exit

4.1.2.9 bool App.getRemoveLocalStorage ()

gets the flag value to remove the local storage directory on shutdown

Returns

flag value to remove the local storage directory on shutdown

4.1.2.10 bool App.getRestart ()

gets the flag value to restart the host

Returns

flag value to restart the host

4.1.2.11 bool App.getRestartElevated ()

gets the flag value to elevate the restart

Returns

flag value to elevate the restart

4.1.2.12 bool App.hasBundle (string *bundleName*)

This method checks if a bundle name has already been attached.

Parameters

<i>bundleName</i>	Name of bundle to be checked.
-------------------	-------------------------------

Returns

true if attached, false otherwise.

4.1.2.13 void App.info (string *message*)

Adds an information line to the host log file.

Parameters

<i>message</i>	string to add to the log file.
----------------	--------------------------------

4.1.2.14 void App.loadInterop (string *fileName*)

This method loads an interop library.

Parameters

<i>fileName</i>	Name of interop library to be loaded.
-----------------	---------------------------------------

4.1.2.15 void App.release ()

releases the object

4.1.2.16 void App.setCommandLine (string *value*)

sets the application command line

Parameters

<i>value</i>	application command line
--------------	--------------------------

4.1.2.17 void App.setConfig (string *key*, string *value*)

This method sets a config setting.

Parameters

<i>key</i>	Name of config setting to be retrieved.
<i>value</i>	Value store in key parameter.

4.1.2.18 void App.setImplode (bool *value*)

sets the self delete on exit

Parameters

<i>value</i>	self delete on exit
--------------	---------------------

4.1.2.19 void App.setRemoveLocalStorage (bool *value*)

sets the flag value to remove the local storage directory on shutdown

Parameters

<i>value</i>	flag value to remove the local storage directory on shutdown
--------------	--------------------------------------------------------------

4.1.2.20 void App.setRestart (bool *value*)

sets the flag value to restart the host

Parameters

<i>value</i>	flag value to restart the host
--------------	--------------------------------

4.1.2.21 void App.setRestartElevated (bool *value*)

sets the flag value to elevate the restart

Parameters

<i>value</i>	flag value to elevate the restart
--------------	-----------------------------------

4.1.2.22 void App.warn (string *message*)

Adds a warning line to the host log file.

Parameters

<i>message</i>	string to add to the log file.
----------------	--------------------------------

4.1.3 Member Data Documentation

4.1.3.1 string App.instanceId

Instance id used to bind proxy object to native object

4.1.3.2 event App.onBundleAttached

Triggered when a bundle is attached.

Parameters

<i>bundleName</i>	name of the bundle to attach.
<i>path</i>	path of the bundle to attach.
<i>successful</i>	true if the attach was successful, false otherwise.

4.1.3.3 event App.onInteropLoaded

Triggered when an interop library is loaded.

Parameters

<i>fileName</i>	file name of the interop library.
-----------------	-----------------------------------

<i>successful</i>	true if the load was successful, false otherwise.
-------------------	---------------------------------------------------

4.1.3.4 event **App.onWillConstructWebGetTarget**

This event is triggered whenever a http get is attempted.

Parameters

<i>url</i>	url for the http get.
------------	-----------------------

Returns

string: new url for the http get.

4.1.3.5 event **App.onWillRestart**

Triggered when an restart is required.

The documentation for this class was generated from the following file:

- [app.js](#)

4.2 Array Class Reference

[Array](#) extension.

Public Member Functions

- int [indexOf](#) (void elt)
- void [removeElement](#) (void element)
- void [removeIndex](#) (int index)

4.2.1 Detailed Description

[Array](#) extension.

4.2.2 Member Function Documentation

4.2.2.1 int [Array.indexOf](#) (void *elt*)

gets the index of an object

Parameters

<i>object</i>	object to get the array index of
<i>from</i>	optional starting point

Returns

index of object in array

4.2.2.2 void Array.removeElement (void *element*)

removes an object

Parameters

<i>object</i>	object to remove
---------------	------------------

4.2.2.3 void Array.removeIndex (int *index*)

removes an object at the given index

Parameters

<i>index</i>	index of object
--------------	-----------------

The documentation for this class was generated from the following file:

- jshelper.js

4.3 BundleLoader Class Reference

Helper class to load bundles from JSON file.

Public Member Functions

- void [start](#) ()

Public Attributes

- string [name](#)
- event [onComplete](#)
- event [onProgress](#)
- string [url](#)

4.3.1 Detailed Description

Helper class to load bundles from JSON file.

4.3.2 Member Function Documentation

4.3.2.1 void `BundleLoader.start` ()

starts the bundle loader

4.3.3 Member Data Documentation

4.3.3.1 string `BundleLoader.name`

Name of the bundle in the {LocalStorage} directory

4.3.3.2 event `BundleLoader.onComplete`

Triggered when the bundle loader is complete.

Parameters

<i>successful</i>	true if the download was successful, false otherwise.
-------------------	-------------------------------------------------------

4.3.3.3 event `BundleLoader.onProgress`

Triggered when the bundle loader has made progress.

Parameters

<i>percent</i>	(-1.0: Still calculating) (0.0 to 1.0: Percent completed)
----------------	-----------------------------------------------------------

4.3.3.4 string `BundleLoader.url`

Url location of the bundle

The documentation for this class was generated from the following file:

- [bundleloader.js](#)

4.4 Host Class Reference

[Host](#) Top level container for the host.

Public Member Functions

- void [addBundleAttachedEventHandler](#) (event handler)
- void [addInteropLoadedEventHandler](#) (event handler)
- void [elevate](#) ()
- void [getJSON](#) (string url, event completeCallback)
- string [idFromString](#) (string str)
- void [init](#) (void [skinWindow](#))
- void [loadBundles](#) (string url, event statusCallback, event progressCallback, event completeCallback)
- void [loadJS](#) (string url, event completeCallback)
- void [loadJSArray](#) (array urls, event completeCallback)
- void [release](#) ()
- void [removeBundleAttachedEventHandler](#) (event handler)
- void [removeInteropLoadedEventHandler](#) (event handler)
- string [urlToFileName](#) (string url, string directory, string extension)

Public Attributes

- [app](#) `app`
- [SkinWindow](#) `skinWindow`

4.4.1 Detailed Description

[Host](#) Top level container for the host.

4.4.2 Member Function Documentation

4.4.2.1 void `Host.addBundleAttachedEventHandler (event handler)`

subscribe to the BundleAttached event

Parameters

<i>handler</i>	callback function for BundleAttached event
----------------	--------------------------------------------

4.4.2.2 void `Host.addInteropLoadedEventHandler (event handler)`

subscribe to the InteropLoaded event

Parameters

<i>handler</i>	callback function for InteropLoaded event
----------------	-------------------------------------------

4.4.2.3 void Host.elevate ()

elevates the host

4.4.2.4 void Host.getJSON (string url, event completeCallback)

gets a JSON object from a url

Parameters

<i>url</i>	url of json object
<i>complete-Callback</i>	called when function is complete with json object as parameter

4.4.2.5 string Host.idFromString (string str)

creates an id from a string

Parameters

<i>str</i>	string to create id from
------------	--------------------------

Returns

id created from string

4.4.2.6 void Host.init (void skinWindow)

initializes the host

4.4.2.7 void Host.loadBundles (string url, event statusCallback, event progressCallback, event completeCallback)

load bundles from url containing json file

Parameters

<i>url</i>	url of json object containing bundle information
<i>statusCall-back</i>	called when function changes status with name and state parameters
<i>progress-Callback</i>	called when function changes progress with percent parameter
<i>complete-Callback</i>	called when function is complete with successful parameter

4.4.2.8 void Host.loadJS (string *url*, event *completeCallback*)

dynamically load javascript files from url

Parameters

<i>url</i>	url of javascript
<i>complete-Callback</i>	called when function is complete with successful parameter

4.4.2.9 void Host.loadJSONArray (array *urls*, event *completeCallback*)

dynamically load javascript files from url

Parameters

<i>urls</i>	array of urls for javascript files
<i>complete-Callback</i>	called when function is complete with successful parameter

4.4.2.10 void Host.release ()

releases the host

4.4.2.11 void Host.removeBundleAttachedEventHandler (event *handler*)

unsubscribe from the BundleAttached event

Parameters

<i>handler</i>	callback function for BundleAttached event
----------------	--------------------------------------------

4.4.2.12 void Host.removeInteropLoadedEventHandler (event *handler*)

unsubscribe from the InteropLoaded event

Parameters

<i>handler</i>	callback function for InteropLoaded event
----------------	-------------------------------------------

4.4.2.13 string Host.urlToFileName (string *url*, string *directory*, string *extension*)

converts a url to a filename in the specified directory

Parameters

<i>url</i>	url to convert
<i>directory</i>	output directory
<i>extension</i>	extension of filename

Returns

path to file

4.4.3 Member Data Documentation**4.4.3.1 app Host.app**

[App](#) attached to host

4.4.3.2 SkinWindow Host.skinWindow

Window attached to host

The documentation for this class was generated from the following file:

- [host.js](#)

4.5 HttpDownload Class Reference

HTTP downloader.

Public Member Functions

- int [getExitCode](#) ()
- string [getFileName](#) ()
- string [getUrl](#) ()
- bool [getUseCache](#) ()
- bool [getUseWriteCallback](#) ()
- void [release](#) ()
- void [setFileName](#) (string value)
- void [setUrl](#) (string value)
- void [setUseCache](#) (bool value)
- void [setUseWriteCallback](#) (bool value)
- void [start](#) ()

Public Attributes

- string [instanceId](#)
- event [onComplete](#)
- event [onDownloadRequired](#)
- event [onProgress](#)
- event [onStart](#)
- event [onWrite](#)

4.5.1 Detailed Description

HTTP downloader.

4.5.2 Member Function Documentation

4.5.2.1 `int HttpDownload.getExitCode ()`

gets the exit code for download

Returns

Exit code for download

4.5.2.2 `string HttpDownload.getFileName ()`

gets the file name where url is saved

Returns

File name where url is saved

4.5.2.3 `string HttpDownload.getUrl ()`

gets the url to download

Returns

Url to download

4.5.2.4 `bool HttpDownload.getUseCache ()`

gets the determines if caching is used in the http request

Returns

Determines if caching is used in the http request

4.5.2.5 `bool HttpDownload.getUseWriteCallback ()`

gets the determines if the write callback is used in the http request

Returns

Determines if the write callback is used in the http request

4.5.2.6 `void HttpDownload.release ()`

releases the object

4.5.2.7 `void HttpDownload.setFileName (string value)`

sets the file name where url is saved

Parameters

<i>value</i>	File name where url is saved
--------------	------------------------------

4.5.2.8 `void HttpDownload.setUrl (string value)`

sets the url to download

Parameters

<i>value</i>	Url to download
--------------	-----------------

4.5.2.9 `void HttpDownload.setUseCache (bool value)`

sets the determines if caching is used in the http request

Parameters

<i>value</i>	Determines if caching is used in the http request
--------------	---------------------------------------------------

4.5.2.10 `void HttpDownload.setUseWriteCallback (bool value)`

sets the determines if the write callback is used in the http request

Parameters

<i>value</i>	Determines if the write callback is used in the http request
--------------	--------------------------------------------------------------

4.5.2.11 void `HttpDownload.start` ()

Start downloading

4.5.3 Member Data Documentation

4.5.3.1 string `HttpDownload.instanceId`

Instance id used to bind proxy object to native object

4.5.3.2 event `HttpDownload.onComplete`

Triggered when download is complete.

Parameters

<i>successful</i>	true if the download was successful, false otherwise.
-------------------	-------------------------------------------------------

4.5.3.3 event `HttpDownload.onDownloadRequired`

Triggered when an actual download is required.

4.5.3.4 event `HttpDownload.onProgress`

Triggered when download has made progress.

Parameters

<i>percent</i>	(-1.0: Still calculating) (0.0 to 1.0: Percent completed)
----------------	-----------------------------------------------------------

4.5.3.5 event `HttpDownload.onStart`

Triggered when download is started.

4.5.3.6 event `HttpDownload.onWrite`

Triggered when a write is required.

Parameters

<i>buffer</i>	Hex converted buffer for the download.
---------------	----------------------------------------

Returns

bool: true to continue download, false to cancel

The documentation for this class was generated from the following file:

- [httpdownload.js](#)

4.6 SkinWindow Class Reference

Solid State's HTML Skinned Window.

Public Member Functions

- bool [beginMove](#) ()
- bool [beginSize](#) ()
- bool [close](#) ()
- int [getHeight](#) ()
- int [getWidth](#) ()
- int [getX](#) ()
- int [getY](#) ()
- bool [hide](#) ()
- bool [load](#) (string windowConfigUrl)
- bool [minimize](#) ()
- void [release](#) ()
- bool [restore](#) ()
- bool [setOrigin](#) (int x, int y)
- bool [setSize](#) (int width, int height)
- void [setTitle](#) (string title)
- bool [show](#) ()

Public Attributes

- string [instanceId](#)
- event [onDidClose](#)
- event [onWillClose](#)

4.6.1 Detailed Description

Solid State's HTML Skinned Window.

4.6.2 Member Function Documentation

4.6.2.1 bool SkinWindow.beginMove ()

This method informs the native window of any movement so that the native/browser windows can be synchronized.

Returns

true upon success, false upon failure.

4.6.2.2 bool SkinWindow.beginSize ()

This method informs the native window of any movement so that the native/browser windows can be synchronized.

Returns

true upon success, false upon failure.

4.6.2.3 bool SkinWindow.close ()

This method will exit the launcher application.

Returns

true upon success, false upon failure.

4.6.2.4 int SkinWindow.getHeight ()

gets the height of the window.

Returns

Height of the window.

4.6.2.5 int SkinWindow.getWidth ()

gets the width of the window.

Returns

Width of the window.

4.6.2.6 int SkinWindow.getX ()

gets the x origin of the window.

Returns

X origin of the window.

4.6.2.7 int SkinWindow.getY ()

gets the y origin of the window.

Returns

Y origin of the window.

4.6.2.8 bool SkinWindow.hide ()

This method will hide the window.

Returns

true upon success, false upon failure.

4.6.2.9 bool SkinWindow.load (string *windowConfigUrl*)

This creates a loads a new window.

Parameters

<i>windowConfigUrl</i>	URL to the window configuration file.
------------------------	---------------------------------------

Returns

true upon success, false upon failure.

4.6.2.10 bool SkinWindow.minimize ()

This method will minimize the launcher application.

Returns

true upon success, false upon failure.

4.6.2.11 void SkinWindow.release ()

releases the object

4.6.2.12 bool SkinWindow.restore ()

This method will restore a minimized launcher application.

Returns

true upon success, false upon failure.

4.6.2.13 `bool SkinWindow.setOrigin (int x, int y)`

Set the origin of the launcher window

Parameters

<i>x</i>	The desired x value of the launcher window.
<i>y</i>	The desired y value of the launcher window.

Returns

true upon success, false upon failure.

4.6.2.14 `bool SkinWindow.setSize (int width, int height)`

Set the size of the launcher window

Parameters

<i>width</i>	The desired width of the launcher window.
<i>height</i>	The desired height of the launcher window.

Returns

true upon success, false upon failure.

4.6.2.15 `void SkinWindow.setTitle (string title)`

Set the title of the launcher window

Parameters

<i>title</i>	The title of the launcher window.
--------------	-----------------------------------

4.6.2.16 `bool SkinWindow.show ()`

This method will show the window.

Returns

true upon success, false upon failure.

4.6.3 Member Data Documentation

4.6.3.1 `string SkinWindow.instanceId`

Instance id used to bind proxy object to native object

4.6.3.2 event `SkinWindow.onDidClose`

Triggered when the window is closed.

4.6.3.3 event `SkinWindow.onWillClose`

Triggered before the window is closed.

Returns

bool: true to allow the window to close, false otherwise.

The documentation for this class was generated from the following file:

- [skinwindow.js](#)

Chapter 5

File Documentation

5.1 app.js File Reference

File containing [App](#) class and creation function.

Classes

- class [App](#)
Solid State's Application Interface.

Functions

- void [createApp](#) ()

5.1.1 Detailed Description

File containing [App](#) class and creation function.

5.1.2 Function Documentation

5.1.2.1 void [createApp](#) ()

Create instance of app

5.2 bundleloader.js File Reference

File containing [BundleLoader](#) class and creation function.

Classes

- class [BundleLoader](#)
Helper class to load bundles from JSON file.

Functions

- void [createBundleLoader](#) ()

5.2.1 Detailed Description

File containing [BundleLoader](#) class and creation function.

5.2.2 Function Documentation

5.2.2.1 void [createBundleLoader](#) ()

Create instance of bundle loader

5.3 [host.js](#) File Reference

File containing [Host](#) class.

Classes

- class [Host](#)
[Host](#) Top level container for the host.

Variables

- [Host](#) [host](#)

5.3.1 Detailed Description

File containing [Host](#) class.

5.3.2 Variable Documentation

5.3.2.1 [Host](#) [host](#)

precreated global instance of the host

5.4 httpdownload.js File Reference

File containing [HttpDownload](#) class and creation function.

Classes

- class [HttpDownload](#)
HTTP downloader.

Functions

- void [createHttpDownload](#) ()

5.4.1 Detailed Description

File containing [HttpDownload](#) class and creation function.

5.4.2 Function Documentation

5.4.2.1 void [createHttpDownload](#) ()

Create instance of httpDownload

5.5 skinwindow.js File Reference

File containing [SkinWindow](#) class and creation function.

Classes

- class [SkinWindow](#)
Solid State's HTML Skinned Window.

Functions

- void [createSkinWindow](#) ()

5.5.1 Detailed Description

File containing [SkinWindow](#) class and creation function.

5.5.2 Function Documentation

5.5.2.1 void createSkinWindow ()

Create instance of skinWindow

Index

- addBundleAttachedEventHandler
 - Host, [16](#)
- addInteropLoadedEventHandler
 - Host, [16](#)
- App, [7](#)
 - attachBundle, [8](#)
 - authorizeUrl, [8](#)
 - error, [8](#)
 - expandString, [8](#)
 - getCommandLine, [9](#)
 - getConfig, [9](#)
 - getElevated, [9](#)
 - getImplode, [9](#)
 - getRemoveLocalStorage, [9](#)
 - getRestart, [10](#)
 - getRestartElevated, [10](#)
 - hasBundle, [10](#)
 - info, [10](#)
 - instanceld, [12](#)
 - loadInterop, [10](#)
 - onBundleAttached, [12](#)
 - onInteropLoaded, [12](#)
 - onWillConstructWebGetTarget, [13](#)
 - onWillRestart, [13](#)
 - release, [11](#)
 - setCommandLine, [11](#)
 - setConfig, [11](#)
 - setImplode, [11](#)
 - setRemoveLocalStorage, [11](#)
 - setRestart, [11](#)
 - setRestartElevated, [12](#)
 - warn, [12](#)
- app
 - Host, [19](#)
- app.js, [29](#)
 - createApp, [29](#)
- Array, [13](#)
 - indexOf, [13](#)
 - removeElement, [14](#)
 - removeIndex, [14](#)
- attachBundle
 - App, [8](#)
- authorizeUrl
 - App, [8](#)
- beginMove
 - SkinWindow, [23](#)
- beginSize
 - SkinWindow, [24](#)
- BundleLoader, [14](#)
 - name, [15](#)
 - onComplete, [15](#)
 - onProgress, [15](#)
 - start, [15](#)
 - url, [15](#)
- bundleloader.js, [29](#)
 - createBundleLoader, [30](#)
- close
 - SkinWindow, [24](#)
- createApp
 - app.js, [29](#)
- createBundleLoader
 - bundleloader.js, [30](#)
- createHttpDownload
 - httpdownload.js, [31](#)
- createSkinWindow
 - skinwindow.js, [32](#)
- elevate
 - Host, [16](#)
- error
 - App, [8](#)
- expandString
 - App, [8](#)
- getCommandLine
 - App, [9](#)
- getConfig
 - App, [9](#)
- getElevated
 - App, [9](#)
- getExitCode

- HttpDownload, [20](#)
- getFileName
 - HttpDownload, [20](#)
- getHeight
 - SkinWindow, [24](#)
- getImplode
 - App, [9](#)
- getJSON
 - Host, [17](#)
- getRemoveLocalStorage
 - App, [9](#)
- getRestart
 - App, [10](#)
- getRestartElevated
 - App, [10](#)
- getUrl
 - HttpDownload, [20](#)
- getUseCache
 - HttpDownload, [20](#)
- getUseWriteCallback
 - HttpDownload, [20](#)
- getWidth
 - SkinWindow, [24](#)
- getX
 - SkinWindow, [24](#)
- getY
 - SkinWindow, [24](#)
- hasBundle
 - App, [10](#)
- hide
 - SkinWindow, [25](#)
- Host, [15](#)
 - addBundleAttachedEventHandler, [16](#)
 - addInteropLoadedEventHandler, [16](#)
 - app, [19](#)
 - elevate, [16](#)
 - getJSON, [17](#)
 - idFromString, [17](#)
 - init, [17](#)
 - loadBundles, [17](#)
 - loadJS, [17](#)
 - loadJSONArray, [18](#)
 - release, [18](#)
 - removeBundleAttachedEventHandler, [18](#)
 - removeInteropLoadedEventHandler, [18](#)
 - skinWindow, [19](#)
 - urlToFileName, [18](#)
- host
 - host.js, [30](#)
 - host.js, [30](#)
 - host, [30](#)
 - HttpDownload, [19](#)
 - getExitCode, [20](#)
 - getFileName, [20](#)
 - getUrl, [20](#)
 - getUseCache, [20](#)
 - getUseWriteCallback, [20](#)
 - instanceId, [22](#)
 - onComplete, [22](#)
 - onDownloadRequired, [22](#)
 - onProgress, [22](#)
 - onStart, [22](#)
 - onWrite, [22](#)
 - release, [21](#)
 - setFileName, [21](#)
 - setUrl, [21](#)
 - setUseCache, [21](#)
 - setUseWriteCallback, [21](#)
 - start, [21](#)
- httpdownload.js, [31](#)
 - createHttpDownload, [31](#)
- idFromString
 - Host, [17](#)
- indexOf
 - Array, [13](#)
- info
 - App, [10](#)
- init
 - Host, [17](#)
- instanceId
 - App, [12](#)
 - HttpDownload, [22](#)
 - SkinWindow, [26](#)
- load
 - SkinWindow, [25](#)
- loadBundles
 - Host, [17](#)
- loadInterop
 - App, [10](#)
- loadJS
 - Host, [17](#)
- loadJSONArray
 - Host, [18](#)
- minimize
 - SkinWindow, [25](#)

- name
 - BundleLoader, 15
- onBundleAttached
 - App, 12
- onComplete
 - BundleLoader, 15
 - HttpDownload, 22
- onDidClose
 - SkinWindow, 26
- onDownloadRequired
 - HttpDownload, 22
- onInteropLoaded
 - App, 12
- onProgress
 - BundleLoader, 15
 - HttpDownload, 22
- onStart
 - HttpDownload, 22
- onWillClose
 - SkinWindow, 27
- onWillConstructWebGetTarget
 - App, 13
- onWillRestart
 - App, 13
- onWrite
 - HttpDownload, 22
- release
 - App, 11
 - Host, 18
 - HttpDownload, 21
 - SkinWindow, 25
- removeBundleAttachedEventHandler
 - Host, 18
- removeElement
 - Array, 14
- removeIndex
 - Array, 14
- removeInteropLoadedEventHandler
 - Host, 18
- restore
 - SkinWindow, 25
- setCommandLine
 - App, 11
- setConfig
 - App, 11
- setFileName
 - HttpDownload, 21
- setImplode
 - App, 11
- setOrigin
 - SkinWindow, 25
- setRemoveLocalStorage
 - App, 11
- setRestart
 - App, 11
- setRestartElevated
 - App, 12
- setSize
 - SkinWindow, 26
- setTitle
 - SkinWindow, 26
- setUrl
 - HttpDownload, 21
- setUseCache
 - HttpDownload, 21
- setUseWriteCallback
 - HttpDownload, 21
- show
 - SkinWindow, 26
- SkinWindow, 23
 - beginMove, 23
 - beginSize, 24
 - close, 24
 - getHeight, 24
 - getWidth, 24
 - getX, 24
 - getY, 24
 - hide, 25
 - instanceId, 26
 - load, 25
 - minimize, 25
 - onDidClose, 26
 - onWillClose, 27
 - release, 25
 - restore, 25
 - setOrigin, 25
 - setSize, 26
 - setTitle, 26
 - show, 26
- skinWindow
 - Host, 19
- skinwindow.js, 31
 - createSkinWindow, 32
- start
 - BundleLoader, 15
 - HttpDownload, 21

url
 BundleLoader, [15](#)
urlToFileName
 Host, [18](#)

warn
 App, [12](#)