Paragon (/)

paragon.runtime

Use the paragon.runtime API to manage the runtime. Runtime manages app launch, services and running applications, and app close.

Summary

Events
onAppLaunched
onAppExited
Methods
execService
getRunningApps
getAppByInstanceId
getVersion
launchApp
closeApp
Types
nativePort

Events

onAppLaunched

Fired when an app is launched.

addListener

```
paragon.runtime.onAppLaunched.addListener(function callback)
```

Parameters		
function	callback	The *callback* parameter should be a function that looks like this: `function() {...};`

onAppExited

Fired when an app has exit.

addListener

```
paragon.runtime.onAppExited.addListener(function callback)
```

Parameters		
function	callback	The *callback* parameter should be a function that looks like this: `function() {...};`

Methods

execService

Executes the service by connecting to the native port.

```
paragon.runtime.execService(string application, function callback)
```

Parameters		
string	application	Application service to be executed.
function	callback	Returns the NativePort info. The *callback* parameter should be a function that looks like this: `function(NativePort info) {...};`

getRunningApps

Gets the applications that are running.

```
paragon.runtime.getRunningApps(function callback)
```

<table>
<tr><th colspan="3">Parameters</th></tr>
<tr><td>function</td><td>callback</td><td>Returns the running applications.
The <i>callback</i> parameter should be a function that looks like this:
<code>function(object info) {...};</code>
<table>
<tr><td>object</td><td>info</td><td>
<table>
<tr><td>string</td><td>appId</td><td>A unique application as described in the manifest.</td></tr>
<tr><td>string</td><td>appInstanceId</td><td></td></tr>
<tr><td>string</td><td>browserPid</td><td></td></tr>
<tr><td>string</td><td>workspaceId</td><td></td></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>

getAppByInstanceId

Gets the application corresponding to the Instance Id.

```
paragon.runtime.getAppByInstanceId(string instanceId, function callback)
```

Parameters		
string	instanceId	An instanceId linked to an application.
function	callback	Returns the application. The *callback* parameter should be a function that looks like this:

<table>
<tr><td></td><td></td><td><pre>function(object info) {...};</pre>
<table>
<tr><td>object</td><td>info</td><td>
<table>
<tr><td>string</td><td>appId</td><td>A unique application as described in the manifest.</td></tr>
<tr><td>string</td><td>appInstanceId</td><td></td></tr>
<tr><td>string</td><td>browserPid</td><td></td></tr>
<tr><td>string</td><td>workspaceId</td><td></td></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>

getVersion

Gets the current assembly version.

```
paragon.runtime.getVersion(function callback)
```

Parameters		
function	callback	Returns the current assembly version. The *callback* parameter should be a function that looks like this: `function(string version) {...};`

launchApp

Launches the application specified by a path and Instance Id.

```
paragon.runtime.launchApp(string packageUri, string instanceId, function callback)
```

Parameters		
string	packageUri	A valid path to the package.
string	instanceId	An instanceId linked to an application
function	callback	Returns true/false depending on whether the launching succeeded or not. The *callback* parameter should be a function that looks like this: `function(bool success) {...};`

closeApp

Closes the application.

```
paragon.runtime.applogfiles(string instanceId, function callback)
```

Parameters		
string	instanceId	An instanceId linked to an application
function	callback	Returns true/false depending on whether the application was closed or not.

<table>
<tr><td></td><td></td><td>The <i>callback</i> parameter should be a function that looks like this:
<code>function(bool closed) {...};</code></td></tr>
</table>

Types

NativePort

Properties of the native port

<table>
<tr><th colspan="3">Parameters</th></tr>
<tr><td>event</td><td>onMessage</td><td>Fired when there is a message on the port.
<b>addListener</b>
<code>paragon.runtime.execService.onMessage.addListener(function callback)</code>
<table>
<tr><td>function</td><td>callback</td><td>The <i>callback</i> parameter should be a function that looks like this:
<code>function() {...};</code></td></tr>
</table>
</td></tr>
<tr><td>event</td><td>onDisconnect</td><td>Fired when port is disconnected.
<b>addListener</b>
<code>paragon.runtime.execService.onDisconnect.addListener(function callback)</code>
<table>
<tr><td>function</td><td>callback</td><td>The <i>callback</i> parameter should be a function that looks like this:
<code>function() {...};</code></td></tr>
</table>
</td></tr>
<tr><td>function</td><td>postMessage</td><td>Sends message to a service.
<code>paragon.runtime.execService.postMessage(object message, function callback)</code>
<table>
<tr><td>object</td><td>message</td><td>Message to be posted.</td></tr>
<tr><td>function</td><td>callback</td><td>The <i>callback</i> parameter should be a function that looks like this:
<code>function() {...};</code></td></tr>
</table>
</td></tr>
</table>