Paragon (/)

paragon.app.window

Use the paragon.app.window API to create related browser windows and co-ordinate them being in workspaces. Should be instantiated once per application.

Summary

Methods
create
getCurrent
getAll
getById
Types
CreateWindowOptions
FrameOptions
BoundsSpecification
ApplicationWindow

Methods

create

Creates window

```
paragon.app.window.create(string startUrl, CreateWindowOptions options, function callback)
```

Parameters		
string	startUrl	Location of the window
CreateWindowOptions	options	Properties to Create Window
function	callback	The *callback* parameter should be a function that looks like this: `function() {...};`

getCurrent

Returns an AppWindow object for the current script context (ie JavaScript 'window' object). This can also be called on a handle to a script context for another page, for example: otherWindow.chrome.app.window.getcurrent()

```
paragon.app.window.getCurrent(function callback)
```

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

getAll

Gets an array of all currently created app windows

```
paragon.app.window.getAll(function callback)
```

Parameters		
function	callback	Returns an array of Application Windows The *callback* parameter should be a function that looks like this:

		`function(array of ApplicationWindow){...};`

getById

Returns an AppWindow with the given id. If no window with the given id exists, null is returned

```
paragon.app.window.getCurrent(string id, function callback)
```

Parameters		
string	id	Id associated with the window
function	callback	Returns the appWindow object The *callback* parameter should be a function that looks like this: `function(ApplicationWindow appWindow){...};`

Types

CreateWindowOptions

Options to specify while creating window.

properties		
string	id	Id to identify the window. This will be used to remember the size and position of the window. (OPTIONAL)
BoundsSpecification	outerBounds	Used to specify the initial position, initial size and constraints of the window (including window decorations such as the title bar and frame). If an id is also specified and a window with a matching id has been shown before, the remembered bounds will be used instead. Note that the padding between the inner and outer bounds is determined by the OS. Therefore setting the same bounds property for both the innerBounds and outerBounds will result in an error.
FrameOptions	frame	Window frame options
string	initialState	Initial state of the window, allowing it to be created already fullscreen, maximized or minimized
boolean	hidden	If hidden, show() will make the window appear
boolean	resizable	If true, the window will be resizable by the user. Defaults to true
boolean	alwaysOnTop	If true, the window will stay above most other windows. If there are multiple windows of this kind, the currently focused window will be in the foreground. Requires "alwaysOnTopWindows" permission. Defaults to false
boolean	focused	If true, the window will be focused when created. Defaults to true
boolean	minimizeOnClose	If true, clicking the close button will minimize the window. Defaults to false
boolean	hotKeysEnabled	If true, enables the support of hot keys for this window. Defaults to false

FrameOptions

Defines the specifications for the frame

properties		
boolean	icon	
boolean	minimizeButton	
boolean	maximizeButton	
object	type	enum of "notSpecified", "paragon", "windowsDefault" or "none" | frameType |

<table>
<tr><td>object</td><td>systemMenu</td><td><table><tr><td>array of items</td><td>systemMenuItem</td><td><table>
<tr><td>string</td><td>header</td><td></td></tr>
<tr><td>integer</td><td>id</td><td></td></tr>
<tr><td>boolean</td><td>checkable</td><td></td></tr>
<tr><td>boolean</td><td>isChecked</td><td></td></tr>
<tr><td>boolean</td><td>enabled</td><td></td></tr>
</table></td></tr></table></td></tr>
</table>

BoundsSpecification

Specifications for position, size and constraints of the window.

properties		
double	left	
double	top	
double	width	
double	height	
double	minWidth	
double	minHeight	
double	maxWidth	
double	maxHeight	

ApplicationWindow

Defines JavaScript's perspective of an application window. It is a dynamic plugin; does not require a name

<table>
<tr><td colspan="3"><b>properties</b></td></tr>
<tr><td>event</td><td>onBoundsChanged</td><td>Fired when window is resized.
<b>addListener</b>
<code>paragon.app.window.onBoundsChanged.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>onClosed</td><td>Fired when window is closed.
<b>addListener</b>
<code>paragon.app.window.onClosed.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>onFullScreened</td><td>Fired when window is made full screen.
<b>addListener</b>
<code>paragon.app.window.onFullScreened.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>
</table>

<table>
<tr><td>event</td><td>onMaximized</td><td>Fired when window is maximized.

<b>addListener</b>
<code>paragon.app.window.onMaximized.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>onMinimized</td><td>Fired when window is minimized.

<b>addListener</b>
<code>paragon.app.window.onMinimized.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>onRestored</td><td>Fired when window is restored.

<b>addListener</b>
<code>paragon.app.window.onRestored.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>focus</td><td>Focuses the window; no arguments needed</td></tr>
<tr><td>function</td><td>focus</td><td>Focuses the window.

<code>paragon.app.window.focus(integer x, integer y, function callback)</code><table><tr><td>integer</td><td>x</td><td>Dimension1 of window to focus</td></tr><tr><td>integer</td><td>y</td><td>Dimension2 of window to focus</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>
<tr><td>function</td><td>refresh</td><td>Refreshes the window.</td></tr>
<tr><td>function</td><td>fullscreen</td><td>Fullscreens the window.</td></tr>
<tr><td>function</td><td>minimize</td><td>Minimizes the window.</td></tr>
<tr><td>function</td><td>maximize</td><td>Maximizes the window.</td></tr>
</table>

<table>
<tr><td>function</td><td>restore</td><td>Restores the window.</td></tr>
<tr><td>function</td><td>moveTo</td><td>Moves the window to the position(|left|, |top|).
<pre>paragon.app.window.moveTo(integer left, integer top, function callback)</pre>
<table>
<tr><td>integer</td><td>left</td><td>Position left</td></tr>
<tr><td>integer</td><td>top</td><td>Position top</td></tr>
<tr><td>function</td><td>callback</td><td>The <i>callback</i> parameter should be a function that looks like this:
<pre>function() {...};</pre></td></tr>
</table></td></tr>
<tr><td>function</td><td>resizeTo</td><td>Resizes the window to |width|x|height| pixels in size.
<pre>paragon.app.window.resizeTo(integer width, integer height, function callback)</pre>
<table>
<tr><td>integer</td><td>width</td><td>Width of the window</td></tr>
<tr><td>integer</td><td>height</td><td>Height of the window</td></tr>
<tr><td>function</td><td>callback</td><td>The <i>callback</i> parameter should be a function that looks like this:
<pre>function() {...};</pre></td></tr>
</table></td></tr>
<tr><td>function</td><td>drawAttention</td><td>Draws attention to the window.
<pre>paragon.app.window.drawAttention(boolean autoclear, function callback)</pre>
<table>
<tr><td>boolean</td><td>autoclear</td><td>If false, draws attention to window</td></tr>
<tr><td>function</td><td>callback</td><td>The <i>callback</i> parameter should be a function that looks like this:
<pre>function() {...};</pre></td></tr>
</table></td></tr>
<tr><td>function</td><td>clearAttention</td><td>Clears attention to the window.</td></tr>
<tr><td>function</td><td>close</td><td>Closes the window.</td></tr>
<tr><td>function</td><td>show</td><td>Shows the window.</td></tr>
<tr><td>function</td><td>hide</td><td>Hides the window.</td></tr>
<tr><td>function</td><td>getId</td><td>Gets the Id the window was created with.</td></tr>
<tr><td>function</td><td>getTitle</td><td>Gets the Id to uniquely identify the window instance within the application.</td></tr>
<tr><td>function</td><td>getInnerBounds</td><td>Gets the position, size and constraints of the window's content, which does not include window decorations.
<pre>paragon.app.window.createWindow.getInnerBounds(function callback)</pre>
<table>
<tr><td>function</td><td>callback</td><td>The <i>callback</i> parameter should be a function that looks like this:
<pre>function(BoundsSpecification spec) {...};</pre></td></tr>
</table></td></tr>
</table>

<table>
<tr><td>function</td><td>getOuterBounds</td><td>Gets the position, size and constraints of the window's content, which includes window decorations, such as the title bar and frame.
<code>paragon.app.window.createWindow.getOuterBounds(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(BoundsSpecification spec) {...};</code></td></tr></table></td></tr>
<tr><td>function</td><td>setOuterBounds</td><td>Sets the outer bounds-position, size and constraints of the window's content, which includes window decorations, such as the title bar and frame.
<code>paragon.app.window.createWindow.getOuterBounds(BoundsSpecification spec, 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>
</table>