



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# Service Client API Reference

System namespace

---

## Contents

- **1 Methods**
  - 1.1 amVisible
  - 1.2 closeDialog
  - 1.3 closeToast
  - 1.4 closeViewInApplicationMenuBar
  - 1.5 getAllowedServices
  - 1.6 isFrameLeading
  - 1.7 isFrameFollowing
  - 1.8 isFrameNegotiating
  - 1.9 isFrameLeadingOrNegotiating
  - 1.10 isLastActiveFrame
  - 1.11 openDialog
  - 1.12 popupToast
  - 1.13 triggerActivity
  - 1.14 updateViewInApplicationMenuBar
  - 1.15 updateToast

---

Learn about the System namespace methods in the Service Client API.

### Important

Depending on your environment, you might need to contact your Genesys representative to complete the configuration described on this page.

## Methods

The System namespace includes the following methods:

- amIVisible
- closeDialog
- closeToast
- closeViewInApplicationMenuBar
- getAllowedServices
- isFrameLeading
- isFrameFollowing
- isFrameNegotiating
- isFrameLeadingOrNegotiating
- isLastActiveFrame
- popupToast
- openDialog
- triggerActivity
- updateViewInApplicationMenuBar
- updateToast

### amIVisible

<b>Signature</b>	amIVisible(succeeded, failed) → {boolean}
<b>Description</b>	Get the current visibility state of the frame.

<b>Signature</b>	amIVisible(succeeded, failed) → {boolean}		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
	succeeded	function	A function called when the operation succeeds.
	failed	function	A function called when the operation fails.
<b>Returns</b>	true if the frame is visible.		

### Sample request

```
setTimeout(function() {
  genesys.wwe.service.system.amIVisible(succeeded, failed);
}, 3000); // This gives 3 seconds to switch the panel to test.
```

### Sample response

The asynchronous answer is included in the `data` attribute:

```
{
  "request": "system.amIVisible",
  "data": true,
  "userAgent": "WWE Server",
  "protocolVersion": 2
}
```

### closeDialog

<b>Signature</b>	closeDialog(dialogId, succeeded, failed) → {boolean}		
<b>Description</b>	Close a previously opened dialog.		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
	dialogId	string	The dialog identifier (returned in the response of openDialog).
	succeeded	function	A function called when the operation succeeds.
	failed	function	A function called when the operation fails.

<b>Signature</b>	<code>closeDialog(dialogId, succeeded, failed) → {boolean}</code>
<b>Returns</b>	true if the dialog is closed; false if the dialog is not found.

## closeToast

<b>Signature</b>	<code>closeToast(id, succeeded, failed) → {boolean}</code>														
<b>Description</b>	Closes the specified toast.														
<b>Parameters</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>string</td> <td>The identifier of the toast to close. The identifier is returned by the <code>popupToast</code> method.</td> </tr> <tr> <td>succeeded</td> <td>function</td> <td>A function called when the operation succeeds.</td> </tr> <tr> <td>failed</td> <td>function</td> <td>A function called when the operation fails.</td> </tr> </tbody> </table>	Name	Type	Description	id	string	The identifier of the toast to close. The identifier is returned by the <code>popupToast</code> method.	succeeded	function	A function called when the operation succeeds.	failed	function	A function called when the operation fails.		
Name	Type	Description													
id	string	The identifier of the toast to close. The identifier is returned by the <code>popupToast</code> method.													
succeeded	function	A function called when the operation succeeds.													
failed	function	A function called when the operation fails.													
<b>Returns</b>	true if the toast has been updated; false if the toast identifier has not been found.														

## closeViewInApplicationMenuBar

<b>Signature</b>	<code>closeViewInApplicationMenuBar(parameters, succeeded, failed) → {boolean}</code>														
<b>Description</b>	Removes the given view from the <b>Application Menu</b> bar region.														
<b>Parameters</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>string</td> <td>The name of the custom view to be removed.</td> </tr> <tr> <td>succeeded</td> <td>function</td> <td>A function called when the operation succeeds.</td> </tr> <tr> <td>failed</td> <td>function</td> <td>A function</td> </tr> </tbody> </table>	Name	Type	Description	name	string	The name of the custom view to be removed.	succeeded	function	A function called when the operation succeeds.	failed	function	A function		
Name	Type	Description													
name	string	The name of the custom view to be removed.													
succeeded	function	A function called when the operation succeeds.													
failed	function	A function													

<b>Signature</b>	closeViewInApplicationMenuBar(parameters, succeeded, failed) → {boolean}		
	<b>Name</b>	<b>Type</b>	<b>Description</b>
			called when the operation fails.
<b>Returns</b>	true if the view is removed; false if the view name is not found.		

### Sample request

```
genesys.wws.service.system.closeViewInApplicationMenuBar("view1", succeeded, failed)
```

### Sample response

```
{
  "request": "system.closeViewInApplicationMenuBar",
  "data": true,
  "userAgent": "WWE Server",
  "protocolVersion": 2
}
```

### Sample request

```
genesys.wws.service.system.closeDialog("wweCustomDialog1", succeeded, failed)
```

### Sample response

The asynchronous answer is included in the data attribute:

```
{
  "request": "system.closeDialog",
  "data": true,
  "userAgent": "WWE Server",
  "protocolVersion": 2
}
```

## getAllowedServices

<b>Signature</b>	getAllowedServices(succeeded, failed) → {Array.}		
<b>Description</b>	Gets the list of allowed services, as determined by the Security configuration. If the domain of the web application that calls this method isn't listed in the service-client-api.accepted-web-content-origins option, then this method fails.		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
	succeeded	function	A function called when the operation succeeds.

<b>Signature</b>	getAllowedServices(succeeded, failed) → {Array.}						
	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>failed</td> <td>function</td> <td>A function called when the operation fails.</td> </tr> </tbody> </table>	Name	Type	Description	failed	function	A function called when the operation fails.
Name	Type	Description					
failed	function	A function called when the operation fails.					
<b>Returns</b>	Array.						

## isFrameLeading

<b>Signature</b>	isFrameLeading( <i>succeeded, failed</i> ) → {boolean}									
<b>Description</b>	Find out if the browser tab is leading.									
<b>Parameters</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>succeeded</td> <td>function</td> <td>A function called when the operation succeeds.</td> </tr> <tr> <td>failed</td> <td>function</td> <td>A function called when the operation fails.</td> </tr> </tbody> </table>	Name	Type	Description	succeeded	function	A function called when the operation succeeds.	failed	function	A function called when the operation fails.
Name	Type	Description								
succeeded	function	A function called when the operation succeeds.								
failed	function	A function called when the operation fails.								
<b>Returns</b>	true if the browser tab is the leader.									

## isFrameFollowing

<b>Signature</b>	isFrameFollowing( <i>succeeded, failed</i> ) → {boolean}									
<b>Description</b>	Find out if the browser tab is following.									
<b>Parameters</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>succeeded</td> <td>function</td> <td>A function called when the operation succeeds.</td> </tr> <tr> <td>failed</td> <td>function</td> <td>A function called when the operation fails.</td> </tr> </tbody> </table>	Name	Type	Description	succeeded	function	A function called when the operation succeeds.	failed	function	A function called when the operation fails.
Name	Type	Description								
succeeded	function	A function called when the operation succeeds.								
failed	function	A function called when the operation fails.								
<b>Returns</b>	true if this browser tab is following.									

---

## isFrameNegotiating

<b>Signature</b>	isFrameNegotiating( <i>succeeded, failed</i> ) → {boolean}									
<b>Description</b>	Find out if there is an election in progress and the browser tab state is not yet set to leading or following (the tab is "negotiating.")									
<b>Parameters</b>	<table border="1"><thead><tr><th>Name</th><th>Type</th><th>Description</th></tr></thead><tbody><tr><td>succeeded</td><td>function</td><td>A function called when the operation succeeds.</td></tr><tr><td>failed</td><td>function</td><td>A function called when the operation fails.</td></tr></tbody></table>	Name	Type	Description	succeeded	function	A function called when the operation succeeds.	failed	function	A function called when the operation fails.
Name	Type	Description								
succeeded	function	A function called when the operation succeeds.								
failed	function	A function called when the operation fails.								
<b>Returns</b>	true if the tab is negotiating.									

## isFrameLeadingOrNegotiating

<b>Signature</b>	isFrameLeadingOrNegotiating( <i>succeeded, failed</i> ) → {boolean}									
<b>Description</b>	Find out if the browser tab is leading or there is an election in progress and the tab state is not yet set to leading or following (the tab is "negotiating.").									
<b>Parameters</b>	<table border="1"><thead><tr><th>Name</th><th>Type</th><th>Description</th></tr></thead><tbody><tr><td>succeeded</td><td>function</td><td>A function called when the operation succeeds.</td></tr><tr><td>failed</td><td>function</td><td>A function called when the operation fails.</td></tr></tbody></table>	Name	Type	Description	succeeded	function	A function called when the operation succeeds.	failed	function	A function called when the operation fails.
Name	Type	Description								
succeeded	function	A function called when the operation succeeds.								
failed	function	A function called when the operation fails.								
<b>Returns</b>	true if the browser tab is leading or negotiating.									

## isLastActiveFrame

<b>Signature</b>	isLastActiveFrame( <i>succeeded, failed</i> ) → {boolean}						
<b>Description</b>	Find out if this is the last active browser tab.						
<b>Parameters</b>	<table border="1"><thead><tr><th>Name</th><th>Type</th><th>Description</th></tr></thead><tbody><tr><td>succeeded</td><td>function</td><td>A function called when</td></tr></tbody></table>	Name	Type	Description	succeeded	function	A function called when
Name	Type	Description					
succeeded	function	A function called when					

---

<b>Signature</b>	isLastActiveFrame( <i>succeeded, failed</i> ) → {boolean}		
	<b>Name</b>	<b>Type</b>	<b>Description</b>
			the operation succeeds.
	failed	function	A function called when the operation fails.
<b>Returns</b>	true if this is the last active browser tab.		

## openDialog

<b>Signature</b>	openDialog( <i>url, options, succeeded, failed</i> ) → {string}		
<b>Description</b>	Open an iframe in a dialog, based on the configured parameters.		
	<b>Name</b>	<b>Type</b>	<b>Description</b>
	url	string	The URL of the iframe to load in the dialog.
<b>Parameters</b>	options	object	<p>Optional parameters to configure the dialog. This value can't be null, so you must pass {} if there are no specific options. You can include any of the following options:</p> <ul style="list-style-type: none"> <li>label - Set a custom value for the aria-label attribute on the dialog. When the dialog pops up, this value identifies it</li> </ul>

Signature	openDialog( <i>url, options, succeeded, failed</i> ) → {string}														
	<table border="1"> <thead> <tr> <th data-bbox="824 323 1032 373">Name</th> <th data-bbox="1032 323 1240 373">Type</th> <th data-bbox="1240 323 1446 373">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="824 373 1032 1056"></td> <td data-bbox="1032 373 1240 1056"></td> <td data-bbox="1240 373 1446 1056">           to accessibility tools like screen readers.           <ul style="list-style-type: none"> <li>width - The initial width of the dialog. Valid formats are px or %.</li> <li>height - The initial height of the dialog. Valid formats are px or %.</li> </ul> </td> </tr> <tr> <td data-bbox="824 1056 1032 1182">succeeded</td> <td data-bbox="1032 1056 1240 1182">function</td> <td data-bbox="1240 1056 1446 1182">A function called when the operation succeeds.</td> </tr> <tr> <td data-bbox="824 1182 1032 1308">failed</td> <td data-bbox="1032 1182 1240 1308">function</td> <td data-bbox="1240 1182 1446 1308">A function called when the operation fails.</td> </tr> </tbody> </table>			Name	Type	Description			to accessibility tools like screen readers. <ul style="list-style-type: none"> <li>width - The initial width of the dialog. Valid formats are px or %.</li> <li>height - The initial height of the dialog. Valid formats are px or %.</li> </ul>	succeeded	function	A function called when the operation succeeds.	failed	function	A function called when the operation fails.
Name	Type	Description													
		to accessibility tools like screen readers. <ul style="list-style-type: none"> <li>width - The initial width of the dialog. Valid formats are px or %.</li> <li>height - The initial height of the dialog. Valid formats are px or %.</li> </ul>													
succeeded	function	A function called when the operation succeeds.													
failed	function	A function called when the operation fails.													
Returns	The dialog identifier or null if the url parameter is not defined.														

### Sample request

```
genesys.wwe.service.system.openDialog("", {
  label: "Dialog $Agent.FullName$",
  width: "430px",
  height: "325px"
}, succeeded, failed)
```

### Sample response

The asynchronous answer is included in the data attribute:

```
{
  "request": "system.openDialog",
  "data": "wweCustomDialog1",
```

```

    "userAgent": "WWE Server",
    "protocolVersion": 2
}

```

## popupToast

Signature	popupToast( <i>parameters</i> , succeeded, failed) → {string}																											
Description	Pops up a new custom toast.																											
<b>Parameters</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>title</td> <td>string</td> <td>The title</td> </tr> <tr> <td>iconUrl</td> <td>string</td> <td>The URL of the icon you want to display in the title bar of the custom toast popup.</td> </tr> <tr> <td>subject</td> <td>string</td> <td>Optional. The subject</td> </tr> <tr> <td>message</td> <td>string</td> <td>Optional. The message</td> </tr> <tr> <td>keyValues</td> <td>array</td> <td>Optional. JSON object used to fill the key value pair list. For example:</td> </tr> </tbody> </table> </td> </tr> <tr> <td>parameters</td> <td>object</td> <td></td> </tr> </tbody> </table>	Name	Type	Description			<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>title</td> <td>string</td> <td>The title</td> </tr> <tr> <td>iconUrl</td> <td>string</td> <td>The URL of the icon you want to display in the title bar of the custom toast popup.</td> </tr> <tr> <td>subject</td> <td>string</td> <td>Optional. The subject</td> </tr> <tr> <td>message</td> <td>string</td> <td>Optional. The message</td> </tr> <tr> <td>keyValues</td> <td>array</td> <td>Optional. JSON object used to fill the key value pair list. For example:</td> </tr> </tbody> </table>	Name	Type	Description	title	string	The title	iconUrl	string	The URL of the icon you want to display in the title bar of the custom toast popup.	subject	string	Optional. The subject	message	string	Optional. The message	keyValues	array	Optional. JSON object used to fill the key value pair list. For example:	parameters	object	
	Name	Type	Description																									
		<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>title</td> <td>string</td> <td>The title</td> </tr> <tr> <td>iconUrl</td> <td>string</td> <td>The URL of the icon you want to display in the title bar of the custom toast popup.</td> </tr> <tr> <td>subject</td> <td>string</td> <td>Optional. The subject</td> </tr> <tr> <td>message</td> <td>string</td> <td>Optional. The message</td> </tr> <tr> <td>keyValues</td> <td>array</td> <td>Optional. JSON object used to fill the key value pair list. For example:</td> </tr> </tbody> </table>	Name	Type	Description	title	string	The title	iconUrl	string	The URL of the icon you want to display in the title bar of the custom toast popup.	subject	string	Optional. The subject	message	string	Optional. The message	keyValues	array	Optional. JSON object used to fill the key value pair list. For example:								
Name	Type	Description																										
title	string	The title																										
iconUrl	string	The URL of the icon you want to display in the title bar of the custom toast popup.																										
subject	string	Optional. The subject																										
message	string	Optional. The message																										
keyValues	array	Optional. JSON object used to fill the key value pair list. For example:																										
parameters	object																											

Signature	popupToast( <i>parameters</i> , succeeded, failed) → {string}														
	Name	Type	Description												
			<table border="1"> <thead> <tr> <th data-bbox="1256 394 1312 428">Name</th> <th data-bbox="1312 394 1367 428">Type</th> <th data-bbox="1367 394 1463 428">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="1256 428 1312 638"></td> <td data-bbox="1312 428 1367 638"></td> <td data-bbox="1367 428 1463 638">{"key1" ; "value one", "key2" ; "value two", "key3" ; "value three"}.</td> </tr> <tr> <td data-bbox="1256 638 1312 1520"></td> <td data-bbox="1312 638 1367 1520">Array</td> <td data-bbox="1367 638 1463 1520">Optional. Each character string in this array becomes a button. All buttons are displayed as buttons, not hyperlinks, in the following order: [Button 2] [Button 3] ... [Button N] [Button 1].</td> </tr> <tr> <td data-bbox="1256 1520 1312 1839"></td> <td data-bbox="1312 1520 1367 1839">boolean</td> <td data-bbox="1367 1520 1463 1839">Optional. If set to true, displays the <b>Show and Dismiss</b></td> </tr> </tbody> </table>	Name	Type	Description			{"key1" ; "value one", "key2" ; "value two", "key3" ; "value three"}.		Array	Optional. Each character string in this array becomes a button. All buttons are displayed as buttons, not hyperlinks, in the following order: [Button 2] [Button 3] ... [Button N] [Button 1].		boolean	Optional. If set to true, displays the <b>Show and Dismiss</b>
	Name	Type	Description												
		{"key1" ; "value one", "key2" ; "value two", "key3" ; "value three"}.													
	Array	Optional. Each character string in this array becomes a button. All buttons are displayed as buttons, not hyperlinks, in the following order: [Button 2] [Button 3] ... [Button N] [Button 1].													
	boolean	Optional. If set to true, displays the <b>Show and Dismiss</b>													

Signature	popupToast( <i>parameters</i> , succeeded, failed) → {string}														
	Name	Type	Description												
			<table border="1"> <thead> <tr> <th data-bbox="1256 394 1312 426">Name</th> <th data-bbox="1312 394 1367 426">Type</th> <th data-bbox="1367 394 1463 426">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="1256 426 1312 1222"></td> <td data-bbox="1312 426 1367 1222"></td> <td data-bbox="1367 426 1463 1222">           buttons and pops up the current iframe if the <b>Show</b> button is pushed. If set to false, displays "OK" or custom buttons based on the parameter's buttons.         </td> </tr> <tr> <td data-bbox="1256 1222 1312 1682">autoClose</td> <td data-bbox="1312 1222 1367 1682">boolean</td> <td data-bbox="1367 1222 1463 1682">Optional. If set to greater than 0, the popup is automatically closed after the specified milliseconds.</td> </tr> <tr> <td data-bbox="1256 1682 1312 1839">sendTo</td> <td data-bbox="1312 1682 1367 1839">string</td> <td data-bbox="1367 1682 1463 1839">Optional. If set to</td> </tr> </tbody> </table>	Name	Type	Description			buttons and pops up the current iframe if the <b>Show</b> button is pushed. If set to false, displays "OK" or custom buttons based on the parameter's buttons.	autoClose	boolean	Optional. If set to greater than 0, the popup is automatically closed after the specified milliseconds.	sendTo	string	Optional. If set to
	Name	Type	Description												
		buttons and pops up the current iframe if the <b>Show</b> button is pushed. If set to false, displays "OK" or custom buttons based on the parameter's buttons.													
autoClose	boolean	Optional. If set to greater than 0, the popup is automatically closed after the specified milliseconds.													
sendTo	string	Optional. If set to													

Signature	popupToast( <i>parameters</i> , succeeded, failed) → {string}																							
	<table border="1"> <thead> <tr> <th data-bbox="824 323 1032 369">Name</th> <th data-bbox="1032 323 1240 369">Type</th> <th data-bbox="1240 323 1448 369">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="824 369 1032 1453"></td> <td data-bbox="1032 369 1240 1453"></td> <td data-bbox="1240 369 1448 1453"> <table border="1"> <thead> <tr> <th data-bbox="1256 394 1312 428">Name</th> <th data-bbox="1312 394 1367 428">Type</th> <th data-bbox="1367 394 1448 428">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="1256 428 1312 831"></td> <td data-bbox="1312 428 1367 831"></td> <td data-bbox="1367 428 1448 831">true, sends the <b>subject</b>, <b>iconUrl</b>, <b>title</b>, <b>keyValues</b>, and <b>message</b> parameters to the <b>MyMessage</b> panel.</td> </tr> <tr> <td data-bbox="1256 831 1312 1453">width</td> <td data-bbox="1312 831 1367 1453">number</td> <td data-bbox="1367 831 1448 1453">Optional. The width of the custom toast popup, in pixels. This value takes precedence over the service-client-api.toast.width configuration option.</td> </tr> </tbody> </table> </td> </tr> <tr> <td data-bbox="824 1453 1032 1575">succeeded</td> <td data-bbox="1032 1453 1240 1575">function</td> <td data-bbox="1240 1453 1448 1575">A function called when the operation succeeds.</td> </tr> <tr> <td data-bbox="824 1575 1032 1709">failed</td> <td data-bbox="1032 1575 1240 1709">function</td> <td data-bbox="1240 1575 1448 1709">A function called when the operation fails.</td> </tr> </tbody> </table>			Name	Type	Description			<table border="1"> <thead> <tr> <th data-bbox="1256 394 1312 428">Name</th> <th data-bbox="1312 394 1367 428">Type</th> <th data-bbox="1367 394 1448 428">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="1256 428 1312 831"></td> <td data-bbox="1312 428 1367 831"></td> <td data-bbox="1367 428 1448 831">true, sends the <b>subject</b>, <b>iconUrl</b>, <b>title</b>, <b>keyValues</b>, and <b>message</b> parameters to the <b>MyMessage</b> panel.</td> </tr> <tr> <td data-bbox="1256 831 1312 1453">width</td> <td data-bbox="1312 831 1367 1453">number</td> <td data-bbox="1367 831 1448 1453">Optional. The width of the custom toast popup, in pixels. This value takes precedence over the service-client-api.toast.width configuration option.</td> </tr> </tbody> </table>	Name	Type	Description			true, sends the <b>subject</b> , <b>iconUrl</b> , <b>title</b> , <b>keyValues</b> , and <b>message</b> parameters to the <b>MyMessage</b> panel.	width	number	Optional. The width of the custom toast popup, in pixels. This value takes precedence over the service-client-api.toast.width configuration option.	succeeded	function	A function called when the operation succeeds.	failed	function	A function called when the operation fails.
Name	Type	Description																						
		<table border="1"> <thead> <tr> <th data-bbox="1256 394 1312 428">Name</th> <th data-bbox="1312 394 1367 428">Type</th> <th data-bbox="1367 394 1448 428">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="1256 428 1312 831"></td> <td data-bbox="1312 428 1367 831"></td> <td data-bbox="1367 428 1448 831">true, sends the <b>subject</b>, <b>iconUrl</b>, <b>title</b>, <b>keyValues</b>, and <b>message</b> parameters to the <b>MyMessage</b> panel.</td> </tr> <tr> <td data-bbox="1256 831 1312 1453">width</td> <td data-bbox="1312 831 1367 1453">number</td> <td data-bbox="1367 831 1448 1453">Optional. The width of the custom toast popup, in pixels. This value takes precedence over the service-client-api.toast.width configuration option.</td> </tr> </tbody> </table>	Name	Type	Description			true, sends the <b>subject</b> , <b>iconUrl</b> , <b>title</b> , <b>keyValues</b> , and <b>message</b> parameters to the <b>MyMessage</b> panel.	width	number	Optional. The width of the custom toast popup, in pixels. This value takes precedence over the service-client-api.toast.width configuration option.													
Name	Type	Description																						
		true, sends the <b>subject</b> , <b>iconUrl</b> , <b>title</b> , <b>keyValues</b> , and <b>message</b> parameters to the <b>MyMessage</b> panel.																						
width	number	Optional. The width of the custom toast popup, in pixels. This value takes precedence over the service-client-api.toast.width configuration option.																						
succeeded	function	A function called when the operation succeeds.																						
failed	function	A function called when the operation fails.																						
Returns	A unique identifier																							

---

## triggerActivity

<b>Signature</b>	triggerActivity(succeeded, failed)									
<b>Description</b>	Triggers a fake activity to prevent the inactivity timer from closing the agent session.									
<b>Parameters</b>	<table border="1"><thead><tr><th>Name</th><th>Type</th><th>Description</th></tr></thead><tbody><tr><td>succeeded</td><td>function</td><td>A function called when the operation succeeds.</td></tr><tr><td>failed</td><td>function</td><td>A function called when the operation fails.</td></tr></tbody></table>	Name	Type	Description	succeeded	function	A function called when the operation succeeds.	failed	function	A function called when the operation fails.
Name	Type	Description								
succeeded	function	A function called when the operation succeeds.								
failed	function	A function called when the operation fails.								

## updateViewInApplicationMenuBar

<b>Signature</b>	updateViewInApplicationMenuBar(parameters, succeeded, failed) → {string}									
<b>Description</b>	Creates a custom view in the <b>Application Menu bar</b> region.									
<b>Parameters</b>	<table border="1"><thead><tr><th>Name</th><th>Type</th><th>Description</th></tr></thead><tbody><tr><td>name</td><td>string</td><td>A unique name for the custom view of the <b>Application Menu bar</b> that is to be created or updated. If a view with the given name already exists, it will be updated, otherwise, a new view will be created.</td></tr><tr><td>iconUrl</td><td>string</td><td>The URL of the icon you want to display in the custom view of the <b>Application Menu bar</b> region. This parameter is</td></tr></tbody></table>	Name	Type	Description	name	string	A unique name for the custom view of the <b>Application Menu bar</b> that is to be created or updated. If a view with the given name already exists, it will be updated, otherwise, a new view will be created.	iconUrl	string	The URL of the icon you want to display in the custom view of the <b>Application Menu bar</b> region. This parameter is
Name	Type	Description								
name	string	A unique name for the custom view of the <b>Application Menu bar</b> that is to be created or updated. If a view with the given name already exists, it will be updated, otherwise, a new view will be created.								
iconUrl	string	The URL of the icon you want to display in the custom view of the <b>Application Menu bar</b> region. This parameter is								

---

Signature	updateViewInApplicationMenuBar(parameters, succeeded, failed) → {string}		
	Name	Type	Description
			mandatory if <b>label</b> is not provided.
	label	string	The main textual content to be displayed for the custom view. This parameter is mandatory if <b>iconUrl</b> is not provided.
	shortLabel	string	Optional. A shorter version of the <b>label</b> that will be used in the shortened mode if <b>iconUrl</b> is not available.
	tooltip	string	Optional. The tooltip content to be shown when the mouse is hovered on the custom view.
	labelColor	string	Optional. The color of the label text in case-insensitive hex color code format, for example, #FFFFFF.
	backgroundColor	string	Optional. The background color of the region where icon and title are displayed. The format of the background color is case-

Signature	updateViewInApplicationMenuBar(parameters, succeeded, failed) → {string}		
	<b>Name</b>	<b>Type</b>	<b>Description</b>
			insensitive hex color code, for example, #FFFFFF. By default, it is usually the same color as the navigation bar.
	succeeded	function	A function called when the operation succeeds.
	failed	function	A function called when the operation fails.
Returns	View name if successful.		

### Sample request

```
genesys.wwe.service.system.updateViewInApplicationMenuBar({
  name: "view1",
  iconUrl: "https://cdn1.iconfinder.com/data/icons/free-social-media-12/32/
RSS_social_media-128.png",
  label: "Main content text",
  shortLabel: "Short text",
  tooltip: "Tooltip text",
  labelColor: "#FFFFFF",
  backgroundColor: "#000000"
}, succeeded, failed)'
```

### Sample response

```
{
  "request": "system.updateViewInApplicationMenuBar",
  "data": "view1",
  "userAgent": "WWE Server",
  "protocolVersion": 2
}
```

### updateToast

Signature	updateToast( <i>id</i> , <i>parameters</i> , succeeded, failed) → {boolean}
Description	Updates the specified toast.

Signature	updateToast( <i>id</i> , <i>parameters</i> , succeeded, failed) → {boolean}																			
<b>Parameters</b>	<table border="1"> <thead> <tr> <th data-bbox="824 323 1032 365">Name</th> <th data-bbox="1032 323 1240 365">Type</th> <th data-bbox="1240 323 1448 365">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="824 365 1032 604">id</td> <td data-bbox="1032 365 1240 604">string</td> <td data-bbox="1240 365 1448 604">The identifier of the toast to update. The identifier is returned by the popupToast method.</td> </tr> </tbody> </table>	Name	Type	Description	id	string	The identifier of the toast to update. The identifier is returned by the popupToast method.													
	Name	Type	Description																	
id	string	The identifier of the toast to update. The identifier is returned by the popupToast method.																		
<table border="1"> <thead> <tr> <th data-bbox="824 604 1032 646">Name</th> <th data-bbox="1032 604 1240 646">Type</th> <th data-bbox="1240 604 1448 646">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="824 646 1032 739">title</td> <td data-bbox="1032 646 1240 739">string</td> <td data-bbox="1240 646 1448 739">The title</td> </tr> <tr> <td data-bbox="824 739 1032 1251">iconUrl</td> <td data-bbox="1032 739 1240 1251">string</td> <td data-bbox="1240 739 1448 1251">The URL of the icon you want to display in the title bar of the custom toast popup.</td> </tr> <tr> <td data-bbox="824 1251 1032 1344">subject</td> <td data-bbox="1032 1251 1240 1344">string</td> <td data-bbox="1240 1251 1448 1344">Optional. The subject.</td> </tr> <tr> <td data-bbox="824 1344 1032 1436">message</td> <td data-bbox="1032 1344 1240 1436">string</td> <td data-bbox="1240 1344 1448 1436">Optional. The message.</td> </tr> <tr> <td data-bbox="824 1436 1032 1822">keyValues</td> <td data-bbox="1032 1436 1240 1822">object</td> <td data-bbox="1240 1436 1448 1822">Optional. JSON object used to fill the key value pair list. For</td> </tr> </tbody> </table>	Name	Type	Description	title	string	The title	iconUrl	string	The URL of the icon you want to display in the title bar of the custom toast popup.	subject	string	Optional. The subject.	message	string	Optional. The message.	keyValues	object	Optional. JSON object used to fill the key value pair list. For	object	
Name	Type	Description																		
title	string	The title																		
iconUrl	string	The URL of the icon you want to display in the title bar of the custom toast popup.																		
subject	string	Optional. The subject.																		
message	string	Optional. The message.																		
keyValues	object	Optional. JSON object used to fill the key value pair list. For																		

Signature	updateToast( <i>id</i> , <i>parameters</i> , <i>succeeded</i> , <i>failed</i> ) → {boolean}								
	Name	Type	Description						
			<table border="1"> <thead> <tr> <th data-bbox="1256 394 1312 428">Name</th> <th data-bbox="1312 394 1367 428">Type</th> <th data-bbox="1367 394 1446 428">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="1256 428 1312 663"></td> <td data-bbox="1312 428 1367 663"></td> <td data-bbox="1367 428 1446 663">example: {"key1" : "value one", "key2" : "value two", "key3" : "value three"}.</td> </tr> </tbody> </table>	Name	Type	Description			example: {"key1" : "value one", "key2" : "value two", "key3" : "value three"}.
	Name	Type	Description						
		example: {"key1" : "value one", "key2" : "value two", "key3" : "value three"}.							
	Array buttons	Each character string in this array becomes a button. All buttons are displayed as buttons, not hyperlinks, in the following order: [Button 2] [Button 3] ... [Button N] [Button 1].							
	boolean	If set to true, displays <b>Show</b> and <b>Dismiss</b> buttons and							

Signature	updateToast( <i>id</i> , <i>parameters</i> , succeeded, failed) → {boolean}								
	<table border="1"> <thead> <tr> <th data-bbox="824 327 1032 369">Name</th> <th data-bbox="1032 327 1240 369">Type</th> <th data-bbox="1240 327 1448 369">Description</th> </tr> </thead> </table>			Name	Type	Description			
	Name	Type	Description						
	<table border="1"> <thead> <tr> <th data-bbox="824 369 1032 432">Name</th> <th data-bbox="1032 369 1240 432">Type</th> <th data-bbox="1240 369 1448 432">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="824 432 1032 1157"></td> <td data-bbox="1032 432 1240 1157"></td> <td data-bbox="1240 432 1448 1157">pops up the current iframe if the <b>Show</b> button is pushed. If set to false, displays "OK" or custom buttons based on the parameter's buttons.</td> </tr> </tbody> </table>			Name	Type	Description			pops up the current iframe if the <b>Show</b> button is pushed. If set to false, displays "OK" or custom buttons based on the parameter's buttons.
	Name	Type	Description						
		pops up the current iframe if the <b>Show</b> button is pushed. If set to false, displays "OK" or custom buttons based on the parameter's buttons.							
succeeded	function	A function called when the operation succeeds.							
failed	function	A function called when the operation fails.							
Returns	true if the toast has been updated; false if the toast identifier has not been found.								