

# socialize.showLoginUI JS

## Description

Displays Gigya's Social Login plugin, which lists all the available providers' logos as login options, enabling the user to log into your site via their social network or webmail account.

### Notes:

- All new implementations of the login plugin must use version 2. Apply this by setting the *version* parameter in this API to 2.
- Make sure that all pages using **socialize.showLoginUI** have a `<!DOCTYPE html>` tag preceding the `<html>` tag of your pages for backward browser compatibility.

## Supporting Providers

- By default, the plugin displays the logos of the following providers: **Facebook, Twitter, Google+, LinkedIn, Yahoo!, Microsoft, Instagram, Odnoklassniki, Foursquare, renren, Tencent QQ, Sina Weibo, Vkontakte, AOL, WordPress, Blogger, Line, and WeChat.**
- The following providers are also supported and may be added explicitly: **Amazon, Kakao, Livedoor, mixi, Naver, Netlog, Orange France, PayPal, PayPalOAuth, Spiceworks, Xing, Yahoo! Japan .**
- Some providers require that the user provide permission to access their data on the first sign-in. Yahoo will require the user to provide explicit permission every time they sign in.
- You may customize the list of providers' logos that appear on the plugin using the *enabledProviders* and *disabledProviders* parameters or by using the *requiredCapabilities* parameter.

## Plugin Design and Options

This is the default design of the plugin:

To view the popup version, [click here](#).

The plugin may be embedded in the page or displayed as a popup at the center of the page. If the plugin is opened as a popup, then it automatically disappears at the end of the login process.

The plugin's design is customizable through the use of the parameters listed below.

In the [Social Login demo](#) page you will find a complete working example which uses the `socialize.showLoginUI` method. You may view the code, run it and view the outcome.

**Note:** If you plan on integrating the Login plugin, we highly recommend reading the [Social Login](#) guide, which is a step-by-step guide for implementing the Social Login process in your site.

## Syntax

# Parameters

The following table lists the available parameters:

Required	Name	Type	Description
	version	integer	Gigya offers a new improved version of the Login plugin. To use the new plugin set this parameter to "2". All new implementations are required to use the new version, although existing implementations without this parameter will continue to work.
	width	string	Sets the width you desire for the plugin. You may also use percentages to set the width, e.g., width: 80%, in which case the plugin will be responsive. The login plugin supports the percentage option from version 2, so using the <b>version</b> parameter is required.
	height	integer	Determines the height you desire for the plugin.
	containerID	string	An ID of a <DIV> element on the page in which you want to display the plugin. If this parameter is not provided then the plugin will be displayed in the body of the page. Please refer to <a href="#">Dialog</a> or <a href="#">Embedded</a> section for extended explanation.
	captionText	string	Sets the caption text. This parameter is relevant only when the plugin is in a popup mode (i.e. if the containerID/container class is not provided). If the containerID/container class is provided then the caption is not presented.
	context	object	A reference to a developer created object that will be passed back unchanged to the event handlers of any event triggered by the plugin. This object can be used to pass information to the event handlers. It will be passed as one of the fields of the <b>eventObj</b> received by the event handler (see extended explanation in the <b>"Plugin Events"</b> section).
	actionAttributes	JSON object	In Gamification your users receive points for actions they perform on your site, in this case logging in grants the user points. Action Attributes may be used to annotate actions with additional information, such as the section of the web site the action was performed on. <b>actionAttributes</b> contain a JSON object comprised of a series of attribute keys (categories) with associated values. You can also use a generic "tags" key.  No more than three values can be given, they can be with a single key or each have their own key.  For more information see <a href="#">Variants and Action Attributes</a> . Action attributes are later used to filter GM Plugins by a certain attribute. Example: [{"attribute key1": "value1"}, {"attribute key2": "value2"}, {"attribute key3": "value3"}]
	authCodeOnly	Boolean	This parameter is intended only for developers who wish to implement the "Web Server Flow" of the OAuth 2.0 standard. The default value of this parameter is 'false'. If set to 'true', you will <b>not</b> receive the user data in the onLogin event. The authCode contains a code that is intended to be used for invoking the OAuth 2.0 getToken end-point along with the grant_type parameter set to authorization_code.  The license could not be verified. License Certificate has expired!
	authFlow	string	Using this parameter you may specify that the login flow will use page redirects instead of using a popup. This gives a solution for environments where popups are unavailable (e.g., mobile web view controls). This parameter accepts two values: <ul style="list-style-type: none"> <li><b>"popup"</b> (default)</li> <li><b>"redirect"</b> - the login flow uses page redirects. When the login process completes successfully, the user is redirected to the URL specified by the <i>redirectURL</i> parameter (see above). If the <i>redirectURL</i> parameter is not specified, the user is redirected to the URL specified by the <i>redirectURL</i> parameter.</li> </ul>
	autoDetectUserProviders	string	This parameter, when specified and loading Facebook Connect is possible, and the user is detected as a Facebook user, the login plugin displays a large provider button at the top and the rest of the login buttons at the bottom. Currently it supports: <ul style="list-style-type: none"> <li><b>"facebook"</b> (requires Native SDK Capabilities).</li> </ul>
	buttonSize	integer	A number denoting the height of the buttons in pixels. Width will be set in proportion to the height.
	buttonsStyle	string	This parameter enables selecting one of the pre-defined design styles for the network buttons on the Login UI. Currently Gigya offers the following design styles: <ul style="list-style-type: none"> <li><b>"standard"</b> (default):   </li> <li><b>"fullLogo"</b>:   </li> <li><b>"fullLogoColored"</b>:   </li> <li><b>"signInWith"</b>:   </li> </ul> <p>The <b>buttonSize</b> parameter when using <b>signInWith</b> buttonStyle only supports the following sizes (defined in vertical height):</p> <ul style="list-style-type: none"> <li>20 px (translated to 40px in retina screens)</li> <li>25 px (translated to 50px in retina screens)</li> <li>30 px (translated to 60px in retina screens)</li> <li>35 px (translated to 70px in retina screens)</li> <li>40 px (translated to 80px in retina screens)</li> </ul> <p>Attempting to set sizes not listed here will result in a corrupted button being displayed within the UI.</p>
	cid	string	A string of maximum 100 characters length. The CID sets categories for transactions that can be used later for filtering reports generated by Gigya in the "Context ID" combo box. The CID allows you to associate the report information with more than 100 different context IDs.

**Note:** This parameter overrides the value of the identical parameter in [Global Conf](#) (the `globalConf` parameter is not set for the method, the value from Global Conf is used).

customButtons

an array of JSON objects

This parameter is used to add multiple customButton objects to the Gigya Social Login widget either directly or when using Screen-Sets. You can use customButton objects to add custom login options for any combination of OpenID Cor

[Expand a code example](#)

```
customButtons= [  
  {  
    // customButton object #1  
    "type": "saml",  
    "providerName": "Gateway One",  
    "idpName": "testIdp-gig01",  
    "iconURL": "//developers.gigya.com/download/attachment",  
    "logoURL": "",  
    "lastLoginIconURL": "//developers.gigya.com/download/attachment",  
    "position": "3"  
  },  
  {  
    // customButton object #2  
    "type": "saml",  
    "providerName": "Gateway Two",  
    "idpName": "testIdp-gig02",  
    "iconURL": "//developers.gigya.com/download/attachment",  
    "logoURL": "",  
    "lastLoginIconURL": "//developers.gigya.com/download/attachment",  
    "position": "4"  
  },  
  {  
    // customButton object #3 - IMPORTANT: openID Only works in  
    "type": "openID",  
    "providerName": "Flickr",  
    "iconURL": "https://s3.amazonaws.com/wikifiles.gigya.com",  
    "logoURL": "https://s3.amazonaws.com/wikifiles.gigya.com",  
    "lastLoginIconURL": "https://s3.amazonaws.com/wikifiles.gigya.com",  
    "position": "5",  
    "openIDURL": "https://me.yahoo.com/$USERNAME$"br/>  },  
  {  
    // customButton object #4 - IMPORTANT: oidc Only works in  
    "type": "oidc",  
    "providerName": "Gateway Four",  
    "opName": "testOIDC-gig04",  
    "iconURL": "http://developers.gigya.com/download/attachment",  
    "logoURL": "",  
    "lastLoginIconURL": "//developers.gigya.com/download/attachment",  
    "position": "6"  
  }  
];  
gigya.accounts.showScreenSet({screenSet: 'Default-RegistrationI
```

It is important to note that any unique user can only be connected to a single SAML or OIDC provider (though they may be connected to one of each). If using showLoginUI version 1 to support OpenID, you can only pass

When using a customButtons array with showAddConnectionsUI, you should only include a single button object in the array. If you include multiple custom button objects, only the first button defined will be displayed in th

For more information see [Login](#).

customButton

customButton object

This parameter is deprecated. Use customButtons. above.

You may define any OpenID provider as an authentication destination - read more about [Adding a Custom OpenID Provider](#).

You may also define a SAML login provider - read more about [Adding a SAML Login Provider](#).

deviceType	string	<p>Specifies the type of the device the UI is to be displayed on (see the "pagingButtonStyle" parameter). The possible values are:</p> <ul style="list-style-type: none"> <li>• <b>desktop</b> (default)</li> <li>• <b>mobile</b></li> <li>• <b>auto</b> - When this value is used, you must also include the following meta tag in the &lt;head&gt;-section of your site for mobile devices to be automatically recognized:</li> </ul> <pre>// This must be the first item beneath the // &lt;title&gt;&lt;/title&gt; section of your page! &lt;meta name="viewport" content="width=device-width"&gt;</pre> <p>Otherwise, the device will be recognized as a <b>desktop</b> device.</p>
disabledProviders	string	<p>A comma-delimited list of provider names to exclude in the method execution. This parameter gives the possibility to specify providers to which you do not want this method to apply. If you do not set this parameter, by default, no provider. For example, if you would like the method to apply only to Twitter, define: <code>disabledProviders: "twitter"</code>.</p> <p>Valid provider names include: <code>amazon, aol, blogger, facebook, foursquare, googleplus, instagram, kakao, line, linkedin, livedoor, messenger, mixi, naver, netlog, odnoklassniki, orangefrance, paypaloauth, qq, renren, sina, spiceworks, tv</code> however, for backward compatibility, either can be used).</p> <p><b>Note:</b> This parameter overrides the value of the identical parameter in <b>Global Conf</b> (the <code>g</code> parameter is not set for the method, the value from Global Conf is used).</p>
enabledProviders	string	<p>A comma-delimited list of provider names to include in the method execution. This parameter gives the possibility to apply this method only to a subset of providers of your choice. If you do not set this parameter, by default all the provider. For example, if you would like the method to apply only to Twitter, define: <code>enabledProviders: "twitter"</code>.</p> <p>Valid provider names include: <code>amazon, aol, blogger, facebook, foursquare, googleplus, instagram, kakao, line, linkedin, livedoor, messenger, mixi, naver, netlog, odnoklassniki, orangefrance, paypaloauth, qq, renren, sina, spiceworks, tv</code> however, for backward compatibility, either can be used).</p> <p><b>Note:</b> This parameter overrides the value of the identical parameter in <b>Global Conf</b> (the <code>g</code> parameter is not set for the method, the value from Global Conf is used).</p>
extraFields	string	<p>This parameter accepts a comma-separated list of additional data fields to retrieve. The current valid values are: <code>languages, address, phones, education, honors, publications, patents, certifications, professionalHeadline, bio, in_likes, followersCount, followingCount, name, username, educationLevel, locale, verified, frank, timezone, and samlData</code>.</p> <p><b>Note:</b> Before your application can retrieve Facebook data, the user must grant your application with access. Please make sure you have checked the check boxes that enable retrieving the relevant fields from Facebook in the <b>Permissions</b> section of the Facebook app.</p>
facebookExtraPermissions	string	<p>A comma-delimited list of Facebook extended permissions to request from the user. This parameter gives the possibility to request extended permissions in addition to the permissions that Gigya is already requesting. Please refer to Facebook's extended permissions page for the complete list of permissions. For example, if you wish to RSVIP to events on the user's behalf and to send text messages to the user define: <code>facebookExtraPermissions: "rsvp_event,sms"</code></p> <p><b>Note:</b> This parameter overrides the value of the identical parameter in <b>Global Conf</b> (the <code>g</code> parameter is not set for the method, the value from Global Conf is used).</p>
forceAuthentication	Boolean	<p>The default value of this parameter is 'false'. If it is set to 'true', the user is forced to provide their social network credentials during login - even if the user is already connected to the social network. This parameter is currently supported by different: Facebook expects the current user to enter their password, and will not accept a different user name. Other networks prompt the user to re-authorize the application or allow a different user to log in.</p>
googleExtraPermissions	string	<p>This parameter gives the possibility to request extended permissions in addition to the permissions that Gigya is already requesting. The supported values are: "wallet" - for Google wallet permissions.</p> <p><b>Note:</b> This parameter overrides the value of the identical parameter in <b>Global Conf</b> (the <code>g</code> parameter is not set for the method, the value from Global Conf is used).</p>
googlePlayAppID	string	<p>The objective of this parameter is to support Over The Air app installs for Android devices during Google+ login. Set this parameter with the package name of your Android app (for example: "com.yourdomain.app"). As a result, after sign up website. As a preliminary step you'll need to Utilize Google+ Native Android Sign-on on your Android app. The package name passed to this parameter is the same one you enter when enabling the Google+ API.</p>
headerText	string	<p>Sets the plugin's header text.</p>
includeAllIdentities	Boolean	<p>The default value of this parameter is 'false'. If set to 'true', you will receive all the user's identities, including those with expired sessions. Each entry will have an attribute that will be 'true' when the session has expired for that provider (</p>
lastLoginButtonSize	integer	<p>This parameter is relevant only if the <code>lastLoginIndication</code> parameter is set to 'welcome'. This parameter defines the size of the login network button on the 'welcome back' screen. The parameter defines the button height in pixels.</p>
lastLoginIndication	string	<p>The Login plugin gives an indication of the last provider to which the user was logged in. You may change the way the last login provider is presented using this parameter. Currently supported values for this parameter are:</p> <ul style="list-style-type: none"> <li>• <b>border</b> (default) - Last login provider's logo will be displayed first and will be surrounded with a green border.</li> <li>• <b>welcome</b> - This option changes the view of the plugin. The plugin will show a "Welcome back" view, with the user name and with only the last login provider's logo as a login option. The view includes a "Not you?" link next to the "Not you?" is clicked, the plugin sets <b>forceAuthentication</b> to true for the following login attempts on the same page.</li> <li>• <b>none</b> - There will be no indication for last login provider.</li> </ul> <p>You may find more information about this feature and screenshots in the Last Logged In section of the Developer's Guide.</p>
onButtonClicked	function ref	<p>A reference to an event handler function that will be called when one of the plugin's buttons is clicked.</p>
onClose	function ref	<p>A reference to an event handler function that will be called in one of the following scenarios:</p> <ul style="list-style-type: none"> <li>• If the plugin is displayed as a popup the event will be fired when the plugin closes.</li> <li>• If the plugin is embedded in page the event will be fired when the user clicks a button that indicates the dialog should be closed. In this case the dialog will not hide itself automatically.</li> </ul>
onError	function ref	<p>A reference to an event handler function that will be called when an error occurs.</p>
onLoad	function ref	<p>A reference to an event handler function that will be called when the plugin has finished drawing itself.</p>
onLogin	function ref	<p>A reference to a function that is called when the user is successfully authenticated by Gigya.</p>
pagingButtonStyle	string	<p>Controls the looks of the "next/previous" buttons. The possible values are:</p> <ul style="list-style-type: none"> <li>• <b>arrows</b> - The familiar arrow buttons:  </li> <li>• <b>newArrows</b> (default) - Arrows that are slightly further away from the login buttons, for better usability in a touch input:  </li> <li>• <b>floating</b> - Floating buttons optimized for touch input in small screen sizes:  </li> <li>• <b>auto</b> - This value sets the buttons to <b>floating</b> if the <code>deviceType</code> parameter has been set to 'mobile' or the device has been recognized as mobile (see <code>deviceType</code> for details), and the width of the Login Plugin UI has been set to</li> </ul>

pendingRegistration	Boolean	The default value of this parameter is <b>true</b> . The default behavior - when a new user logs-in (registers), his new Gigya account is not considered final until <code>socialize.notifyRegistration</code> is called. While being not-final, the identities associate a new user logs-in (registers), his new Gigya account is final immediately.
		<b>Note:</b> The value of this parameter overrides the value of the <code>newUsersPendingRegistr</code> object.
redirectURL	URL	A URL to which to redirect the user when the login completes successfully. The following additional parameters will be appended to the URL string: <code>UID, UIDSig, timestamp, loginProvider, loginProviderUID, nickname, photoURL, thumbnailURL, firstName, lastName, gender, birthDay, birthMonth, birthYear, email, country, state, city, zip, profileURL, proxiedEmail, provider</code> . These parameters are described in the <a href="#">Redirect URL</a> page.  If the <code>redirectURL</code> parameter is supplied, the <code>onLogin</code> global event will not be called.
		<b>Note:</b> We strongly advise providing a secure <b>https</b> URL.
requiredCapabilities	string	A comma separated list of any of the following capabilities: <b>Login, Notifications, Actions, Friends, Status, Photos, Contacts</b> . Only providers that are available and support all the required capabilities will be visible.
sessionExpiration	integer	This parameter defines the length of time that Gigya should keep the user's login session valid. It can be configured via Global Configuration, via an individual API call, or left empty. If no value is specified, the default values are <b>0</b> or <b>-2</b> , value via individual API calls override the global configuration.  The expected values are: <ul style="list-style-type: none"> <li><b>0</b> - Session expires when the browser closes. This is the default behavior when RaaS is enabled in your site. This behavior is dependent upon the browser's cookie handling procedures, i.e., Chrome keeps processes running if processes are terminated.</li> <li><b>-1</b> - Session ends after a 60 second period: Gigya gives you the option of creating a cookie that is stored on the site visitor's client (browser), allowing the site to control the session length within this 60 second window, after which (such as credit card details), and the session should be short, with the option of restarting the duration when users perform actions. Useful if you always set the session expiration via individual API methods or with each request session expiration cookie.</li> <li><b>-2</b> - Session is valid forever. This is the default behavior when RaaS is not enabled in your site.</li> <li>Any custom integer - Defines the number of seconds the session is active, e.g., <b>3600</b> (one hour).</li> </ul>
showTermsLink	Boolean	Show or hide the <b>Terms</b> link. Clicking the "Terms" link opens Gigya's Legal Notices page.
showTooltips	Boolean	The default value of this parameter is <b>true</b> . Determines whether a tooltip will be displayed when the mouse hovers over a social network icon. The tooltip presents the social network's full name.
showWhatsThis	Boolean	Indicates whether to show or hide a <b>What's this</b> link. When the user rolls over the link, a hint will pop up describing the Login plugin. By default the value of this parameter is 'false'. In order for the "What's this" link to appear, you must set this parameter to <b>true</b> .  
whatsThisText	string	This parameter is relevant only if the <code>showWhatsThis</code> parameter is set to <b>true</b> . Using this parameter you may define the text that will appear in the pop-up hint (see screenshot above). The default text is: "You can use your account from".
facepilePosition	string	<b>Deprecated</b> . This parameter is ignored by the Gigya API.
useHTML	Boolean	<b>Deprecated</b> . The Login plugin is available in both Flash and HTML versions. By default, the HTML version is used. You may override the default behavior. Setting this parameter with the value 'false', will force Gigya to use the Flash version.
includeIRank	Boolean	<b>Deprecated</b> . This parameter's default value is 'false'. If set to 'true' you will receive the user's <code>iRank</code> (influencer rank) in the response <code>User</code> object .

Unable to render {include} The included page could not be found.

## Plugin Events

An Event Handler is a JavaScript function with the following signature:

**functionName(eventObj)**

The single argument, **eventObj**, contains information about the event and has different fields for different events.

The following tables specify the list of fields available in the **eventObj** for each event:

### onLoad Event Data

Field	Type	Description
eventName	string	The name of the event.
source	string	The source plugin that generated this event. The value of this field is the name of the plugin's API method, e.g., 'showLoginUI', 'showCommentsUI', etc.
context	object	Any user defined data passed as a value of the context parameter by the application to the API method, it will be returned as an accessible property in the response. If no context parameter was passed, it will not be returned.

### onError Event Data

Field	Type	Description
-------	------	-------------

eventName	string	The name of the event.	
source	string	The source plugin that generated this event. The value of this field is the name of the plugin's API method, e.g., 'showLoginUI', 'showCommentsUI', etc.	
context	object	Any user defined data passed as a value of the context parameter by the application to the API method, it will be returned as an accessible property in the response. If no context parameter was passed, it will not be returned.	
errorCode	integer	The result code of the operation. Code '0' indicates success, any other number indicates failure. For a complete list of error codes, see the <a href="#">Error Codes table</a> .	
errorMessage	string	A short textual description of an error, associated with the errorCode, for logging purposes.	
errorDetails	string	This field will contain the exception info, if available.	

## onClose Event Data

Field	Type	Description
eventName	string	The name of the event.
source	string	The source plugin that generated this event. The value of this field is the name of the plugin's API method, e.g., 'showLoginUI', 'showCommentsUI', etc.
context	object	Any user defined data passed as a value of the context parameter by the application to the API method, it will be returned as an accessible property in the response. If no context parameter was passed, it will not be returned.

## onLogin Event Data

Field	Type	Description
eventName	string	The name of the event: 'login'.
source	string	The source plugin that generated this event. The value of this field is the name of the plugin's API method, in this case 'showLoginUI'. Note: this field will not be available if the source of this event is not a plugin (e.g., if the source is a <a href="#">socialize.login</a> API call).
context	object	The context object passed as a parameter to the plugin/method that triggered this event, or null if no object was passed.
loginMode	string	The type of login: <ul style="list-style-type: none"> <li><i>standard</i> - the user is logging into an existing account.</li> <li><i>reAuth</i> - the user is proving ownership of an account by logging into it.</li> </ul>
provider	string	The name of the provider that the user used in order to login (e.g., "Facebook"). Note: If this event is fired as a result of a <a href="#">socialize.notifyLogin</a> call, i.e., the user was authenticated by your site, the <i>provider</i> field will be set to "site".
UID	string	The User ID that should be used for login verification*. <div style="border: 1px solid #007bff; padding: 5px; margin-top: 10px;"> <p><b>Note:</b> The UID string must be encoded using the <code>encodeURIComponent()</code> function before sending it from your client to your server.</p> </div>
UIDSignature	string	The signature that should be used for login verification, as described under the <a href="#">Validate the UID Signature in the Social Sign-On Process</a> section.
user	User object	A <a href="#">User</a> object with updated information for the current user.
signatureTimestamp	string	The GMT time of the response in UNIX time format (i.e., the number of seconds since Jan. 1st 1970). The time stamp should be used for login verification, as described under the <a href="#">Validate the UID Signature in the Social Sign-On Process</a> section.

\* To learn more about login verification, please refer to the [Validate the UID Signature in the Social Sign-On Process](#) section in the [Security](#) page of the Developer's Guide.

## onButtonClicked Event Data

Field Name	Type	Description
eventName	string	The name of the event.
source	string	The source plugin that generated this event. The value of this field is the name of the plugin's API method, e.g., 'showLoginUI', 'showCommentsUI', etc.
context	object	Any user defined data passed as a value of the context parameter by the application to the API method, it will be returned as an accessible property in the response. If no context parameter was passed, it will not be returned.
button	customButton object	The object representing the custom button that has been clicked. If the clicked button is not a custom button, no customButton object will be returned.

## Global Event Triggered

By using this plugin, the `onLogin` global event may be triggered (the `onLogin` global event is fired when a user successfully logs in to Gigya). To register an event handler, use the `socialize.addEventHandlers` API method.

You can read more about [onLogin event data](#) and [Events](#).

## Code Sample

```
var context = {
  msg: 'This is my params.context.msg'
};
var params = {
  captionText: 'This is my caption text',
  headerText: 'This is my header content',
  showTermsLink: false,
  redirectURL: 'http://www.yourNewSite.com',
  context: context,
};
params['onLoad'] = function(evt) {
  evt['msg'] = 'After onLoad';
  var msg = 'Event name is : ' + evt.eventName + '\n';
  msg+= 'evt[\'msg\'] is : ' + evt['msg'] + '\n';
  msg+= 'context.msg is : ' + evt['context']['msg'];
  alert(msg);
};
gigya.socialize.showLoginUI(params);
```

### Notes:

- This sample is not meant to be fully functional code. For brevity's sake, only the code required for demonstrating the API call itself is presented.
- To run the code on your own domain, add your Gigya API key to the `gigya.js` URL. A Gigya API key can be obtained on the [Site Dashboard](#) page on Gigya's website. Please make sure that the domain from which you are loading the page is the same domain name that you used for generating the API key.