Login

The Login add-on allows users to log into your site using their social identity. Use it if you want to use Gigya as your Social Login service.

The add-on displays the logos of all the available providers and enables the user to log into your site through the social network or webmail account of his choice.

When the user logs in successfully, the onLogin global event is fired. Refer to the Events page for information about event handling.

Implementation

The API method that displays the add-on is socialize.showLoginUI.

If you plan on integrating the Login add-on, please read the Social Login guide, which is a step-by-step guide for implementing the social login process in your site.

Working Example

In the Login Demo page you will find a working example that implements the framework of a login algorithm using the Login add-on. You may view the code, run it and view the outcome.

UI Customizations

The simplest way to customize the Login add-on is through the usage of the **Login Wizard** . The wizard lets you customize the visual aspects of the Login add-on design, view, and grab the customized code.

Alternatively, you may customize the add-on through the use of the socialize.showLoginUI method parameters (see the params table in the reference page).

Some of the add-on's adjustable parameters:

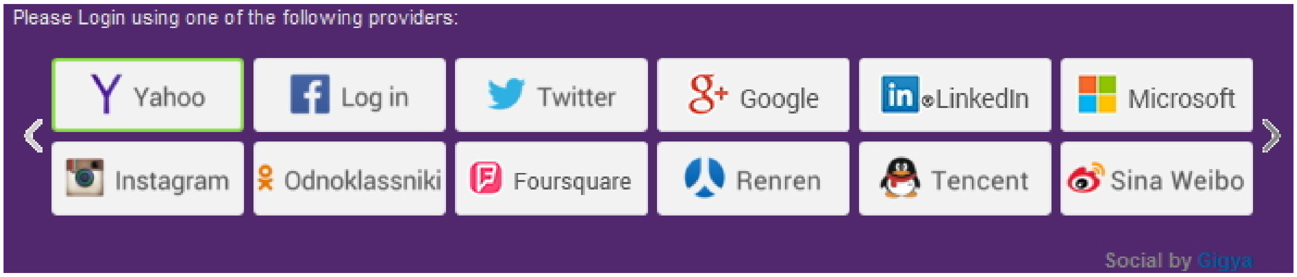
- The size of the add-on.
- Which providers' logos it displays, out of the available providers: **Amazon**, **AOL**, **Blogger**, **Facebook**, **Foursquare**, **GooglePlus**, **Instagram**, **Kakao**, **LINE**, **LinkedIn**, **Livedoor**, **Messenger**, **mixi**, **Naver**, **Netlog**, **Odnoklassniki**, **Orange France**, **PayPalOAuth**, **Tencent QQ**, **renren**, **Sina Weibo**, **Spiceworks**, **Twitter**, **VKontakte**, **WeChat**, **WordPress**, **Xing**, **Yahoo!**, **Yahoo! Japan**.(Note: messenger has been replaced by microsoft, however, for backward compatibility, either can be used)..
- The size of the logo buttons.
- The style of the logo buttons (you can select one of the predefined styles).
- Whether the add-on is shown as a popup (default) or embedded in the web page.
- Colors: add-on background, frame, text, link, navigation arrow, caption background, caption text, button text.
- Version - Version 2 of the login add-on introduced improved and responsive design for the add-on. While new add-on implementations are automated to version 2, it is also possible to upgrade to version 2 by passing the version parameter to the Socialize.showLoginUI method, or through the 'properties' section of the UI Builder. Note that upgrading to version 2 may change the add-on's layout.

Changing the Design - Code Example:

In the following example we display on screen a Login add-on with changed design and settings:

```
<html>
<head>

<!-- Including the gigya.js script: -->
<script type="text/javascript"
src="https://cdns.gigya.com/js/gigya.js?apikey=INSERT-YOUR-APIKEY-HERE"
></script>
<script type="text/javascript">

// Changing the default look and behavior of the add-on using the 'params'
object:
var params =
{
    showTermsLink:false, // 'terms' link is hidden
    headerText:"Please Login using one of the following providers:", //
adding header text
    height:145, // changing default add-on size
    width:700,  // changing default add-on size
    cid:'',
    containerID:"loginDiv", // The add-on will embed itself inside the
"loginDiv" DIV (will not be a popup)

    // Changes to the default design of the add-on's design

    //     Background color is changed to purple, text color to gray and
button size is set to 40 pixels:
    UIConfig:'<config><body><texts
color="#DFDFDF"></texts><controls><snbuttons
buttonsize="40"></snbuttons></controls><background
background-color="#51286D"></background></body></config>',

    // Change the buttons design style to the 'fullLogo' style:
    buttonsStyle:'fullLogo',

    // After successful login - the user will be redirected to
"https://www.MySite.com/welcome.html" :
    redirectURL: "https://www.MySite.com/welcome.html"
}
</script>
</head>
<body>
<div id="loginDiv"></div>
<script type="text/javascript">
    gigya.socialize.showLoginUI(params);
</script>
</body>
</html>
```

The output of this code:



Please Login using one of the following providers:

In the **JavaScript** implementation:

Following a successful login, the user will be redirected to "www.MySite.com/welcome.html". The following additional parameters will be appended to the URL string: *UID, UIDSig, timestamp, loginProvider, nickname, photoURL, thumbnailURL, firstName, lastName, gender, birthDay, birthMonth, birthYear, email, country, state, city, zip, profileURL, proxiedEmail, provider.*

For example:

```
http://www.MySite.com/welcome.html?zip=&
photoURL=http%3A%2F%2Fprofile%2Eak%2Efbcdn%2Enet%2Fprofile5%2F1719%2F30%2F
s722723209%5F4728%2Ejpg&
nickname=Assaften%20Esh&profileURL=http%3A%2F%2Fwww%2Efacebook%2Ecom%2Fpro
file%2Ephp%3Fid%3D722723209&
birthMonth=10&loginProvider=twitter&loginProviderUID=65e&country=&thumbnai
lURL=&lastName=Esh&signature=XGcp4tmIVvQNCqG9pET7XbHaies%3D&
firstName=Assaften&provider=twitter&gender=&birthYear=1965&timestamp=2009%
2D12%2D01%2011%3A05%3A33&
UIDSig=&state=&UID=123&email=&city=&birthDay=11&
proxiedEmail=apps%2B51244333578%2E722723209%2Edb978474083d884bcb0ba35d313c
4c8f%40proxymail%2Efacebook%2Ecom
```

The parameters are equivalent to the User object fields. Please find the parameters' description in the User object reference page.

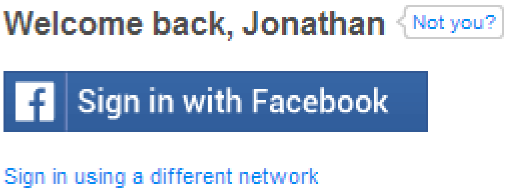
Last Login Indication

The Login add-on gives an indication of the last provider to which the user was logged in, so as to provide an improved user experience. The add-on displays the relevant provider's logo first (before the rest of the logos) and that logo is surrounded by a green border. In the following example, the user has previously logged in via Facebook:

You may change this default behavior by using the *lastLoginIndication* parameter (see the socialize.showLoginUI API method parameters reference):

1. Set the parameter to 'none' to supply no indication as to the last login provider.

Set it to 'welcome' to replace the default add-on view with a personalized 'welcome back' view:



Welcome back, Jonathan

In this view, only the last login provider's logo appears as a login option (in this example, Facebook). You may change the button size using the *lastLoginButtonSize* parameter, which defines the button height in pixels. In addition, this view shows:

- The user's name.
- A "Not you?" link next to the user name.
- A "Connect using a different service" link.

Both links lead to the default view of the add-on.

Adding a Custom OpenID Provider

You can add to the Login add-on your own custom OpenID provider, by defining a customButtons array in the socialize.showLoginUI params.

Running Example

In this example we add Flickr as the new custom provider:

The Example Code

```
<html>
<head>

<!-- including the gigya.js script -->
<style>
    body {font-family:arial;font-size:11px}
</style>
<SCRIPT type="text/javascript" lang="javascript"

src="https://cdns.gigya.com/js/gigya.js?apikey=2_Y82PzwJ_chSFImHXaIDJClnLy
Jzmk-VFOavSsaNTzl6m901s_NNxRAS0xJ3bd3_N">
    {
       enabledProviders:
'facebook,twitter,googleplus,linkedin,yahoo,messenger,aol,foursquare,vkont
akte,renren,QQ,Sina'
    }
</SCRIPT>
<script type="text/javascript">

// Adding Flickr as a custom OpenID provider using the 'params' object:
customButtons= [
    {   // Flickr Button
        "type": "openID",
        "providerName":"Flickr",
        "iconURL":
"https://s3.amazonaws.com/wikifiles.gigya.com/images/flickr.png",
        "logoURL":
"https://s3.amazonaws.com/wikifiles.gigya.com/images/flickr_large.png",

"lastLoginIconURL":"https://s3.amazonaws.com/wikifiles.gigya.com/images/fl
ickr.png",
        "position":"4",
        "openIDURL": "https://me.yahoo.com/$USERNAME$"
    }
];
var params =
{
    'customButtons': customButtons,
    "showTermsLink":false, // 'terms' link is hidden
    "containerID":"loginDiv" // The component will embed itself inside the
"loginDiv" DIV (will not be a popup)
}
</script>
</head>
<body>
<div id="loginDiv"></div>
<script type="text/javascript">
    gigya.socialize.showLoginUI(params);
</script>
</body>
</html>
```

Notes:

In order to make the above code work in your environment, please note:

- The API key in the sample will only work on *http://localhost/...*
- To load the page from your domain, modify the value of the "APIKey" field in the code to your own Gigya API Key. A Gigya API Key can be obtained from 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.
- If you are using https, be sure to further adjust the JS API url to: **https**://**cdns**.gigya.com/js/gigya.js?apikey=<Your_API_Key>.

Custom Button Parameters

To view the customButtons array documentation, click here.

Adding a SAML Login Provider

SAML (Security Assertion Markup Language) is a protocol for exchanging authentication and authorization data between separate systems. Read more about SAML Login and its Configuration .

You can add to the Login add-on a custom SAML Login button by defining a custom button using the customButtons array in the socialize.showLoginUI params.

Running Example

In this example we add Media ID as the new custom SAML provider:

The Example Code

```
<html>
<head>

<!-- including the gigya.social.js script -->
<style>
    body {font-family:arial;font-size:11px}
</style>
<SCRIPT type="text/javascript" lang="javascript"

src="https://cdns.gigya.com/js/gigya.js?apikey=2_Y82PzwJ_chSFImHXaIDJClnLy
Jzmk-VFOavSsaNTzl6m901s_NNxRAS0xJ3bd3_N">
        {
            enabledProviders:
'facebook,twitter,googleplus,linkedin,saml,yahoo,messenger,aol,foursquare,
vkontakte,renren,QQ,Sina'
        }
</SCRIPT>
<script type="text/javascript">

// Adding testShib-gig02 as a SAML login provider using the 'params'
object:
customButtons= [
    {
     "type": "saml",
     "providerName":"testShib-gig02",
        "idpName":"testShib-gig02",
        "iconURL":
"//developers.gigya.com/download/attachments/8563756/loginButton.png",
        "logoURL":
"//developers.gigya.com/download/attachments/8563756/loginButton.png",

"lastLoginIconURL":"//developers.gigya.com/download/attachments/8563756/lo
ginButton.png",
        "position":"4"
    }
];
var params =
{
    "customButtons": customButtons, // Includes the customButtons array
    "showTermsLink":false, // 'terms' link is hidden
    "containerID":"loginDiv", // The component will embed itself inside the
"loginDiv" DIV (will not be a popup),
    width:200
}
</script>
</head>
<body>
<div id="loginDiv"></div>
<script type="text/javascript">
    gigya.socialize.showLoginUI(params);
</script>
```

```
</body>
</html>
```

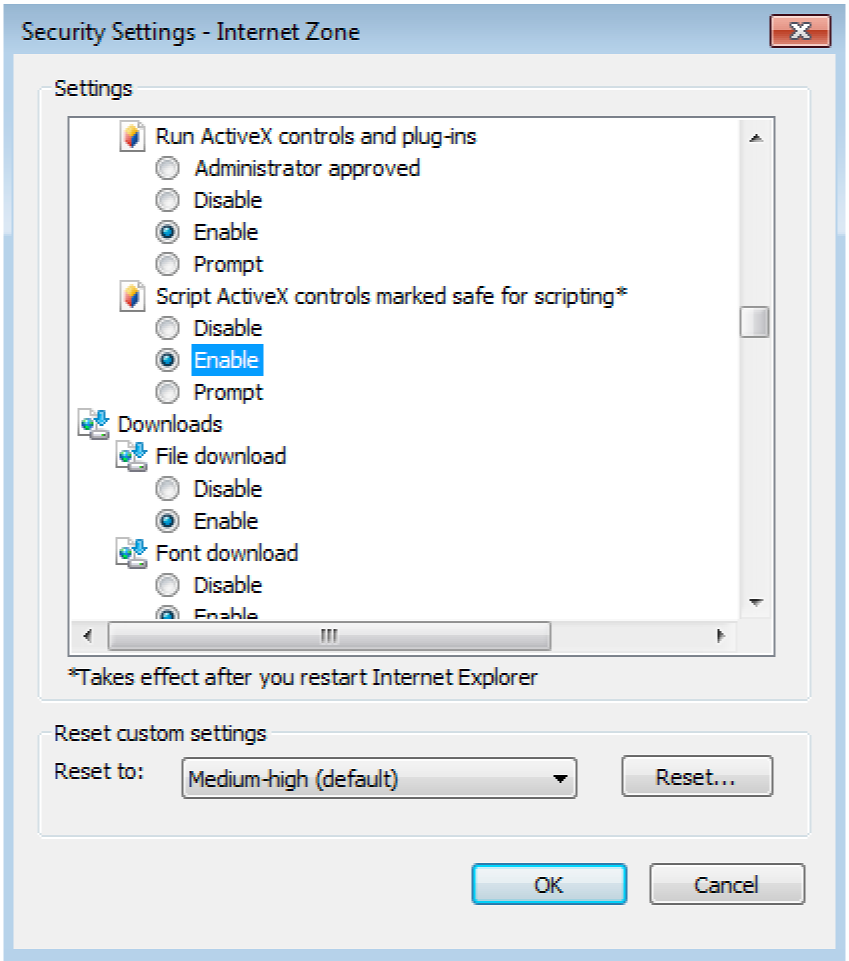
Notes:

In order to make the above code work in your environment, please note:

- The API key in the sample will only work on *http://localhost/...*
- To load the page from your domain, modify the value of the "APIKey" field in the code to your own Gigya API Key. A Gigya API Key can be obtained from 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.
- If you are using https, be sure to further adjust the JS API url to: **https**://**cdns**.gigya.com/js/gigya.js?apikey=<Your_API_Key>.

Login UI on Internet Explorer

In order for the your Login add-on to work, you must have the "Script ActiveX Controls Marked as Safe For Scripting" option enabled in your Internet Explorer security settings:



Security Settings - Internet Zone

Settings

Run ActiveX controls and plug-ins
○ Administrator approved
○ Disable
● Enable
○ Prompt

Script ActiveX controls marked safe for scripting*
○ Disable
● Enable
○ Prompt

Downloads

File download
○ Disable
● Enable

Font download
○ Disable
● Enable

*Takes effect after you restart Internet Explorer

Reset custom settings

Reset to: Medium-high (default)