

# Social Login

Gigya's Social Login is an authentication system that allows users to register and login to your site using their social network accounts, such as Facebook, Twitter, Google, Yahoo, LinkedIn and more. The login is secure and simple, providing users with an easy way to use your site without having to create yet another password, granting you permissions to store their publicly-available user data. The implementation proposed in this document is targeted for sites that already have their own user management system, and wish to offer Social Login (login/register through social networks) side by side with their existing site login/registration.

Some of the steps in this implementation guide are marked as optional, but we recommend implementing all the steps.

For a quick start implementation and a live demo of the social login flow, see the [Basic Social Login](#) working code example.

## Implementation

When a user opts to login or register to your site, they are offered to authenticate using either Gigya's Social Login or the site Login/Register form. Each of these two options leads to a separate logic flow.

The following two sections provide step-by-step instructions for implementing these two logic flows:

- [Integrating Gigya's Social Login](#) - the flow deriving from clicking on Gigya's Login plugin. Before implementing, please refer to the [Screensets](#), which illustrate this flow from the user point of view.
- [Site Login - Synchronizing with Gigya Service](#) - the flow deriving from using existing site Login/Register form.

Below these two sections you may find sections dealing with complementary implementation issues:

- [Logging Out](#)
- [Sessions Expiration](#)
- [Using Social Plugins to Initiate Site Login](#)
- [Adding Connections](#)
- [Error Handling](#)

We also recommend reading the documentation for [socialize.showLoginUI](#) before implementation, to review the options for customizing the Social Login's user interface.

The screenshot shows a registration interface with two main sections. On the left, a box titled "Quickly register with your social network:" contains six buttons: Facebook Log in, Twitter, LinkedIn, Google, Yahoo, and SMS Login. Below this box, a red arrow points to the text "Social Login". On the right, a section titled "Register with your email and password:" contains input fields for Email, First name, Last name, Password, and Confirm password. Below these fields is a checkbox for "I have read and understood the Terms of Use" and a blue "Submit" button. At the bottom right, there is a link "Have an account already?". A red arrow points from the "Social Login" text to the "Site Registration" text, which is positioned below the "or" separator between the two registration options.

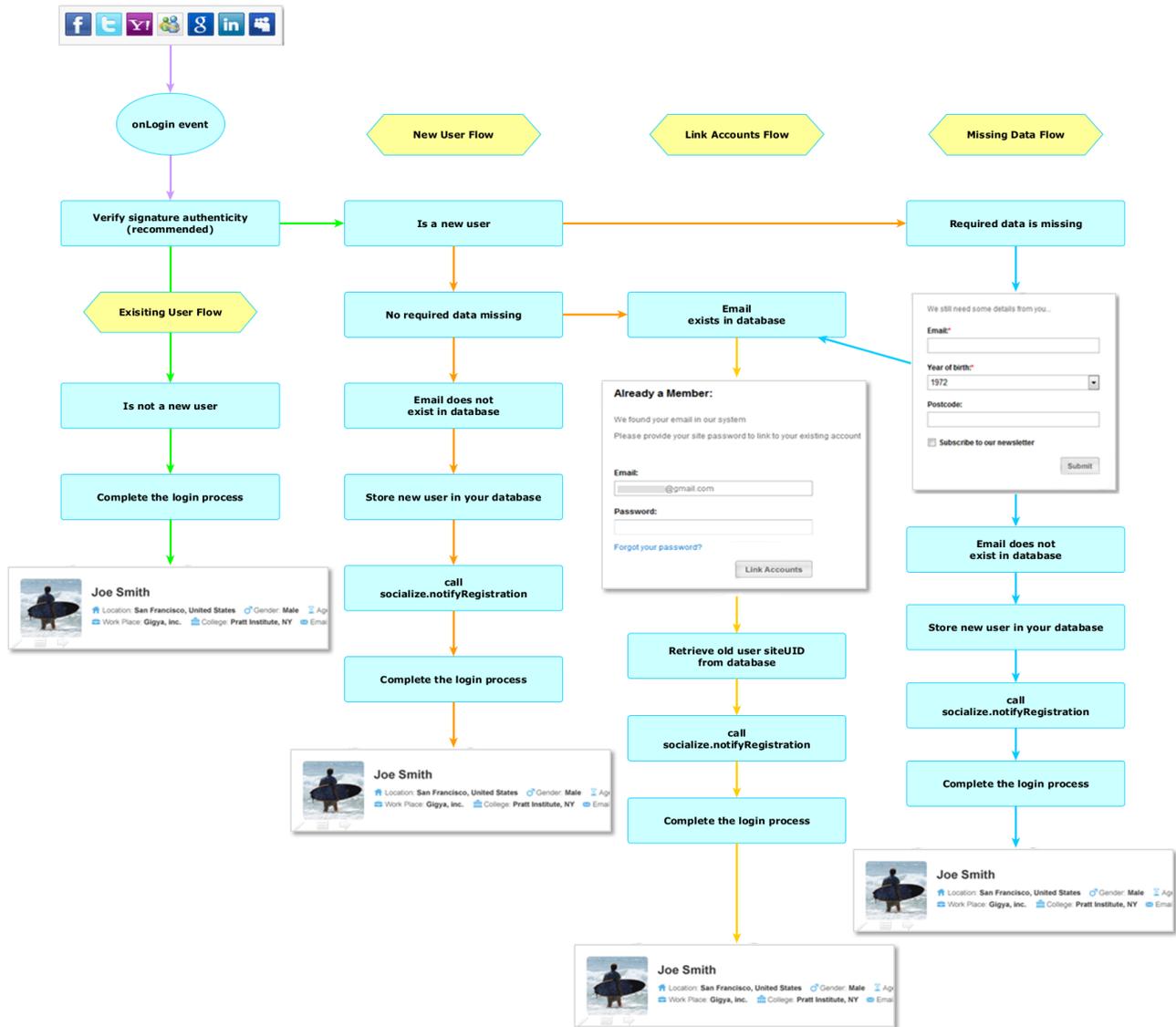
**Note:** The implementation includes usage of Gigya's API. Most Gigya API methods are supported both for client side (Web SDK) and server side (REST API). We encourage you to work server to server whenever applicable. Please use one of our [Server Side SDKs](#) for server side API calls. If there is no SDK available for your preferred language, you may use [direct REST API](#) calls.

# Integrating Gigya's Social Login

Before delving into the implementation, please take a look at the [Screen-Sets](#) illustrating the following implementation from the user's point of view.

Please note, this implementation proposes using the user's email address as the user identifier, in other words the email field is required and unique per user account. If this assumption is incorrect for your site, for example if you are using any string as the "username" and an email is not a required field, you can easily adjust the implementation accordingly.

To integrate Gigya's social login/registration, please implement the flow presented in the following chart (the steps are described below):



There are four possible flows that could occur depending upon the current status of the user and their information.

## Initiate the Login Plugin

These initial steps will always occur regardless of which sub-flow ultimately succeeds.

- Add the **Login plugin** to your Sign-Up (registration) and Login pages. The easiest and quickest way is to use the **Login Plugin wizard**. This wizard lets you customize the plugin, while previewing the changes as you make them. Based on your modifications, the wizard generates code that you can insert into your application. Alternatively, you may design your own user interface and use the **Login API** method.

- Upon successful Login you will be notified with an `onLogin` event. The `onLogin` event data includes a `User` object enfolding the user data, some of this data is used in the next steps. Please make sure to register to the `onLogin` event, using the `socialize.addEventHandlers` method, during [page load](#). To learn more about how to register and handle events generated by the Gigya service, please refer to the [Events](#) page.
- To avoid fraud, we highly recommend verifying the authenticity of the signature received from Gigya (with the `onLogin` event data from the previous step). To learn more about this subject, please refer to the [Validate the UID Signature in the Social Login Process](#) section of the [Security](#) guide.
- Check whether the user is new. To implement this, retrieve the `UID` field from the `onLogin` event data and look up the `UID` in your user management system.
- An alternative and more efficient way to check if the user is new, is by inspecting the `isSiteUID` field, which is part of the `User Object` (received within the `onLogin` event data).

• **Note:** The UID string must be encoded using the `encodeURIComponent()` function before sending it from your client to your server.

- If `isSiteUID===true`, this means that the UID in the `UserObject` is an ID that your site has provided using the `socialize.notifyRegistration` API method (see below) in a previous login flow, hence the user is a returning user.
- If `isSiteUID===false`, this means the UID was generated by Gigya, hence the user is new.

## "Existing User" Sub-Flow

If the UID already exists in your user management system:

- Identify the user
- Complete the login process on your site.
- Mark the user as logged-in.
- Move to your post-login page.

## "New User" Sub-Flow

If the UID is new, you should:

- Check whether any required data is missing. In many cases you have user data fields that you consider to be required. An example may be the email address. During the `onLogin` event you received the `User` object that enfolds various user data. This user data may or may not include all the data that you consider to be required. The user data that is available in the `User Object` may vary from one provider to another and from one user to another. For example, Facebook provides email address, while Twitter does not.

- If data is missing:

If you don't wish to implement this sub-flow (you don't require any data), you can skip this step.

- Proceed to the [Missing Required Data](#) sub-flow below.

- If no data is missing:

If you don't require email uniqueness in you system, or you don't wish to implement account linking, you can skip this step.

- Check whether the user's email address (received within the `User object`) exists in your system.
  - If the email exists in your database:
    - View the [Link Accounts Flow](#). This provides the user with an option to link to his site account (by entering his credentials and pressing the "Link Accounts" button).
    - If the user pressed the "**Link Accounts**" button, you should authenticate the user's credentials in your database. If authenticated (i.e., the user identified as an existing site user), move to the next step.
    - Retrieve the old user siteUID from the database. In the previous step, you have identified the user, thus found his record in your database. Retrieve his site user ID from his record. It will be used in the next step. From here you complete the login process on your site - mark the user as logged-in, and move to your post-login page.
  - If the email does not exist in your database.
    - store the new user in your database. The site user ID you designate for the user in this step will be used when you call `socialize.notifyRegistration`.
- Call the `socialize.notifyRegistration` API method. This method receives a parameter named `siteUID`. Set the `siteUID` parameter now or use the user ID that you have designated to this user in either of the previous steps.
- Complete the login process on your site.  
Mark the user as logged-in.  
Move to your post-login page.

## "Link Account" Sub-Flow

If the user's email already exists in your database and you need him to link his existing account to the site:

- Present them with the "Already a Member" screen.
  - View the [Link Accounts Flow](#). This provides the user with an option to link to their existing site account (by entering his credentials and pressing the "Link Accounts" button).  
The user must press the "**Link Accounts**" button, you should then authenticate the user's credentials in your database.
    - If authenticated (i.e., the user identified as an existing site user).
      - Retrieve the old user siteUID from your database.
        - In the previous step, you have identified the user, thus found his record in your database.
        - Retrieve his site user ID from his record. It will be used when you call `socialize.notifyRegistration`.
- Call the `socialize.notifyRegistration` API method. This method receives a parameter named `siteUID`. Set the `siteUID` parameter now or use the user ID that you have designated to this user in either of the previous steps.
- Complete the login process on your site.
- Mark the user as logged-in.
- Move to your post-login page.

## "Missing Required Data" Sub-Flow

If the user's email does not exist in your database, or the current User object does not contain the necessary required information for your site:

- Present them with the "We still need some details from you" screen.
  - View the Profile Completion option in the [Advanced Registration Flow](#). this form allows the user to fill in required fields (e.g., email address). You can use your standard registration form, pre-populated with data.  
The user must fill out the form and press "Submit".
    - The [User Object](#) that you receive from Gigya as part of the `onLogin` event data in the initial steps holds user-profile information extracted from social networks. You can use the information that is relevant to your site to pre-populate the form.
- Call the `socialize.notifyRegistration` API method. This method receives a parameter named `siteUID`. Set the `siteUID` parameter now or use the user ID that you have designated to this user in any of the previous steps.
- Complete the login process on your site.
- Mark the user as logged-in.
- Move to your post-login page.

## Site Login - Synchronizing with Gigya Service

When a user authenticates using your existing Login form or when a new user registers using site registration, it is important to notify Gigya of the user's new state, so as to provide consistent user experience. To implement this, call `socialize.notifyLogin` at the end of your existing login flow. This is illustrated in the following flowchart diagram:

## Site Login

Or login with your account:

User name:

Password:

Authenticate  
Credentials

Call  
socialize.notifyLogin

 Hello Karen site  
Log out

Connect to networks:

## Site Registration

Or create a new account:

Full name:

Email:

Password:

Confirm Password:

Create New Site  
User

Call  
socialize.notifyLogin  
(newUser='true')

 Hello Karen site  
Log out

Connect to networks:

The `socialize.notifyLogin` method receives a required parameter named **siteUID**. Please set this parameter with the user ID that you have designated for this user in your database. The `notifyLogin` call registers a new user in Gigya in case the `siteUID` provided is new, or reconnects a returning user in case the `siteUID` already exists in our records.

If it is a new user, call the `socialize.notifyLogin` API method with the **newUser** parameter set to `'true'`. This will enable Gigya to distinguish between a new site user and a returning site user, allowing Gigya to analyze users' login/registration behavior with or without Social Login and compare the ratio.

When receiving the `notifyLogin` response on server side, please make sure to create a **session cookie**, so as to maintain the client application synchronized with the user state. The `notifyLogin` response data includes the following fields: `cookieName`, `cookieValue`, `cookiePath`, `cookieDomain`. Please create a cookie using these fields.

For example, in PHP:

```
setcookie(cookieName, cookieValue, 0, cookiePath, cookieDomain)
```

Please make sure that the page following the login includes Gigya's library i.e., `gigya.js`, in order for Gigya to read the cookie before it expires.

Synchronizing Gigya with the user's state has several benefits:

- In calls to `socialize.getUserInfo` and `socialize.getFriendsInfo` Gigya will return your own site user ID as the UID for this user (as opposed to Gigya's user ID).
- Gigya will set the `isSiteUser` flag for this user to `"true"`, thus even if this user is referred to as a friend of another user you can easily tell he is a user of your site and not just a friend of the visiting user.
- Any connections the user makes to social networks will be linked with the site account. The social graph will automatically be made available the next time the user logs in to the site.

## Logging Out

When a user logs out from your site, it is important to notify Gigya of the user's new state, otherwise Gigya will still consider the user logged in and someone else who uses the same computer could gain access to that user's personal information and perform actions on his behalf.

Within your site logging out flow, please call the [socialize.logout](#) API method. We recommend calling the [socialize.logout](#) method from your client application, this way, Gigya will clear the relevant cookies, and your client application will know when the user is logged out without calling the server.

When calling [socialize.logout](#) method, Gigya will attempt to logout the user from all the providers to which the user is connected. In order to force logout from Facebook, you have to [configure a Domain Alias \(CNAME\)](#) for your site and [enable automatic session renewal](#) in our site's [Settings page](#) in the Facebook Configuration dialog.

## Sessions Expiration

When a user logs in via Gigya, Gigya creates a login session for the user. By default the session stays valid forever, or until [socialize.logout](#) method is called.

Gigya gives you the option to change the default behavior and decide when to terminate a login session. Read more in [Controlling Session Expiration](#) section of the [Security](#) guide.

## Using Social Plugins to Initiate Site Login

Please read the [Using Plugins to Initiate Site Login](#) to learn how to integrate Gigya's Plugins with the Social Login Process and leverage the Plugins to acquire new site users.

## Adding Connections

After the user has logged-in to your site, you can give your user an option to add connections to multiple social networks, hence giving the possibility to interact with friends on multiple social networks. You can do that by adding the [Add Connections plugin](#) to your site pages. This is broadly explained in the [Adding Connections to Social Networks](#) guide.

## Error Handling

Gigya uses an asynchronous programming model in which operations are triggered and then run in the background until they are completed. Upon successful or unsuccessful completion you receive a method response that includes the results of the operation. In case of [using the Web SDK](#), the operation invokes a **callback function**, which should be provided as a parameter to the API method call. The callback function receives a [response object](#) that includes the results of the operation.

To prevent an inconsistent user state it is vital to make sure that all the steps of the Login flow complete successfully. For this purpose, we recommend employing the following steps when calling an API method (i.e., [showLoginUI](#), [notifyRegistration](#), [notifyLogin](#), [logout](#)):

1. Define a timeout.
2. If after the timeout you did not receive a response, or if you received an error in the response, then retry calling the API method.
3. In case of [notifyRegistration](#), if after several retries the method has not successfully completed, rollback the database changes.

### Important Notes

- When a user logs-in through Gigya you should not use the [socialize.notifyLogin](#) call because the user is already logged-in in Gigya's platform at that point.
- Please do not call [socialize.notifyRegistration](#) nor [socialize.setUID](#) API methods after logging-in a returning user.
- In most cases, a user only needs to be authenticated once per session, either with Gigya or with your site's Login system. Therefore, do not display Gigya's Login Plugin for users who have already logged in.

## Supported Providers

## Capabilities Definitions

Gigya supports three broad tiers of capabilities across all social networks. Different social networks support different capabilities. The three tiers of supported features are as follows, and the below examples show each in action:

- **Login** - The provider can be used for logging in.
- **Actions** - The provider allows sending newsfeed updates.
- **Friends** - The provider supports obtaining the list of friends of the current user.

## Capabilities By Network

This chart shows which capabilities are supported by Gigya for each social network.

Some of these abilities require receiving approved extended permissions from the network.

Social Network	Network	Login	Actions	Friends
	Facebook			
	Twitter			
	Google			
	LinkedIn			
	Yahoo			
	Messenger			
	Foursquare			
	Renren			

 Tencent	QQ			
 Sina Weibo	Sina			
 VKontakte	VKontakte			
 Mixi	mixi			
 Yahoo Japan	Yahoo Japan			
 Spiceworks	Spiceworks			
 Instagram	Instagram			
 Odnoklassniki	Odnoklassniki			
 Amazon	Amazon			
 Xing	Xing			
 WeChat	WeChat			
 Wordpress	Wordpress			
 Blogger	Blogger			
 Paypal	PayPal ( <i>deprecated</i> )			

 PayPal	PayPalOAuth			
 Netlog	Netlog			
 Line	LINE			
 Naver	Naver			
 Log In	Kakao			
 Livedoor	Livedoor			
 Aol.	AOL			
 Orange	OrangeFr			

To see how different social network permissions are mapped within Gigya via the [Permissions](#) page of the Dashboard, see the [Permissions To Scope Table](#).

The license could not be verified: License Certificate has expired!

## Working Examples

We offer several Social Login working code examples, from the most simplified to the most comprehensive:

1. **Basic Social Login:** for a quick start implementation, please refer to the [Basic Social Login](#) working code example.
2. **Social Login Demo:** in the [Social Login Demo](#) page you may find a working example that implements a simplified version of the algorithm that is described in [Integrating Gigya's Social Login](#).  
The missing parts in the implementation are: the "Link Account" Sub-Flow, the "Missing Required Data" Sub-Flow and the parts that require interaction with your site's server and database.
3. **Comprehensive demo site:** The [Daily Recipe](#) is a Gigya demo site written in [PHP](#) and JavaScript. This demo outlines how to make a web site social using Gigya's platform, and implements the best practice as described in this page. The demo site's code is available for you to [download](#), use and [learn about the site implementation](#). Examine how Gigya's Social Login best practice is implemented within the site.

## Guidelines

- **Position the Login plugin in a prominent place on your registration and login forms:** Many sites have different forms for registration and logging-in, this is reflected in the mock-up shown above. We advise you to include the login plugin in a prominent position on both forms to increase the user registration rate. If your site has both registration and login on the same page we recommend adding the Gigya login plugin in both sections of the page, so it would appear once under each section. This makes it easier for users to understand their options, thus increasing registration rates.
- **Make sure your design helps users understand that they can either link to an existing account or register:** We recommend placing the linking option in a prominent spots in your step two registration page, for example side by side with your registration option.
- **Provide a consistent and coherent user experience:** Make sure you sync your login system with Gigya, so that when a user login through your system or logs-out you notify Gigya of the user's state.
- **Store social network user info properly:** To store user data, it is recommended to use Gigya [Profile Management](#).
- **Follow the social networks' terms of use:** Each social network has its own terms of use, developer's guidelines, and site policies and requires sites to implement their solutions in accordance to those. Gigya does most of the work for you via our [Identity Compliance](#), but there are a few basic rules that you should keep in mind when designing your site's social network integration:
  - Don't publish feed items to user's streams without first getting the user's consent.
  - Setup a dedicated social network external application per site, so that users can control their privacy settings.