Linking Social Accounts

Overview

Gigya's plugins, such as the social login plugin, enable users who wish to *register or log in* to your site to use their social identities to create accounts.

These identities are managed and identified using login identifiers, which are, by default, the email addresses associated with an account's identit y.

When two accounts have conflicting login identifiers, Gigya assumes that the same user has ownership of both accounts, and enables linking the accounts into a single account, associated with two identities, rather than maintaining two separate accounts.

Gigya's default state is to allow and support linking of social identities to site identities. This guide explains how to implement support for linking between social identities.

Note: Linking a social account to a site account is supported by default and can be implemented in the same manner as described in this document. Linking between site accounts is not supported.

Contents

This guide covers the following subjects:

- Account linking flows - Situations which trigger the possibility for account linking.
- Implementation - How to implement support for account linking for your website.
    - Policies - Setting your site's policies to support account linking.
    - UI implementation - Adding account linking support for existing UI Builder.
    - API implementation - Linking accounts using Gigya's API.

Link Accounts Flows

Account linking can occur in the following cases:

New Social Account Conflict

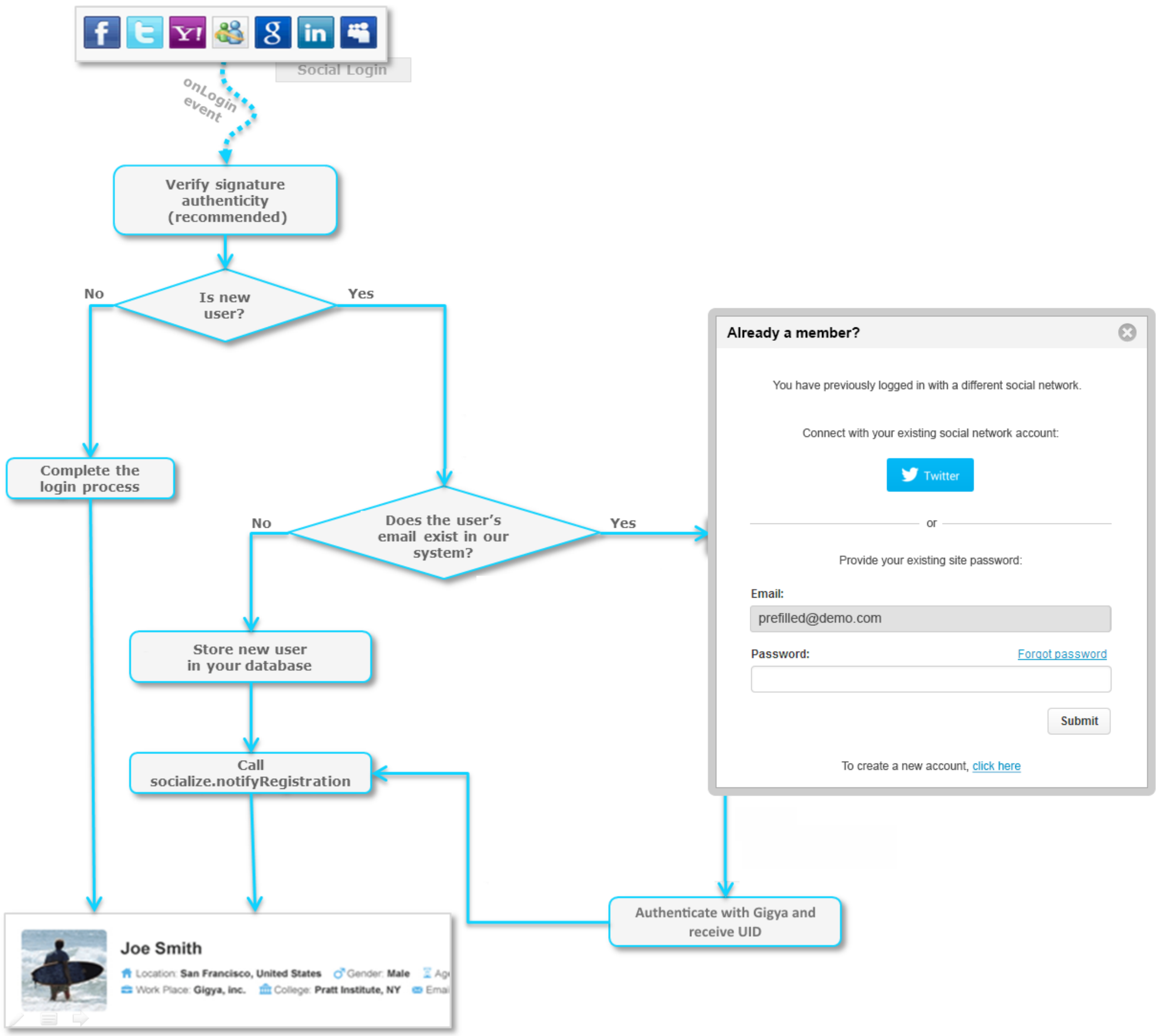
Since the registration flow is seamless with social login, some users are likely to want to log in with any given social identity.

When a member of your site registers with a new social identity for the first time, Gigya recognizes a conflict of email addresses and suggests linking the accounts.

If the user chooses to link the accounts, an authentication screen appears to enable the user to prove ownership of the previous account in order to complete the account linking.

For example:

- User has both a *Facebook* and a *Twitter* account, both registered to the email address - *me@gmail.com*. Now let's say that this user has registered to your site **using his *Facebook* account,** and has used it to log in (and engage in other social activities) several times since.
- A while has passed, and the user returns to your site, **this time using his *Twitter* account** to log in (which actually creates a new account with a *Twitter* identity). As the new account is being created, Gigya recognizes a conflict of email addresses and suggests that the user link the accounts together. The user is then required to authenticate the Facebook account to provide proof of ownership of the account, and the two accounts are linked.
- The user can now log in with **either** his *Facebook* or*Twitter* account and perform actions as a single user.


Social Login
onLogin event
Verify signature authenticity (recommended)
Is new user?
No
Yes
Complete the login process
Does the user's email exist in our system?
No
Yes
Store new user in your database
Call socialize.notifyRegistration
Authenticate with Gigya and receive UID
Already a member?
You have previously logged in with a different social network.
Connect with your existing social network account:
Twitter
or
Provide your existing site password:
Email:
prefilled@demo.com
Password:
Forgot password
Submit
To create a new account, click here
Joe Smith
Location: San Francisco, United States
Gender: Male
Work Place: Gigya, inc.
College: Pratt Institute, NY

Conflict on Completing Registration

A conflict can be created for an existing account, if a change in the loginID is applied. This can happen when a new account is missing some key fields to complete its registration. When these fields are added using the accounts.setAccount.Info API, and a change in the login identifier creates a conflict, Gigya will alert the conflict according to the conflict handling policy, and if required, trigger the account linking flow, similar to account linking on a new registration.

If the user **chooses not to link** the account, a new account will be created with the conflicting field, however that field could not be used as a login identifier.

Implementation

This section explains the required steps for implementing both methods.

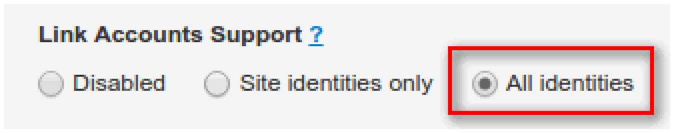
Account linking support can be implemented using Gigya's API, or through using UI Markup Extensions.

Linking Accounts Policy

First, in order to support social-to-social account linking, your site's policy regarding linking needs to be defined to explicitly support social-to-social linking. By default, only social-to-site linking is supported.

Console Settings

To be able to link between social accounts, a change of policy is required. Make sure to check the **All Identities** radio button on the Console's Policies page.



Link Accounts Support ?

Disabled [✓ All identities] Site identities only

Setting the Policy via API

To change the site's policy manually via API, you must set the **Conflict handling** policy to allow **Social-to-Social Linking**.
This is done using the accounts.setPolicies API. When calling the API set the **accountOptions** object's **loginIdentifierConflict** field to **failOnAnyConflictingIdentity**.

```
var accountOptionsObj = {
        //Allow social-to-social account linking
        loginIdentifierConflict : 'failOnAnyConflictingIdentity',
}
var policies = {
    accountOptions : accountOptionsObj
};
accounts.setPolicies(policies);
```

Building Account Linking UI

The following instructions are using the Gigya WebSDK, for information about using REST APIs, see accounts.login REST Errors.

This section explains how to add support for social-to-social account linking to an existing screen-set:

- **Account Linking Screen-Set**
  - Fortunately, the new RaaS-screen-set collections come with built-in support for social-to-social account linking. So if you are new to RaaS and are just equipping your site with screen-sets for the first time, you do not need to take any further measures to support social-to-social linking apart from adding the screen-sets to your site and enabling **Link All Accounts** in the site Policies. To read more on using screen-sets go to the RaaS best practice implementation section.

    If you are using generically created screen-sets (i.e., you haven't made any changes to the markup created by Gigya), you can simply create a new screen-set collection and load it, instead of your previous screen-set, when calling accounts.showScreenSet in your code.

    If you are using the old screen-sets, or have custom built your screen-sets, and would like to add social-to-social account linking support, continue to the **Adding Social-to-Social Linking Support** section.

- **Adding Social-to-Social Linking Support to an existing Screen-Set**
    - To enable social-to-social linking you need to add Gigya's social login widget to your existing link-accounts screen's markup, with the required **loginMode='link'** parameter.
      In order to do so you will have to export the markup from the console's screen-set page. If you are new to Gigya's markup extensions, you are advised to read the related guide prior to reading this section.

```
<div class="gigya-social-login">
    <param name="version" value="2" />
    <param name="loginMode" value="link">
</div>
```

In order to display the social login widget only in relevant situations (i.e., when the user has existing social accounts to link to), place the widget inside a Gigya conditional container with the *data-login-identities* attribute of *social*, to indicate the container will be displayed when the user has a social network identity.

Read more on conditional containers here.

```
<div class="gigya-container" data-login-identities="social">
    <div class="gigya-social-login">
        <param name="version" value="2" />
        <param name="loginMode" value="link">
    </div>
</div>
```

Linking Accounts Using API

This section explains how to perform account linking manually, using Gigya's API.

- **Step 1 - Identifying A Conflict**
    - When a login is performed via API and a conflict is identified by the system, the method will return an error code indicating the conflict.

      Along with the error, the response will contain a registration token that IDs the current registration flow and is required in order to link the new account to the existing one.

```
//Global registration token for cases of login error
var globalRegToken;

function loginHandler(response){
    //In case login failed due to conflicting login-identifiers
    if (response.errorCode == 403043){
        //Call a method that handles account linking and pass it
the regToken returned from the initial login
        globalRegToken = response.regToken;
        linkAccounts(globalRegToken);
    }
};
//start the login / registration process
accounts.socialLogin({
    // let's assume the user is logging in with Facebook
    provider: 'facebook',
    //Use loginHandler as callback
    callback: loginHandler
});
```

- **Step 2 - Get Conflicting Account Details**
  - Once the initial login has failed, you can call accounts.getConflictingAccount with the user's **regToken** in order to get a list of the providers associated with the conflicting account.

    These providers can be used to authenticate the user with the existing account.

```
function linkAccounts(regToken){
    //Get the list of provider for the conflicting account
    accounts.getConflictingAccount({
        regToken : regToken,
        callBack : linkHandler
    });
};
```

- **Step 3 - Link Accounts**
  - The Social Providers returned from the **conflictHandling** method can be presented to the user to select from. Once you have found a provider for authentication, call the login method again (either socialize.login for social login clients, accounts.socialLogin for RaaS clients who are linking social to social accounts, or accounts.login for RaaS clients who are linking social to site account) with **loginMode='link'** and the **regToken** returned from the initial call as a parameter to initiate the account linking.

```
function linkHandler(response){
    //Check response for success
    if (response.errorCode == 0){
        //Present a linking choice UI for the user to select a
provider to authenticate with
        // ....
        //suppose the loginProviders=['facebook', 'twitter',
'site']
        var provider =
getSelectedProvider(response.conflictingAccount.loginProviders);
        //Check the user's selection (social or site identity)
and call the corresponding method
        if (isSiteIdentity(provider)){
            //Alert the server to link the site accounts using
accounts.login
            makeAjaxLoginCall(provider, globalRegToken, link);

        }
        else {
            //link the social accounts. Make sure to call the
login method with loginMode='link'
            accounts.socialLogin({
                provider: provider,
                regToken : globalRegToken,
                loginMode: 'link'
            });
        }
    }
};
```

- **Step 4 - Finalize The Registration**
    - After completing the previous steps and receiving an approval from the server that the accounts have been linked, call accounts.finalizeRegistration to complete the process.

      You're done and the accounts are now linked.

Limitations

Current limitations of the Account-Linking feature:

- **Email Only**
    - The Link Accounts feature only functions when using **Email** as the site's **Login Identifier**.
- **Unlinking Accounts**
    - Once accounts have been linked, they cannot be unlinked. This means that taking an account with two (or more) identities and **splitting it into two separate accounts** is not supported.
      This should be distinguished from adding / removing connections. Adding / removing connections allows you to manage the identities associated with a given account. While a social or site identity can be added or removed from an account, it cannot be migrated into a new separate account. So one of the identities can be removed from the linked account, but the accounts cannot be unlinked. What this means, is that once an account is linked to another account, and their data is merged, removing the identity will not un-merge this data and it will persist on the account that the identity was removed from.
- **Site-To-Site and Site-To-Social Linking**
    - Site to social linking is currently not supported. Meaning that creating a site account (registering via form without using a social identity) and linking it to an already-existing social account is not possible.
      Site-to-site linking is also not supported.
- **Social Login Linking**
    - Support for social accounts linking is currently available for RaaS clients only. Clients implementing social login cannot link

between social accounts.
If you do not currently have RaaS enabled on your system, please contact your Implementation Consultant in order to activate it.