FAQs

Find answers to common questions regarding Gigya's technology.
Expand/Collapse All

General

What do I need to get started?

We recommend that you start with our RaaS Guide and Site Setup guide. You may also refer to our Live Working Examples, which also include the code you can download and run from localhost.

How do I get an API key?

Gigya plugins and API calls require passing an API key. The API key is a unique key, which is used to verify that API calls are made from an authorized domain. You may obtain your Gigya API key from the Site Dashboard page on the Gigya website (see also the Site Setup documentation).

Development on localhost: For development and testing purposes, it is possible to run Gigya on '*localhost*' and with any valid API key. You do not have to sign up in order to run a test on your localhost. You may use the API key that is provided in the various code samples throughout the wiki (see Working Code Examples). Copy the code, and run it on your localhost without any change. After signing up for the domain on which you will deploy, you may use your API key for development on localhost.

*Some API's will not generate an error with code 500000 and not complete from localhost if have not added **localhost** as a **Trusted Site URL** to the API Key you are using.

Deployment: Before deploying, you must complete your Site Setup, including Configuring Social Network Application Keys. Please make sure that all the pages that use Gigya API include your unique API key. Make sure that the domain name from which you load the pages is the same domain name that you used for generating the API key.

Do I have to establish a Gigya API key for a development and testing environment?

For development and testing purposes, it is possible to run Gigya on '*localhost*' and with any valid API key. You do not have to sign up in order to run a test on your localhost. You may use the API key that is provided in the various code samples throughout Gigya's documentation (see Working Code Examples). Copy the code, and run in on you localhost without any change.
After signing up for the domain on which you will deploy, you may use your API key for development on localhost.

What are the Supported Versions of Browsers?

See the JS Supported Browsers page.

Does Gigya Support 3rd Party Cookie Blocking?

Gigya's single sign-on, as well as social login and raas, require 3rd party cookies to be enabled within the user's browser.

How do I implement Gigya's Social Sign-on?

Review the following guides:

1. Social Login - An overview of Gigya's Social Login.
2. Adding Connections - This page provides logged-in users in your site with an option to add connections to multiple social networks, hence enabling to interact with friends on multiple social networks.
3. Using Plugins to Initiate Site Login - On this page you can learn how to integrate Gigya Engagement Add-Ons with the social login process and leverage the plugins to acquire new site users.

What is the difference between Login and Adding Connection?

Gigya offers two main services - authentication and social interaction. These two services are related but actually represent two different concepts.

Social interaction is an on-going process that requires a long term relationship with the Social Network. Gigya represents this long term relationship as a "**connection**" between the social networks and a user. Technically speaking a connection is equivalent to an established session with the social network and it may expire according to the social network policy. A valid and active connection is required in order to perform social interaction using Gigya, such as publishing a newsfeed report and accessing the user's social graph. The Gigya API calls for establishing connections are socialize.addConnection, socialize.showAddConnectionsUI and socialize.removeConnection (for terminating connections).

Authentication, on the other hand, is the process of identifying a person and assigning an ID in a secure, reliable and repeatable way. It ensures the same person will get the same ID every time he or she is authenticated. This process is a discrete operation with a very clear beginning and end. It only happens when the user is logging in and doesn't require any long term interaction with the user or the authentication providers beyond that point it time. The Gigya API calls related to authentication are socialize.login, socialize.showLoginUI, socialize.logout, socialize.notifyLogin, socialize.setUID and socialize.deleteAccount. A broad explanation of implementing Gigya authentication and the usage of these methods appears in the Authentication guide.

In Gigya, whenever a user logs in a connection is automatically established with the authentication provider, if that provider supports it.

The actual process of logging in and adding connection is the same for the user, in both cases he is asked to login to a social network and authorize the site. However the goal is different, only a login operation authenticates the user in a way that retrieves a persistent user ID.

The rule of thumb for when to use the *connect* family of APIs are:

- If you only plan to use Gigya to tap into social network APIs (i.e. getting user info, sending messages through the social networks, etc.) and you are not considering using the authentication service.
- When you want to add another connection to a user who is already logged in (even if that user used your own login system but as long as you called the socialize.notifyLogin API method). This subject is discussed in the Authentication guide.
- When you want to provide social services in a section of your site that doesn't require a login. If a user logs-in afterwards his connection will be imported automatically to his logged-in account.

How do I force logout from Facebook?

Make sure you load the gigya.js with your API key as part of the <head> section of your site's pages:

```
<SCRIPT type="text/javascript" lang="javascript"

src="http://cdn.gigya.com/js/gigya.js?apikey=INSERT-YOUR-APIKEY-HERE">
</SCRIPT>
```

Next, make sure you configured a Domain Alias (CNAME) for your site and enabled automatic session renewal in your site's Settings. Then when you call our client-side socialize.logout API method, Gigya will logout the user from all the providers to which he is connected, including Facebook.

Can I get user information that is not directly supported by Gigya?

Yes. When the social network supports additional information that is not available directly from Gigya, you may use the getRawData API call to request the additional data items directly from the social network.

Can I use Gigya and make direct API calls to the social networks at the same time?

Yes. Gigya allows you to get all the benefits of a unified social API while still giving you the flexibility to make direct API calls to the social networks if you wish to.

In order to make direct calls you need to get two pieces of information:

- The user's session token and session secret on the social network - This is obtained by calling socialize.getSessionInfo.
- The user's userID on the social network - The user ID is available in the providerUID field of the identity object associated with the social network in the response of socialize.getUserInfo.

How come it doesn't work when I use HTTPS?

If you choose to communicate with the Gigya service over a secure connection, you will need to include the Socialize JS file from our https domain:

Every page that uses the Gigya API must include Gigya's Java-script library in the section. On secured pages, the line of code should be:

```
<script type="text/javascript"
src="https://cdns.gigya.com/js/gigya.js?apikey=INSERT-YOUR-APIKEY-HER
E" ></script>
```

What user information should I expect from each network?

Different providers supply different data. You can view the availability of profile data by the main providers in the User Object table. Please note that the availability is also dependent on user permissions.

Does Gigya support localization?

Yes, Gigya's platform supports localization. Gigya's default language is English, but you may change the language to one of the supported languages. This is done by assigning a string language code to the *lang* field of the global Conf object. The language code table can assist you in finding the code of your desired language.

Gigya also supports localization for 3rd-party plugins such as Facebook's ***Like*** button. This is done by adding a new query string to the socialize JS file called *lang*, with the code of the desired language. Read more about Gigya's 3rd-party localization.

Are the statusCode and statusReason response fields supported?

Yes, although the statusCode and statusReason fields are deprecated, they are supported for backwards compatibility.

Authentication

Can I keep my existing login system and still benefit from using Gigya for authentication?

Yes. Actually, the most common integration of the Gigya service is by adding it as an additional authentication option, on top of the site's existing login system.

How do I deal with signatures?

You have to construct a signature and validate it.

Follow these steps to construct a signature:

Your partner's "**Secret Key**" is provided in BASE64 encoding, at the bottom of the Site Setup page on the Gigya website (please make sure that you have logged in to Gigya's website and that you have completed the Gigya Site Setup process).

1. Construct a "base string" for signing: "%Timestamp%_%UID%" replacing %Timestamp% and %UID% with the corresponding values.
2. Convert the base string into a binary array using UTF-8 encoding.
3. Optional - If you have a mechanism for storing and verifying cryptographic nonces it is recommend that you store the base string as a nonce and verify that you only get the same base string once.
4. Convert your "**Secret Key**" from its BASE64 encoding to a binary array.
5. Use the HMAC-SHA1 algorithm to calculate the cryptographic signature of the "base string" constructed in step 1, with your binary "**Secret Key**" calculated in step 4 as the key. The HMAC-SHA1 algorithm is implemented in many standard libraries and is readily available in any web development environment. The HMAC-SHA1 method usually receives two parameters: a *binary key*, and a *buffer* to be signed. It returns a binary array containing the signature.
6. Convert the signature to a BASE64 string.

For more details and pseudo code example, please refer to Constructing a Signature.

Please follow these steps to implement signature validation:

1. Validate that the timestamp is within 3 minutes of your current server time. Note that the timestamp is in "Unix time format" meaning the number of seconds since Jan. 1st 1970 and in GMT/UTC timezone. The timestamp is provided in the signatureTimestamp field of the User object.
2. Construct a signature from the UID, signatureTimestamp and your secret key. Follow the instructions in the Constructing a Signature section below.
3. Compare the signature you have calculated to the one generated by Gigya. If the UID signature is valid the two would be exactly the same. The Gigya signature is provided in the UIDSignature field of the User object.

For more details and pseudo code, please refer to Signature Validation Process.

Why is the browser popup killer blocking the window Gigya is trying to open when I call login() or addConnection()?

The authentication popup window is usually blocked if it's triggered without the user clicking a button. If you call the login method within an onClick callback, the window will typically be shown without any issues. Issues will arise if you call login outside of the onClick function handler.

When should I use setUID vs. notifyLogin?

Most sites choose to use the Gigya service as an additional authentication mechanism on top of their existing authentication mechanism. You may find an example of an existing registration form, alongside with Gigya's Plugin on the Login page of Gigya's website (if you are logged in, log out to view the page).

When a user opts to login or register to your site, you can offer the user to authenticate using either Gigya Plugin or your existing Login/Registration form. Each of these two options leads to a separate logic flow.

notifyLogin: When a user authenticates using your existing Login/Registration form, it is important to notify Gigya of the user's new state, so as to provide consistent user experience. To implement that, call the socialize.notifyLogin API method at the end of your existing authentication flow.

setUID: This method replaces the Gigya UID in the user account of Gigya's DB, with a site user ID that you provide. This operation practically links the current user's Gigya account to his account on your user management system. Use this method when a new user has registered through Gigya (using a social network). Call setUID immediately after you have stored the new user in your database. Set the "siteUID" parameter with the user ID which you have designated to this user in your database.

Both scenarios are fully described in our Authentication developer's guide. We highly recommend that you read this guide if you plan on implementing Gigya authentication in you site.

Error Codes

Why am I getting the "Request has expired" error (code 403002)?

Please make sure you have the correct time, timezone and daylight savings configured on your server. Make sure that the time on your

server is roughly synced with the NIST atomic clock (please refer to http://time.gov/).

For security reasons, Gigya requires every REST API call to be signed, so as to guarantee that it originated from an authorized partner and was not tampered with in transit. The signature's timestamp and the time on Gigya's server when the request is received are compared, and if the time difference is more than two minutes, the request is rejected and you will receive the "Request has expired" error (code 403002). The most common cause for this error is when your server's clock is not accurately set, which causes a gap between your server time and Gigya's server time.

Learn more in Signing the Request guide.

Why are my users receiving an Error Code 400097 when attempting to log in from Android?

Error code 400097 is returned with the message "User is attempting to access the Gigya service from an unsecure browser that is not supported. User should switch browsers." This is due to a major security issue with Androids original browser. Due to the security issue, Gigya no longer supports AOSP browser in Android OS versions 4.0 - 4.4. Please see here for more information.

Why are my REST API calls failing with the "Invalid request signature" error (code 403003)?

For security reasons, Gigya requires every REST API call to be signed, so as to guarantee that it originated from an authorized partner and was not tampered with in transit. For this purpose all the REST API methods include a set of required parameters. One of these parameters is the *sig* parameter. You should assign this parameter with a signature that you generate. To learn how to generate a signature, please read the Signing the Request guide.

Why are my REST API calls failing with the "Duplicate nonce" error (code 403004)?

Every REST API method includes a required parameter called 'nonce'. In security engineering, a **nonce** is an abbreviation of *number used once*. The purpose of this parameter is to to ensure that old communications cannot be reused in *replay attacks* . Gigya requires that in each REST API call the nonce string is unique. If Gigya receives two API calls with the same nonce, the second API call will be rejected with the "Duplicate nonce" error (code 403004).

Why are my REST API calls failing with error code 500000?

Error code 500000 indicates a server communication problem. For more information, see Troubleshooting Server Connectivity.

Why are calls to getAccountInfo failing with Unauthorized User errors after the user is logged in?

One of the most common reasons this may be occurring is if the global Conf object is not loading properly. If you have any undefined variables as values inside the global Conf object, this will cause the script to fail without logging any errors and it will not finish loading. When this happens, numerous API calls to Gigya will return an Unauthorized User error.

User Interface

How do I get rid of the Terms link in the UI?

To remove the "Terms" link, set the *showTermsLink* parameter of the add-on to *false* .

```
var params = {
    showTermsLink: 'false', // removing the "Terms" link
    // ... other parameters
};
gigya.socialize.showLoginUI(params);
```

How do I get rid of the Gigya logo in the UI?

You may remove the "powered by Gigya" logo from Gigya's Add-ons only if you are a valid subscription customer. You can contact us by filling in a support form on our site. You can also access the support page by clicking "Support" on the upper menu of the Gigya Console.

Reporting

What reporting functions does Gigya support?

Reporting is supported via the Audit Log and Gigya's API's. See reports.getSocializeStats for more information.

Can Gigya's getSocializeStats API be customized?

Yes, the API is completely customizable.

Can I create a scheduled email report that can be sent on a regular basis - per brand?

At this time creating unique scheduled email reports is not supported.

Under the new Registered Users reporting tab, is it possible to see the sum of all within a given period?

You can view the total amount of new Registered Users using the Export Users API, however, you must have **Link Accounts Support: All Identities** enabled, because the export feature uses the **Site** provider only.

Can I view what other Brands a user is connected to in order to compare cross-Brand registrations?

Yes, you can view this information via the Raw Data in the Audit Log.

Can I easily view the number of new registrations on my site for a specific time period?

Yes, you can view this data using the Identity Query Tool within the Identity Access tab of the Gigya Console.

Can we have unique email verifications for each Brand?

Yes, you can create individual emails for each unique Brand.

Is there anything that Gigya can not support within an HTML email?

Normal HTML email restrictions apply; anything generally supported in HTML emails will function correctly.

If a user registers on my site, then logs out and then logs back in using a social network, are these counted as separate registrations?

No, if a user registers via the site registration and logs out, the user is registered. If they then revisit the site and login with a different provider, this will be counted as a login, but the user is only registered the initial time. Furthermore, if the user needs to verify his email to gain access to the site after registration, the user is still counted as a 'registered user' even if they never complete verification.

If a user logs in to my site and then leaves and returns while the session is active, is that counted as two logins?

No, if the session for the user is still active and the user does not pass through the login screen on recurring visits, they are not counted as a separate login. This functionality is the same within a Single Sign-on situation as well.

Can we view unique user logins per day?

No, Gigya only remembers the last, most recent, login date/time for any particular user.