

APIs and SDKs

Integration Overview

Gigya supports multiple environments:

Core REST API

The core of the Gigya service is a [REST-like interface](#). Please refer to the [REST API reference](#) for the list of methods and parameters. In addition, Gigya offers various developer tools that wraps this API for various environments; thus in most cases you will not need to use the REST API directly.

Web SDK

Gigya's Web SDK is your access to Gigya's add-ons. It provides a wrap around Gigya's REST API, in addition to implementing the add-ons listed [here](#). We recommend reading the [Using the Web SDK](#) guide, which provides the basic knowledge needed to start developing using the Gigya Web SDK. The guide includes a basic example and instructions for using methods and events. Refer to the [Web SDK Reference](#) for the list of methods and parameters.

Server Side SDKs

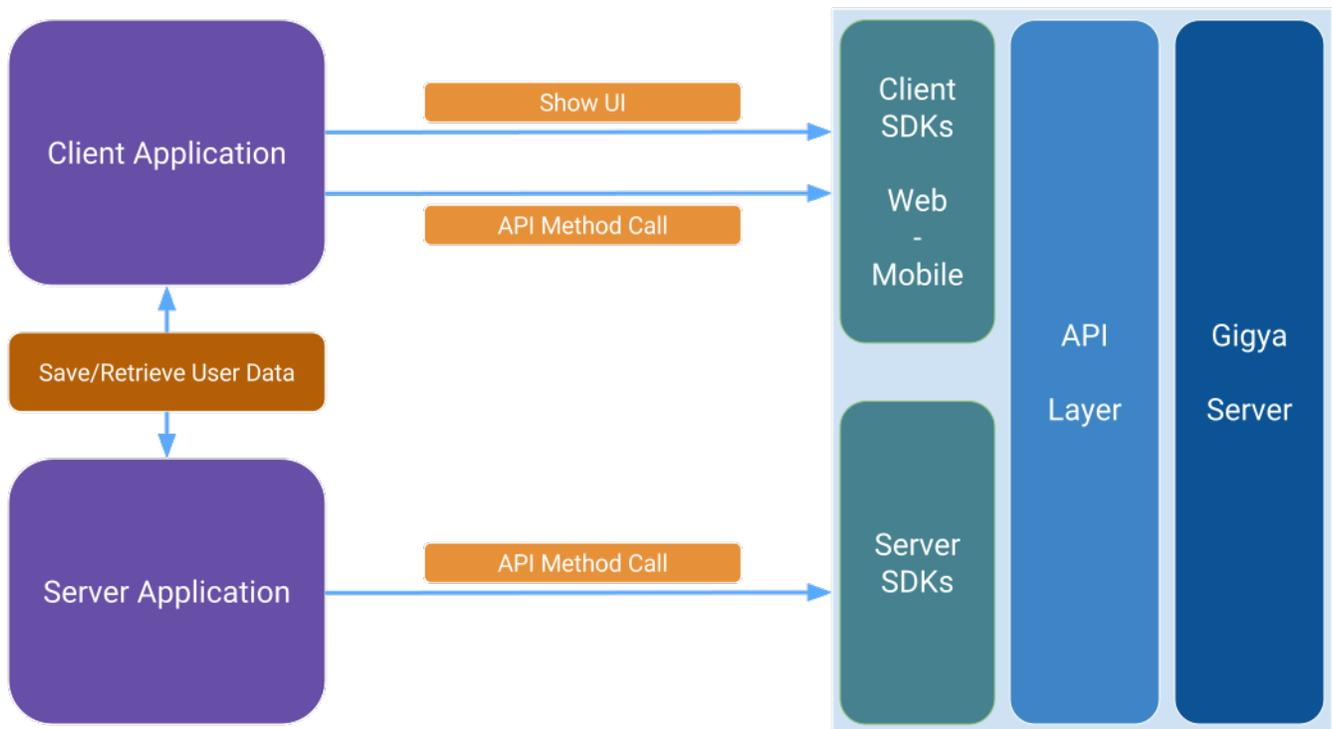
Gigya offers a set of [Server Side SDKs](#), not part of our core offering, that provide a wrap around Gigya's REST API. The SDKs make it simple to integrate Gigya's service in various development environments. Check out our [Server Side SDKs](#) documentation and find out if there is an SDK available for your preferred language.

Mobile SDKs

Gigya offers SDKs, not part of our core offering, for [iOS](#), [Android](#), and [Cordova](#).

Common Integrations Diagram

The following diagram shows the most common use-case of a web site that integrates the Gigya service:



Your website client application uses the [Web SDK](#) to integrate the plugins and client site API method calls.

Your website server application uses a server side SDK for server API method calls. Choose the [Server Side SDKs](#) that matches your environment.

If there is no SDK available for your platform (e.g., Perl, Ruby), you may use an external OAuth2 SDK as a wrap around our REST API. Learn more [here](#).

Handling The Response

When developing logic to parse any data returned from Gigya it is important to ensure that your code does not use strict serialization/deserialization and is able to iterate through the response's properties, as opposed to expecting a finite number of specific properties. Gigya may add properties without notice and your code should be designed to handle/ignore any unexpected properties gracefully.

Internal Methods

Gigya services use numerous methods within our SDKs and GConnectors to perform specific internal functions to support our public APIs and methods. These methods are subject to change at any time without notice. You should never use any undocumented APIs or methods within your implementations or your integration may fail if these methods are updated.

API Key and Site Setup

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](#) of the [Gigya Console](#) (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.

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.

Create Custom Sample Requests

You can use our [Custom Sample Requests Tool](#) to generate simple sample requests for use with our most popular SDKs.