

GConnector - CMS and E-Commerce Integrations

Overview

GConnector is a tool that makes it easy to implement Gigya within a CMS or e-commerce platform, while leveraging the best of Gigya functionality without compromising the CMS capabilities. When using GConnectors, the user's session and user data are synced in real time between Gigya and the CMS, providing a smooth, cohesive user experience, and providing site admins with up-to-date, workable data. GConnectors share the same design principles.

Benefits

- Scalable - including in multi-site, multi-CMS scenario
- Seamless, real-time integration between Gigya and CMS identities for a smooth user experience
- Fast implementation
- Based on Gigya's best practices
- Support for new CMS version releases
- Cross-platform support of SSO (across different CMS systems)
- Full support for Gigya's screen-sets and for Web SDK implementations with custom screens

Core Features

- Support for RaaS
- Session synchronization
- Secure by design
- UID-based sync
- Field mapping and hooks for data transformation
- Easy logging and debugging
- Built-in localization support, including inheriting language from the CMS, and fallback language definition

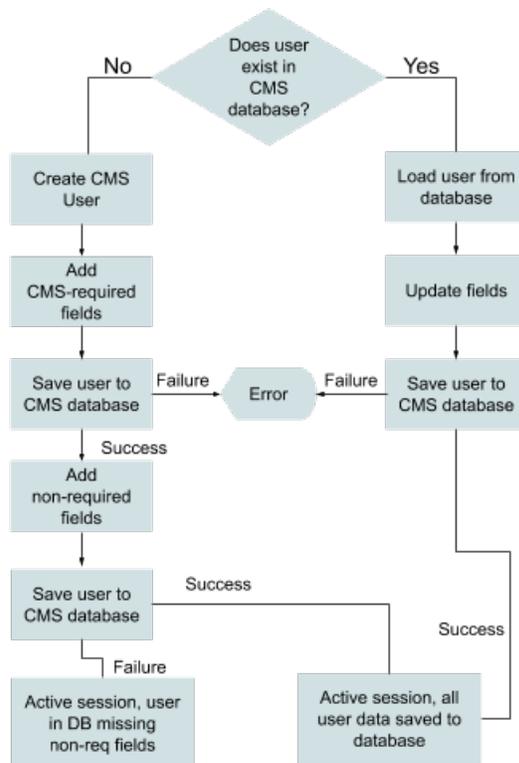
Extended Functionality

- Support for Gigya plugins
- Session management configuration
- Back office sync so as to display up-to-date user data in the CMS admin panel
- Sync with Gigya's [Data Store](#) (DS)
- Personalization

GConnector Identity Data

The main RaaS actions (i.e. login, register and edit profile) are mirrored in the CMS user management system. This is achieved by registering to Gigya's [onLogin](#) event, and creating or updating a user account in the CMS. In the case of mismatching data (for example, the user's country is registered as France in Gigya's database and Germany in the CMS database), the Gigya data overrides that of the CMS.

The flow when a user logs in is as follows:



UID-Based Sync

Data sync between Gigya and the CMS is based on Gigya's UID. This way, user information is independent of any input provided by the user (such as name or email), and passed consistently between systems, so as to provide a seamless integration and improve the users' experience. The UID field is created automatically in the CMS when installing the GConnector. When configuring Gigya for CMS integrations, the email is always the login identifier (included in the **loginID**).

We differentiate between two types of CMS:

- **Email** is required and unique (a user identifier in the CMS).
- **Username** is required and unique (a user identifier in the CMS).

When Email is the CMS Identifier

When the user's email is an identifier in the CMS, by default, the CMS email field is mapped to the emails stored in Gigya's **loginID**. This cannot be overridden by field mapping.

The login/registration flows in this case require that at least one email that is recognized in Gigya as a login identifier is unique to the user. After a user logs in or registers, the flow is as follows:

1. Following Gigya's **onLogin** event, the UID is passed.
2. The GConnector checks if the UID exists in the CMS database:
 - Login: If the UID exists in the database, the system makes sure that at least one of the emails in the **loginID** is unique to this user, logs the user in and sets that email address in the CMS. If **all** of the user's **loginID** emails are saved to other users (other UIDs), the user will not be logged in, and will receive an error.
 - Register: If the UID does not exist, the system makes sure that at least one of the emails in the **loginID** is unique to this user, registers a new user and sets that email address in the CMS. If **all** of the user's **loginID** emails are saved to other users (other UIDs), the user will not be logged in, and will receive an error.

If you are using the **profile.email** Gigya data field, it should be mapped to a custom email field in the CMS, and not to the default required email field in the CMS.

When Username is the CMS Identifier

1. Following Gigya's **onLogin** event, the UID is passed.
2. The GConnector checks if the UID exists in the CMS database:
 - Login: If the UID exists, the user is logged in. Data is updated according to [this flow](#) (required fields are updated before non-required).
 - Registration: If the UID does not exist, a new user is registered in the CMS. Data is saved according to [this flow](#) (required fields are updated before non-required).

Data Store Sync

When implementing Gigya's [Data Store \(DS\)](#) to store user-related data, GConnectors may offer a DS Sync module that adds the ability to pull data from Gigya's Data Store and map it into the user account database in the CMS. This is provided through an optional, separate package that is installed separately.

This capability is currently available for the following connectors:

- [Sitefinity](#)
- [Drupal 7](#)
- [Umbraco 7](#)
- [Magento 1.0](#)

Back End Sync

In some cases, the GConnector includes a module for synchronizing Gigya data to the CMS back end or back office, where CMS admins manage user data.

This capability is currently available for the following connectors:

- [Magento 1.0](#)

Field Mapping

Fields in one database are dynamically mapped to the corresponding field in the other database. For examples, see [Gigya Connector for Drupal 8](#), [Gigya Extension for Magento 2.0](#), and [Gigya Connector for Sitefinity 9](#).

By default, the GConnector usually maps the following fields for every user in every CMS: **first name**, **last name**, and **email**.

Hooks

GConnectors usually offer the option of registering to specific events and adding custom code. Gconnectors offer hooks that are fired after the following events:

- Field mapping: A common use case is transforming the user's birthday from 3 Gigya fields (**profile.birthDay**, **profile.birthMonth**, **profile.birthYear**) into one CMS field (e.g., **birthDate**).
- Language: Enables making on-the-fly exceptions to the GConnector language settings and loading screen-sets in different languages.
- Screen-sets: Passes into the screen-set specific values for the parameters of [accounts.showScreenSet](#).
- Global parameters: Enables overriding the [global parameters](#) defined for the GConnector, as needed.

Security

Gigya is committed to maintaining the highest level of security. Our platform is designed to protect customer data with a series of strict security and compliance standards which are applied to our GConnectors as well. These include the following principles:

- The Application Secret is never used or stored in the CMS (session /cache/cookie etc.), but rather loaded from the database for each call.
- The Application Secret is encrypted in the database.
- Validates using [accounts.exchangeUIDSignature](#).
- Gigya's [Web SDK](#) is loaded over HTTPS.
- Session expires automatically.
- Uses server-side calls when verified user data is required.
- Server-side calls are signed.

Configuration Options

Generally, the GConnector configuration options (sometimes available within the CMS Admin area) align with those of Gigya's [Global Configuration](#). In addition, they include

- Field mapping
- Session management and session expiration settings. See [Managing Session Expiration](#) for more information about Gigya's session management.
- Redirect and logout URLs

Multi-Site and SSO Support

GConnectors support multi-site implementations. Configuration for each site is independent, and in some cases sites can inherit configurations from a master site. In addition, SSO is supported by Gigya. For more information, see [Site Groups and Single Sign-On](#).

When configuring an SSO segment, the session management of all sites in the segment should share the same definition:

- **Dynamic:** This is the best practice configuration for SSO. When all sites in the segment are set to dynamic session management, the session is restarted with every call to the server. The best practice is to set the session duration to the same length for all sites that share an SSO segment.
- **Fixed:** when all sites in the segment are set to fixed session management, the start of the session is counted from the first site to which the user logged in.

Consent

For customers who implement Gigya's [Consent Management](#), enables syncing the user's consent status to a consent object or custom field in the target platform. Connectors who have this feature can dynamically call any Gigya screen during the various customer interactions with their site.

This is supported for the following GConnectors:

- SAP Commerce Cloud
- Magento 2

Lite Screen-Set

Support showing the [Lite Registration](#) screen-set, for registering passwordless users via the connector integration. Lite registrations is used for newsletter subscriptions, lead generation and other customer communication use cases. The information is saved to Gigya's database.

This is supported for the following GConnectors:

- SAP Commerce Cloud
- Magento 2

GConnector Data Flow

The following diagram illustrates the typical data flow between Gigya ([RaaS accounts namespace](#) and [Data Store](#)) and the CMS (both client and server side), using the GConnector extension of the CMS.

- The [Accounts](#) namespace is where core Gigya user and identity data is stored, and includes the basic identifiers such as first and last name, email, UID and any other fields stored in the [profile](#) and [data](#) objects.
- Gigya's [Data Store](#) (DS) is a premium product, an additional database with a flexible, dynamic schema. The DS part of the flow below is relevant only to customers who have implemented the Data Store.

For a list of links to the APIs participating in this flow and other relevant documentation, see below.

This flow is relevant both to implementations that use Gigya's [Screen-Sets](#) and custom screens with Gigya's Web SDK.

APIs and Objects Used in GConnectors

- [accounts.login JS](#)

- accounts.getAccountInfo JS
- accounts.exchangeUIDSignature REST
- accounts.getAccountInfo REST
- accounts.setAccountInfo REST - manually called in the Backoffice sync module; otherwise this API is automatically used by RaaS Screen-Sets.
- Profile REST
- accounts.getPolicies - used for validating the application key and secret entered when configuring the GConnector.
- ds.get REST - optional
- ds.search REST - optional

Products

GConnectors



Other Gigya-CMS Connectors

In addition, Gigya offers the following CMS connectors. These connectors support some of the GConnector functionality but not all of it. For more details, click on the relevant CMS icon:



Adobe Experience Manager



Adobe Experience Manager Mobile

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.

Feature Overview

All GConnectors support the following features:

- Language configuration
- Multi-site support
- Dynamic field mapping

In addition, the following matrix outlines supported features per connector:

CMS	Data Store Sync	Back Office (Bi-Directional) Sync	UID-Based Sync	Gigya-Led Session Management	User Deletion	Lite Screen Set	User Consent
AEM							
Drupal 7							
Drupal 8							
SAP Commerce Cloud							
Magento 1							
Magento 2							
Salesforce Commerce Cloud							
Sitecore							
Sitefinity							
Umbraco							
Wordpress							

Product Updates

You can view the change log of the following GConnectors:

- [Change Log - Drupal 7.x](#)
- [Change Log - Drupal 8.x](#)
- [Change Log - Magento 1.x](#)
- [Change Log - Magento 2.X](#)
- [Change Log - Sitefinity](#)
- [Change Log - Umbraco 7](#)

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