

Single Sign-on Integration (V2)

www.AnswerCart.com



1. General Information

The purpose of this document is to define the process of the single sign-on for community solution powered by AnswerCart.

1.1. Intended Audience

This document is intended for community administer and other IT professionals responsible for integrating the internal process with AnswerCart Communities.

2. Single Sign-On Settings on Community Control Panel

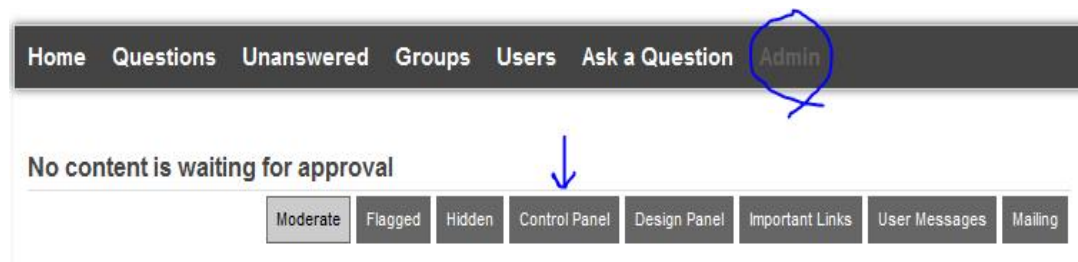
2.1. Add Enterprise Login Details

2.1.1 Login to the community

Log-in in to your community (<http://community.example.com>) as an administrator

2.1.2 Locate the control panel

Go to **Admin > Control Panel** Tab to open configuration setting as follows –



2.1.3 Add website signin/signout/signup links

1. Locate **Single Sign-On Configuration** Option
2. Check **Enterprise Login Required**
3. Add the required **URL**'s in the textbox to redirecting your link.
 - 1) Signin (login) Link
 - 2) Signout (logout) Link
 - 3) SignUp (register)Link

Single Sign-On Configuration

☒ Enterprise Login Required:

Enterprise Signin URL:

Enterprise Signout URL:

Enterprise Signup URL :

☐ Disable Direct Login

2.1.4 Disable Direct Login

Check Disable Direct Login if we don't want to show the direct login at top of the page (recommended with the single sign-on).



A horizontal login bar with a dark background. It contains an 'Email' input field, a 'Password' input field, a 'Remember' checkbox with a checked mark, a blue 'Login' button, and an orange 'Register' button.

2.1.5 Add Enterprise Secret Key

Secret key is a method of exchanging information between the community and master website i.e. example.com (this parameter shall be kept private).

Secret Key Configuration

Secret Key:



A text input field containing a series of 'x' characters, representing a secret key.

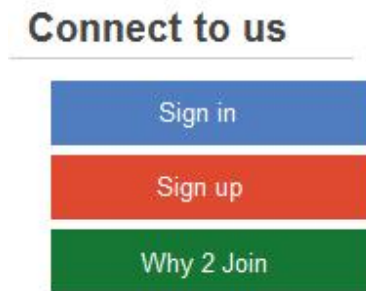
2.1.6 Save the Configuration

Locate the save button at the bottom and press it.

2.2. API Integration

2.2.1 New Register/Login Method

After the Step 2 user will see the following options for the login/register.



2.2.2 URL Transfer to the new Login or Register

Pressing of the signin page or signup page control will move to the login/signup page as stated in the section 2.1.3 and the calling of the url would be as follows -

A screenshot of a web browser's address bar. It contains the URL: `www.example.com/login?redirect_uri=http%3A%2F%2Fcommunity.example.com%2F`

2.2.3 Obtain the Login Token

Login token can be obtained by calling the “login-token” json-api along with the email and secret key.

API Input Parameters:

Key=md5(secret_key)

Email= <email of the user for which token to be obtained>

API Output Parameters:

error = false <if API is successfully processed>

email = <user email>

token = <generated 32byte token>

time = <timestamp>

Example API Call

`http://community.example.com/json-request/login-`

`token?key=58fadd7bb100785b7ff9f11ee4801891&email=sample@answercart.com`

2.2.4 Internal login/register to the community from master website

Login Parameters

1. keypass

keypass is calculated by concatenating secret key (2.1.5) with the email and then by taking the md5 of result. i.e. –

keypass=md5(concat(<secretkey>, <email>);

2. token

token obtained in step 2.2.3, a token can be used only for once. **After successful login token gets expired for the given user.**

token=<token obtained in the step 2.2.3>

3. email

Verified email of the user.

4. fullname

Full Name of the user which can contain special character but in the encoded way.

5. photo_url (optional)

Photo address of the user.

6. redirect_uri (optional)

Web address at which control would be transferred after the internal login/registration is completed. If this parameter is not present then default action at the community would take place.

Note: If email is in caps but email in keypass is not then keypass will fail. So recommended way is to have both in small case with trailing spaces removed.

HTTP API

http://community.example.com/login?keypass=<keypass>&token=<token>&email=<email>&fullname=<fullname>&photo_url=<photo_url>&redirect_url=<redirect_uri>

2.2.5 When and How to call the API

Case 1: User is logging on the master website (login to community internally)

In this case master website shall call the login API of community as stated in section 2.2.4 just after the login/registration process is completed but with the redirect_uri so that control will come back to the master website after the internal login/registration.

Case 2: User is on the community section and calling the login/signup

In this case community would call login/signup page as stated in the section 2.1.3, and once login/registration process is completed, master website shall transfer the control to the community by calling the API as stated in section 2.2.4 but **the API shall not have redirect_uri.**

2.2.6 Logout Process**Case 1: User is logging out from the master website**

Master website shall logout first from the community internally before proceeding to the logout from the master website. Logout call shall be as follows –

http://community.example.com/logout?redirect_uri=<masterwebsite_url>

Case 2: User is logging out from the community website

Community will call the logout link as stated in the section 2.1.3 with the redirect_uri as community logout url. In this case master website shall complete the logout first without calling the logout process of community followed by calling the url stated in redirect_uri.