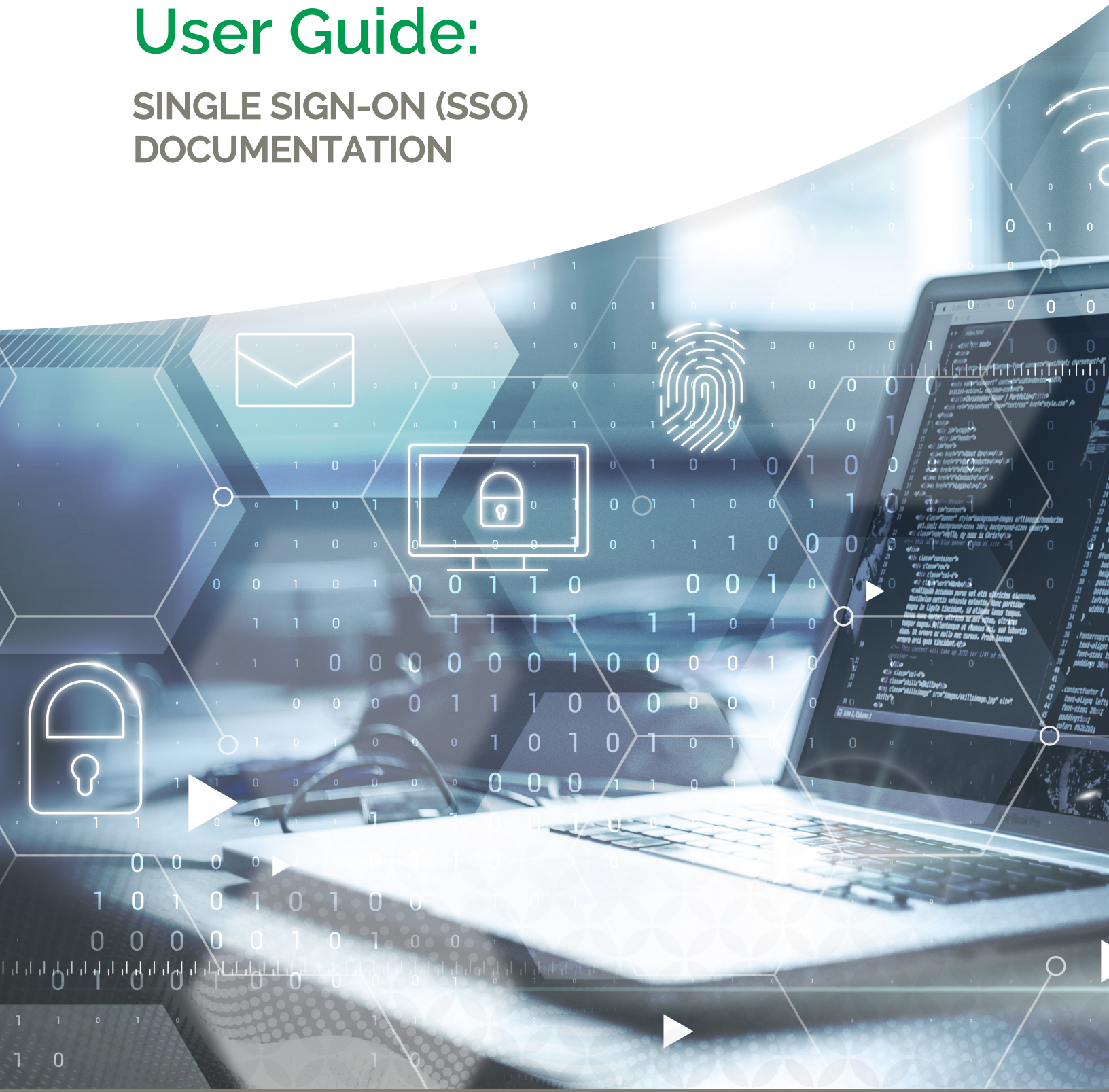
MemberSuite

User Guide:

SINGLE SIGN-ON (SSO) DOCUMENTATION



47 Perimeter Center E, Suite 300 | Atlanta, Georgia 30346 | sales@membersuite.com | (678) 606-0310 | www.membersuite.com

SSO Documentation

Overview

Single Sign-On (SSO) is used with MemberSuite to provide a seamless transition for members from an organization's password protected environment into the MemberSuite Portal without the user having to re-enter a password or a completely different set of credentials.

In this document, we will address the different workflows that a given client may wish to pursue in order to fully implement SSO with MemberSuite. This ranges from authentication paths for login, logout, reset password, and forgot password to features involving linking your Content Management System (CMS) to the MemberSuite portal for event registration and much more.

Single Sign-On

As part of the single sign on process it will be necessary to obtain an AuthToken using SOAP API credentials. This process is something that must be done involving an API user. This user can be set up in the console. You will also need the SSO Sample SDK to generate the token for you to initiate the calls based on a given user. This can be used interchangeably with the JWT token that is mentioned more prominently in the REST API documentation.

Here I have included a sample cURL that can be used to emulate the initial REST call to authenticate with the SOAP token generated from our signing certificate and secret access key specific to your association. For further information on this please see the SOAP documentation.

To verify that you are gaining the correct information you can also use Postman to emulate the appropriate calls to generate the response as it should be blow.

```
curl -X POST \
  https://rest.membersuite.com/platform/v2/regularSSO \
  -H 'Content-Type: application/x-www-form-urlencoded' \
  -H 'Postman-Token: 2c0ef72c-538b-4d10-8d32-330594fb5c73' \
  -H 'cache-control: no-cache' \
  -d 'Token=ValueOfTokenGeneratedFromSDKGoesHere&ReturnText=&LogoutUrl=&undefined='
```

The call above will return a URL with a TokenGUID which can be used to sign in seamlessly into the site. You can see the GUID and the appropriate URL return from the REST API in Postman when you attempt to make the call. (**Help Tip:** If using postman, set the Settings to not automatically redirect to the URL in order to see the response URL below) The format should look something like below:
https://associationAcronym.users.membersuite.com/auth/sso?tokenGUID=2f863665-d197-4664-a498-500b245fbfe3&nextUrl=&returnText=&logoutUrl=

Single Sign-On: Specific Use Case Flows

Any word in `` is a variable name: eg: `eventId` stands for the actual id of the event. This id will vary from event to event.

Event Flow:

Step - 1: As a non-logged in user, user hits the URL: https://associationAcronym.users.membersuite.com/event/`eventId`/details?entry=0

Step - 2: The user is redirected to the CMS login URL (Plugin Provider) with query params: nextUrl=events, entry=1, id=`eventId`

Step - 3: If the user logs in: The user is redirected to https://associationAcronym.users.membersuite.com/auth/sso?tokenGUID=`guidToken`&nextUrl=events&eventId=`eventId`&entry=1

Next the user is navigated to event registration flow

Step - 4: If the user signs up: The user is redirected to https://associationAcronym.users.membersuite.com/auth/sso?nextUrl=events&eventId=`eventId`&entry=1&signup=1

Next the user is navigated to Cognito sign up and then event registration flow

*Note:

- In case of signup, user won't be logged in the CMS site
- In case of signup, tokenGUID should not be there
- In case of forgot password, user won't be logged in the CMS site
- In case of forgot password, tokenGUID should not be there
- In case of sign up, the user has to navigate through the Cognito sign up flow (and can't be redirected to CMS login) because the token won't be available with CMS until the user completes the first flow of registration
- In case of forgot password, the user must navigate through the Cognito forgot password flow (and can't be redirected to CMS login) because the Cognito redirect_uri does not work for Cognito forgotPassord page. The Cognito forgotPassord page is designed to navigate to the Cognito login page only

Normal Flow:

If the user logs in: The user is redirected to https://associationAcronym.users.membersuite.com/auth/sso?tokenGUID=`guidToken`&nextUrl=`nextUrl`

If the user signs up: The user is redirected to https://associationAcronym.users.membersuite.com/auth/sso?nextUrl=`nextUrl`&signup=1

*Note:

- If the nextURL is a complete URL including https protocol, then it will be treated as a public website URL and the user would be redirected to the nextURL in the CMS site.
- If the nextURL is a partials URL (eg: store/browse), then it will be treated as a portal URL and the user would be redirected to the nextURL in the portal site.

Logout:

If the user logs out from public website, he is redirected to: https://associationAcronym.users.membersuite.com/auth/sso?logOut=1

The user logs out of portal and is redirected back to CMS logout URL

If the user logs out from portal, he is redirected to the CMS logout URL, where he would be logged out of the CMS site as well

Forgot Password:

Event Flow: If the user needs to reset the password: The user is redirected to: https://associationAcronym.users.membersuite.com/auth/sso?nextUrl=events&id=`eventId`&entry=1&forgotPassword=1

Normal Flow: If the user needs to reset the password: The user is redirected to: https://associationAcronym.users.membersuite.com/auth/sso?nextUrl=`nextUrl`&forgotPassword=1

Notes:

- In case of forgot password, the user has to navigate through the Cognito forgot password flow (and can't be redirected to CMS login) because the Cognito redirect_uri does not work for Cognito forgotPassord page. The Cognito forgotPassord page is designed to navigate to the Cognito login page only
- In case of forgot password, user won't be logged in the CMS site
- In case of forgot password, tokenGUID should not be there

SSO Configs:

The home URL, login URL and logout URL can be configured in SSO settings in the portal console.

*If the URL protocol is not mentioned (without http or https), https is considered as the default protocol

APIs Used:

/regularSSO: POST Method: CMS site submits the token to this REST API. The response header of this API redirects the user to portal along with tokenGuid

/regularSSO: GET Method: Portal takes the tokenGuid from the query params and hits this API and gets the actual SSO token which is used as the access token going forward.