Logging into Go.Learn Using a Custom Single Sign On

Introduction

Even if the Go.Learn mobile app supports many different Single Sign On (SSO) protocols (namely SAML, OKTA, Google Apps, Gmail, Auth0 and OpenID Connect), other SSO protocols or some custom implemented SSO schemas may not be supported or may not work as expected in Go.Learn.
That's why Docebo introduced an advanced support for deep links and JWT tokens and a few customization options on the login page in order to enable all of the clients to implement and secure the login phase in the way they prefer.

Customizing the Go.Learn Login

In order to be able to customize the login to your Go.Learn mobile app, follow the following four steps:

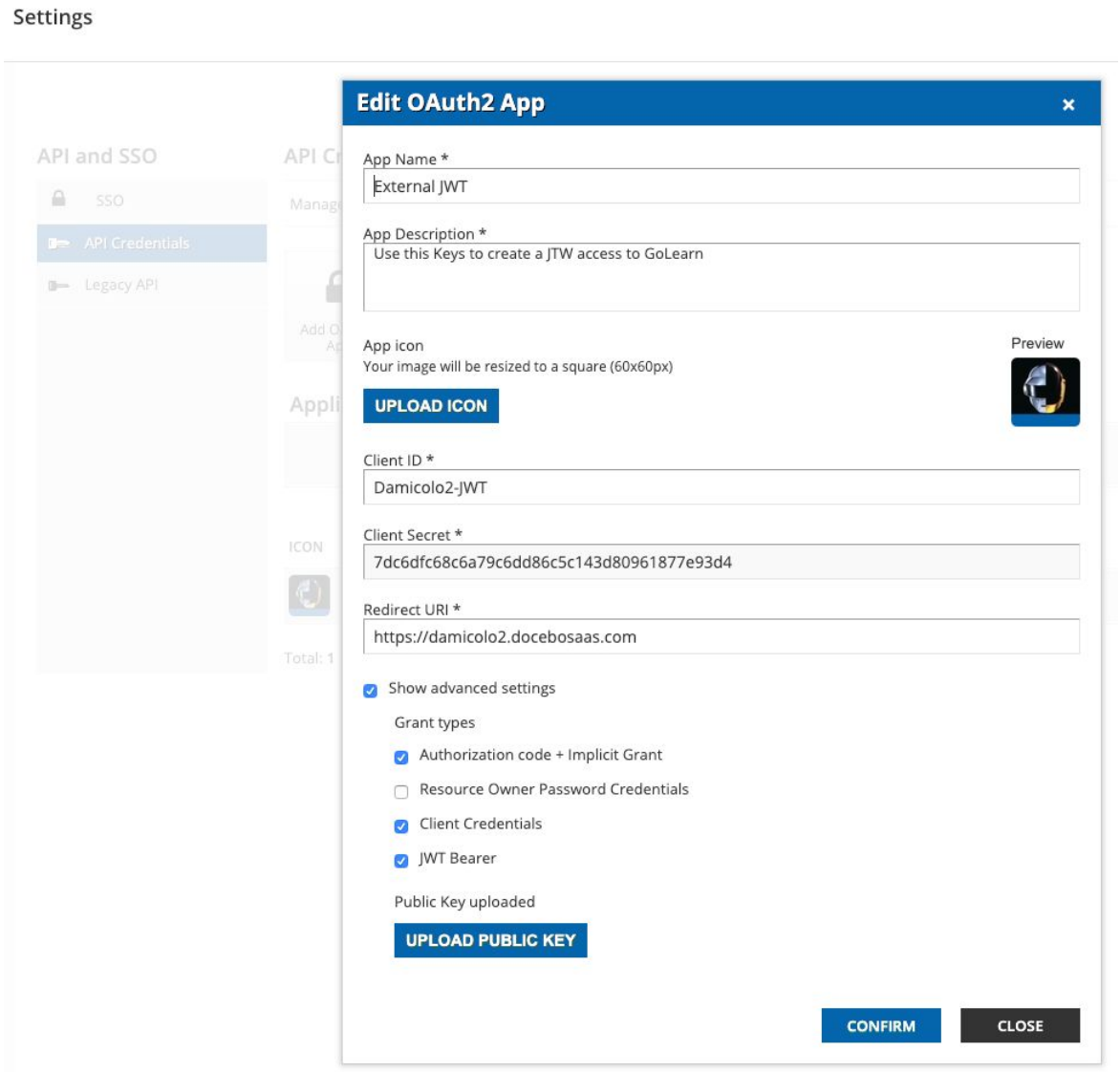
1. Create a new OAuth2 app in your Docebo platform and enable the support for JWT tokens.
2. Implement the actual user authentication in a web portal that you control or within any app which is used and developed by your organization.
3. Once the user authentication has been completed successfully, redirect the user to the Go.Learn app (or to your branded version of Go.Learn) using a deep link and passing a proper JWT token as a parameter.
4. Optimize the app experience to prevent confusion in the users.

1. Create a New OAuth2 App

First of all, you need a couple of RSA private/public key in .pem format. You are free to create them in the way you prefer, for example using OpenSSL or this free online service.

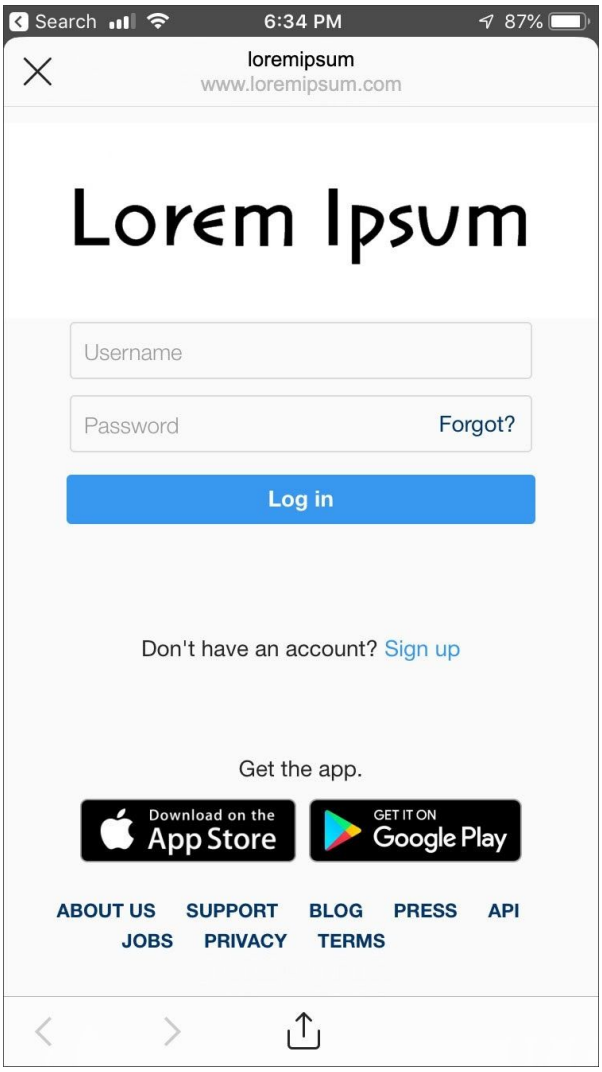
Next, access the **Admin Menu** of your Docebo platform from the **gear icon** in the top right corner of the page. Select the **Manage** option in the *API and SSO* section. On the page that opens, access the *API Credentials* tab and press the **Add OAuth2 App** button. A pop-up box will appear. You will need to fill in its fields according to the instructions provided in this article about the API App, and in this article about the APIs authentication.

Please Note: Remember to flag the *JWT Bearer* option in the advanced settings at the bottom of the pop-up box. Finally, you will be required to upload the public key you created at the beginning of this procedure. Once finished, press **Confirm.**

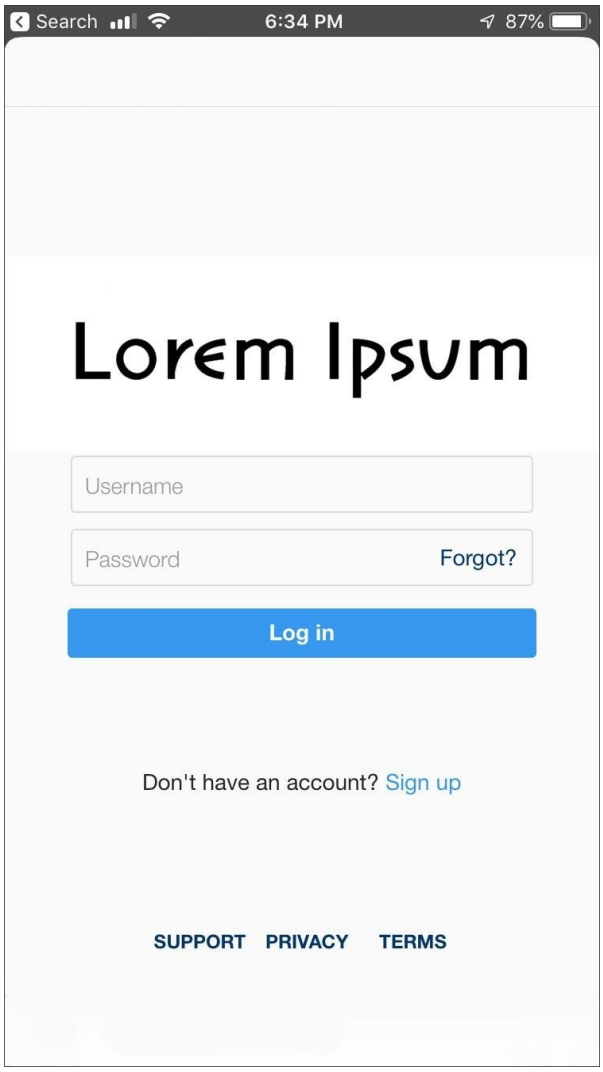


2. Implement the Login Procedure

Feel free to implement the login in a web portal managed by you, or in any app used by your organization.



Sample of Web-based Login Page



Sample of App Login Page

On the login page, your user will type his or her username and password. When the user taps on the **Log In** button, you can verify his or her credentials using the standard procedures you implemented.

3. Redirect to the App Using a JWT Token

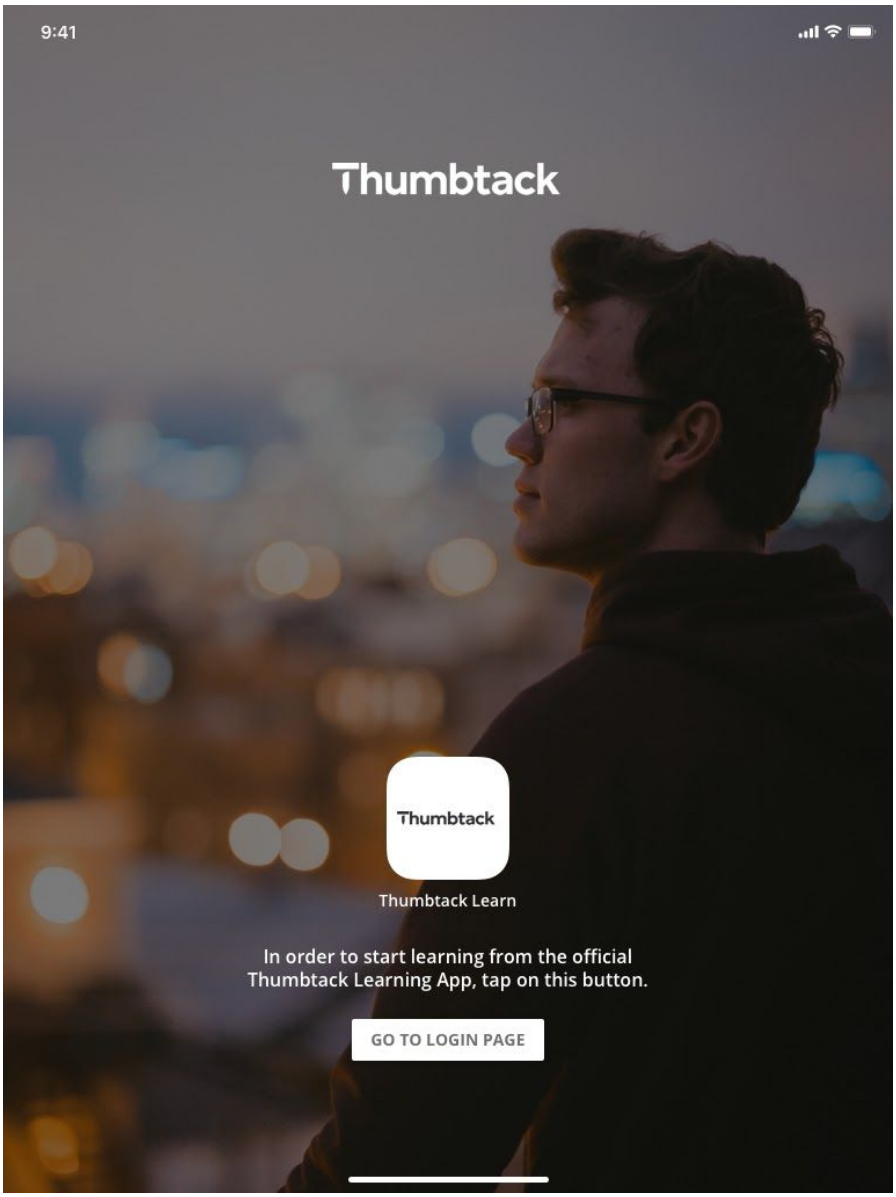
At the end of the authentication phase, instead of moving your user to any landing page, you now need to create a JWT bearer containing the username and other necessary information.
Remember to sign the JWT token using the RSA private key you created at the beginning of step 1 *Create a New OAuth2 App* above in this document. For more info about how to build a proper JWT token to be used in the Docebo platform, please refer to this article.
Finally, call the Go.Learn deep link to start the app (it can be the Go.Learn app or your own branded app) passing the JWT bearer you just created as a parameter. Read this article for further information about the deep links supported by Go.Learn.

Below, you'll find an example of the structure of a Go.Learn deep link:
golearn://allchannelspage?token=dfeyfdsfSFfsdFSefesJhbGc.jM5MHrtFDeDIyf.EoFdeEwmOx6Iy

4. Optimize the in-App Experience

Given that you now implemented the login procedure **outside** of the app, it could be misleading for a user to see the login options within the app. For this reason, it is recommended to hide these options within the app and give the correct instructions so that the user is able to login in the proper way.

For example, you can arrange the login page of the app to look like this one:

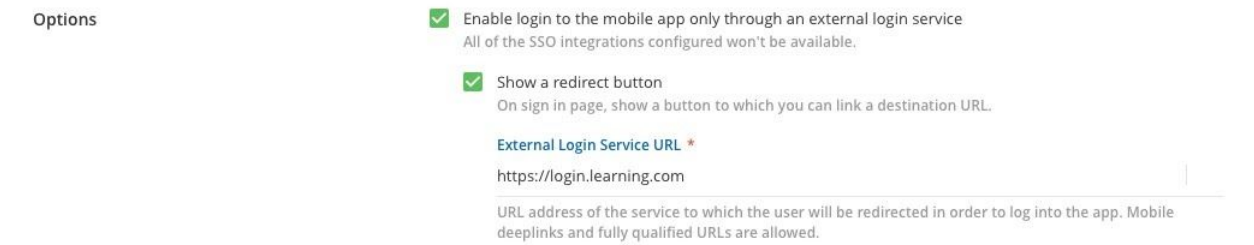


On this login page, the user doesn't see any username/password boxes or any links to the SSO logins, but there is only a button (**Go to Login Page**) that redirects the user to the actual login page.

In order to configure the login page in this way, access the **Admin Menu** of your Docebo platform from the **gear icon** in the top right corner of the page. Select the **Configure Branding and Look** option in the *Settings* section. On the page that opens, access the **Mobile App** tab, then reach the *Sign In Page* section.

In the *Sign In Page Background* subsection of the *Sign In Page* section, you can customize the background image of this page. Please note that in order to provide your users with more detailed information, you can also embed any text you want in the background image.

In the *Options* subsection of the *Sign In Page* section, you can configure the following settings:



Here, you can choose to hide all of the login with SSO buttons by enabling the login to the mobile app only through an external login service. You can also decide to show/hide a button to redirect the user to the proper login page by flagging or unflagging the corresponding option (*Show a redirect button*), that appears when selecting the *Enable login to the mobile app only through an external login service* option above. When selecting the redirect button option, you will need to add the URL (or a deep link) to redirect the user to the correct login page in the *External Login Service URL* field that appears below.

Finally, remember to switch to the **Desktop** tab on the **Configure Branding and Look** page, open the *Sign In Page* section and activate the *Show only SSO buttons and hide login form* toggle in the *Login Form* subsection.