Single Sign-On

OutSystems provides single sign-on capabilities by default: after authenticating in one of the applications, the end-users can access all other applications without having to provide the credentials. Single Sign-On is supported only on modules that are enabled to use cookies.

Using Single Sign-On

All created applications have single sign-on capabilities by default, because their User Provider module is set to Users. However, if you are using a different User Provider:

- Identify the **User Provider module** which provides the end-users and sessions to other modules and open the module in Service Studio. Set the module property **Is User Provider** to **Yes**, and publish the module.

- Identify the User Subscriber modules that share the end-users and sessions with the User Provider module. Open each one of the subscriber modules and select the **User Provider module** in the drop-down list of the User Provider module property (found in the module property).

In a Single Sign-On scenario, check out your unified modules in Service Center: edit a module and select the **Single Sign-On** tab to see the User Provider module and User Subscriber modules.

Sharing Users

The User Provider module contains an entity that stores the end-user information. This entity is set with the **Public** property to **Yes** so that it can be reused by User Subscriber modules.

To authenticate an end-user, you should check if the provided credentials match a record in the entity of the User Provider module.
Sharing Sessions

The session is created the first time the end-user accesses the server to request a page of any unified module. However, since the session is shared by the unified modules, the first On Session Start action to be invoked is the one of the User Provider module. Only then, if there is an On Session Start action in the requested module, it is invoked.

From then on, the On Session Start actions of the User Subscriber modules are invoked the first time each module is accessed via a screen, web block, or public action.

In a Single Sign-On situation, there is only one session shared by the unified modules which consists of all the session variables defined in these modules. If you need to reference or change a session variable from another module, you must use public actions, because there is no other way to add and remove session references between modules.

Note that sharing session variables through public actions only works for unified modules of the same set. Otherwise, as you will have different sessions, you will refer to different variables from each module.