Two Factor Authentication Component Plugin

Overview

The Two Factor Authentication component plugin allows you to add a T-OTP (Time based OTP) authentication within Appian interfaces. It can be incorporated into Appian interfaces to conditionally handle visibility of interface sections based on T-OTPs generated from Google Authenticator or Microsoft Authenticator.

Features

The component plugin includes two components namely

1. Authenticator Field
2. Registration QR Field

Authenticator Field

The Authenticator Field allows the users to input a 6-digit T-OTP and validate the entered T-OTP to get the validation status.

Parameters

Sl No	Name	Type	Description
1	align	Text (Optional)	Specify the horizontal alignment of the component. Valid values are: START(default), CENTER and END.
2	connectedSystem	Connected System (Required)	The Authenticator client connected system object constant.

Sl No	Name	Type	Description
3	onValidate	Saveinto	Executes when the authenticate button is clicked. Saves the authentication status as a boolean value on authenticate button click.
4	pseudoSecret	Text (Required)	Provide a partial secret phrase to control the generation of the secret.

The pseudoSecret must be provided as a non-empty value that was used in the registration component. **Fox example:** If user A uses a pseudoSecret value of "Test Secret $$" in the Registration QR Field component, then the generated QR will contain parts of the pseudo secret value. Thus **the same value** has to be provided as the value to the **pseudoSecret parameter** in the **Authenticator Field** in order to properly validate the entered T-OTP.

Considerations

The T-OTP generation and validation is based on the generated secret and the moving factor that is, time. Thus, several issues may arise if the **clocks** of the client system and the server are **not synchronized**. In such cases the component field will throw an error indicating the **step count** and whether the server's clock is behind or ahead of the client's clock.

The step count difference can be used to synchronize the clocks by calculating the exact time difference. To calculate the time difference from the step count, use the following formula:

Time difference = Step Count * 30 (seconds)

Screenshots

Note: The working of the authenticator component is that it gets the T-OTP as input from the user and invokes the client API from the Authenticator client connected system plugin and executes a saveinto indicating the validation status as a boolean value.

Registration QR Field

The Registration QR Field component generates a QR image for registering with the authenticator application. The generated QR code can be scanned either using the Google or Microsoft Authenticator apps.

Parameters

Sl No	Name	Type	Description
1	align	Text (Optional)	Specify the horizontal alignment of the component. Valid values are: START(default), CENTER and END.
2	connectedSystem	Connected System (Required)	The Authenticator client connected system object constant.
3	onValidate	Saveinto	Executes when the authenticate button is clicked. Saves the authentication status as a boolean value on authenticate button click.
4	user	Text (Optional)	The user for whom the QR code will be generated. The value of this parameter will show up in the authenticator as the requester account name. Defaults to empty string.
5	pseudoSecret	Text (Required)	Provide a partial secret phrase to control the generation of the secret.

The pseudoSecret value provided will create variations in the generated QR Code. Thus, this field can be provided with the same value to create an application-wide common code generation where all registered users will get the same T-OTPs based on their clocks.

This value can be made unique to each user thus, creating a separate T-OTP generation cycle for each registered user.

Considerations

The Registration QR field contains the generated secret encoded as a URL within the QR Image. The generated secret is not back traceable and hence the secret is not exposed. But the QR code can be scanned to register and receive T-OTPs if any user's account is exposed and the actual user and the imitator will receive the same T-OTPs thus, eliminating the restriction.

One way to avoid the mentioned issue is to use an assigned pseudo secret value to each user which they will have to manually enter to generate the QR code and to validate. This restricts the access to the QR Code thus, improving the security.

Screenshots

Note: The Two Factor Authentication Plugin requires the Authenticator client Connected System plugin.