



# Singpass Login

Support with Salesforce

**Emmanuel Schweitzer**

[eschweitzer@salesforce.com](mailto:eschweitzer@salesforce.com)

Principal Platform Solution Engineer





# What is Singpass Login

The Singpass logo, featuring the word "singpass" in a bold, sans-serif font. The "i" is black, and the rest of the letters are red. The logo is positioned on a light blue background that curves upwards from the bottom left corner of the slide.

**singpass**

Login enables Singapore residents' easy access to government and private sector digital services. Using the Singpass app, residents can securely log in to digital services without passwords.

This service provides businesses with an accessible and established authentication gateway for all their digital services.

# Objectives

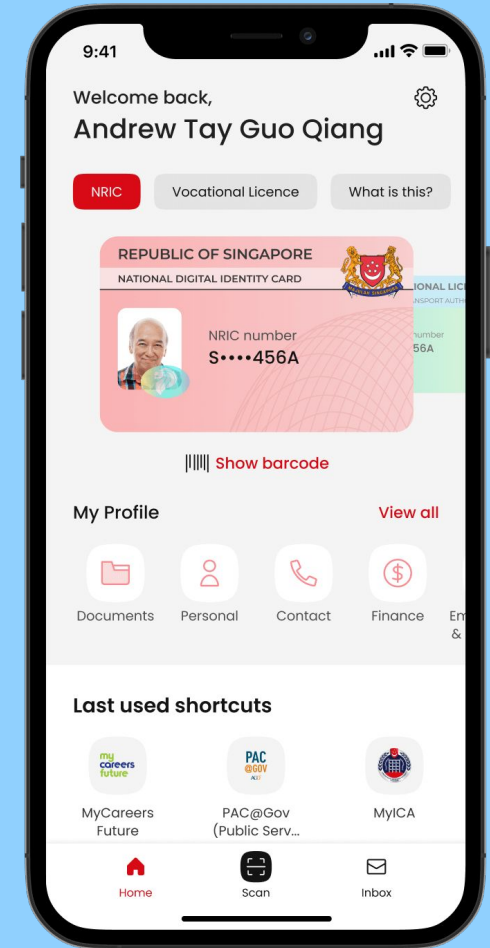
## What did we aim for?

This guide is meant to give you a high-level view of how you could implement an end-to-end login workflow for Salesforce Experience Cloud leveraging SingPass Login for B2C or G2C use cases in Singapore.

It's based on hands-on experiences conducted in the second half of 2021 and early 2022.

Use this as a base or source of inspiration, making your own enquiries and decisions based on what's right for your particular use case.

Lessons learned would apply for signing-on the Salesforce enterprise app as well.



# Foreword

salesforce

Singpass Login is not a trivial SSO/OIDC and it has marquee security features to make it extra safe to use.

This adds a few provisions you need to be aware of, including owning your own domain and SSL certificate from a tier 1 vendor.

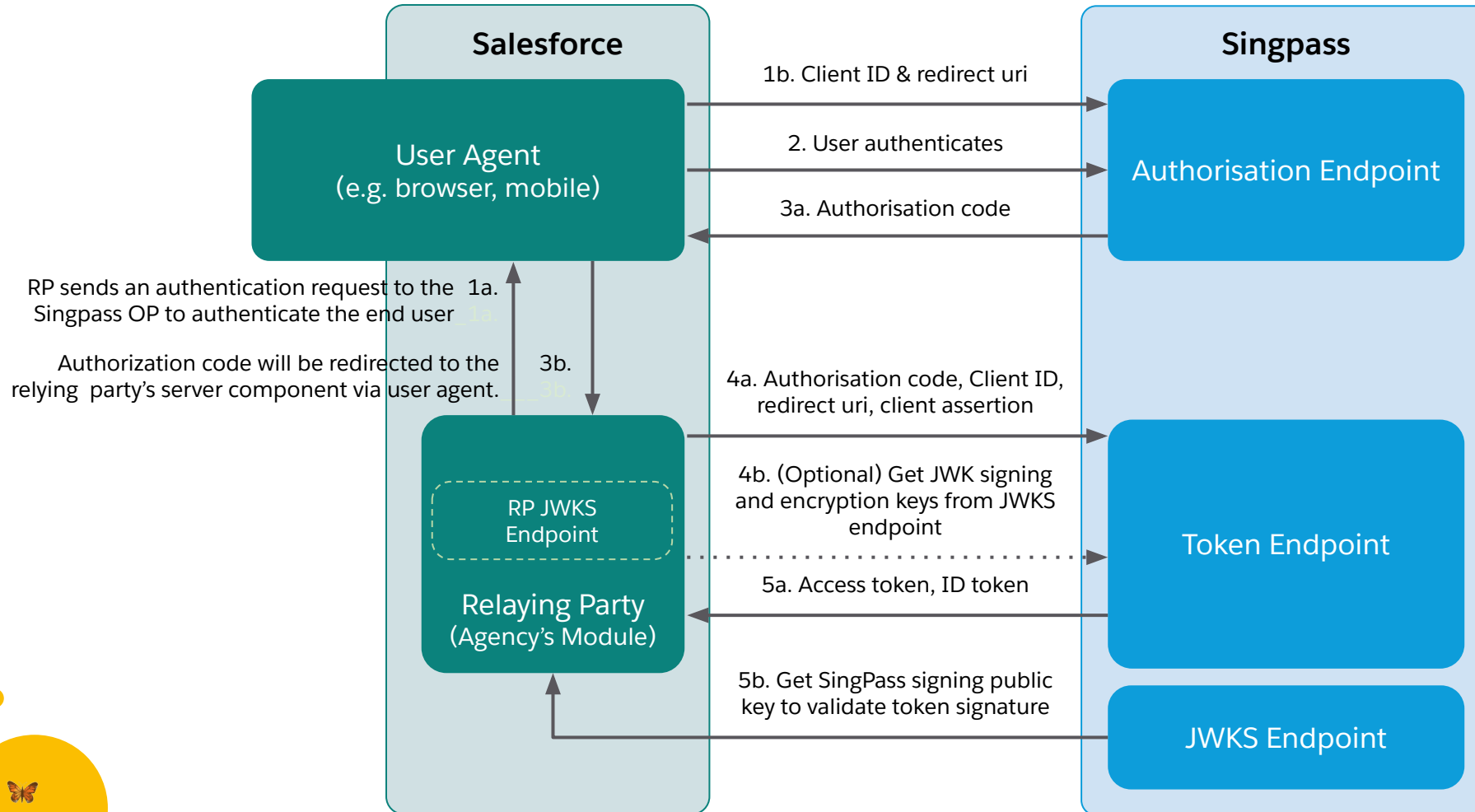
We urge you to thoroughly review the available [documentation](#) from the Government of Singapore (including detailed specifications) before attempting to go through this guide and starting implementation.

A good understanding of OIDC, JWT, Encryption and Signature is of an advantage.

# End to end login workflow



Summary, from Salesforce (Relaying Party) to Singpass OP (Identity Provider)



Inspired by <https://api.singpass.gov.sg/library/login/developers/overview-at-a-glance>



# End to end login workflow

## From Salesforce (RP) to Singpass OP (IdP) – Breakdown

SingPass is essentially following OIDC with a twist. Our suggestion is hence to break down the end to end workflow as follows:

1. Create a Custom Auth Provider named **Singpass** and implemented as the **Singpass\_AuthProvider** Apex class
2. Leverage a Visualforce page named **ndi\_auth** to initiate SingPass authentication
3. Have a second Visualforce page named **ndi\_auth\_cb** mediating SingPass' callback with expectation of that of the Custom Auth. Provider (additional logic required)
4. Have the **Singpass\_AuthProvider** Apex class get an access token with a signed request
5. Have a last visualforce page named **jwks** to serve public signing and encryption keys to SingPass

**SingPass**  
Custom Auth. Provider

**SingPassAuthProvider**  
Apex Class

**ndi\_auth**  
Visualforce Page

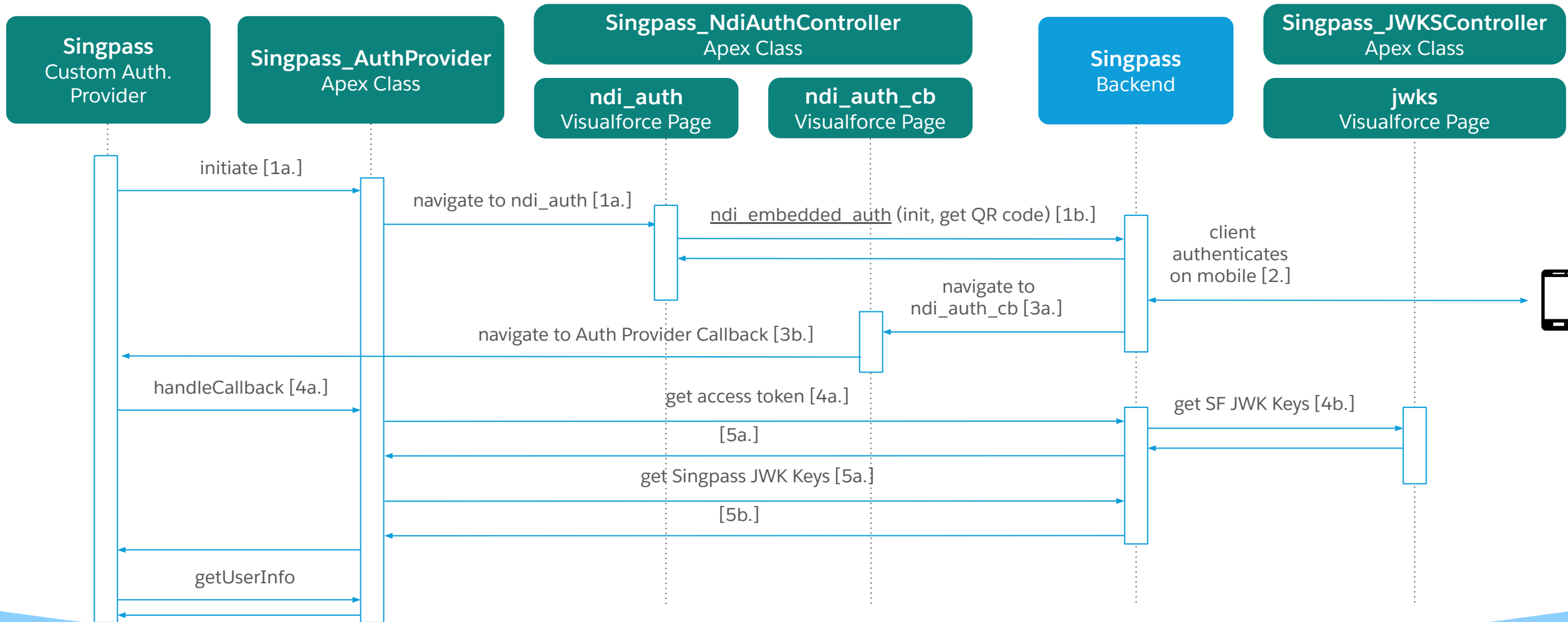
**ndi\_auth\_cb**  
Visualforce Page

**jwks**  
Visualforce Page

# End to end login workflow



From Salesforce (RP) to Singpass OP (IdP) – Sequence diagram





# 1a. and 1b. Authentication request

From Salesforce (RP) to Singpass OP (IdP) – Details

- Visualforce page named **ndi\_auth** to...
  - make note of the login state generated by Salesforce (as it is too large for SingPass to process)
  - generate a Singpass-compliant state that will be associated with the Salesforce state using an entry in the platform cache
  - pass these details and display a SingPass login QR code by leveraging Singpass' NDI embedded auth, e.g.  
[https://stg-id.singpass.gov.sg/static/ndi\\_embedded\\_auth.js](https://stg-id.singpass.gov.sg/static/ndi_embedded_auth.js)

**Singpass**  
Custom Auth. Provider

**SingpassAuthProvider**  
Apex Class

**ndi\_auth**  
Visualforce Page



## 2. User authenticates

From Salesforce (RP) to Singpass OP (IdP) – Details

This happens out of band in the Singpass mobile app



**SingPass**  
Backend

# 3a. Authorisation code

## From Salesforce (RP) to Singpass OP (IdP) – Details

- Second Visualforce page named **ndi\_auth\_cb** mediating Singpass' callback
  - gets a call back with the authorisation code and state info it was provided
  - verifies the state send to Singpass by checking that it's present in the platform cache
  - retrieves the associated Salesforce state from the platform cache
  - performs the actual callback as set in the Custom Auth Provider



**SingPass**  
Custom Auth. Provider

**SingpassAuthProvider**  
Apex Class

**ndi\_auth\_cb**  
Visualforce Page

**Singpass**  
Backend



## 4a. Access token request

From Salesforce (RP) to Singpass OP (IdP)

Custom Auth Provider named **Singpass** and implemented as the **Singpass\_AuthProvider** Apex class to:

- Prepare an access token request, which is embodied by a jwt that must be signed with your private signing key as listed in jwks
- Keys must be large Elliptic Curve (EC)
- Signing a JWT using one of these wasn't possible at the time we wrote this guide due to a bug in `Crypto.sign` / `Crypto.verify`
- Note that the EC signature algorithm is too CPU intensive to consider an Apex implementation as an alternative; it would not finish within the allocated CPU slice under governor limits, even in batch
- The suggestion is to offload the signature to Salesforce functions (preferred), or Heroku (acceptable with strong security measures, e.g. two way certificate, IP whitelisting...). Implementation should be lightweight with the JOSE library in JavaScript or Java.

**Singpass**  
Custom Auth. Provider

**Singpass\_AuthProvider**  
Apex Class

**JWT Signature**  
Heroku or Salesforce  
Functions

**Singpass**  
Backend



## 4b. Access token request

From Salesforce (RP) to Singpass OP (IdP)

**Singpass** Backend queries the jwks endpoint which we suggest to expose as a Visualforce page:

- Could be a very simple implementation that returns a static JSON content with both public signature and encryption keys in JWKS format
- Alternatively the keys could be stored somewhere else (e.g. custom metadata), with the Visualforce page performing a query and rendering dynamically
- Note that Certificate Management in setup can only be used with RSA keys at this stage



**jwks**  
Visualforce Page

**Singpass**  
Backend





# 5a. and 5b. Access token response

From Salesforce (RP) to Singpass OP (IdP)

The **SingPass** backend returns an Access token response that **Singpass\_AuthProvider** must...

- Decrypt using your private key and verify using **SingPass**' public key from **Singpass**' jwks endpoint
- Similarly to 4a. decryption and verification should be offloaded to Salesforce Functions (preferred) or Heroku
- Have a second Visualforce page named **ndi\_auth\_cb** intercepting **Singpass**' callback to...
  - verify the state send to Singpass
  - retrieve the associated Salesforce state from the platform cache
  - perform the actual callback as set in the Custom Auth Provider

**SingPass**  
Custom Auth. Provider

**SingPassAuthProvider**  
Apex Class

**JWT Decrypt/Verify**  
Heroku or Salesforce  
Functions

**SingPass**  
Backend



# Final step: GetUserInfo

From Salesforce (RP) to Singpass OP (IdP)

SingPassAuthProvider must...

- Use the (little) information contained in the Access token response to decide which user will be logging in (or provisioned JIT if required)
- At best you'll receive the NRIC and a UUID uniquely identifying the SingPass customer
- Either of these would enable you to check for an existing user by leveraging a SQL query and a custom field on User
- However, to provision a user you would most likely need to combine this workflow with a call to Singpass MyInfo in order to retrieve useful data points like names and contact details to populate the user record; this could be done Just-In-Time or as a prior step, e.g. self-service.
- MyInfo could also be used to automatically update the User and Contact records.

**Singpass**  
Custom Auth. Provider

**Singpass\_AuthProvider**  
Apex Class

# Further Resources



Code Snippets for  
Salesforce and Heroku

[GitHub Repository](#)



Single-Sign On with  
Salesforce as the Service  
Provider

[Documentation](#)



JWT, JWS, etc...

[Beginner's guide](#)



Salesforce Functions

[Documentation](#)  
[Trailhead](#)



Heroku

[Documentation](#)  
[Trailhead](#)



A vibrant, stylized illustration of a forest scene. The background is a clear blue sky with three small, white, fluffy clouds. The top and bottom edges of the frame are decorated with lush green foliage, including various leaves and small, colorful flowers in shades of pink, yellow, and purple. On the left and right sides, the brown trunks of large trees are visible. In the lower-left area, a small orange butterfly is shown in flight. The central focus of the image is the text "Thank You" in a large, bold, dark blue font.

# Thank You



# Auth. Provider



SETUP

Auth. Providers

Auth. Provider

Help for this Page

Auth. Provider Detail

EditDeleteClone

Auth. Provider ID	0SO5j000000wuTC
Provider Type	Custom
Name	SingPass
URL Suffix	SingPass
Plugin	<a href="#">SingPass_AuthProvider</a>
Client ID	SingPass assigned clientId
Key ID	Your signature keyId from jwks
Token Url	<a href="https://stg-id.singpass.gov.sg/token">https://stg-id.singpass.gov.sg/token</a>
Custom Error URL	
Custom Logout URL	
Registration Handler	<a href="#">SingPass_RegistrationHandler</a>
Execute As	<a href="#">Emmanuel Schweitzer</a>
Icon URL	<a href="https://your-domain/resource/1629189946000/Sing...">https://your-domain/resource/1629189946000/Sing...</a>

Salesforce Configuration

Test-Only Initialization URL	<a href="https://eschweitzer-220321-12-demo.my.salesforce.com/services/auth/test/SingPass">https://eschweitzer-220321-12-demo.my.salesforce.com/services/auth/test/SingPass</a>
Single Sign-On Initialization URL	<a href="https://eschweitzer-220321-12-demo.my.salesforce.com/services/auth/ss0/SingPass">https://eschweitzer-220321-12-demo.my.salesforce.com/services/auth/ss0/SingPass</a>
Existing User Linking URL	<a href="https://eschweitzer-220321-12-demo.my.salesforce.com/services/auth/link/SingPass">https://eschweitzer-220321-12-demo.my.salesforce.com/services/auth/link/SingPass</a>
OAuth-Only Initialization URL	<a href="https://eschweitzer-220321-12-demo.my.salesforce.com/services/auth/oauth/SingPass">https://eschweitzer-220321-12-demo.my.salesforce.com/services/auth/oauth/SingPass</a>
Callback URL	<a href="https://eschweitzer-220321-12-demo.my.salesforce.com/services/auth/callback/SingPass">https://eschweitzer-220321-12-demo.my.salesforce.com/services/auth/callback/SingPass</a>
Single Logout URL	<a href="https://eschweitzer-220321-12-demo.my.salesforce.com/services/auth/rp/oidc/logout">https://eschweitzer-220321-12-demo.my.salesforce.com/services/auth/rp/oidc/logout</a>

► Experience Cloud Sites

EditDeleteClone

# Experience Cloud Login

## Workspace setup




### Login & Registration

Brand, configure, and customize your site's login experience, which includes pages used to log in users, verify identities, reset passwords, register members, and for login flows. Tip: To view your login pages as you work, use your browser's private browsing mode to access your site.

#### Branding Options

Customize your site's login experience to reflect your brand. You can use dynamic branding URLs to change how login and related pages appear at runtime. [Learn about Dynamic Branding URLs](#)

Logo Type	File	<a href="#">i</a>
Logo File	<div>Choose file   No file chosen</div>	
	JPG, GIF or PNG, 100 KB max.	
	<div></div>	
	250 px max	
	125 px max	
Background Type	Color	<a href="#">i</a>
Background	<div>#B1BAC1</div>	<a href="#">i</a>
Login Button	<div>#1797C0</div>	<a href="#">i</a>
Right Frame URL	<div></div>	<a href="#">i</a>
Footer Text	<div>Partner Central</div>	<a href="#">i</a>

#### Login Page Setup

Choose a login page type to create a branded login experience. Depending on the login page type, your users can log in with their username, email, phone number, or other user identifier. [Learn more](#)

Login Page Type	Default Page	<a href="#">i</a>
<input checked="" type="checkbox"/> Allow employees to log in directly to an Experience Cloud site <a href="#">i</a>		
Select login options to display on the login page. To add more login options, visit <a href="#">Single Sign-On Settings</a> or <a href="#">Auth. Providers in Setup</a> . <a href="#">i</a>		
<input checked="" type="checkbox"/> Salesforce Demo username and password		
<input checked="" type="checkbox"/> SingPass		