

Summary

This package includes 2 components - Custom Self Registration and Custom Login. It is designed for use in Experience Cloud sites where the out of the box components do not meet your business needs and configurability is required. The Custom Self Registration component requires that Self Registration is enabled on your site. The components can be used together to complement each other, or individually.

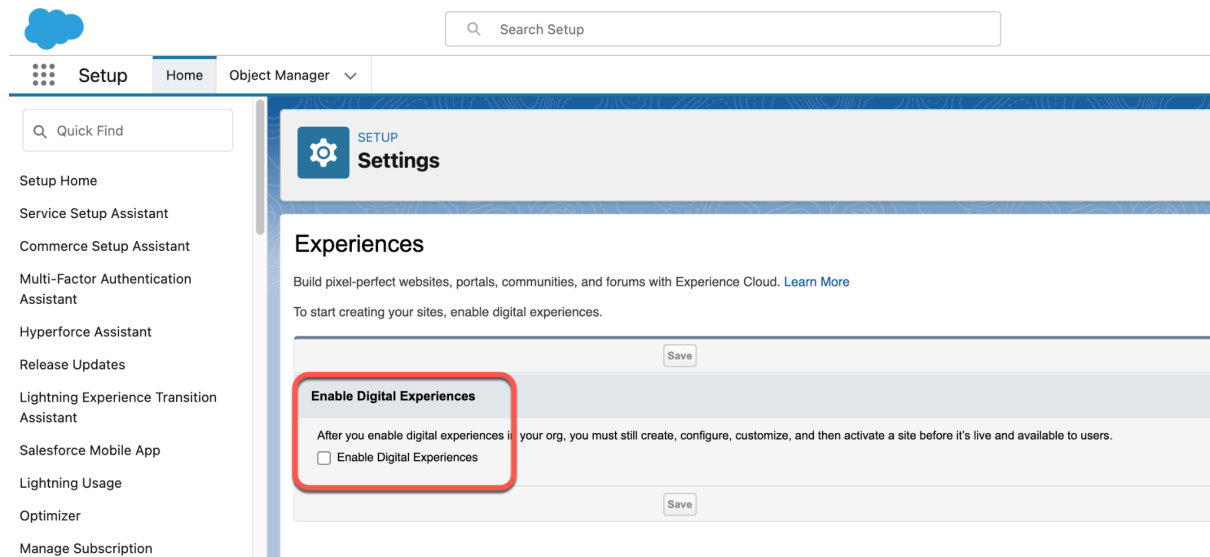
The components have been designed to accommodate the average Salesforce Administrator who wishes to simply install and configure with no code involved. You are only required to know a little bit about SOQL queries for the Custom Self Registration component, everything else is deliberately admin friendly! The components offer many configuration options which are described throughout this document and they may or may not fit your use cases exactly, but aims to fill the gaps and extend existing functionality to provide a more flexible solution.

Leave comments on your thoughts for improvements, or if you find bugs then raise an Issue via the [GitHub repository](#) so I can investigate and fix. I will aim to maintain the solution as much as possible as time allows Please note that unfortunately LWC development is not my full time job and this was created in my spare time.

Prerequisites

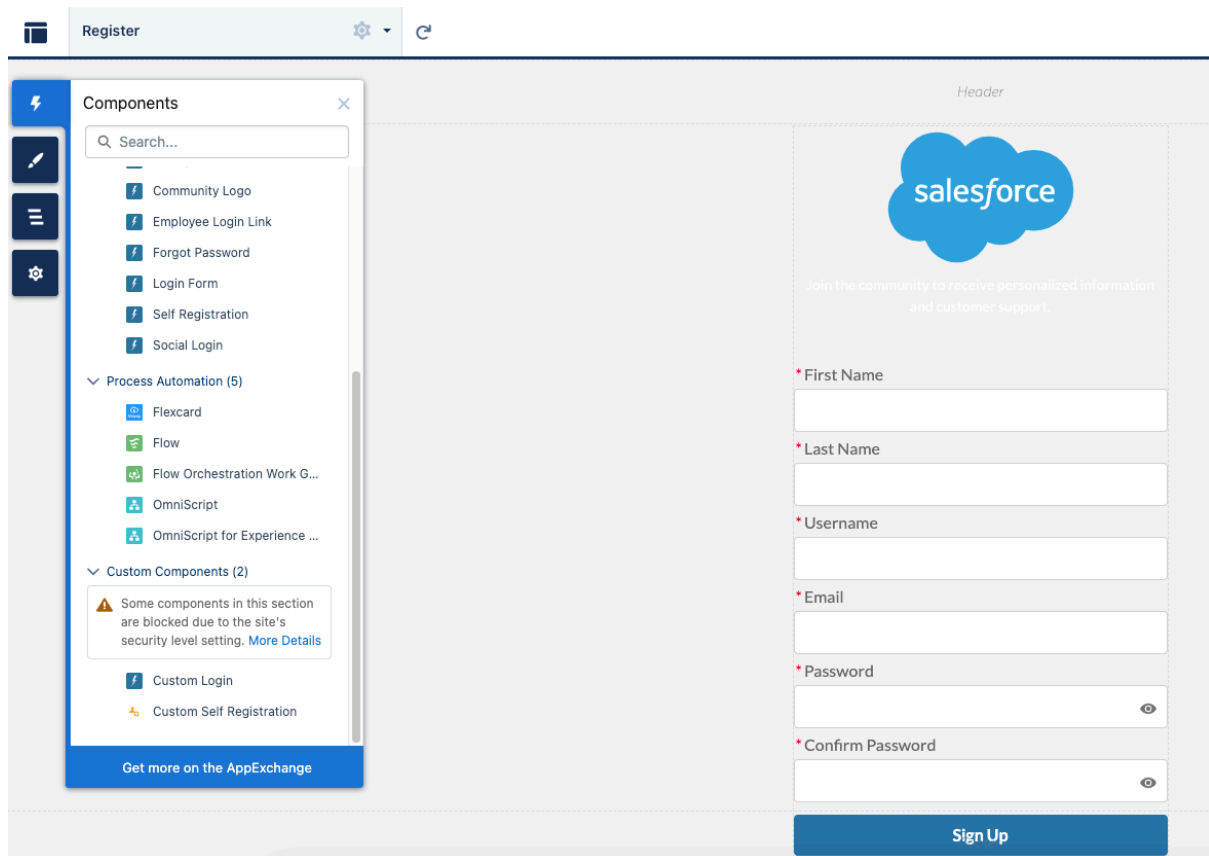
The package install requires that your Salesforce Org has the following features enabled before installing the AppExchange package:

NetworksEnabledOnce: This refers to Experience Cloud and enabling the Digital Experiences feature from Setup > Digital Experiences > Settings. Enable the setting, and save then try the installation again.



Installation

From the AppExchange, click Get it Now on the listing. Install in your organisation for Admin Users. Once the package has been installed on your org, two new Lightning Web Components will become available to you within the Experience Cloud Builder under **Custom Components**



Navigate to the Login or Register page, then simply drag and drop the relevant component onto the canvas to begin configuring it for use.

Permission Sets

There are three Permission Sets provided as part of the package:

- Custom Login - Guest Access
- Custom Login - Internal Access
- Custom Self Registration - Guest Access
- Custom Self Registration - Internal Access

These Permission Sets should be assigned appropriately to allow access to the necessary Objects & Fields, Custom Metadata Types and Apex Classes which control the component's various features.

Permission Set Name	Object & Field	Custom Metadata Type	Apex Class
Custom Login - Guest Access	Experience Cloud Event User (Custom Field only)	Custom Experience Cloud Setting	SiteLoginController SiteUtilities LoginHistory
Custom Login - Internal Access	Experience Cloud Log Experience Cloud Event User (Custom Field only)	Custom Experience Cloud Setting	SiteLoginController SiteUtilities LoginHistory
Custom Self Registration - Guest Access	Experience Cloud Event User (Custom Field only)	Custom Experience Cloud Setting Custom Registration Configuration	SiteRegistrationController SiteUtilities RetrieveRecordsForQuery RetrieveNetworkMembers
Custom Self Registration - Internal Access	Experience Cloud Log Experience Cloud Event User (Custom Field only)	Custom Experience Cloud Setting Custom Registration Configuration	SiteLoginController SiteRegistrationController SiteUtilities LoginHistory RetrieveRecordsForQuery RetrieveNetworkMembers

To assign a Permission Set to a User, open the relevant Permission Set, click Manage Assignments and then follow the wizard. Search for 'guest' in the All Users list will return a

Guest User for each Experience Cloud portal on your org, so ensure you choose the correct user if more than one is available:

The screenshot shows the Salesforce Setup interface. The left sidebar contains navigation links for Release Updates, Lightning Experience Transition Assistant, Salesforce Mobile App, Lightning Usage, Optimizer, and ADMINISTRATION. Under ADMINISTRATION, the 'Users' section is expanded, showing links for Permission Set Groups, Permission Sets, Profiles, Public Groups, Queues, Roles, User Management Settings, and Users. The main content area is titled 'Custom Self Registration - Guest Access' and shows a 'Select Users to Assign' section. Below this, there is a table of users. The user 'LWC Testing Site Guest User' is highlighted with a red box. The table has columns for Full Name, Username, Role, and Profile. The user 'LWC Testing Site Guest User' has the username 'lwc_testing@00dho0000018...' and is assigned the 'LWC Testing Profile'.

Full Name	Username	Role	Profile
Jim Lowe	jlowe	jimlowe@0dbho000000treu...	Custom Experience Clo...
LWC Testing Site Guest User	guest	lwc_testing@00dho0000018...	LWC Testing Profile
Mimi Webb	mwebb	mimiwebb@0dbho000000tr...	Custom Experience Clo...

Guest User Sharing Rules

The Custom Self Registration component runs as the Guest User when accessing it via an Experience Cloud site. The component has been secured so that it runs in a 'with sharing' context, but executes queries in a 'without sharing' context. Thus you should use the Access Level parameter with care - System Mode ignores Field Level/Object Level to the Guest User to run the specified query. User mode will enforce Field Level/Object Level Access as per your org settings. Criteria Based Sharing Rules can be used to share any relevant Accounts, Contacts or Cases with the Guest User to open up further access if needed.

Component Parameters

The following parameters are available for configuration once you have added the Custom Self Registration component to the Register page.

- **Custom Query:** Write a custom SOQL query to find a matching Contact, Account (if using Person Accounts) or Case to relate a newly registered Portal User to (External User). See below for more information.
- **Access Level for SOQL Query:** Controls the access level for the query above. Change to 'User' if the query should run as the Guest User or 'System' to run the query with elevated access and see all data.
- **Create Record on Registration (if not found):** If a record is not found using the above custom query, then a new record can be created and related to the External User instead.
- **Object Type to Create:** If the above setting is set to TRUE, set whether the component should create a Contact or Person Account. Note that Person Accounts must be switched on and configured before you can use this functionality otherwise you will experience all kinds of weird behaviour!
- **Account Id:** If the "Object Type to Create" setting is set to Contacts, then set the Account Id of an existing Account to link a Contact to. This ensures that visibility can be enforced for any Contacts that have self registered.
- **Person Account Record Type:** If the "Object Type to Create" setting is set to Person Accounts, then set the Record Type of the Person Account Record Type in your Org. Person Accounts must already be enabled in the org for this option to be configured successfully.
- **Send Email Confirmation:** When the user successfully registers, an email notification is sent to the user from Salesforce. You may wish to turn this off if using another tool such as Marketing Cloud to send out registration emails. You could use the "Is Customer Portal" checkbox to trigger a journey into Marketing Cloud. If enabled, change the content of the email easily by navigating to the template set in Welcome New Member setting within Experience Cloud.
- **Sign Up Button Label:** This is the text shown in the "Sign Up" button when the page loads and before the button is clicked.
- **Sign Up Button Waiting Message:** This is the text shown briefly on the Sign Up button once the form is submitted and before the page redirects to the portal home page (if the registration was successful).
- **Password Match Error Message:** When registering, if the Password and Confirm Password fields do not match then this message is displayed.
- **Username Taken Error Message:** If the username chosen when registering is already taken, then display this message.
- **Record Not Found Error Message:** If a record is not found based on the submitted details and the configured custom query then this message is displayed.
- **Multiple Records Found Error Message:** If a record is not found based on the submitted details and the configured custom query then this message is displayed.
- **Login After Registration Error Message:** If there is a problem logging into the portal after registration, then this message is displayed.
- **User Creation Error Message:** If there is a problem creating an external user, then this message is displayed.

- **Already Registered Error Message:** If the matched record from the Custom Query is already registered for the portal, then this message is displayed.
- **Redirect URL After Registration:** After registration, the user is redirected to this Portal page. The default is “/” which takes the user to the configured Home Page. However, this could be another page such as the default Case List View using “case/Case/Default”.

Basic Setup - Self Registration

When the component is first added to the canvas, the Custom Query parameter is blank and must be configured as a minimum to use the component. You can write your own SOQL query to identify an existing Contact, Person Account or Case during registration to link the registration to, if a record is not found then you could later decide to create a record. Updates are not supported to an existing record.

The query can be dynamic and supports values submitted on the registration form. For example, you can use the submitted email address of the user registering to find a Contact, Person Account or Case in Salesforce. To do this, include one of the following bind variables in your query:

- :FirstName
- :LastName
- :Username
- :Email

The query must return exactly 1 record to ensure a unique match. If the query finds more than 1 result, then the component will error during registration with the parameter value set in “Multiple Records Found Error Message” and the site visitor will be unable to login to your Experience Cloud portal. Due to this, use LIMIT 1 within your chosen query as per the examples. If more than one record is found, only the first one found is used, ignoring any others.

Important

Do not use spaces between field names for the SELECT part of the SOQL statement otherwise the component will not validate the query correctly.

AccountId is a required field when querying the Contact object. If you omit the field from the Custom Query parameter, the Custom Self Registration component will error.

Example 1 (Contacts):

SELECT Id,AccountId,Email, FROM Contact WHERE Email = :Email LIMIT 1

This query attempts to find a Contact where the email address submitted matches a Contact in the Salesforce instance. It returns 1 result.

Example 2 (Person Accounts):

*SELECT Id,PersonEmail FROM Account WHERE PersonEmail = :Email AND
IsPersonAccount = true LIMIT 1*

This query attempts to find a Person Account where the email address submitted matches an Account and the record is a Person Account type.

If your Org's data quality isn't great, you may also wish to enable logging so you can log each registration and monitor for problems. This will ensure that errors are caught and you have visibility of the failures in the Self Registration Logs custom object. See "Enabling Custom Logging" section for more information.

If you have configured Custom Fields to show on your form using the Custom Self Registration Configuration metadata type (see below), then these fields can also be passed to your SOQL query using the API Name of the field. For example, if you create a Custom Field called "Reference Number", the API Name is set to "Reference_Number__c" by default by Salesforce on field creation (NOTE: This could have been manually overridden). Simply use ":Reference_Number__c" as part of your query as shown:

Example 3 (Contacts with Custom Field):

```
SELECT Id,Email,Reference_Number__c FROM Contact WHERE Reference_Number__c =  
:Reference_Number__c LIMIT 1
```

The default behaviour of the component at this point is to display a message to the user as configured in the **Record Not Found Error Message** parameter if your query returns no results, or if there is more than 1 result, then the **Multiple Records Found Error Message** parameter is shown instead.

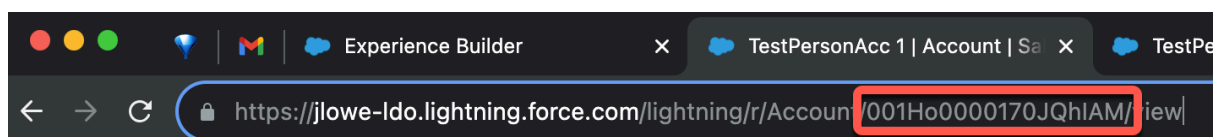
The Custom Query can be run in User, or System Mode which affects the results that are found. Change the Access Level for SOQL Query setting accordingly. The registration form runs as a Guest User, and thus so does your Custom Query so you **MUST** carefully consider which option you wish to use based on the security measures enforced by your org.

Create Record on Registration (if not found)

This option does exactly what it says on the tin! If your Custom Query does not find a record and this mode is enabled, then a new record is created. This setting works in tandem with the Object Type to Create setting where you can choose to create a Contact linked to a pre-existing Account, or a Person Account (if your org is configured to support Person Accounts).

Turn this setting on by ticking the checkbox, then choose the appropriate object type to create.

If creating a Contact, ensure you have an Account to attach any new registrations to which will enforce any security settings configured in your org. Without setting an Account Id, the Contact remains private and may not be visible to your users - see [here](#). Navigate to an Account, copy the Id from the browser URL bar (starting 001) into the component configuration panel.



If creating a Person Account, select the appropriate Person Account record type from the Picklist. You do not need to set the Account Id parameter for this object type.

Send Email Confirmation

By default, the component sends an email from Salesforce upon successful registration using the Welcome Email template configured in Workspaces.

Switch this feature off if you plan to use some other tool for successful registration. For example, you may wish to use Marketing Cloud and add new registered users to a specific journey rather than relying on standard templates.

Configuring Form Fields on the Form

By default, the form will show the following standard fields once installed:

- First Name
- Last Name
- Username
- Email
- Password
- Confirm Password

These are controlled from Setup > Custom Metadata Types > Custom Registration Configuration. From here, you amend each record to change the field behaviour, for example show/hide field, make a field required or not, change length etc.

IMPORTANT NOTE

Whilst you can hide all fields, it is recommended that you do not do this otherwise the registration will fail. The provided fields are the minimum required to create an External User. It is however supported to hide Username from the form as long as you capture the Email address. The component will populate the Username and Nickname fields with the submitted email if not present on the form.

It is possible to capture Custom Fields during registration and store them on the Contact/Account record created as part of the registration. Fields can also be used in the Custom Query as described earlier.

IMPORTANT NOTE

Data captured during registration is stored on the Contact/Person Account when creating a new Contact/Account only, not if an existing record is matched.

Start by creating a new Custom Field on the Contact or Account object from Setup > Object Manager.

Next, navigate to Setup > Custom Metadata Types > Manage Record beside the Custom Registration Configuration option. Click New to begin configuring a new field.

Fill out the following options:

- **Label & Custom Registration Configuration Name:** These fields are not used by Experience Cloud. Set them to something suitable based on any customisation standards defined by your org.
- **Active:** Ensure you set the new configuration to Active to display it on the registration form.
- **Portal API Name:** The easiest way to discover the API Name of your Experience Cloud portal is to use a simple SOQL query which can be run from the Developer Console of your Salesforce org or using an external tool such as Workbench. Run the following query: *SELECT Id,Name,UrlPathPrefix FROM Site* -
 - *NOTE: The query may return 2 results, use the value with "1" appended to the end.*
 - *NOTE: You can also use the value "ALL" if you have one site configured on your Org, or you wish to share this configuration deliberately across multiple sites using this component.*
- **Field Label:** Set this to the label that will be displayed to the user on the registration form.
- **Field API Name:** Set this to the API Name of the field that you created on the User object.
- **Field Type:** The default is 'text', however other data types are supported. Pick the most appropriate from the list for the field you wish to display to prevent submission errors.
- **Display Order:** Set this to a number. The component sorts fields by this value. The default fields are set with values 1-6. For custom fields enter a number >6 to display the field after the standard ones, or amend all records to create a custom sort order.
- **Field Help Text:** This value is used to guide the user during registration. It shows in a small help bubble beside the Field Label on the form.
- **Field Placeholder:** This value displays on the form field before a value is entered to provide guidance to the user on what to enter.
- **Field Class:** At the moment, this option is only used within the Password/Confirm Password fields to identify them for Custom Validation that is applied. This can be used as required when extending the component for other use cases.
- **Field Parent Class:** Style the field's parent container with a CSS class to help with icon positioning (See the section "Configuring CSS for Icons" for more information) or other custom use cases
- **Field Show Icon:** Show/hide the defined icon within the field.
- **Field Icon Class:** Add the icon below to the left or right of the field using CSS classes. See the section "Configuring CSS for Icons" for more information.
- **Field Icon Name:** Add an icon within the field using SLDS icons. See [here](#) for more information.
- **Field Label Variant:** Show the field label above the field (standard & default), to the left (inline) or not at all (hidden).
- **Field Date Style:** Allows a date field to be configured with different date formats - supported as short (dd/mm/yyyy), medium (Jan 7, 2020), long (January 7, 2020).
- **Field Show Password Visibility:** Add a Show/hide feature on password fields to reveal/mask the value typed.

Customisable Error Messages:

- **Field Validation Regex:** Choose the required format of data entered into the field during registration.
- **Field REGEX Message:** This is the error message displayed if the Field Validation Regex is not met or the field is set to required and has not been populated.
- **Field Required:** Choose whether the field must be populated to submit the registration form.
- **Field Required Message:** This message is displayed when the field is required, and the user navigates away from the field when it is empty.
- **Field Min Characters:** Specify the minimum length of a field on the registration form
- **Field Message Too Short:** Display a message when the field length is below the minimum characters specified above.
- **Field Max Characters:** Set the max length of the field on the registration form. It must not exceed the field length of the Custom Field otherwise failures during registration will occur.
- **Field Message Too Long:** Display a message when the field length is above the maximum characters specified above. NOTE: In some browsers, the maximum characters behaviour blocks a user from entering more than the specified characters so this message may not fire. Use Field Validation REGEX & Field REGEX Message instead to e.g. “.{0,3}” to fire a message when more than 3 characters are entered.
- **Minimum Value:** The lowest value or earliest date/time/datetime that this field will accept. This can be set to a number, date, datetime or time value and is not supported with other Field Types.
- **Field Under Minimum Range Message:** Set the message that will be displayed if the field is below the Minimum Value.
- **Maximum Value:** The highest value or latest date/time/datetime that this field will accept. This can be set to a number, date, datetime or time value and is not supported with other Field Types.
- **Field Over Max Range Message:** Set the message that will be displayed if the field is above the Maximum Value.

Custom Registration Configuration

Custom Registration Configuration Edit

SaveSave & NewCancel

Information

LabelFirst NameProtected Component☒ 1
Custom Registration Configuration NameFirst_NameNamespace Prefix

Configuration

Active☒
Portal API NameALL
Field LabelFirst Name
Field Label Variant--None--
Field API NameFirstName
Field Typetext

Display Order1
Field Help Text
Field Placeholder
Field Show Password Visibility
Field Date Style--None--

Icon Configuration

Field Show IconField Icon Name

CSS Overrides

Field Parent ClassField Class
Field Icon Class

Field Message Configuration

Field Validation REGEX
Field Required☒
Field Min Characters
Field Max Characters40
Minimum Value
Maximum Value

Field REGEX Message
Field Required MessagePlease enter a First Na
Field Message Too Short
Field Message Too LongMaximum of 40 charact
Field Under Minimum Range Message
Field Over Max Range Message

SaveSave & NewCancel

Once you are happy with the configuration, save the Custom Metadata, then go back to Experience Cloud and refresh the page to show the custom fields. You do not need to republish the site for these changes to take effect. Add as many Custom Fields as required, but do consider the user experience before capturing too many fields!

Pre-populate Form Fields

The component supports the ability to pre-populate defined fields on the form from values in the URL Parameters. When a user navigates to your Self Registration page, add the API Name of a given field and its value at the end of the page URL in the format:

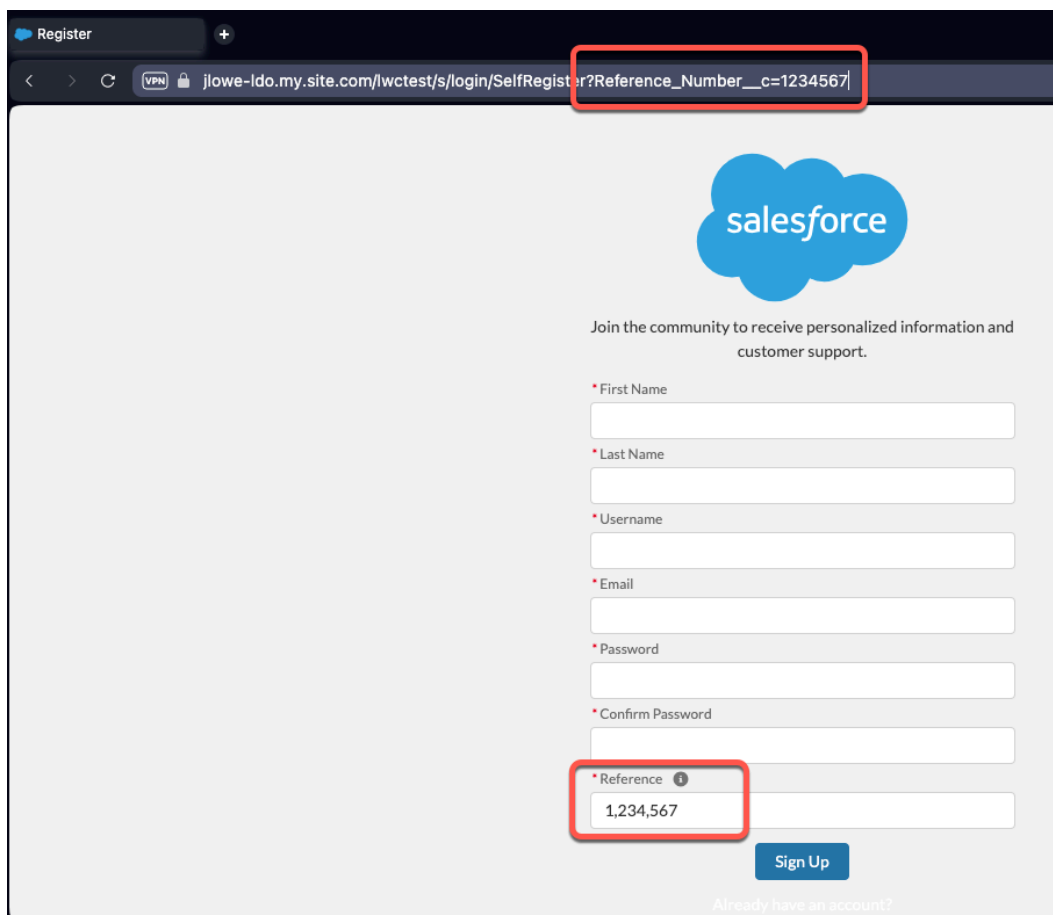
`?Reference_Number__c=1234567`

If the API Name provided exists on the form when the page loads, the value for that parameter is pre-populated on the form. Multiple fields can be populated by separating the parameters with the "&" symbol e.g:

`?Reference_Number__c=1234567&FirstName=Test`

NOTE

There is no validation of the data set at the point of the page loading. The form is only validated when a user navigates away from a field, or the form is submitted - at which point, the component will fire any configured validation messages.



The screenshot shows a web browser window with the address bar displaying the URL: `jlowe-ldo.my.site.com/lwctest/s/login/SelfRegister?Reference_Number__c=1234567`. The page features the Salesforce logo and a registration form. The form includes fields for First Name, Last Name, Username, Email, Password, and Confirm Password. A red box highlights the 'Reference' field, which is pre-populated with the value '1,234,567'. Another red box highlights the URL in the address bar. A 'Sign Up' button is located at the bottom right of the form, and a link 'Already have an account?' is visible below it.

Basic Setup - Login

The login component configuration is set up solely in the Experience Cloud Builder tool. Within the property panel for Custom Login, the following parameters are available:

- **Username Field Label:** The default is Username. But this can be overridden.
- **Username Label Variant:** The default is “standard” which shows the label above the field. Can be hidden or shown “inline” (to the left of the field) if required.
- **Username Help Text:** Shows in an “I” bubble beside the Field Label if populated. Only supported currently when the field label is set to “standard”
- **Username Placeholder:** Shows a temporary value within the field itself and is hidden when a user types in the field
- **Username Field Icon:** Shows an icon within the field, either to the left or right depending on the CSS Class applied in Parent Container CSS / Icon CSS Class parameters. Currently set to “utility:email”. Other icons available [here](#).
- **Username Field Icon CSS Class:** Currently set to “input-icon-label-shown-right”
 - Other classes that can be used are “input-icon-label-hidden-X” or “input-icon-label-inline-X” where X is left or right to place the icon based on the field label variant configuration.
- **Username Parent Container CSS Class:** Currently set to “slds-input-has-icon_right”
 - Other classes that can be used are “slds-input-has-icon_left” to place the icon on the left. When using the Field Label Variant of “label-hidden”, you can also specify “top” as an additional class to add spacing between the fields e.g. “slds-input-has-icon_left top”.
- **Username Field Validation Message:** The field is required to login. This is the message that is shown to the user if the field is left blank.
- **Username Field REGEX Pattern Match:** A REGEX pattern is used to enforce an email address format for the username field. This is defaulted to the REGEX pattern: `^[a-zA-Z0-9._+\-]+@[a-zA-Z0-9.\-]+\.[a-zA-Z]{2,4}$` which enforces the format of a common email address e.g. first.last+test@mail.com
- **Username Field REGEX Pattern Match Message:** This message is shown to the user when the entered username does not meet the requirements. By default, this is configured to Please enter an email address in the correct format e.g. "example@mail.com"

Most of the above are also available for the Password field too. Also available for configuration are:

- **Password Field: Show Icon Name:** The default is “utility:preview”. Uses SLDS Icons, see here for other supported icons. The icon that is displayed when the password value is hidden. Clicking the icon shows the password.
- **Password Field: Hide Icon Name:** The default is “utility:hide”. Uses SLDS Icons, see here for other supported icons. The icon that is displayed when the password value is shown. Clicking the icon hides the password.
- **Enable Show Password function:** This shows or hides the preview icon in the field.
- **Login Button Label:** Set to Login by default, but can be changed.
- **Login Waiting Button Label:** Set to “Logging in...Please Wait” by default. Can be overridden as required.
- **Redirect URL after Login:** The URL that the user is redirected to after a successful login. The default is the home page.

Lastly, error messages are configurable with the following parameters:

- **Blocked User Error Message:** If an admin uses the “Freeze” button on a user and they try to login, this message is displayed.
- **Locked User Error Message:** If the user tries to login with incorrect credentials more than the profile’s specified password policy then they are temporarily locked out. When this is the case, this error message is displayed.
- **Incorrect Username/Password Error:** If the user tries to login with incorrect credentials, or the account details they are used are not recognised then this error message is displayed.

Configuring CSS for Icons

The Self Registration and Login components allow the configuration of icons that appear within a configured field (Self Registration) or within the provided Username/Password fields (Login). The icon can be positioned using one of the provided CSS classes documented below.

Self Registration classes

The following table documents the Custom Metadata configuration needed to display an icon in the specified position. Also set Show Field Icon to TRUE, and set an Icon Name to a valid SLDS utility icon e.g. "utility:email".

Label Variant	Desired Icon Position	Field Parent Class	Icon Class
Standard	Left	slds-input-has-icon_left	input-icon-label-shown-left
Standard	Right	slds-input-has-icon_right	input-icon-label-shown-right
Inline	Left	slds-input-has-icon_left	input-icon-label-inline-left
Inline	Right	slds-input-has-icon_right	input-icon-label-inline-right
Hidden	Left	slds-input-has-icon_left	input-icon-label-hidden-left
Hidden	Right	slds-input-has-icon_right	input-icon-label-hidden-right

A Field Parent Class is needed to add padding around the field's icon to stop typed text overlaying the icon.

In addition to the Icon Class, you can use the class "top" for the Hidden variants above to place padding between each of the fields when no label is displayed.

Login classes

Similar to the Self Registration component, the Username and Password fields can also have icons but as the form is solely configured in the Experience Cloud builder, these settings are applied in the property panel, not Custom Metadata Types.

Username / Password Label Variant	Desired Icon Position	Username / Password Parent Container CSS Class	Username / Password Field Icon CSS Class
Standard	Left	slds-input-has-icon_left	input-icon-label-shown-left
Standard	Right	slds-input-has-icon_right	input-icon-label-shown-right
Inline	Left	slds-input-has-icon_left	input-icon-label-inline-left
Inline	Right	slds-input-has-icon_right	input-icon-label-inline-right
Hidden	Left	slds-input-has-icon_left	input-icon-label-hidden-left
Hidden	Right	slds-input-has-icon_right	input-icon-label-hidden-right

For the Password Field, there can be two possible icons which show or hide the password value typed. There are parameters for each where the above classes are applied to:

- **Password Field: Show Icon Name**
- **Password Field: Hide Icon Name**

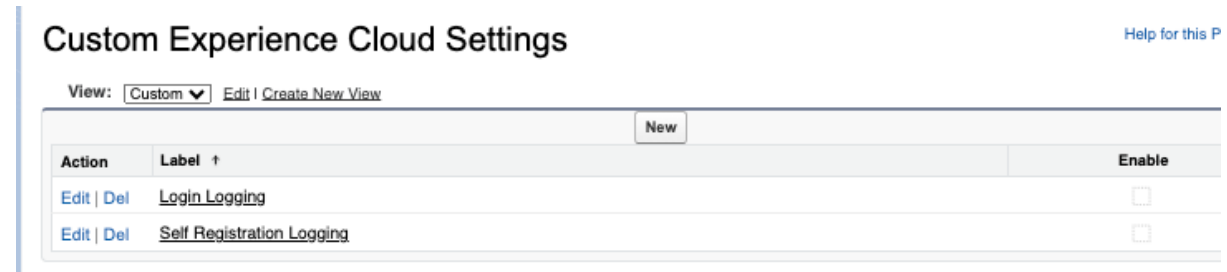
As per the Self Registration component, the **Parent Container CSS Class** is needed to add padding around the field's icon to stop typed text overlaying the icon.

In addition to the Icon Class, you can use the class "top" for the Hidden variants above to place padding between each of the fields when no label is displayed.

Further styling can be configured via the Experience Builder CSS Override feature. However, this document does not cover that in detail. Refer to the Salesforce help for more information.

Create Log Entry for Self Registrations & Logins

By default, this feature is off for both components. Turn it on in Setup > Custom Experience Cloud Setting. Enable Logging by setting the “Activate Setting” value to TRUE.



Custom Experience Cloud Settings [Help for this P](#)

View: Custom [Edit](#) [Create New View](#) New

Action	Label ↑	Enable
Edit Del	Login Logging	☐
Edit Del	Self Registration Logging	☐

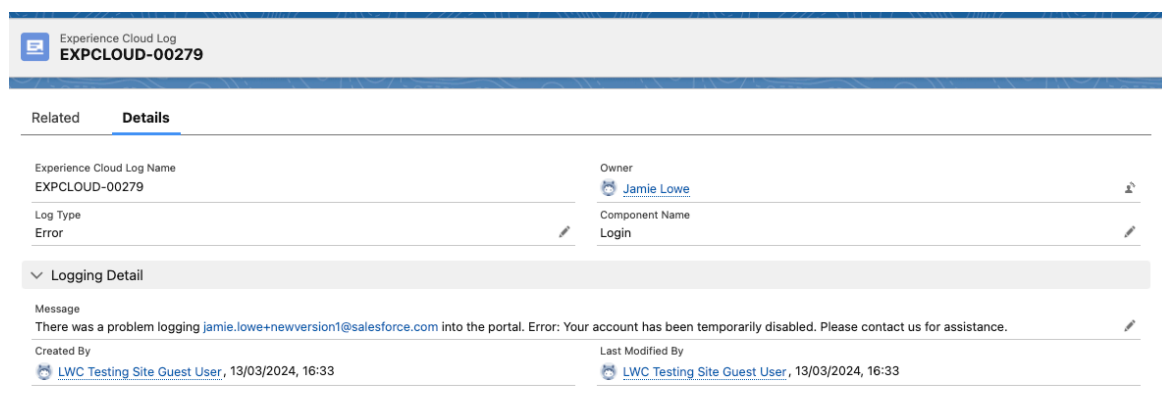
This setting creates noisy logging for each registration / login and thus it is not recommended to leave this on continuously. This setting is particularly useful when first configuring the components as it gives you a detailed trail of what happened without having to navigate complex Debug Logs.

A single log record is created at the end of processing the Self Registration form submission with details of:

- Validating Username - is the chosen name available or taken
- Validating Password - was it valid and met the SF org requirements
- Actual registration i.e. locating a record, creating a new record, creating an external user etc.

For Login, the record is more simple and logs only a success message, or in the instance of a failure the message that is configured in the Experience Cloud builder for the scenario encountered e.g. if the user is blocked the message will say “There was a problem logging {username} into the portal.” Followed by the message “Your account has been temporarily disabled. Please contact us for assistance.” assuming the default has not been changed.

There are two Log Types used - Error and Information. The Component Name is also stored. Use the Message field to understand the steps completed.



Experience Cloud Log
EXPCLOUD-00279

Related **Details**

Experience Cloud Log Name EXPCLOUD-00279	Owner Jamie Lowe
Log Type Error	Component Name Login

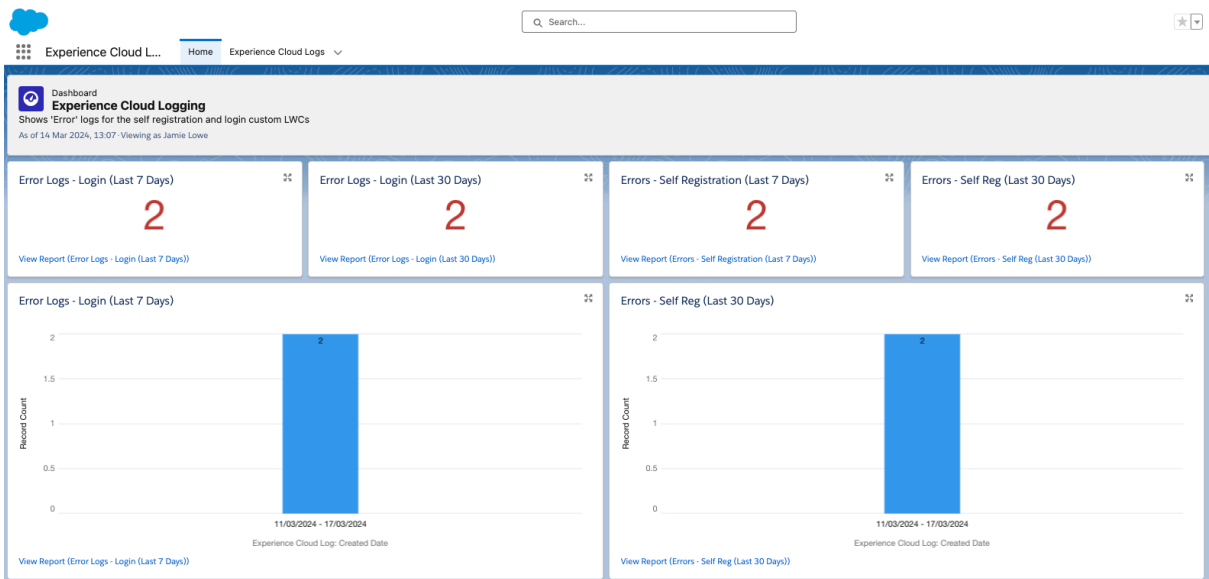
Logging Detail

Message
There was a problem logging [jamie.lowe+newversion1@salesforce.com](#) into the portal. Error: Your account has been temporarily disabled. Please contact us for assistance.

Created By
[LWC Testing Site Guest User](#), 13/03/2024, 16:33

Last Modified By
[LWC Testing Site Guest User](#), 13/03/2024, 16:33

Some basic Reports & Dashboards on the error data are installed with the package and accessible via the Experience Cloud Logging app or via the Reports/Dashboards tabs directly. These can be used to quickly identify problems that arise and proactively investigate them.



Experience Cloud Log records are created via a Platform Event (Experience Cloud Event) & Flow subscription and are accessible to System Admins by default. Configure another Permission Set as required to give other users access.

Expected Errors:

The following errors are shown in Experience Cloud builder and are expected behaviour.
The table below summarises

Error	Description / Solution
Only Contact or Account objects are supported with the Custom SOQL Query on this component.	If you try to query other object types over the ones supported e.g. Case, then you will see this message. Update the Custom Query parameter to query Contacts or Accounts.
Person Accounts are not enabled on this org so you cannot use Accounts in a Custom Query.	If your Custom Query uses the Account object and Person Accounts are not configured in your org. Either use Contacts, or configure Person Accounts before using this component.
Object Type to Create cannot be "N/A" when the Create Record function is set to TRUE.	You've enabled the "Create Record During Registration (if not found) feature but not configured the Create Record Type picklist. Change it from N/A to the required type - remember that Person Accounts must be enabled to use Person Accounts as a creatable record type with this component.
Please specify an Account Id parameter when creating a Contact.	When creating a new Contact, you must specify an Account Id to link that Contact to for sharing & visibility purposes. The contact can be manually updated later to a new Account if required.
Account Id parameter must be a Salesforce 15 or 18 character reference	The Id you've entered is not 15 or 18 characters and therefore not the expected length for a Salesforce Id so it is likely incorrect. Please check the value, or navigate to the Account and use the Id shown in the browser URL.
The Account Id parameter must start with 001 (Account Object Type).	The Id you've entered does not start with the characters 001 and therefore not the expected format for a Salesforce Account Id so it is likely incorrect. Please check the value, or navigate to the Account and use the Id shown in the browser URL.

Error	Description / Solution
Person Accounts are not enabled on this org.	Enable Person Accounts if they are a suitable solution for your Org but consider the implications carefully and what it means for your org. This feature cannot be turned off in Salesforce once enabled so proceed with care. See here .
Please select a Person Account Record Type from the list to create a Person Account during registration.	You've selected to create a Person Account if a record is not found during registration, but you have not selected the appropriate Record Type from the list. Configure this additional parameter to proceed.

During registration, most errors are hidden from the registering user where possible with the generic message to hide technical Salesforce issues which may confuse the user. The user should see "An unknown error has occurred, please contact us for further assistance." The best course of action is to turn on the Create Log Entry for Registrations feature and monitor for issues (see above for more information).

You may also see the error: You do not have access to the Apex class named 'SiteRegistrationController'. - to resolve this issue, simply assign the "Custom Self Registration - Guest Access" Permission Set to the Guest User.