← Back to Applications

vite-react-heroui-auth0-template

Single Page Applications Client ID `SNUKCOsPFNi9sLzxkSHR0CZy5YotWb8m`

Quickstart | Settings | Addons | Connections | Organizations

Basic Information

Name *
vite-react-heroui-auth0-template

Domain
sctg-development.eu.auth0.com

Client ID
SN[redacted]Wb8m

Client Secret
••

The Client Secret is not base64 encoded.

Description
Add a description in less than 140 characters

A free text description of the application. Max character count is 140.

Application Properties

Application Logo

⌘ + Return Cancel Save Changes

https://sctg-development.github.io/flow-dilution/logo.svg

The URL of the logo to display for the application, if none is set the default badge for this type of application will be shown. Recommended size is 150×150 pixels.

Application Type
Single Page Application

The type of application will determine which settings you can configure from the dashboard.

Application URIs

Application Login URI
https://sctg-development.github.io/vite-react-heroui-auth0-template

In some scenarios, Auth0 will need to redirect to your application's login page. This URI needs to point to a route in your application that should redirect to your tenant's `/authorize` endpoint. Learn more

Allowed Callback URLs
https://sctg-development.github.io/vite-react-heroui-auth0-template

After the user authenticates we will only call back to any of these URLs. You can specify multiple valid URLs by comma-separating them (typically to handle different environments like QA or testing). Make sure to specify the protocol (`https://`) otherwise the callback may fail in some cases. With the exception of custom URI schemes for native clients, all callbacks should use protocol `https://`.You can use Organization URL parameters in these URLs.

Allowed Logout URLs
https://sctg-development.github.io/vite-react-heroui-auth0-template

Comma-separated list of allowed logout URLs for redirecting users post-logout. You can use wildcards at the subdomain level (`*.google.com`). Query strings and hash information are not taken into account when validating these URLs. Learn more about logout

Allowed Web Origins
https://sctg-development.github.io

Comma-separated list of allowed origins for use with Cross-Origin Authentication, Device Flow, and web message response mode, in the form of `<scheme>://<host>[:<port>]`, such as `https://login.mydomain.com` or `http://localhost:3000`. You can use wildcards at the subdomain level (e.g.: `https://*.contoso.com`). Query strings and hash information are not taken into account when validating these URLs.

OpenID Connect Back-Channel Logout

ENTERPRISE

Learn more about OpenID Connect Back-Channel Logout

Back-Channel Logout URI
https://myapp.org/backchannel-logout

Logout URI that will receive a `logout_token` when selected Back-Channel Logout initiators occur. URIs with a querystring will be re-encoded properly.

Back-Channel Logout Initiators

- ○ Selected initiators only
- ○ All supported initiators

Send a `logout_token` on selected Back-Channel Logout initiators only.

- ☐ IdP-Logout REQUIRED
- ☐ RP-Logout REQUIRED
- ☐ Password Changed
- ☐ Session Expired
- ☐ Account Deleted
- ☐ Email Changed
- ☐ Session Revoked
- ☐ Account Deactivated

Cross-Origin Authentication

Allow Cross-Origin Authentication
○

When allowed, cross-origin authentication allows applications to make authentication requests when the Lock widget or custom HTML is used.

Allowed Origins (CORS)

List additional origins allowed to make cross-origin resource sharing (CORS) requests. Allowed callback URLs are already included in this list.

- URLs can be comma-separated or added line-by-line
- Use wildcards (*) at the subdomain level (e.g. https://*.contoso.com)
- Query strings and hash information are ignored
- Organization URL placeholders are supported

Cross-Origin Verification Fallback URL

Fallback URL when third-party cookies are not enabled in the browser. URL must use https and be in the same domain as the embedded login widget.

ID Token Expiration

Maximum ID Token Lifetime *
36000 seconds

Time until an `id_token` expires regardless of activity.

Refresh Token Expiration

Set Idle Refresh Token Lifetime
●

Require `refresh tokens` to expire after a set period of inactivity.
Learn more about refresh token expiration

Idle Refresh Token Lifetime *
1296000 seconds

Set Maximum Refresh Token Lifetime
●

Require `refresh tokens` to expire after a set period regardless of activity.
Required for refresh token rotation. Learn more about refresh token expiration

Maximum Refresh Token Lifetime *
2592000 seconds

Refresh Token Rotation

Allow Refresh Token Rotation
●

When allowed, `refresh tokens` will automatically be invalidated after use and exchanged for new tokens to prevent replay attacks. Requires a maximum refresh token lifetime. Learn more about refresh token rotation

Rotation Overlap Period *
0 seconds

Period of time the most recently-used `refresh token` can be reused without triggering breach detection.

Token Sender-Constraining

Require Token Sender-Constraining
○

When required, `access tokens` must be constrained to this application to prevent unauthorized use of leaked or stolen tokens.
Learn more about token sender-constraining

Authorization Requests

Require Pushed Authorization Requests (PAR)
○

When required, authorization request parameters must be sent using back-channel communication for confidentiality and integrity protection. Requires tenant to allow PAR. Learn more about Pushed Authorizarion Requests (PAR)

Require JWT-Secured Authorization Requests (JAR)
○

When required, authorization request parameters must be wrapped in a signed JSON Web Token (JWT) for cryptographically confirmed non-repudiation.
Learn more about JWT-Secured Authorizarion Requests (JAR)

Advanced Settings ⌄

Danger Zone

Delete this application
All your apps using this client will stop working.
Delete

Rotate secret
All authorized apps will need to be updated with the new client secret.
Rotate