



Exploring The Demo

[Canton Network Quickstart Guide | 2025](#)

Version: 1.0.3-2025-03-21

Contents

[Exploring the Demo](#)
[Prerequisites](#)
[Walkthrough](#)
 [Canton Console](#)
 [Daml Shell](#)
 [Connect to DevNet](#)
 [Important: Migration ID for DevNet Connections](#)
 [Configuring Non-Default DevNet Sponsors](#)
 [SV UIs](#)
 [Canton Coin Scan](#)
 [Observability Dashboard](#)
[Keycloak in the CN-QS](#)
 [Realm Structure](#)
 [Keycloak Configuration](#)
 [Customizing Keycloak for Business Needs](#)
 [Accessing the Admin Console](#)
 [Customization Scenarios](#)
 [Add a New User](#)
 [Modify Client Settings](#)
 [Add a New Client](#)
 [Update Environment Variables](#)
 [Troubleshooting](#)
[Next Steps](#)

Exploring the Demo

The CN-QS and its guides are a work-in-progress (WIP). As a result, the CN-QS guides may not accurately reflect the state of the application. If you find errors or other inconsistencies, please contact your representative at Digital Asset.

This section works through a complete business operation within the CN-QS.

Prerequisites

You should have successfully installed the CN-QS before beginning this demonstration.

Access to the [CN-Quickstart Github repository](#) and [CN Docker repository](#) is needed to successfully pull the Digital Asset artifacts from JFrog Artifactory.

Access to the *Daml-VPN* connection or [a SV Node](#) that is whitelisted on the CN is required to connect to DevNet. The GSF publishes a [list of SV nodes](#) who have the ability to sponsor a Validator node. To access DevNet, contact your sponsoring SV agent for VPN connection information.

If you need access, email support@digitalasset.com.

The CN-QS is a Dockerized application and requires [Docker Desktop](#). Running CN-QS on LocalNet is resource intensive. It is recommended to allocate 25 GB of memory and 3 GB of Swap memory to properly run the required Docker containers. If you witness unhealthy containers, please consider allocating additional resources, if possible.

DevNet is not as intensive because the SVs and other LocalNet containers are hosted outside of your local machine.

Walkthrough

After the QS is installed and running, confirm that you are in the quickstart subdirectory of the CN-QS.

Open an incognito browser.

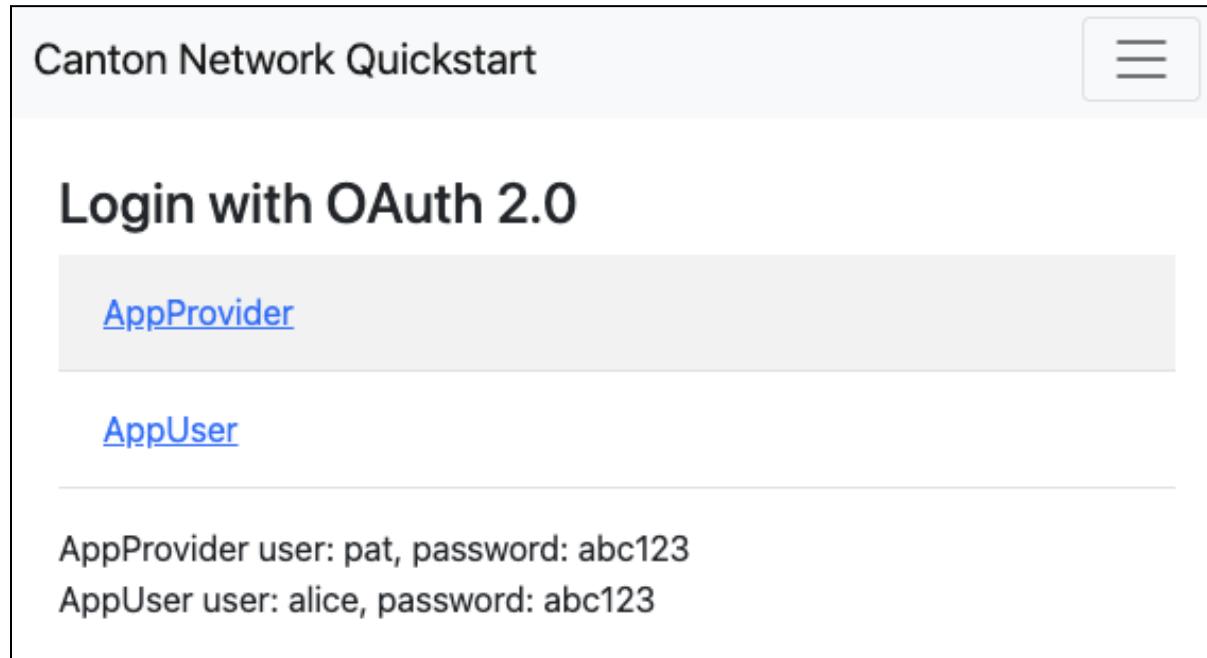
Navigate to:

localhost:3000/login

 Currently, localhost URLs do not work in Safari. We are working on a solution and apologize for the inconvenience.

Alternatively, in the terminal, from quickstart/ run:

```
make open-app-ui
```



Canton Network Quickstart

Login with OAuth 2.0

[AppProvider](#)

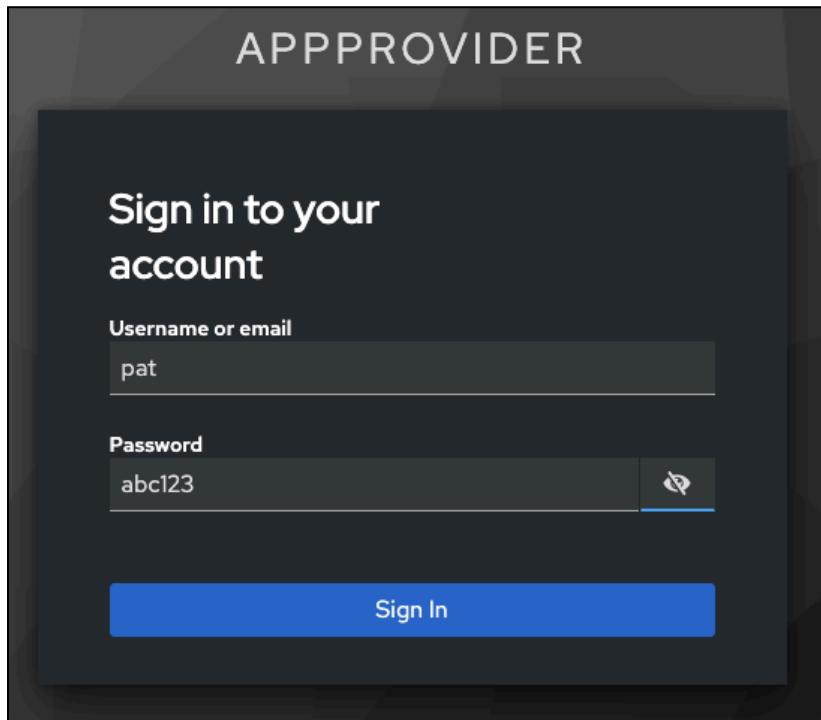
[AppUser](#)

AppProvider user: pat, password: abc123
AppUser user: alice, password: abc123

Make note that the AppProvider's username is "pat" and the password is "abc123" (all lowercase).

Login as the AppProvider.

Fill in the login credentials: username: pat, password: abc123



Select “AppInstalls” in the menu.

A screenshot of a web-based interface titled "Canton Network Quickstart". The top navigation bar includes links for "Home", "AppInstalls", "Licenses", and "Tenants". On the right side of the header, there is a link "Pat the provider". The main content area is titled "App Installs". Below the title, a note says "Note: Run make create-app-install-request to submit an AppInstallRequest". A table is displayed with columns: Contract ID, Status, DSO, Provider, User, Meta, # Licenses, and Actions. The table currently has no data rows.

Open a terminal.

From /quickstart/ run:

```
make create-app-install-request
```

This command creates an App Installation Request on behalf of the Participant.

```
((base) quickstart ~ % make create-app-install-request
docker compose -f docker/app-user-shell/compose.yaml --env-file .env run --rm create-app-install-request || true
get_token ledger-api-user AppProvider
get_user_party AppProvider participant-app-provider
http://participant-app-provider:7575/v2/users/AppProvider
get_token ledger-api-user Org1
get_user_party Org1 participant-app-user
http://participant-app-user:7575/v2/users/Org1
get_token administrator Org1
http://validator-app-user:5003/api/validator/v0/scan-proxy/dso-party-id
http://participant-app-user:7575/v2/commands/submit-and-wait
--data-raw {
    "commands" : [
        { "CreateCommand" : {
            "template_id": "#quickstart-licensing:Licensing.AppInstall:AppInstallRequest",
            "create_arguments": {
                "dso": "DSO::12209a3af80af3fa93853be8a8b9f5887055edc3b0a94f4b198f486d08d197784c09",
                "provider": "AppProvider::1220b3de80de523473aa2ca56745ef1fb29a2203dfd13029e0448336bbcfc382aa86",
                "user": "Org1::1220e69bc6115cd8ed360aa6caafce122c2a24561262c2f6ce6a6070e170e9e8244d",
                "meta": {"values": []}
            }
        }
    ]
},
"workflow_id" : "create-app-install-request",
"application_id": "ledger-api-user",
"command_id": "create-app-install-request",
"deduplication_period": { "Empty": {} },
"act_as": ["Org1::1220e69bc6115cd8ed360aa6caafce122c2a24561262c2f6ce6a6070e170e9e8244d"],
"read_as": ["Org1::1220e69bc6115cd8ed360aa6caafce122c2a24561262c2f6ce6a6070e170e9e8244d"],
"submission_id": "create-app-install-request",
"disclosed_contracts": [],
"domain_id": "",
"package_id_selection_preference": []
}
{"update_id":"1220aebcd64fc960c2aeecc834d415967899994f21a69ef9cab5639b8521169a0ca67", "completion_offset":81}
```

If your machine is not powerful enough to host LocalNet or if the docker containers are not responsive then the response may show a failure with status code 404 or 000.

Increasing Docker memory limit to at least 25 GB should allow the LocalNet containers to operate properly.

```
((base) quickstart ~ % make create-app-install-request
docker compose -f docker/app-user-shell/compose.yaml --env-file .env run --rm create-app-install-request || true
[+] Building 0.0s (0/0)
[+] Building 0.0s (0/0)
get_token ledger-api-user AppProvider
get_user_party AppProvider participant-app-provider
http://participant-app-provider:7575/v2/users/AppProvider
get_token ledger-api-user Org1
get_user_party Org1 participant-app-user
http://participant-app-user:7575/v2/users/Org1
get_token administrator Org1
http://validator-app-user:5003/api/validator/v0/scan-proxy/dso-party-id
Request failed with HTTP status code 404
Response body: The requested resource could not be found.
```

Return to the browser.

The install request appears in the list.

Click “Accept”.

Canton Network Quickstart Home AppInstalls Licenses Tenants Pat the provider

App Installs

Note: Run `make create-app-install-request` to submit an AppInstallRequest

Contract ID	Status	DSO	Provider	User	Meta	# Licenses	Actions
00a2f3c34cc5e...	REQUEST	DSO::12206b3c...	app_provider_q...	app_user_quick...	{"data":{}}	0	<button>Accept</button> <button>Reject</button> <button>Cancel</button>

The AppInstallRequest is Accepted. The actions update to create or cancel the license.

Canton Network Quickstart Home AppInstalls Licenses Tenants Pat the provider

App Installs

Note: Run `make create-app-install-request` to submit an AppInstallRequest

Success

```
Success: Accepted AppInstallRequest
00a2f3c34cc5e6df660aeb6658cd7000541349f3e
5b2e3268330ff1307067aa9fbc1012204022da1
3d54f9c9321e2401364a3123671ebcaddc47deb6
2ee39f9c81e4f0c9f
```

Contract ID	Status	DSO	Provider	User	Meta	# Licenses	Actions
00502121cb22e...	INSTALL	DSO::12206b3c...	app_provider_q...	app_user_quick...	{"data":{}}	0	<button>Create License</button> <button>Cancel Install</button>

Click “Create License”.

The license is created and the “# Licenses” field is updated.

Canton Network Quickstart Home AppInstalls Licenses Tenants Pat the provider

App Installs

Note: Run `make create-app-install-request` to submit an AppInstallRequest

Success

```
Success: Created License:
00be4bac78d731505d04d5e4964c9a297d9e479
54942c54809bc24d3b65cd0342bca101220a263
af0d31f1043ccb903c58b0479db70935d107ae0a
0a6299e510c84c402909
```

Contract ID	Status	DSO	Provider	User	Meta	# Licenses	Actions
00fdf01728922...	INSTALL	DSO::12206b3c...	app_provider_q...	app_user_quick...	{"data":{}}	1	<button>Create License</button> <button>Cancel Install</button>

In the AppProvider, “Pat the provider’s,” account, navigate to the **Licenses** menu and select “Actions.”

Canton Network Quickstart Home AppInstalls Licenses Tenants Pat the provider

Licenses

License Contract ID	DSO	Provider	User	Expires At	License #	Renew Fee	Extension	Actions
00be4bac78d731505d04...	DSO::122...	app_provide...	app_user_q...	2025-03-13T21:20:...	1			<button>Actions</button>

An “Actions for License” modal opens with an option to renew or expire the license. Per the Daml contract, licenses are created in an expired state. To activate the license, it must be renewed.

Actions for License 00be4bac78d731505d04d5e4 X

Renew License

Extension: 30 days (P30D), **Payment Acceptance:** 7 days (P7D), **Fee:** 100 CC

Description:
e.g. "Renew for next month"

Issue Renewal Payment Request

Expire License

Description:
e.g. "License expired"

Expire

Close

To renew the license, enter a description then click the green “Issue Renewal Payment Request” button.

Actions for License 00c81bd63b6d139f58e45d5c X

Renew License

Extension: 30 days (P30D), **Payment Acceptance:** 7 days (P7D), **Fee:** 100 CC

Description:
Activate for one month

Issue Renewal Payment Request

Expire License

Description:
e.g. "License expired"

Expire

Close

The license renewal process is initiated and ultimately successful.

Canton Network Quickstart Home AppInstalls Licenses Tenants Pat the provider

Licenses

License Contract ID	DSO	Provider	User	Expires At	License #	Renew Fee	Extension	Actions
00be4bac78d731505d04...	DSO::122...	app_provide...	app_user_q...	2025-03-13T21:20:...	1	100	30 days	<button>Actions</button>

Success
Success: License Renewal initiated successfully

The license is now available for a 30-day extension for a flat fee of \$100 CC.

Canton Network Quickstart Home AppInstalls Licenses Tenants Pat the provider

Licenses

License Contract ID	DSO	Provider	User	Expires At	License #	Renew Fee	Extension	Actions
00be4bac78d731505d04...	DSO::122...	app_provide...	app_user_q...	2025-03-13T21:20:...	1	100	30 days	<button>Actions</button>

Pat the provider has done as much as they are able until Alice pays the renewal fee.

For the next step we recommend opening a separate browser in incognito mode. Each user, AppProvider, and Org1, should be logged into separate browsers for most consistent results. For example, if you logged into AppProvider using Chrome, you would use Firefox when logging into Org1.

Navigate to `http://localhost:3000/login` using a separate browser in incognito or private mode.

Canton Network Quickstart

Login with OAuth 2.0

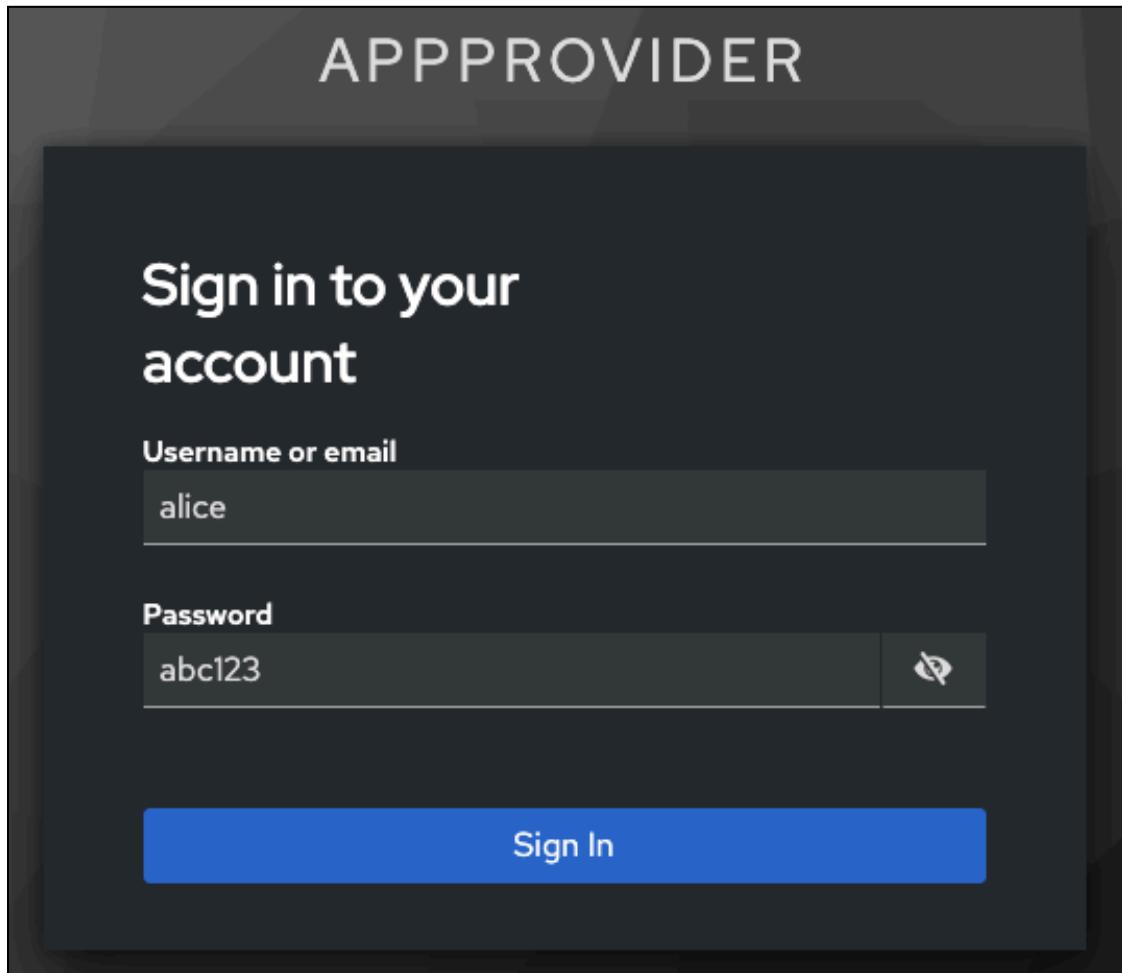
[AppProvider](#)

[AppUser](#)

AppProvider user: pat, password: abc123
AppUser user: alice, password: abc123

Login as AppUser alice.

Note that AppUser's username is "alice" and the password is "abc123".



Go to the **Licenses** View and click the "Pay renewal" button.

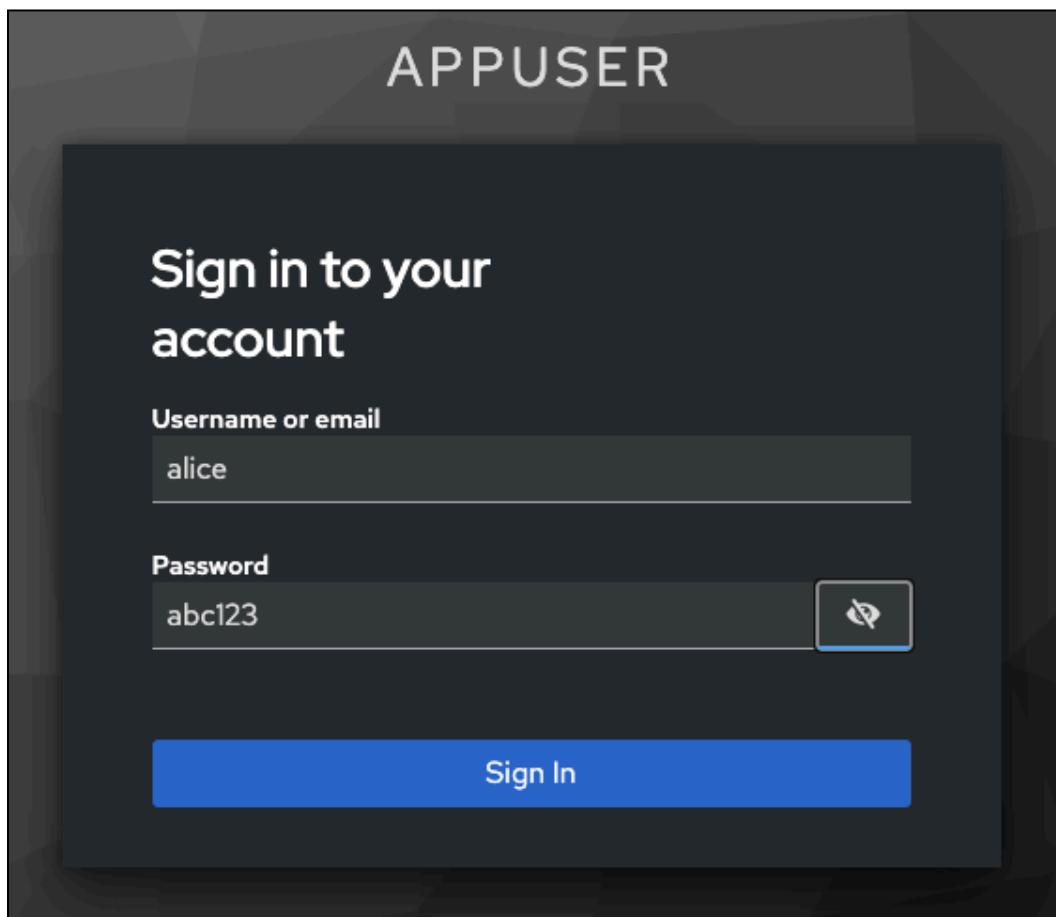
Canton Network Quickstart									Alice the user
Licenses									
License Contract ID	DSO	Provider	User	Expires At	License #	Renew Fee	Extension	Actions	
00be4bac78d731505d04...	DSO::122...	app_provide...	app_user_q...	2025-03-13T21:20:...	1	100	30 days	<button>Pay Renewal</button>	

Click on the Pay Renewal button. This navigates to the Canton Coin Wallet log in. Click "LOG IN WITH OAUTH2".

If you have any issues with log in, navigate directly to <http://wallet.localhost:2000/>.



This navigates to a keycloak login.
Enter the same username and password as before.



Signing in directs to the Canton Coin Wallet.

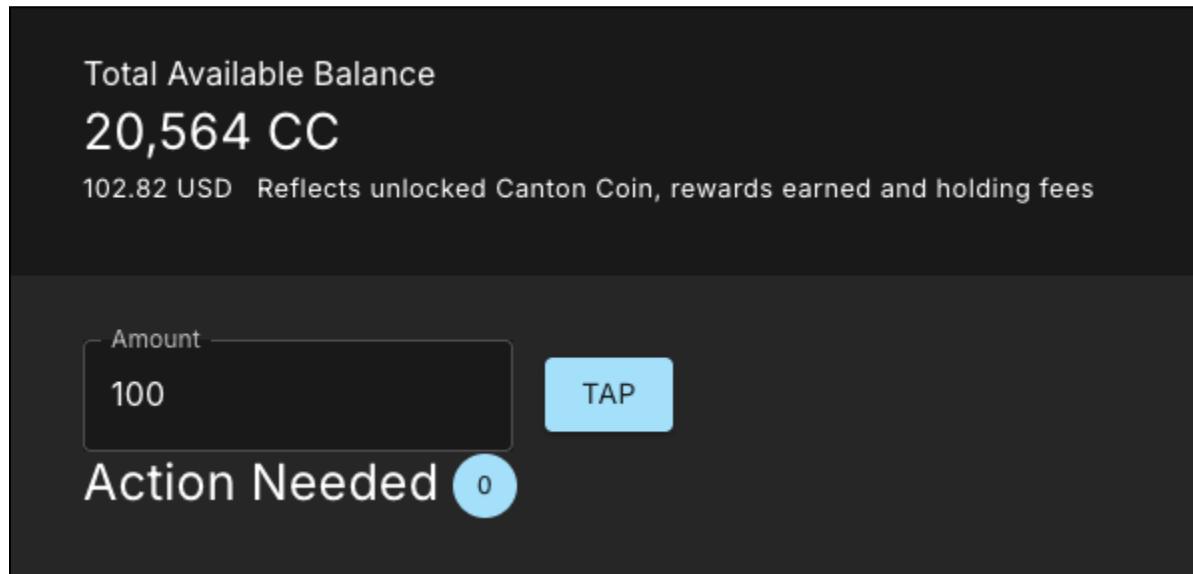
The screenshot shows the 'CANTON COIN WALLET' interface. At the top, there are navigation links: Transactions, Transfer, Subscriptions, FAQs, app_user_qui..., a lock icon, SELF-GANT FEATURED APP RIGHTS, PRE-APPROVE INCOMING DIRECT TRANSFERS OF CANTON COIN, and a Logout button. Below this, the 'Total Available Balance' is displayed as 564 CC, with a note: '2.82 USD Reflects unlocked Canton Coin, rewards earned and holding fees'. A form field for 'Amount' contains '0', with a 'TAP' button next to it. An 'Action Needed' button shows '0'. Below this, a message says 'No transfer offers available'. The 'Transaction History' section has columns: TYPE, DATE, SENDER OR RECEIVER, REWARDS CREATED, and BALANCE CHANGE. One entry is shown: Sent (Automation) on 2025-03-14 at 09:54:50, via app_user_quickstart-j..., Automation Validator Rewards: 570 CC, resulting in +564 CC and +2.82 USD @ 200 CC/USD. A 'Load More' button is at the bottom. Copyright notice: Copyright © Digital Asset 2025.

The wallet must be populated with CC in order to fulfill the transaction.

In CC Wallet, populate the wallet with \$100 USD, or the equivalent of 20,000 CC.

The screenshot shows the 'CANTON COIN WALLET' interface. At the top, there are navigation links: Transactions, Transfer, Subscriptions, FAQs, app_user_qui..., a lock icon, SELF-GANT FEATURED APP RIGHTS, PRE-APPROVE INCOMING DIRECT TRANSFERS OF CANTON COIN, and a Logout button. Below this, the 'Total Available Balance' is displayed as 564 CC, with a note: '2.82 USD Reflects unlocked Canton Coin, rewards earned and holding fees'. A form field for 'Amount' contains '100', with a 'TAP' button next to it. An 'Action Needed' button shows '0'. Below this, a message says 'No transfer offers available'. The 'Transaction History' section has columns: TYPE, DATE, SENDER OR RECEIVER, REWARDS CREATED, and BALANCE CHANGE. One entry is shown: Sent (Automation) on 2025-03-14 at 09:54:50, via app_user_quickstart-j..., Automation Validator Rewards: 570 CC, resulting in +564 CC and +2.82 USD @ 200 CC/USD. A 'Load More' button is at the bottom. Copyright notice: Copyright © Digital Asset 2025.

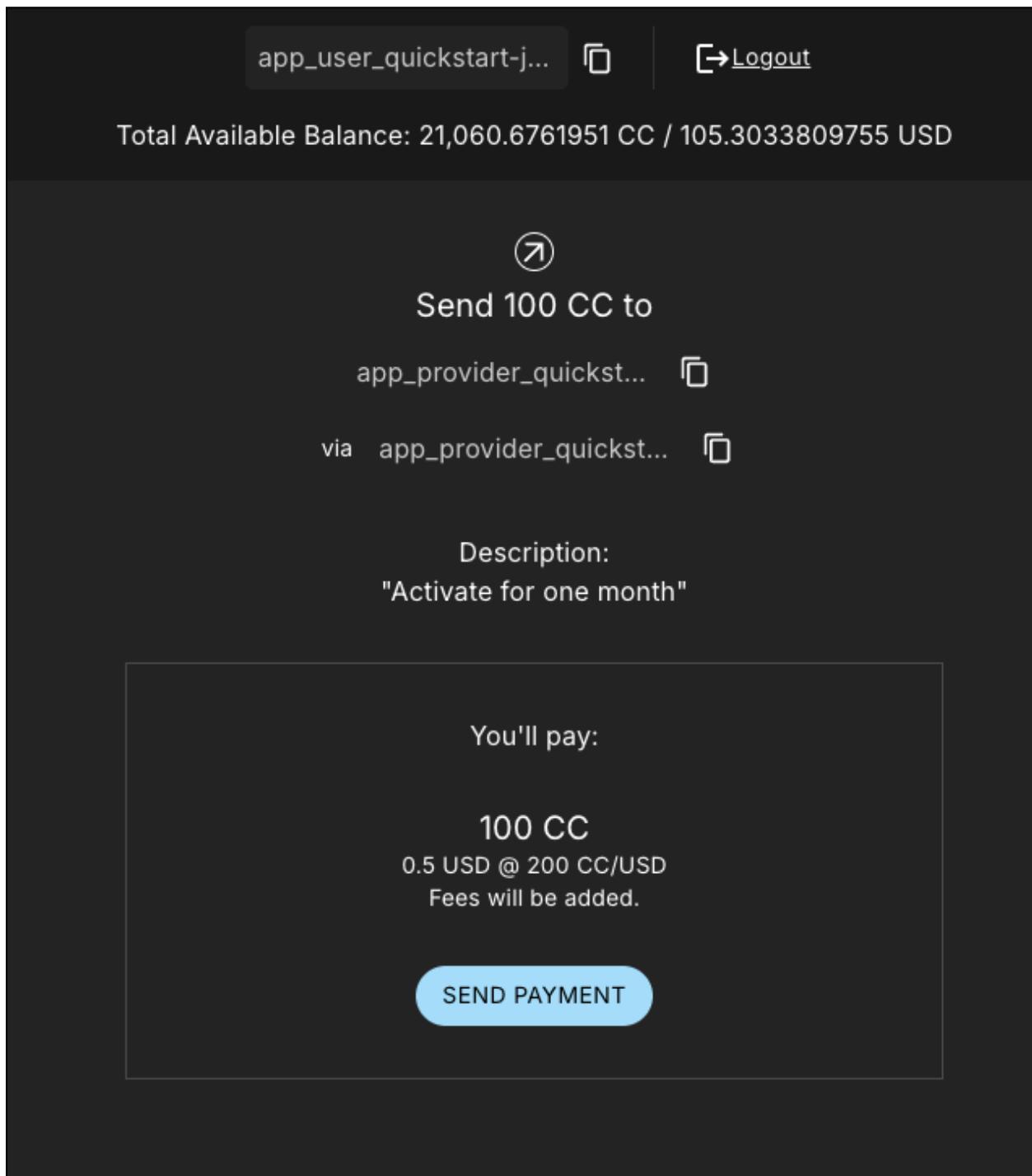
The wallet was prepopulated with 564 CC so it now contains 20,564 CC.



Return to the License Renewal Request as Org1. Click “Pay Renewal”.

Licenses								
License Contract ID	DSO	Provider	User	Expires At	License #	Renew Fee	Extension	Actions
00a01c1718dbe1a0f5e45...	DSO::122...	app_provide...	app_user_q...	2025-03-14T14:36:...	1	100	30 days	<button>Pay Renewal</button>

The CC Wallet balance is sufficient to send payment to the Provider.



Return to the AppProvider's License Renewal Requests View.
The AppProvider may now Complete the Renewal.

License Contract ID	DSO	Provider	User	Expires At	License #	Renew Fee	Extension	Actions
00a01c1718dbe1a0f5e45...	DSO::122...	app_provide...	app_user_q...	2025-03-14T14:36:...	1	100	30 days	<button>Complete Renewal</button>

Clicking "Complete Renewal" results in a Success.

License Contract ID	DSO	Provider	User	Expires At	License #	Renew Fee	Extension	Actions
00a01c1718dbe1a0f5e45...	DSO::122...	app_provide...	app_user_q...	2025-03-14T14:36:...	1			<button>Actions</button>

Alice's License view shows the activated license.

License Contract ID	DSO	Provider	User	Expires At	License #	Renew Fee	Extension	Actions
00f77a38c88b2ce54687...	DSO::122...	app_provide...	app_user_q...	2025-04-13T15:11:1...	1			<button>Actions</button>

Congratulations. You've successfully created and activated a license with a payment transfer!

Canon Console

The Canon Console connects to the running application ledger. The console allows a developer to bypass the UI to interact with the CN in a more direct manner. For example, in Canon Console you can connect to the Participant to see the location of the Participant and their domain.

The app provider and the app user each have their own console. To activate the app provider's Canon Console in a terminal from the `quickstart/` directory. Run:

```
make console-app-provider
```

Open the participant's Canon Console with

```
make console-app-user
```

After the console initiates, run the `participant` and `participant.domains` commands, respectively.

`participant`

Returns their location in the ledger.

```
[@ participant
  res0: com.digitalasset.canton.console.RemoteParticipantReference = Participant 'participant'
```

`participant.domains`

Shows the Participant's synchronizer.

```
[@ participant.domains
  res1: participant.domains.type = com.digitalasset.canton.console.commands.ParticipantAdministration$domains$@4f4c3934
```

`participant.health.ping(participant)`

Runs a health ping. The ping makes a round trip through the CN blockchain. Pinging yourself validates communication throughout the entire network.

```
[@ participant.health.ping(participant)
  res0: Duration = 4979 milliseconds
```

Daml Shell

The Daml Shell connects to the running PQS database of the application provider's Participant. In the Shell, the assets and their details are available in real time.

Run the shell from `quickstart/` in the terminal with:

```
make shell
```

Run the following commands to see the data:

```
active
```

Shows unique identifiers and the asset count

```
postgres-splice-app-provider:5432/scribe> active
```

Identifier	Type	Count
quickstart-licensing:Licensing.AppInstall:AppInstall	Template	1
quickstart-licensing:Licensing.License:License	Template	1
splice-amulet:Splice.Amulet:Amulet	Template	1
splice-amulet:Splice.Amulet:ValidatorRight	Template	1
splice-wallet:Splice.Wallet.Install:WalletAppInstall	Template	1

```
active quickstart-licensing:Licensing.License:License
```

List the license details.

```
postgres-splice-app-provider:5432/scribe 6d → f7> active quickstart-licensing:Licensing.License:License
```

Created at	Contract ID	Contract Key	Payload
a9	000cd68ca7d2cf95454b...		dso: DS0::12203a329668884fac6377f41c924df6a11c59f3be909737d27799252930e537c42b user: Org1::12209d2965deec586b4a6d12b80e535bb52407fad54dc2dd88575291780ed5fd9ff4 params: meta: values: provider: AppProvider::12206e5249b12cd9fd05e9b25894c0663b73a18c90baccd7f2d48e7958157b510358 expiresAt: 2025-03-16T21:59:38.403435Z licenseNum: 1

```
active quickstart-licensing:Licensing.License:LicenseRenewalRequest
```

Displays license renewal request details.

```
archives quickstart-licensing:Licensing.AppInstall:AppInstallRequest
```

Shows any archived license(s).

```
postgres-splice-app-provider:5432/scribe 6d → f7> archives quickstart-licensing:Licensing.AppInstall:AppInstallRequest
```

Created at	Archived at	Contract ID	Contract Key	Payload
6f	75	00e906b2720a9b7965a3...		dso: DS0::12203a329668884fac6377f41c924df6a11c59f3be909737d27799252930e537c42b meta: values: user: Org1::12209d2965deec586b4a6d12b80e535bb52407fad54dc2dd88575291780ed5fd9ff4 provider: AppProvider::12206e5249b12cd9fd05e9b25894c0663b73a18c90baccd7f2d48e7958157b510358

Connect to DevNet

Stop the LocalNet containers to change the connection from LocalNet to DevNet.

In the terminal, run:

```
make stop && make clean-all
```

To edit the connection and observability parameters run:

```
make setup
```

When prompted to enable LocalNet, enter “n”. This enables DevNet

Optionally, enter “Y” to enable observability. This starts additional containers which may require more memory for Docker.

You may leave the party hint as the default value by tapping ‘return’ on the keyboard.

```
(base) quickstart ~ % make setup
Starting local environment setup tool...
./gradlew configureProfiles --no-daemon --console=plain --quiet
Enable LocalNet? (Y/n): n
  LOCALNET_ENABLED set to 'false'.

Enable Observability? (Y/n): Y
  OBSERVABILITY_ENABLED set to 'true'.

Specify a party hint (this will identify the participant in the network) [quickstart-... -1]:
  PARTY_HINT set to 'quickstart-... -1'.

.env.local updated successfully.
(base) quickstart ~ %
```

 Running `make setup` regenerates `.env.local` but preserves the contents of the `.env` file settings.

The application is now connected to DevNet.

Important: Migration ID for DevNet Connections

When connecting to DevNet, verify that the `MIGRATION_ID` value in `.env` matches the current network migration ID for your DevNet Super Validator.

Check the current migration ID at <https://sync.global/sv-network/> under the GSF DevNet information section.

For example, if the Super Validator Node Information shows the `migration_id` value as “0” then update `MIGRATION_ID` to “0” in your `.env`.

GSF DevNet Super Validator Node Information

```
{  
  "network": "devnet",  
  "sv": {  
    "migration_id": 0,  
    "version": "0.3.15"  
  },  
  "synchronizer": {  
    "active": {  
      "chain_id_suffix": "5",  
      "migration_id": 0,  
      "version": "0.3.15"  
    },  
    "legacy": null,  
    "staging": null  
  }  
}
```

In .env:

```
ONBOARDING_SECRET_URL=https://sv.sv-1.dev.global.canton.network.digitalasset.com/api/sv/v0/devnet/onboard/validator/prepare  
MIGRATION_ID=0  
APP_PROVIDER_VALIDATOR_PARTICIPANT_ADDRESS=participant-app-provider  
APP_USER_VALIDATOR_PARTICIPANT_ADDRESS=participant-app-user
```

Configuring Non-Default DevNet Sponsors

In DevNet mode, you can configure a non-default SPONSOR_SV_ADDRESS, SCAN_ADDRESS and ONBOARDING_SECRET_URL or ONBOARDING_SECRET in the quickstart/.env file.

 Connecting to DevNet requires a connection to an [approved SV](#). If your organization provides access to the DAML-VPN, then connect to it to access the Digital Asset-sponsored SV.

Your organization may sponsor another [CN-approved SV](#). If this is the case, speak with your administrator for privileged access.

Review the DevNet Global Synchronizer documentation to learn more about the [SV onboarding process](#).

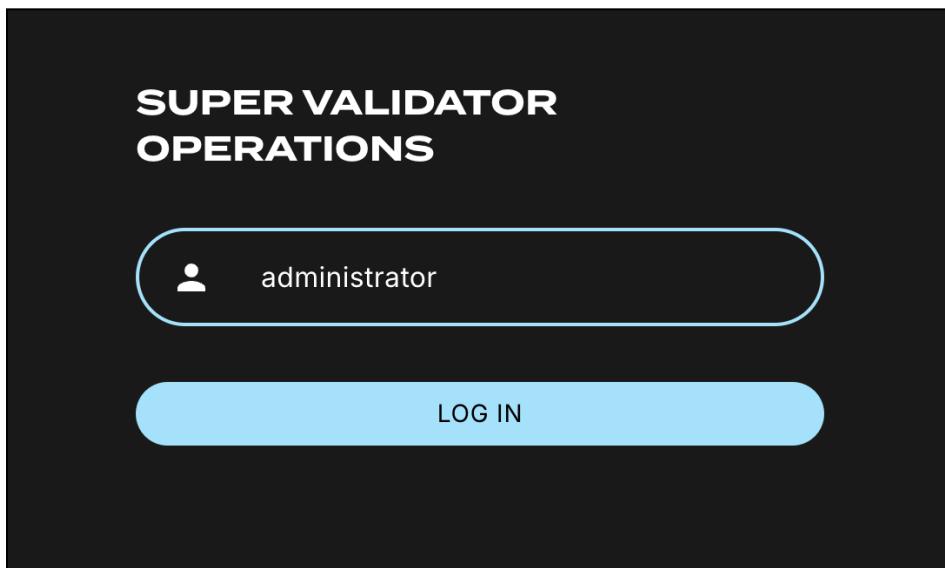
 If you run into errors when making DevNet operations, double check that the DevNet VPN is active. DevNet VPNs may timeout, especially if left unattended for extended periods of time.

In an incognito browser navigate to `localhost:3000/login`. Login as the `Org1 user` and create and archive assets, as before. Logout and do the same as the `AppProvider`.

SV UIs

Navigate to `http://sv.localhost:4000/` for the Super Validator Web UI. The SV view displays data directly from the validator in a GUI that is straightforward to navigate.

Login as ‘administrator’.



The UI shows information about the SV and lists the active SVs.

A screenshot of the "SUPER VALIDATOR OPERATIONS" dashboard. The top navigation bar includes links for "Information", "Validator Onboarding", and "Canton Coin Price". Below the navigation, a sub-navigation bar has "General" selected, along with other tabs: "DSO Info", "Canton Coin Info", "CometBFT Debug Info", and "Domain Node Status".

Super Validator Information
svUser: administrator
svPartyId: sv::1220e29d956452e... [🔗](#)

Active Super Validators
sv: sv::1220e29d956452e... [🔗](#)

Decentralized Synchronizer Operations
dsoLeaderPartyId: sv::1220e29d956452e... [🔗](#)
dsoPartyId: DSO::12200af1e2b814... [🔗](#)
dsoEpoch: 0

The Validator Onboarding menu allows for the creation of validator onboarding secrets.

SUPER VALIDATOR OPERATIONS

Information Validator Onboarding Canton Coin Price Delegate Election Governance [Logout](#)

Validator Onboarding Secrets

[CREATE A VALIDATOR ONBOARDING SECRET](#)

EXPIRES AT	ONBOARDING SECRET
02/13/2025 15:13	quickstart-jpmiller-1::122077f1ae74e... View THIS SV
02/13/2025 15:13	quickstart-jpmiller-1::12203783f656... View THIS SV
02/13/2025 15:10	sv::122016716056ad8277a9220b78... View THIS SV

The CC Price menu option has an option to set the price for open mining rounds.

SUPER VALIDATOR OPERATIONS

Information Validator Onboarding Canton Coin Price Delegate Election Governance [Logout](#)

0.005 USD

Median of Canton Coin prices voted by all Super Validators

Your Desired Canton Coin Price

Amount: [UPDATE](#) [CANCEL](#)

Desired Canton Coin Prices of Other Super Validators

SUPER VALIDATOR	SUPER VALIDATOR PARTY ID	DESIRED CANTON COIN PRICE	LAST UPDATED AT
6	0.005 USD	12/11/2024 11:19	12/11/2024 11:39
5	0.005 USD	12/11/2024 11:09	12/11/2024 11:29
4	0.005 USD	12/11/2024 10:58	12/11/2024 11:18

Open Mining Rounds

ROUND	CANTON COIN PRICE	OPENS AT	TARGET CLOSES AT
6	0.005 USD	12/11/2024 11:19	12/11/2024 11:39
5	0.005 USD	12/11/2024 11:09	12/11/2024 11:29
4	0.005 USD	12/11/2024 10:58	12/11/2024 11:18

Update the price of the CC.

The screenshot shows the 'SUPER VALIDATOR OPERATIONS' section. At the top, there are navigation links: 'Information', 'Validator Onboarding', and 'Canton Coin Price' (which is underlined, indicating it's the current page). Below this, the main content area displays the 'Canton Coin Price for Next Open Mining Round' as '1 USD'. A note below states 'Median of Canton Coin prices voted by all Super Validators'. There is also a field labeled 'Your Desired Canton Coin Price' with the value '1 USD' and a pencil icon for editing.

The updated coin price reflects the new open mining rounds.

The screenshot shows two tables. The first table, titled 'Desired Canton Coin Prices of Other Super Validators', lists four columns: 'SUPER VALIDATOR', 'SUPER VALIDATOR PARTY ID', 'DESIRED CANTON COIN PRICE', and 'LAST UPDATED AT'. The second table, titled 'Open Mining Rounds', lists four columns: 'ROUND', 'CANTON COIN PRICE', 'OPENS AT', and 'TARGET CLOSES AT'. Both tables have a dark background.

SUPER VALIDATOR	SUPER VALIDATOR PARTY ID	DESIRED CANTON COIN PRICE	LAST UPDATED AT

ROUND	CANTON COIN PRICE	OPENS AT	TARGET CLOSES AT
7	1 USD	12/11/2024 11:29	12/11/2024 11:49
6	0.005 USD	12/11/2024 11:19	12/11/2024 11:39
5	0.005 USD	12/11/2024 11:09	12/11/2024 11:29

CC Scan

Navigate to the CC Scan Web UI at <http://scan.localhost:4000/>.

The default activity view shows the total CC balance and the Validator rewards.

Explore, search and find answers to current network configuration details.

The content on this page is computed as of round: 14

TOTAL CANTON COIN BALANCE

7,326,239.93606952 CC
36,631.1996803476 USD

TOTAL APP & VALIDATOR REWARDS

20,520 CC
102.6 USD

Select the Network Info menu to view SV identification.

The screenshot shows a dark-themed web application interface for managing super validators. At the top, there's a navigation bar with tabs: SUPER VALIDATOR OPERATIONS, Information, Validator Onboarding, Canton Coin Price, Delegate Election, and Governance. Below the navigation, there's a sub-navigation bar with tabs: General (which is selected), DSO Info, Canton Coin Info, CometBFT Debug Info, and Domain Node Status. The main content area is divided into sections: **Super Validator Information**, **Active Super Validators**, and **Decentralized Synchronizer Operations**. Each section contains tables with data rows. For example, under "Super Validator Information", there's a row for "svUser" with value "administrator". Under "Active Super Validators", there's a row for "sv" with value "sv::1220e29d956452e...". Under "Decentralized Synchronizer Operations", there's a row for "dsoLeaderPartyId" with value "sv::1220e29d956452e...", and so on.

The Validators menu shows that the local validator has been registered with the SV.

CANTON COIN SCAN		Col 2	Col 3	Col 4	Col 5	Col 6
Created At	Validator	Sponsor	Actions			
02/11/2025 08:06	sv::12201b7f2459db0f8...	sv::12201b7f2...	View	Edit	Delete	THIS SV
02/11/2025 08:05	quickstart-jpmiller-1::12...	sv::12201b7f2...	View	Edit	Delete	THIS SV
02/11/2025 08:05	quickstart-jpmiller-1::12...	sv::12201b7f2...	View	Edit	Delete	THIS SV

Observability Dashboard

In a web browser, navigate to <http://localhost:3030/dashboards> to view the observability dashboards. Select “Quickstart - consolidated logs”.

The screenshot shows the Digital Asset Observability Dashboard at <http://localhost:3030/dashboards>. The left sidebar includes links for Home, Starred, Dashboards (which is selected), Explore, Alerting, Connections, and Administration. The main area is titled 'Dashboards' with the sub-instruction 'Create and manage dashboards to visualize your data'. It features a search bar, a 'Filter by tag' dropdown, and a 'Starred' checkbox. A table lists dashboards with columns for Name and Tags. One dashboard, 'Quickstart - consolidated logs', is highlighted with a green checkmark in the 'Starred' column and is associated with 'logs' and 'loki' tags.

The default view shows a running stream of all services.

Change the services filter from “All” to “participant” to view participant logs. Select any log entry to view its details.

Keycloak in the CN-QS

Keycloak is an open-source Identity and Access Management (IAM) solution that provides authentication, authorization, and user management for modern applications and services. It acts as a centralized authentication server that handles user logins, session management, and security token issuance.

The CN-QS uses Keycloak to provide secure authentication across its distributed architecture. Keycloak maintains separation between authentication concerns and business logic.

Realm Structure

The CN-QS defines two Keycloak realms that mirror its business domains. The AppProvider realm manages authentication for services and users on the provider side of the application. The AppUser realm handles authentication for the consumer side. When components like validators or participant nodes receive requests, they validate the authentication tokens against the appropriate realm.

Keycloak Configuration

The default `.env` configuration includes predefined users in each realm:

- **User "Pat"** (`AUTH_APP_PROVIDER_WALLET_ADMIN_USER_NAME=pat`)
- **UUID:** `553c6754-8879-41c9-ae80-b302f5af92c9`
(`AUTH_APP_PROVIDER_WALLET_ADMIN_USER_ID`)

AppUser Realm:

- **User "Alice"** (`AUTH_APP_USER_WALLET_ADMIN_USER_NAME=alice`)
- **UUID:** `92a520cb-2f09-4e55-b465-d178c6cfe5e4`
(`AUTH_APP_USER_WALLET_ADMIN_USER_ID`)
- **Password:** `abc123` (`AUTH_APP_USER_WALLET_ADMIN_USER_PASSWORD`)

Customizing Keycloak for Business Needs

You can customize the Keycloak configuration to meet your specific business requirements.

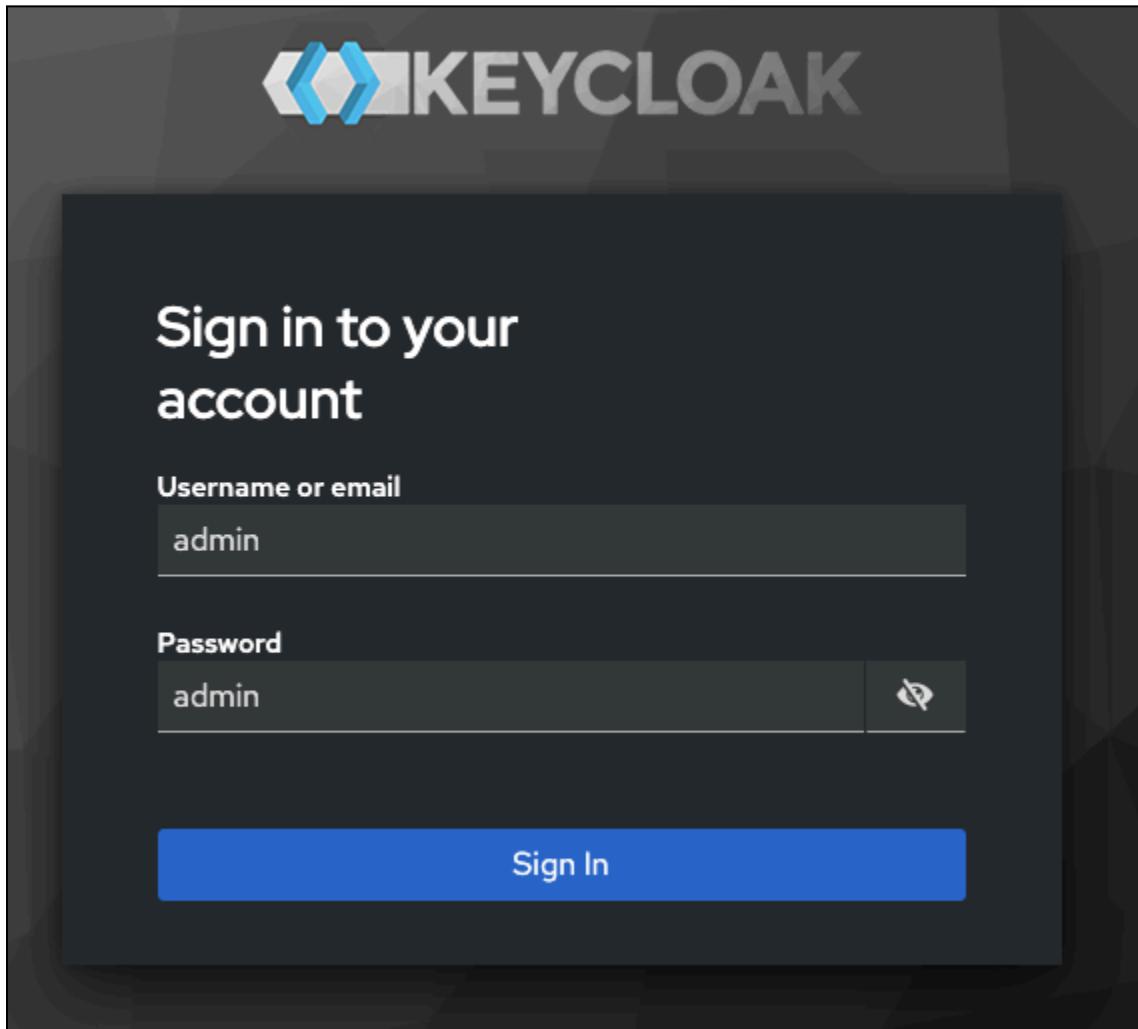
Accessing the Admin Console

The Keycloak Admin Console is available at:

```
http://keycloak.localhost:8082/admin/master/console/#/master
```

To log in use the default credentials:

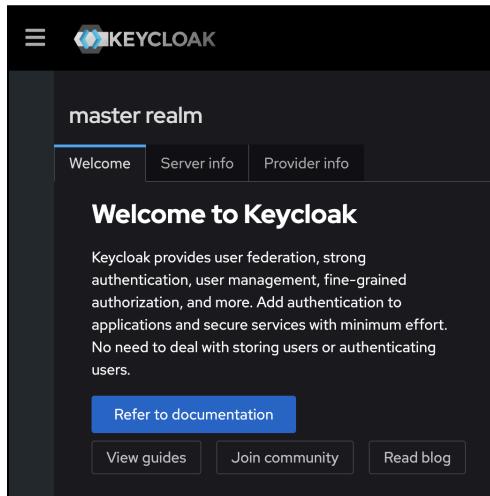
- **Username:** `admin`
- **Password:** `admin`



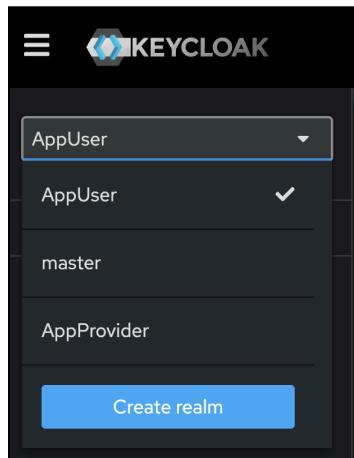
Customization Scenarios

Add a New User

1. Log in to the Keycloak Admin console

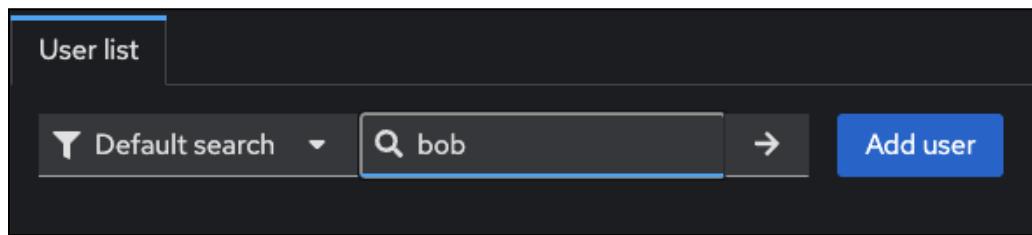


2. Select the appropriate realm (AppProvider or AppUser)



3. Navigate to the “Users” -> “Add user”

The screenshot shows the 'Users' page in Keycloak. On the left, there is a sidebar with navigation links: 'Manage', 'Clients', 'Client scopes', 'Realm roles', 'Users' (which is selected and highlighted in blue), 'Groups', 'Sessions', and 'Events'. The main content area has a title 'Users' and a subtitle 'Users are the users in the current realm.' Below this is a 'User list' table. The table has columns: 'Username', 'Email', 'Last name', and 'First name'. Two users are listed: 'adrian' (Email: adrian@app-user.localhost, Last name: Doe, First name: Adrian) and 'alice' (Email: alice@app-user.localhost, Last name: the user, First name: Alice). At the top right of the table is a blue 'Add user' button. Above the table, there is a search bar with 'Default search' and 'Search user' fields, and a 'Refresh' button.



4. Fill in the user details and click **Create**

A screenshot of a user creation form titled 'General'. The form fields include: 'Username *' with value 'bob'; 'Email' with placeholder 'Email'; 'First name' with placeholder 'First name'; 'Last name' with placeholder 'Last name'; and a 'Groups' section with a 'Join Groups' button. At the bottom are 'Create' and 'Cancel' buttons.

5. Go to the **Credentials** tab to set a password

No credentials

This user does not have any credentials. You can set password for this user.

Set password

Set password for bob

Password * abc123

Password confirmation * abc123

Temporary ⓘ Off

Save **Cancel**

6. Save the password

Set password?

Are you sure you want to set the password for the user bob?

Save password **Cancel**

7. You can now sign in using the new user and their password.

- a. Click **AppUser**

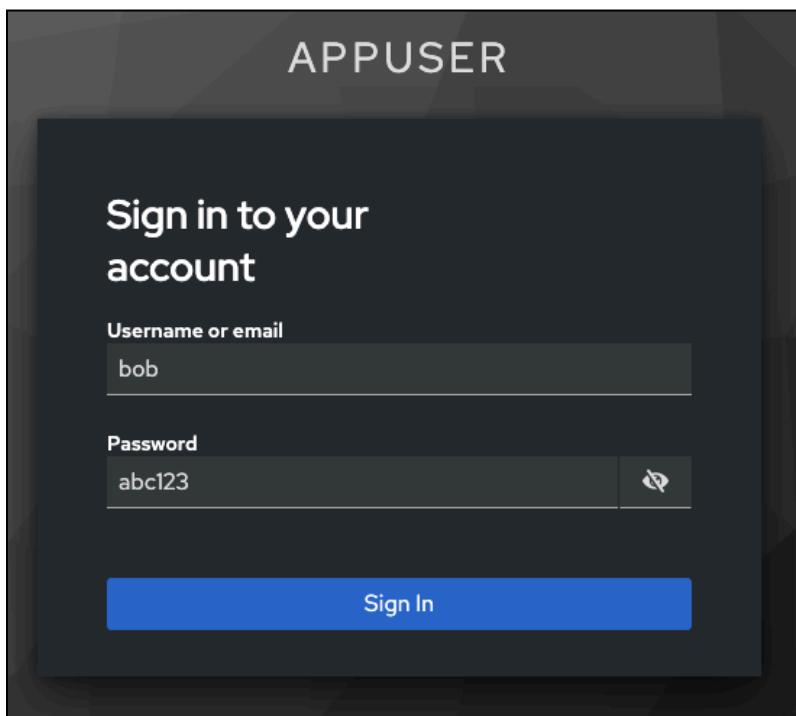
Canton Network Quickstart

Login with OAuth 2.0

[AppProvider](#)

[AppUser](#)

AppProvider user: pat, password: abc123
AppUser user: alice, password: abc123



8. Bob is now a user

Canton Network Quickstart Home AppInstalls Licenses bob doe Logout

App Installs

Note: Run `make create-app-install-request` to submit an AppInstallRequest

Modify Client Settings

1. Select the appropriate realm
2. Navigate to **Clients** -> Select the client to modify

The screenshot shows the Keycloak admin interface for managing clients. On the left, a sidebar menu is open under the 'AppUser' realm. The 'Clients' option is selected, highlighted in blue. The main content area is titled 'Clients' and contains a sub-header: 'Clients are applications and services that can request authentication of a user.' Below this are three tabs: 'Clients list' (which is active), 'Initial access token', and 'Client registration'. A search bar labeled 'Search for client' with a magnifying glass icon is followed by a 'Create client' button and an 'Import client' link. A 'Refresh' button is also present. At the bottom right, there's a pagination indicator '1 - 10' with arrows. The main table lists ten clients:

Client ID	Name	Type	Description	Home URL
accounting	client_1	OpenID Connect	-	http://keycloak.localhost:8082/realm/AppUser/account/
accounting	client_2	OpenID Connect	-	http://keycloak.localhost:8082/realm/AppUser/account/
admin	client_3	OpenID Connect	-	-
app	client_4	OpenID Connect	-	-
app	client_5	OpenID Connect	-	-
app	client_6	OpenID Connect	-	-
app	client_7	OpenID Connect	-	-
broker	client_8	OpenID Connect	-	-
realm	client_r	OpenID Connect	-	-
security	client_s	OpenID Connect	-	http://keycloak.localhost:8082/admin/AppUser/console/

3. Update settings per your needs

The screenshot shows the 'Clients' section of the Digital Asset management interface. A client named 'app-user-wallet' is selected, which is an 'OpenID Connect' type. The client is currently 'Enabled'. The 'General settings' tab is active, displaying fields for Client ID (app-user-wallet), Name (client_app-user-wallet), and Description (empty). An 'Always display in UI' toggle is set to 'Off'. The 'Access settings' tab shows empty fields for Root URL and Home URL, and a Valid redirect URIs field containing 'http://wallet.localhost:2000'. A sidebar on the right lists other configuration sections: General settings, Access settings, Capability config, Login settings, and Logout settings. At the bottom are 'Save' and 'Revert' buttons.

4. Save changes

Add a New Client

1. Select the appropriate realm
2. Navigate to “Clients” -> “Create”

The screenshot shows the 'Clients list' page. It features a navigation bar with tabs for 'Clients list', 'Initial access token', and 'Client registration'. Below the navigation is a search bar with a magnifying glass icon and the placeholder 'Search for client'. To the right of the search bar is a large blue 'Create client' button.

3. Configure the client general settings. Click **Next** for additional configuration options

The screenshot shows the 'Create client' page with a dark theme. On the left, a sidebar lists three steps: 1. General settings (selected), 2. Capability config, and 3. Login settings. The main area contains fields for Client type (set to OpenID Connect), Client ID (empty), Name (empty), and Description (empty). A toggle switch for 'Always display in UI' is set to Off. At the bottom, there are Back, Next, and Cancel buttons.

Clients > Create client

Create client

Clients are applications and services that can request authentication of a user.

1 General settings 2 Capability config 3 Login settings

Client type ⓘ OpenID Connect

Client ID * ⓘ

Name ⓘ

Description ⓘ

Always display in UI ⓘ Off

Back Next Cancel

4. Configure additional settings

The screenshot shows the 'Create client' configuration interface. On the left, a sidebar lists three steps: 1. General settings, 2. Capability config (which is selected and highlighted in blue), and 3. Login settings. The main panel contains several configuration sections. Under 'Client authentication', the toggle switch is set to 'Off'. Under 'Authorization', it is also set to 'Off'. In the 'Authentication flow' section, the 'Standard flow' checkbox is checked, while 'Implicit flow', 'OAuth 2.0 Device Authorization Grant', and 'OIDC CIBA Grant' are unchecked. At the bottom of the panel are 'Back', 'Next', and 'Cancel' buttons.

The screenshot shows the 'Create client' configuration interface. The sidebar indicates 'Login settings' is selected. The main panel displays fields for 'Root URL', 'Home URL', 'Valid redirect URIs' (with a plus icon to add more), 'Valid post logout redirect URIs' (with a plus icon to add more), and 'Web origins' (with a plus icon to add more). At the bottom are 'Back', 'Save' (which is highlighted in blue), and 'Cancel' buttons.

5. Save the client

Update Environment Variables

After making changes to Keycloak configuration, you may need to update the corresponding environment variables in the `.env` file:

1. The Keycloak user must have the same ID as the ledger user's ID.
 2. For client changes, update the corresponding client ID and secret
 3. For user changes, update the corresponding user ID and credentials
 4. Restart the services to apply the changes:

make stop && make start

Troubleshooting

Login Failures:

- ## 1. Verify Keycloak is running: `make status`

Find **keycloak** near **grafana** and **loki** in the list.

Keycloak should show as “healthy”

grafana	grafana/grafana:11.1.5	"/run.sh"	grafana	30 minutes ago	Up 30 minutes	0.8.0.8-3808-3808/tcp
keycloak	quay.io/keycloak/keycloak:26.1.0	"/opt/keycloak/bin/w...	keycloak	30 minutes ago	Up 30 minutes	healthy 8080/tcp, 8443/tcp, 9080/tcp
keycloak-postgres	postgres:13	"docker-entrypoint.s...	keycloak-postgres	30 minutes ago	Up 30 minutes	5432/tcp
loki	grafana/loki:3.1.1	"/usr/bin/loki -con...	loki	30 minutes ago	Up 30 minutes	3108/tcp

- ## 2. Check keycloak credentials in .env file

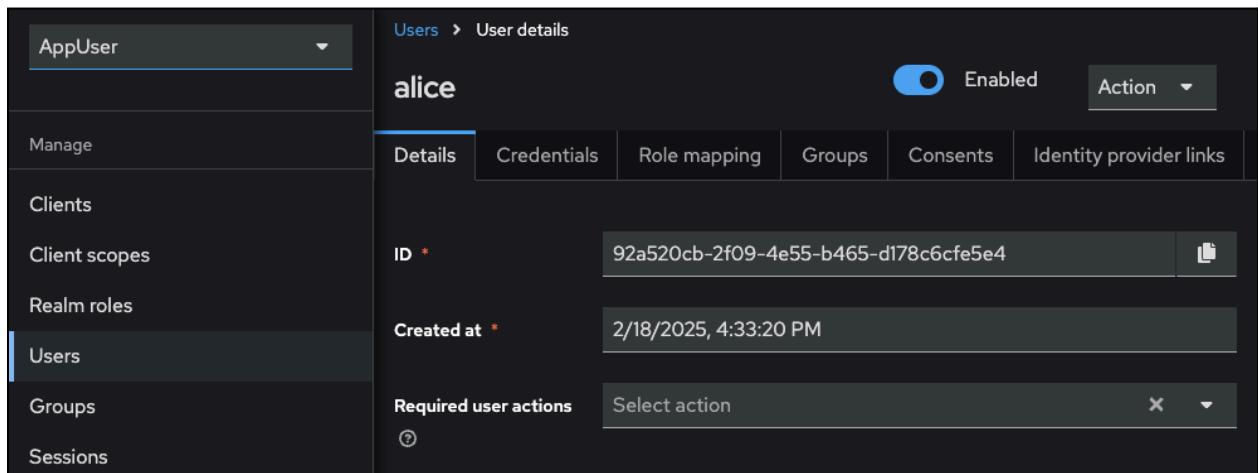
```
AUTH_APP_USER_ISSUER_URL_BACKEND=http://nginx-keycloak:8082/realms/AppUser
# for backend
AUTH_APP_USER_ISSUER_URL=http://keycloak.localhost:8082/realms/AppUser
# for backend, wallet-ui
AUTH_APP_PROVIDER_ISSUER_URL=http://keycloak.localhost:8082/realms/AppProvider
# for backend oidc client conf, wallet-ui
AUTH_APP_PROVIDER_ISSUER_URL_BACKEND=http://nginx-keycloak:8082/realms/AppProvider
# for backends
```

3. Check that the Keycloak user ID matches the ledger user ID

- a. App User

- i. Compare the **ID** value in Keycloak's User Details with the **AUTH_APP_USER_WALLET_ADMIN_USER_ID** value in `.env`.

`AUTH_APP_USER_WALLET_ADMIN_USER_ID=92a520cb-2f09-4e55-b465-d178c6cf5e4`



The screenshot shows the Keycloak User details page for a user named 'alice'. The left sidebar has a dropdown set to 'AppUser' and a 'Users' tab selected. The main area shows the user 'alice' with an 'Enabled' toggle switch. Below it, there are tabs for 'Details', 'Credentials', 'Role mapping', 'Groups', 'Consents', and 'Identity provider links'. The 'Details' tab is active, showing the following fields:

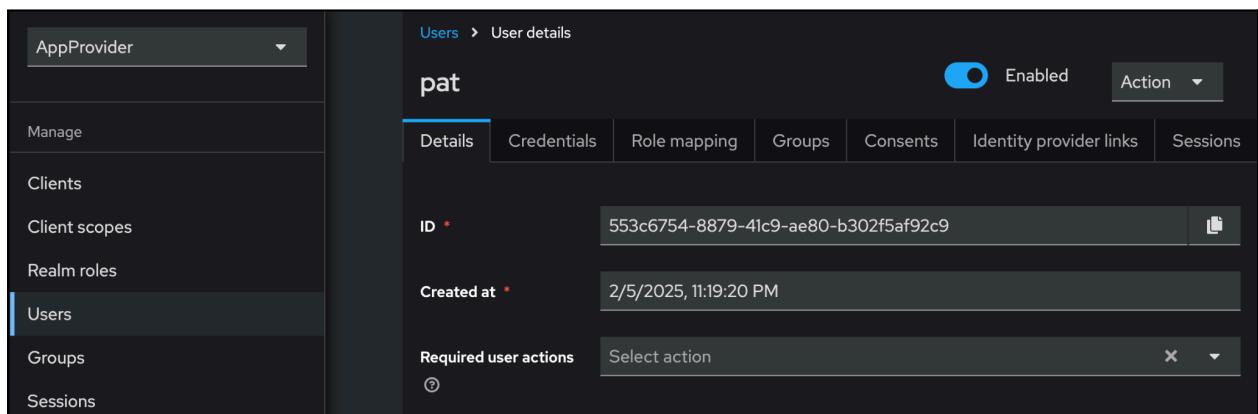
ID *	92a520cb-2f09-4e55-b465-d178c6cf5e4
Created at *	2/18/2025, 4:33:20 PM
Required user actions	Select action

b. App Provider

Compare the **ID** value in Keycloak's User Details with the

AUTH_APP_PROVIDER_WALLET_ADMIN_USER_ID value in `.env`.

`AUTH_APP_PROVIDER_WALLET_ADMIN_USER_ID=553c6754-8879-41c9-ae80-b302f5af92c9`



The screenshot shows the Keycloak User details page for a user named 'pat'. The left sidebar has a dropdown set to 'AppProvider' and a 'Users' tab selected. The main area shows the user 'pat' with an 'Enabled' toggle switch. Below it, there are tabs for 'Details', 'Credentials', 'Role mapping', 'Groups', 'Consents', 'Identity provider links', and 'Sessions'. The 'Details' tab is active, showing the following fields:

ID *	553c6754-8879-41c9-ae80-b302f5af92c9
Created at *	2/5/2025, 11:19:20 PM
Required user actions	Select action

Learn more about using Keycloak through their documentation portal:

[Keycloak Official Documentation](#)

[Keycloak Server Administration Guide](#)

[Securing Applications with Keycloak](#)

Next Steps

You've completed a business operation in the CN-QS and have been introduced to the basics of the Canton Console, Daml Shell, and the SV UIs.

Learn more about Daml Shell and the project structure in the [Project Structure Guide](#).