

Kotlin MultiPlatform App with Nix Build Environment

Perry Kundert

2024-11-25 11:11:00

Building a Kotlin app requires a multitude of dev dependencies, usually supplied by installing random tools like Android Studio, etc. This produces a non-deterministic build environment. (PDF/Text)

Using Nix, we provide a deterministic build environment, replicable quickly on any suitable platform.

Contents

1	Intelli-J Idea IDE	1
2	Create a Demo Kotlin App	1
3	TODO	1
4	Project: User Onboarding and Authentication Flow	2

1 IntelliJ Idea IDE

Run

```
$ nix-shell
```

to gain access to ‘kdoctor’ to analyze your development environment, and ‘idea’ to start IntelliJ Idea.

2 Create a Demo Kotlin App

Follow <https://www.jetbrains.com/help/kotlin-multiplatform-dev/multiplatform-create-first-app.html> to create a demo app. This is how we produced GreetingsKMP.zip.

3 TODO

- Provide Nix Flake environment w/ build tools and SDKs: <https://github.com/tadfisher/android-nixpkgs>

- Add access to Holochain dev environment with <https://github.com/spartan-holochain-counsel/nix-overlay>

4 Project: User Onboarding and Authentication Flow

This project implements a user onboarding and authentication flow where identity and agency are based around Ed25519 keypairs (like Passkeys). The system will be implemented in Holochain HDK v0.5.0 and HDI v0.4.0, with JavaScript binding to allow clients to interact via WebSockets.

Capabilities:

- A simple authentication API compatible with Kotlin client apps.
- A simple onboarding flow that supports email + password for agent key generation.
- Designation of 2+ recovery partners for key revocation/recovery.
- Support for advanced users using external hardware wallets for key generation and recovery.

User Stories:

1. As a new user, I want to create a new agent keypair using an email and password, so that I can initialize my identity and start using the system.
2. As an existing user, I want to associate a new device with my identity, so that I can access the system from multiple devices.
3. As a user, I want to designate 2+ recovery partners during onboarding, so that I can recover my keypair if needed.
4. As an advanced user, I want to use an external hardware wallet to generate my keypair, so that I can have enhanced security.
5. As a user, I want a simple authentication API compatible with Kotlin client apps, so that I can easily integrate the authentication flow into my application.
6. As a user, I want to be able to revoke and recover my keypair using designated recovery partners or an external hardware wallet, so that I can maintain access to my identity securely.

Tasks:

4.0.1 Project Setup

1. Initialize Holochain Project
 - Setup Holochain environment with HDK v0.5.0 and HDI v0.4.0.
 - Create new Holochain DNA for the project.

4.0.2 Onboarding Flow

1. Create Keypair Generation for Basic Users
 - Implement keypair generation using email and password.
 - Store generated agent keypair in Holochain.
2. Designate Recovery Partners
 - Implement functionality to designate 2+ recovery partners during onboarding.
 - Ensure recovery partners create agents using email and password.

4.0.3 Authentication Flow

1. Implement Simple Authentication API
 - Develop API endpoints for authentication compatible with Kotlin client apps.
 - Ensure API supports login using generated agent keypairs.

4.0.4 Device Association

1. Associate New Device with Existing Identity
 - Implement functionality for existing users to associate new devices with their identity.
 - Ensure new devices generate an agent keypair and associate with the existing identity.

4.0.5 Key Revocation and Recovery

1. Implement Key Revocation and Recovery
 - Develop functionality for key revocation using designated recovery partners.
 - Implement recovery process using recovery partners' agents.
 - Support key revocation/recovery using external hardware wallets for advanced users.

4.0.6 Advanced User Support

1. Support External Hardware Wallets
 - Integrate external hardware wallets for keypair generation.
 - Ensure compatibility with key revocation/recovery processes.

4.0.7 Javascript Binding

1. Develop Javascript Binding for Client Interaction
 - Create Javascript bindings to interact with Holochain via WebSockets.
 - Ensure bindings support keypair generation, authentication, device association, and recovery processes.

4.0.8 Testing and Documentation

1. Testing
 - Write unit tests for all functionalities.
 - Conduct integration tests to ensure smooth flow of onboarding, authentication, and recovery processes.
2. Documentation
 - Document the API endpoints and usage.
 - Provide a guide for setting up the system and integrating with client apps.
 - Write user guides for basic and advanced users.

Milestones:

4.0.9 Milestone 1: Project Setup and Keypair Generation

- Initialize Holochain project.
- Implement keypair generation for basic users.
- Designate recovery partners.

4.0.10 Milestone 2: Authentication API and Device Association

- Develop simple authentication API.
- Implement device association functionality.

4.0.11 Milestone 3: Key Revocation and Recovery

- Implement key revocation and recovery using recovery partners.
- Support external hardware wallets for advanced users.

4.0.12 Milestone 4: Javascript Binding and Testing

- Develop Javascript binding for client interaction.
- Complete testing and documentation.