Soroban Toolkit

Soroban Toolkit is a powerful library designed to simplify interactions with Stellar’s Soroban smart contracts. It provides tools for deploying, invoking, and managing smart contracts while handling complex blockchain interactions with ease.

Features

- **Smart Contract Deployment**: Streamline the deployment of Soroban smart contracts.
- **Transaction Management**: Simplify transaction creation and execution.
- **Address Book**: Keep track of deployed contracts and WASM hashes.
- **Verbose Logging**: Customizable verbosity levels (none, some, full) to tailor logs.
- **Airdrop Utility**: Easy access to funding accounts on test networks.

Getting Started

Installation

```
# Install Node.js 22 LTS
nvm install 22

# Use Node.js 22 as default
nvm use 22
nvm alias default 22

# Verify the version
node --version
```

1. Clone the repository:

PROF

```
git clone https://github.com/paltalabs/soroban-toolkit.git
cd soroban-toolkit
```

2. Install dependencies:

```
yarn install
```

3. Link the package for local testing:

```
yarn link
```

4. In your testing project:

```
yarn link <package_name>
```

Development Workflow

Watch Mode

Run the library in watch mode to rebuild changes automatically:

```
yarn start
```

Build for Production

Compile the library for distribution:

```
yarn build
```

Usage

Example Setup

To use the toolkit, initialize it with your network configuration and contract paths:

```
import { createToolkit, airdropAccount } from "<package_name>";

const toolkitLoader = createToolkit({
  adminSecret: process.env.ADMIN_SECRET_KEY!,
  contractPaths: {
  some_contract: "./target/wasm32-unknown-unknown/release/some_contract.wasm"
  },
  customNetworks: [
  {
    network: "standalone",
    friendbotUrl: "http://localhost:8000/friendbot",
    horizonRpcUrl: "http://localhost:8000",
    sorobanRpcUrl: "http://localhost:8000/soroban/rpc",
    networkPassphrase: "Standalone Network ; February 2017",
  },
  ],
  verbose: "full",
});
```

PROF

```
const toolkit = toolkitLoader.getNetworkToolkit("standalone");

await airdropAccount(toolkit, toolkit.admin);

const account = await
toolkit.horizonRpc.loadAccount(toolkit.admin.publicKey());
console.log("Account details:", account);
```

Features and Modules

1. Toolkit Loader

`createToolkit` initializes the toolkit with your configuration:

```
const toolkitLoader = createToolkit({
  adminSecret: 'your_admin_secret',
  contractPaths: {
    contract_name: 'path_to_wasm_file',
  },
  customNetworks: [
    {
      network: 'standalone',
      friendbotUrl: 'http://localhost:8000/friendbot',
      horizonRpcUrl: 'http://localhost:8000',
      sorobanRpcUrl: 'http://localhost:8000/soroban/rpc',
      networkPassphrase: 'Standalone Network ; February 2017',
    },
  ],
  verbose: 'full',
});
```

2. Address Book

PROF

The toolkit integrates an AddressBook to manage deployed contracts:

```
toolkit.addressBook.setContractId('contractKey', 'contractId');
toolkit.addressBook.getContractId('contractKey'); // Returns the
contract ID
toolkit.addressBook.listKeys(); // Lists all keys in the address book
```

3. Contract Deployment

Deploy and manage contracts:

```
await deploySorobanToken(toolkit, toolkit.admin);
```

4. Verbose Logging

Adjust verbosity during toolkit initialization:

- **none**: No logs
- **some**: Basic logs
- **full**: Full details (e.g., XDRs, hashes)

Testing

Run tests with:

```
yarn test
```

Publishing to NPM

1. Build the library:

```
yarn build
```

2. Publish to NPM:

```
yarn publish
```

Directory Structure

```
src
├── config
│   ├── defaultNetworks.ts      # Default network configurations
│   ├── loader.ts               # Toolkit loader
│   └── toolkit.ts              # Toolkit core
├── index.ts                    # Entry point
├── managers
│   ├── contract.ts             # Contract deployment and management
│   ├── token.ts                # Token contract utilities
│   └── transaction.ts          # Transaction utilities
├── soroban_token.wasm          # Token contract WASM file
├── utils
│   ├── accountUtils.ts         # Account utilities (e.g., airdrop)
│   ├── addressBook.ts          # Address book for tracking deployed
contracts
│   └── utils.ts                # General utilities
```

PROF

Additional Notes

- Ensure `soroban_token.wasm` is included in your build. Use the `rollup-plugin-copy` configuration to copy it to `dist`.
- Use `yarn link` for testing local changes and `yarn unlink` to remove links.