

Module 4 - Sui上でのdApp構築

Sui上でのdApp構築

スマートコントラクト開発、フロントエンド統合、ウォレット接続を含む、Sui上でのdApp構築プロセスを理解する。



アジェンダ

1. フロントエンドとスマートコントラクトの接続
2. ウォレット統合
3. NFTミントアプリの構築
4. 基本的なDeFiアプリの構築
5. 実践的な演習

フロントエンドとスマートコントラクトの接続

Sui TypeScript SDK

- ✓ 低レベルブロックチェーン
インタラクション
- ✓ スマートコントラクト
関数の呼び出し
- ✓ ネットワーク用の
SuiClientProvider
- ✓ ウォレット管理用の
WalletProvider
- ✓ ユーザーログイン用の
ConnectButton

React用dApp Kitの使用例

- ✓ アプリをプロバイダーで
ラップし、ConnectButtonを
使用してウォレットアドレス
を取得

ウォレット統合

ウォレットの役割

- ✓ トランザクションの署名、
ユーザーアセットの管理

dApp Kitの使用法

- ✓ `WalletProvider` +
`ConnectButton`でシームレス
な接続を実現

開発者向けヒント

- ✓ `useCurrentAccount`
フックでウォレット情報に
アクセス

NFTミントアプリの構築

スマートコントラクト

- ✓keyとstore属性を持つ
NFT構造体を定義

ミント関数

- ✓新しく作成されたNFTを
送信者に転送

フロントエンド

- ✓SDK/dApp Kitを使用して
メタデータ付きでミント
を呼び出し

*MoveとJavaScriptの実装例が利用可能です。

基本的なDeFiアプリの構築

レンディングプール

スマートコントラクト

- ✓ レンディングプール構造体を定義
- ✓ 預金、借入、返済関数を実装
- ✓ 金利処理を含む

フロントエンド

インターフェース

- ✓ SUI預金ページ
- ✓ 担保を使った借入
- ✓ ローン返済ページ

SDK統合

- ✓ UIからコントラクト関数を呼び出し

フロントエンド統合の最適化

React Hooks

- ✓ 状態とアカウント管理にフックを使用

プロバイダーラッピング

- ✓ アプリをSuiClientとWalletプロバイダーでラップ

ユーザーエクスペリエンス

- ✓ ウォレット接続状況とエラーハンドリングを表示

実践的な演習

NFTミントアプリ

- ✓ NFT構造体用のMoveコード、フロントエンド
ミント呼び出し用のTypeScript

基本的なDeFiアプリの構築

- ✓ SDKを使用：預金、借入、返済のために
フロントエンドからスマートコントラクト関数を呼び出し

(https://mirror.xyz/greymate.eth/_P2NXvVoh9wISj_mqgavDymIERCnW2DgC1gigJNrmUI)

(<https://docs.sui.io/guides/developer/app-examples>)

ありがとうございました。