"CONNECT WALLET" WITHOUT THE RAGE QUIT

Building Web3 UX That Feels Like Magic

How do you do, fellow kids?

ELLIOT VORIS

“

WEB3 UX DOESN’T HAVE TO SUCK!

Let’s prove it.

TABLE OF CONTENTS

01

Smart Wallets are the Future

02

What time is it? Demo Time!

03

Look at that Code

04

Your turn. Get Started!

STELLAR: A CRYPTO OG

Stellar’s got a lot going for it...

🏛️ Been around 10+ years

💨 Fast transaction times

⛽ Low gas fees

⚓ Network of on- and off-ramps

🧮 Soroban: Top-notch smart contract platform

📜 ...the list goes on

BUT ALL THAT'S OLD HAT

Instead, let's see how we can bring peopleon-chain *quickly* & **easily**.

With smart wallets!

Smart Wallet: a smart contract that acts as an account

Passkey Kit: Use passkeys to create and use smart wallets on Stellar

Launchtube: A paymaster service for gasless transactions

CHECK THIS OUT

TODO: video recorded, just needs to be trimmed down a bit

Smart Stellar Demo

Login or create a new account.

THE BREAKDOWN

The experience as a user:

You're already familiar with this onboarding flow

No more remembering seed phrase, secret key, etc.

Entirely non-custodial: Your passkey. Your crypto.

The experience as a developer:

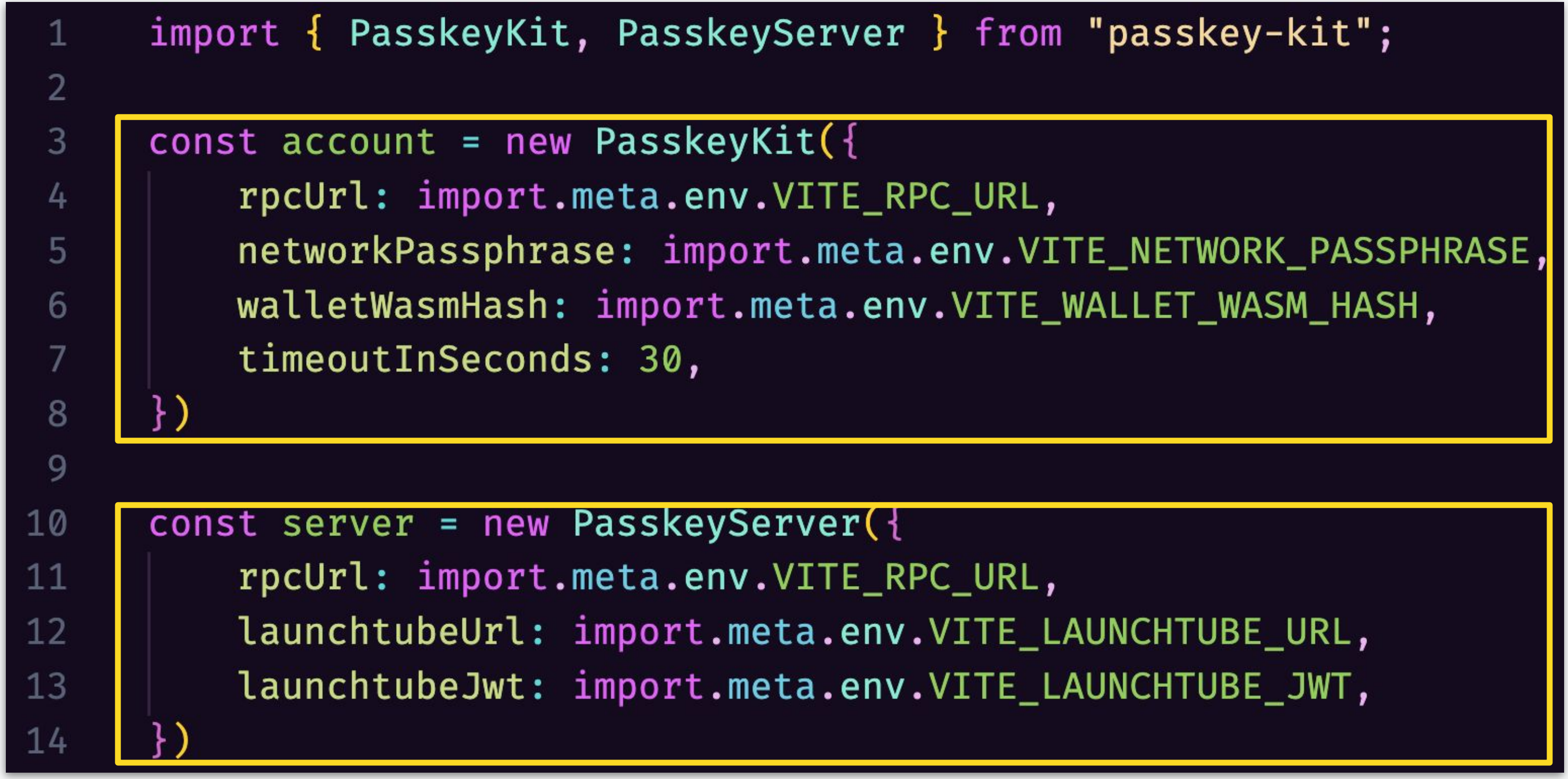
No wallet plugin integration for your frontend

Super simple to setup and get started

Your users are automatically on-chain

LOOK AT THAT CODE!

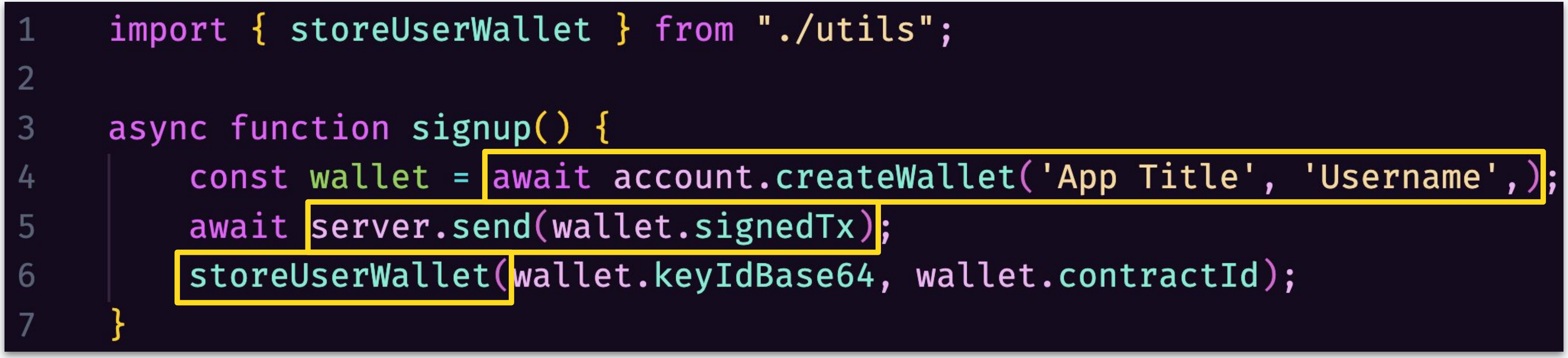
Super Simple Passkey-Kit Setup



```
import { PasskeyKit, PasskeyServer } from "passkey-kit";

const account = new PasskeyKit({
    rpcUrl: import.meta.env.VITE_RPC_URL,
    networkPassphrase: import.meta.env.VITE_NETWORK_PASSPHRASE,
    walletWasmHash: import.meta.env.VITE_WALLET_WASM_HASH,
    timeoutInSeconds: 30,
})

const server = new PasskeyServer({
    rpcUrl: import.meta.env.VITE_RPC_URL,
    launchtubeUrl: import.meta.env.VITE_LAUNCHTUBE_URL,
    launchtubeJwt: import.meta.env.VITE_LAUNCHTUBE_JWT,
})
```

LOOK AT THAT CODE!

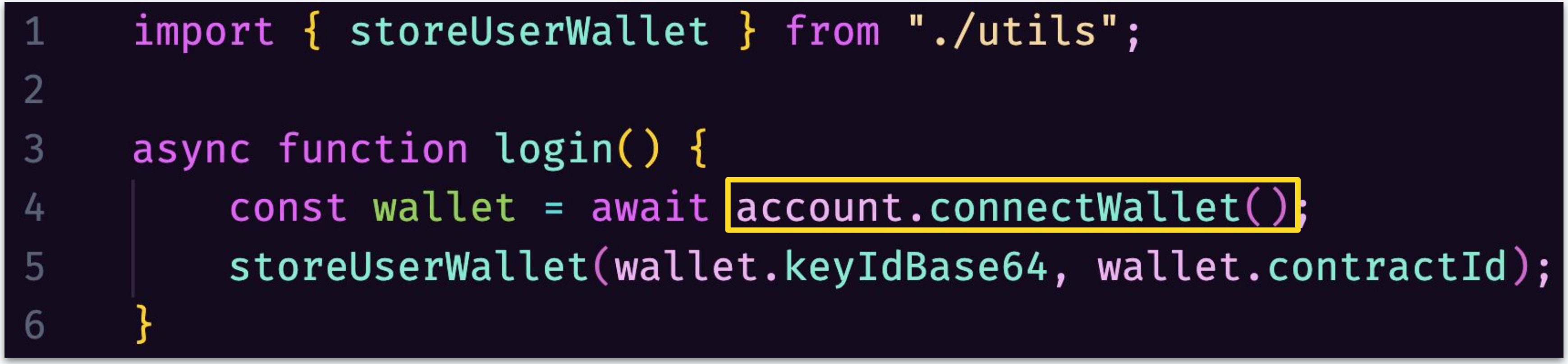
Super Simple Signup Function



```
import { storeUserWallet } from "./utils";

async function signup() {
    const wallet = await account.createWallet('App Title', 'Username',);
    await server.send(wallet.signedTx);
    storeUserWallet(wallet.keyIdBase64, wallet.contractId);
}
```

LOOK AT THAT CODE!

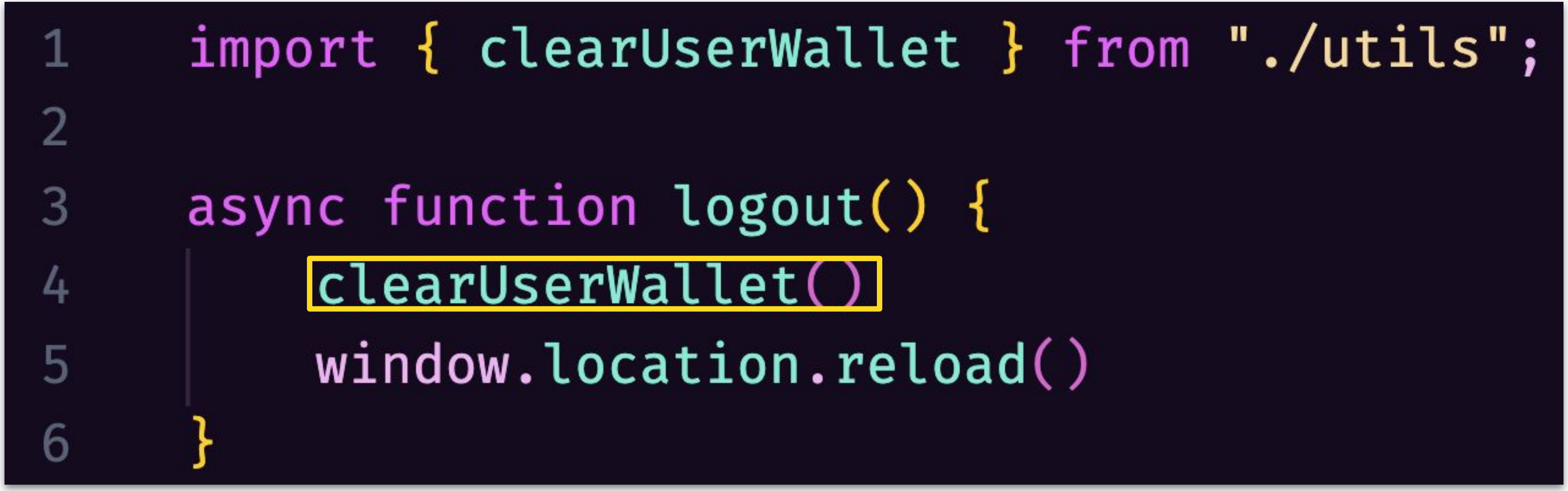
Super Simple Login Function



```
import { storeUserWallet } from "./utils";

async function login() {
    const wallet = await account.connectWallet();
    storeUserWallet(wallet.keyIdBase64, wallet.contractId);
}
```

LOOK AT THAT CODE!

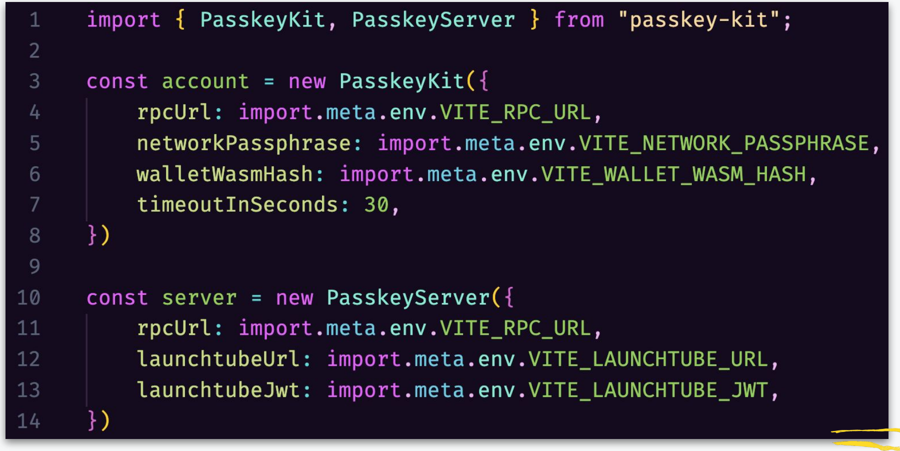
Super Simple Logout Function



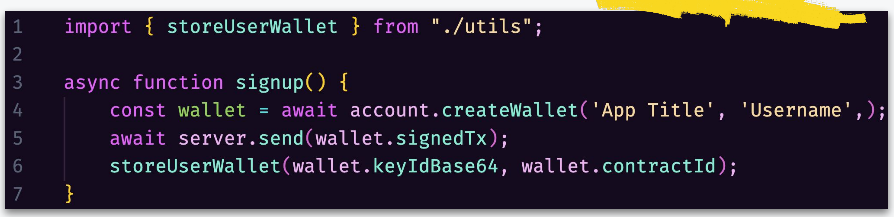
```
import { clearUserWallet } from "./utils";

async function logout() {
    clearUserWallet()
    window.location.reload()
}
```

LOOK AT THAT CODE!

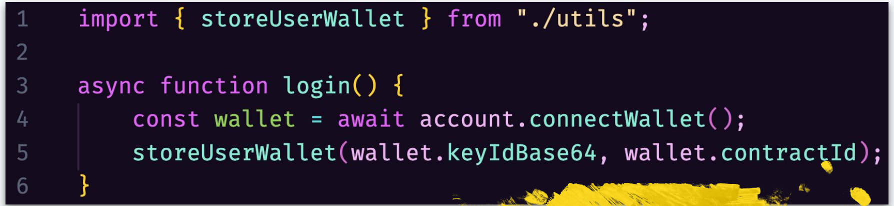
All your wallet packages are belong to us



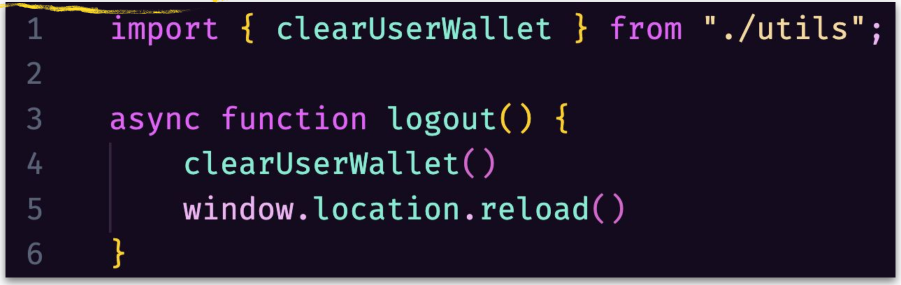
```javascript
import { PasskeyKit, PasskeyServer } from "passkey-kit";

const account = new PasskeyKit({
    rpcUrl: import.meta.env.VITE_RPC_URL,
    networkPassphrase: import.meta.env.VITE_NETWORK_PASSPHRASE,
    walletWasmHash: import.meta.env.VITE_WALLET_WASM_HASH,
    timeoutInSeconds: 30,
})

const server = new PasskeyServer({
    rpcUrl: import.meta.env.VITE_RPC_URL,
    launchtubeUrl: import.meta.env.VITE_LAUNCHTUBE_URL,
    launchtubeJwt: import.meta.env.VITE_LAUNCHTUBE_JWT,
})
```



```javascript
import { storeUserWallet } from "./utils";

async function signup() {
    const wallet = await account.createWallet('App Title', 'Username',);
    await server.send(wallet.signedTx);
    storeUserWallet(wallet.keyIdBase64, wallet.contractId);
}
```



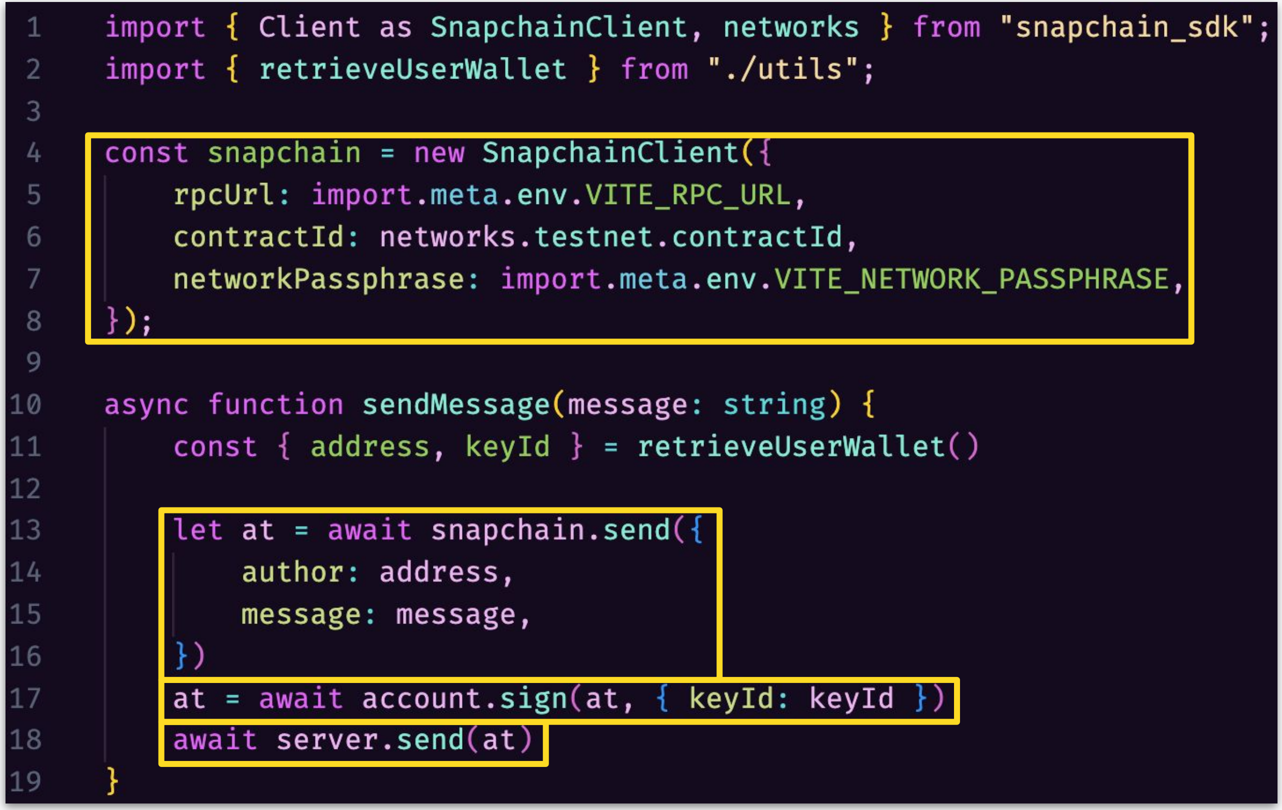
```javascript
import { storeUserWallet } from "./utils";

async function login() {
    const wallet = await account.connectWallet();
    storeUserWallet(wallet.keyIdBase64, wallet.contractId);
}
```



```javascript
import { clearUserWallet } from "./utils";

async function logout() {
    clearUserWallet()
    window.location.reload()
}
```

LOOK AT THAT CODE!

Super Simple Send Message Function



```
import { Client as SnapchainClient, networks } from "snapchain_sdk";
import { retrieveUserWallet } from "./utils";

const snapchain = new SnapchainClient({
    rpcUrl: import.meta.env.VITE_RPC_URL,
    contractId: networks.testnet.contractId,
    networkPassphrase: import.meta.env.VITE_NETWORK_PASSPHRASE,
});

async function sendMessage(message: string) {
    const { address, keyId } = retrieveUserWallet()

    let at = await snapchain.send({
        author: address,
        message: message,
    })
    at = await account.sign(at, { keyId: keyId })
    await server.send(at)
}
```

NOW GET TO IT!

Play with the demos

Smart Stellar Demo (site) (source)
React (source)

Snapchain Demo (site) (source)

Play with passkeys & AI

Smol will generate AI masterpieces!

https://smol.xyz

Farm some KALE

A homegrown Stellar memecoin

https://kalefarm.xyz

Start building!

Developer Documentation

Passkey Kit

Launchtube

DevContainer Template Repo

Hackathon Resources