Hello WebApp

Most interactions with the NEAR ecosystem can be grouped in 2 categories:

1. Interacting with aNEAR smart contract
2. .
3. Interacting with aNEAR component
4. .

In this guide we will show you how to quickly spin-up an application where users canlogin using their wallets and interact with bothcontracts andcomponents .

Furthermore, the application readily integrates aWeb3 wallet , allowing people to useMetamask to interact with multi-chain components.

Searching to integrate NEAR in your App? If you already have an application and want to integrate NEAR into it, we recommend you to first go through this guide and then check our documentation onintegrating NEAR to a frontend

Create NEAR App

If you already haveNode.js installed, simply run:

npx create-near-app@latest Use the interactive menu to set up:

1. A Near Gateway (Web App)
2. .
3. NextJs + React
4. .

Using pnpm While you can use our app with any package manager, we recommend you to skip the installation step and manually install the dependencies usingpnpm i . Once the folder is ready - and all dependencies installed - you can start the development server usingpnpm .

pnpm dev Visithttp://localhost:3000 in your browser to view the dApp. Note that since the dApp uses NextJS the app might take longer to load the pages on first visit.

The app is not starting? Make sure you are usingnode >= v18 , you can easily switch versions usingnvm use 18

Landing Page

Once the app starts you will see the landing page, rendering a navigation bar that allows users to login using their NEAR wallet, and two pathways:

Landing page of Hello NEAR Gateway

Go ahead and sign in with your NEAR account. If you don't have one, you can create one on the fly.

Under the Hood

Next.js uses a template system, where each page is a React component.

Our app's template is defined at./src/app/layout.js . It does two things:

1. Initializes awallet selector
2. , and stores it so other components can access it later.
3. Renders the navigation menu and the page's content.

templates/frontend/next-app/src/app/layout.js loading ... See full example on GitHub What it the wallet selector? The wallet selector is a component that allows users to select their preferred Near wallet to login. Our application implements auseInitWallet hook, that initializes a wallet selector and stores it so other components can access it later.

Navigation Bar & Login

The navigation bar implements buttons tologin andlogout users with their Near wallet.

The code for the navigation bar can be found at./src/app/navigation.js . The login and logout buttons are implemented by using thelogIn andlogOut methods from the wallet selector previously initialized:

templates/frontend/next-app/src/components/navigation.js loading ... See full example on GitHub

Interacting with NEAR

Now that you understand how the landing page works, we can move to theNear Integration page, which retrieves a greeting from thehello.near-examples.testnet contract.

View of theNear Integration page

Login if you haven't done it yet and you will see a simple form that allows you to store a greeting in the smart contract.

Under the Hood

Interactions with NEAR are done using theuseWallet hook to retrieve both theviewMethod andcallMethod methods and thesignedAccountId property from thewallet selector .

templates/frontend/next-app/src/app/hello-near/page.js loading ... See full example on GitHub On load, the firstuseEffect hook will call the contract'sget_greeting method and set thegreeting state to the result.

If the user is logged in, thesaveGreeting method will call the contract'sset_greeting method and set thegreeting state to the result.

Interacting with a Component

Now let's take a look at the Components page. Go ahead and click on the card titled Web3 Components at the bottom on the page. Once you do you'll be taken to the screen below:

The Near Components Page

If you're following along, you should already be logged into your NEAR account. If you aren't, go ahead and do so now.

You'll see that once you're logged in you are able to interact with the components that come included with the app.

Ethereum Components To interact with the Ethereum components (a Lido) you will need to have Metamask, or other web3 compatible wallets Ethereum Mainnet While the component's code is stored in theNear testnet , the Lido component is connected to theEthereum mainnet .

Under the Hood

The source code (located in./src/hello-components/page.js) shows us that the page is rendering components pulled from the SocialDB:

- page.js
- vm-components.js

templates/frontend/next-app/src/app/hello-components/page.js loading ... See full example on GitHub
templates/frontend/next-app/src/components/vm-component.js loading ... See full example on GitHub Particularly, theComponent in the main page are wrappers around theWidget component of the SocialVM.

Moving Forward

That's it for our quickstart tutorial. You have now seen a fully functional frontend that can talk with NEAR contracts and render Web3 components.

If you have any questions, do not hesitate in joining us orDiscord . We regularly host Office Hours, in which you can join our voice channel and ask questions.

Happy coding! Edit this page Last updatedonJan 31, 2024 bygagdiez Was this page helpful? Yes No

Previous What are Web3 Apps?Next Integrating Contracts