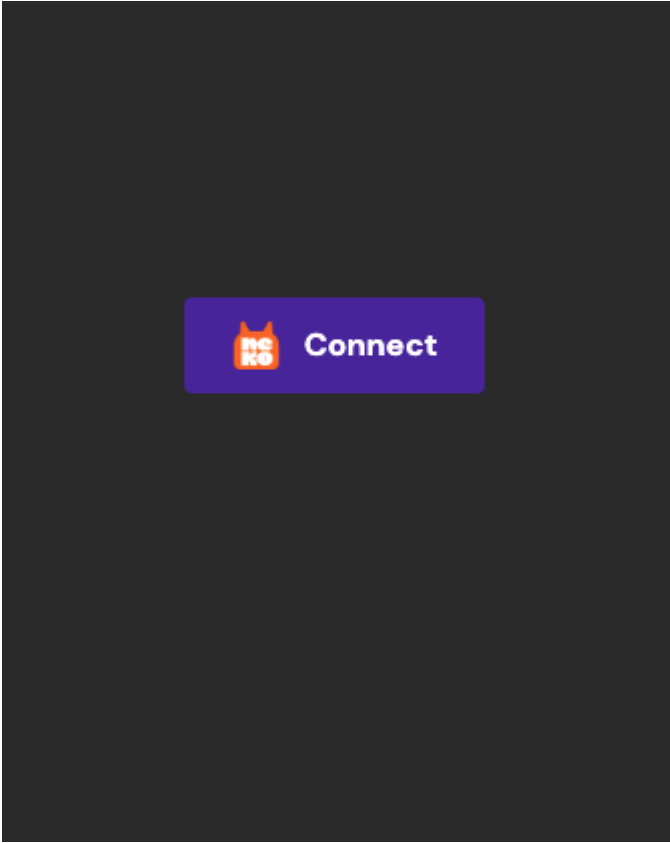
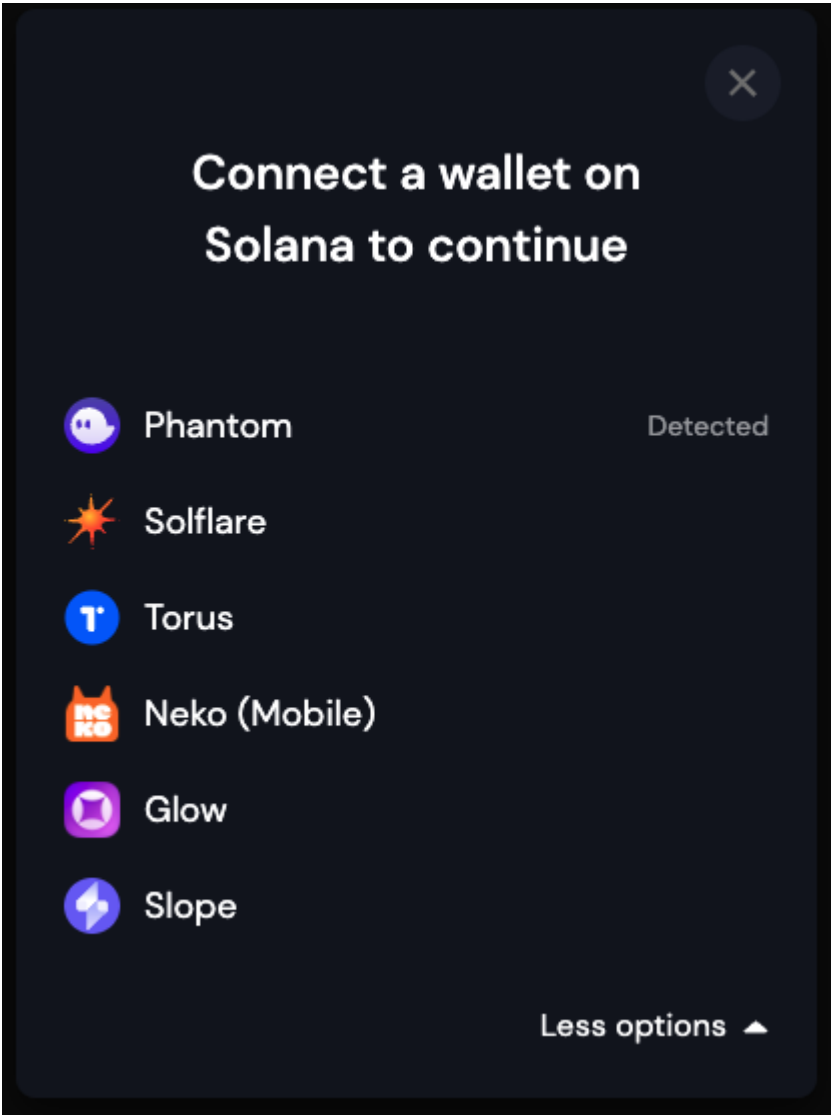


# docs-neko-wallet-adapter

---

Documents for Neko Wallet Adapter



- [Quick Setup \(using React UI\)](#)
  - [Install](#)
  - [Setup](#)
  - [Usage](#)
  - [Demo](#)
- [Build from Source](#)
- [React demo](#)

## Install

Install these dependencies:

```
npm install @solana/wallet-adapter-base \
  @solana/wallet-adapter-react \
  @solana/wallet-adapter-react-ui \
  @solana/web3.js \
  react
```

Install adapter for wallet

Install individual adapter for [Neko Wallet](#)

```
npm install @solana/wallet-adapter-neko
```

Or the [@solana/wallet-adapter-wallets](#) provided adapters for each wallet. You can install it for all adapters supporting now. [Neko Wallet](#) is also contain in this package.

```
npm install @solana/wallet-adapter-wallets
```

## Setup

```
import React, { FC, useMemo } from 'react';
import { ConnectionProvider, WalletProvider } from '@solana/wallet-
adapter-react';
import { WalletAdapterNetwork } from '@solana/wallet-adapter-base';
import { NekoWalletAdapter } from '@solana/wallet-adapter-neko'
// import {
//   NekoWalletAdapter,
//   CoinbaseWalletAdapter,
//   PhantomWalletAdapter,
// } from '@solana/wallet-adapter-wallets';

import {
  WalletModalProvider,
  WalletDisconnectButton,
```

```

    WalletMultiButton
  } from '@solana/wallet-adapter-react-ui';
  import { clusterApiUrl } from '@solana/web3.js';

  // Default styles that can be overridden by your app
  require('@solana/wallet-adapter-react-ui/styles.css');

  export const Wallet: FC = () => {
    // The network can be set to 'devnet', 'testnet', or 'mainnet-beta'.
    const network = WalletAdapterNetwork.Devnet;

    // You can also provide a custom RPC endpoint.
    const endpoint = useMemo(() => clusterApiUrl(network), [network]);

    // @solana/wallet-adapter-neko includes the adapter for neko wallet
    // Only the wallets you configure here will be compiled into your
    application, and only the dependencies
    // of wallets that your users connect to will be loaded.
    const wallets = useMemo(
      () => [
        new NekoWalletAdapter()
      ],
      [network]
    );

    return (
      <ConnectionProvider endpoint={endpoint}>
        <WalletProvider wallets={wallets} autoConnect>
          <WalletModalProvider>
            <WalletMultiButton />
            <WalletDisconnectButton />
            { /* Your app's components go here, nested within the
context providers. */ }
          </WalletModalProvider>
        </WalletProvider>
      </ConnectionProvider>
    );
  };
};

```

## Usage

```

import { WalletNotConnectedError } from '@solana/wallet-adapter-base';
import { useConnection, useWallet } from '@solana/wallet-adapter-react';
import { Keypair, SystemProgram, Transaction } from '@solana/web3.js';
import React, { FC, useCallback } from 'react';

export const SendOneLamportToRandomAddress: FC = () => {
  const { connection } = useConnection();
  const { publicKey, sendTransaction } = useWallet();

  const onClick = useCallback(async () => {

```

```
    if (!publicKey) throw new WalletNotConnectedError();

    const transaction = new Transaction().add(
      SystemProgram.transfer({
        fromPubkey: publicKey,
        toPubkey: Keypair.generate().publicKey,
        lamports: 1,
      })
    );

    const signature = await sendTransaction(transaction, connection);

    await connection.confirmTransaction(signature, 'processed');
  }, [publicKey, sendTransaction, connection]);

  return (
    <button onClick={onClick} disabled={!publicKey}>
      Send 1 lamport to a random address!
    </button>
  );
};
```

## Run demo

1. Clone the project:

```
https://github.com/neko-global/docs-neko-wallet-adapter
```

1. Install

```
cd demo
npm install
```

2. Run

```
npm run start
```