

Wormhole Kit

The Developer's Guide to the Galaxy



Don't Panic

Just three steps are needed.

Count: 3

Installation

```
npm install @zktx.io/wormhole-kit
```

bash

```
yarn add @zktx.io/wormhole-kit
```

bash

Count: 2

Instantiate a **WhProvider** component

```
const root = createRoot(document.getElementById('root') as HTMLElement);
root.render(
  <StrictMode>
    <WhProvider
      network="Testnet"
      chains={['Sui', 'Sepolia', 'Solana', 'Aptos', 'Celo', 'Polygon']}
      mode="dark"
    >
      <QueryClientProvider client={queryClient}>
        <SuiClientProvider
          defaultNetwork="testnet"
          networks={{ testnet: { url: getFullnodeUrl('testnet') } }}
        >
          <WalletProvider autoConnect>
            <App />
          </WalletProvider>
        </SuiClientProvider>
      </QueryClientProvider>
    </WhProvider>
  </StrictMode>,
);
```

Count: 1

Use **WhTransferModal** or **WhRedeemModal**
for use **Wormhole**

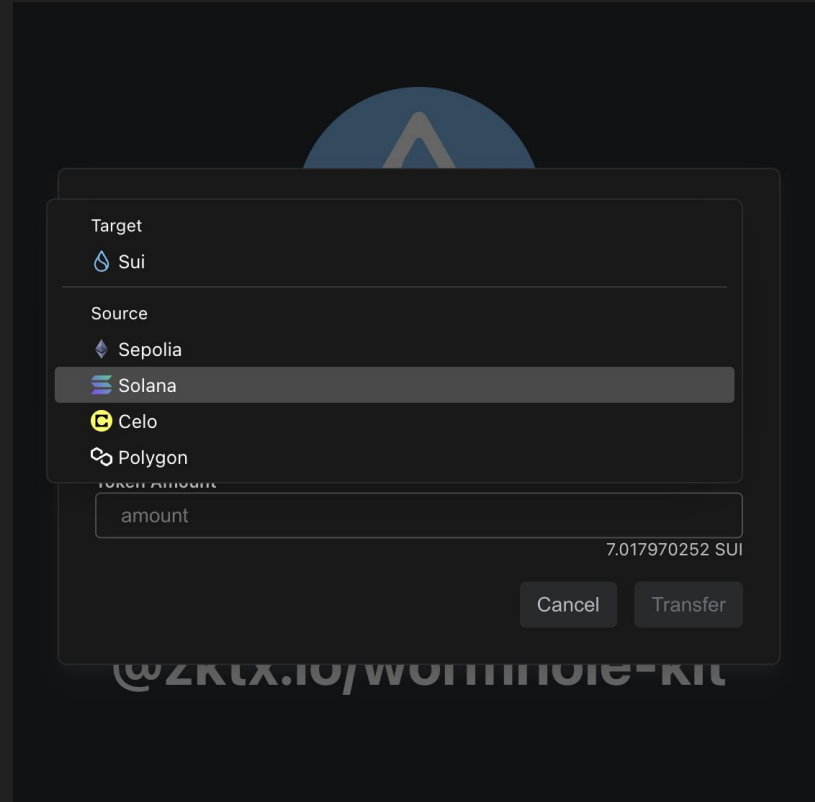
```
const App = () => {
  const [address, setAddress] = useState<string | undefined>(undefined);
  const [open, setOpen] = useState<boolean>(false);

  const handleUnsignedTx = async (unsignedTx: any): Promise<void> => {
    // To execute the transaction,
    // send the unsigned transaction to the wallet.
  };

  return (
    <>
      <button
        onClick={() => {setOpen(true);}}
      >
        Transfer
      </button>
      <WhTransferModal
        chain="Sui"
        address={address}
        handleUnsignedTx={handleUnsignedTx}
        open={open}
        setOpen={setOpen}
      />
    </>
  )
}
```

Launch

You can now freely travel with your assets through the **Wormhole**.



Boost up

Pairing it with '@mysten/dapp-kit' takes your development experience to the next level.

```
import {
  ConnectButton,
  useCurrentAccount,
  useSignAndExecuteTransactionBlock,
} from '@mysten/dapp-kit';
import { WhRedeemModal, WhTransferModal } from '@zktx.io/wormhole-kit';

function App() {

  const { mutate: signAndExecuteTransactionBlock } =
    useSignAndExecuteTransactionBlock();
  const account = useCurrentAccount();

  const handleUnsignedTx = (unsignedTx: any) => {
    try {
      signAndExecuteTransactionBlock(
        {
          transactionBlock: unsignedTx,
        },
        {
          onError: (err) => {
            console.error(err);
          },
          onSuccess: (data) => {
            console.log(data);
          },
        },
      );
    } catch (error) {
      console.log(error);
    }
  };
};
```

Boost your dapp now with the Wormhole kit.

