

Ex1 - Integrating Compound

Preliminaries - Getting into DeFi

1. [Install](#) Metamask chrome extension, make sure you are installing an extension with many downloads. ([Be careful!](#) :))
2. Let's put some fake money in the wallet. Get some Goerli ETH tokens from here: <https://goerlifaucet.com/> (**Note:** You might need to approve test networks in that wallet, so you put some fake tokens)
3. Go to [Compound's](#) website, and lend ("supply") some ETH, you should receive cETH against that, make sure you see the cETH in your wallet.
4. Congrats on your first DeFi transaction! Go and see it in <https://etherscan.io/>
5. The important fields to us are: **Input Data, Value, From, To**. This transaction is in general quite easy to understand, but make sure you do.

Getting our hands dirty - Exercise - Integrating Compound to our own UI

We want to interact with Compound, and make transactions, **but not use "Compound's frontend"**, we want to be smart contract native.

We want to create an html page with the following capabilities:

Compound

Token Symbol:

Value:

DEPOSIT

WITHDRAW

Compound

GET INFO

As can be seen, we can do the following:

"Write API" - Lend / withdraw from compound.

"Read API" - Get current info about the current position in Compound.

Guidance:

1. Begin with understanding how to **lend ETH**, from your own html page.
2. Then get the current **ETH position** in Compound.
3. **Withdraw ETH**.

If you have time, you can support 1 more token. Don't need more than that.

You'll submit the following files (github repo will do):

- Main.html / index.html
- Main.js
- You can obviously add any related files that these files up there depend on.

Good luck!

=====

Helpful materials:

1. <https://docs.metamask.io/guide/create-dapp.html#basic-action-part-1> - **Use that!**
2. Read about **eth_sendTransaction**.
3. <https://compound.finance/documents/Compound.Whitepaper.pdf> - Only if that doesn't confuse you.

Helpful code snippets, think where you should embed it:

(1)

```
// Send transaction
const txHash = await ethereum.request({
  method: 'eth_sendTransaction',
  params: [txParams],
})
```

(2)

```
orig = window.ethereum.request;
window.ethereum.request = async function(e) {
  console.log(e);
  return orig(e).then((result) => {
    console.log(result);
    return result;
  });
}
```