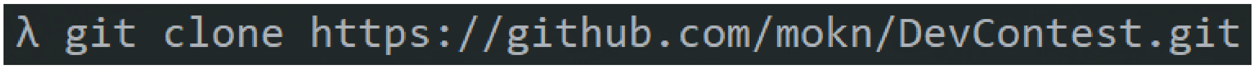
DevContest Startup Guide

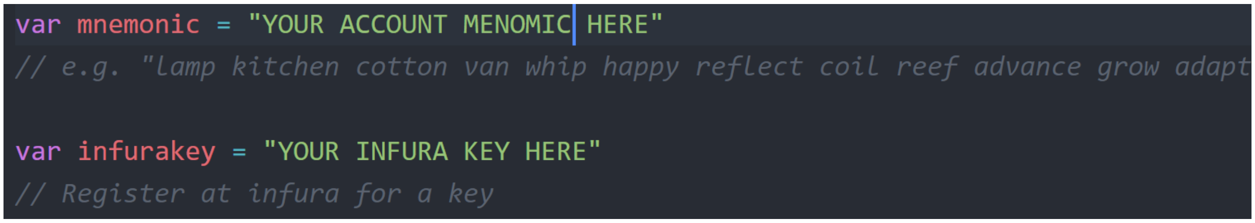
Navigate to https://github.com/mokn/DevContest and clone the repository locally.



```
λ git clone https://github.com/mokn/DevContest.git
```

Ensure all dependencies are installed beforehand (like truffle, node, npm).

1. Navigate into DevContest and view the file called **truffle-config.js**. This file contains the networks available and the infura configuration for the network. This information needs to be replaced for deployment of the contracts from an appropriate account.



```
var mnemonic = "YOUR ACCOUNT MENOMIC HERE"
// e.g. "lamp kitchen cotton van whip happy reflect coil reef advance grow adapt

var infurakey = "YOUR INFURA KEY HERE"
// Register at infura for a key
```

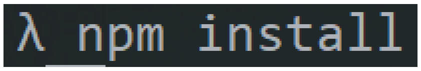
CAUTION: Do not publicly share this information using an important account or risk losing the account.

Truffle-config.js will require a mnemonic to login to metamask and an infura key to deploy the contract to the network using truffle. This mnemonic will correspond to the owner of the contract, and will need to have some ether to be able to deploy to the corresponding network.

2. Next, lets take a look at **migrations\2_deploy_contracts.js**. This file sets the beginning and ending block number respectively as the final two arguments of deployer.deploy().

Once these values are set you are ready to deploy to the contract to the network of your choosing.

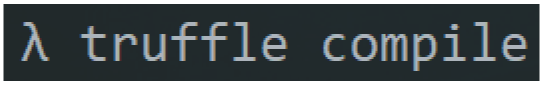
3. To do so navigate to the **src** folder and run the following:



```
λ npm install
```

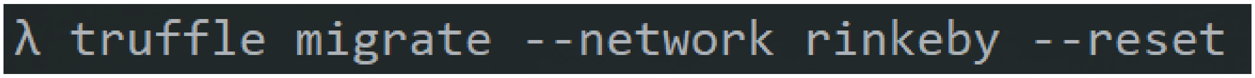
You can use yarn as well.

4. Once that is successful, go back up a folder to **DevContest** and run:



```
λ truffle compile
```

And finally:

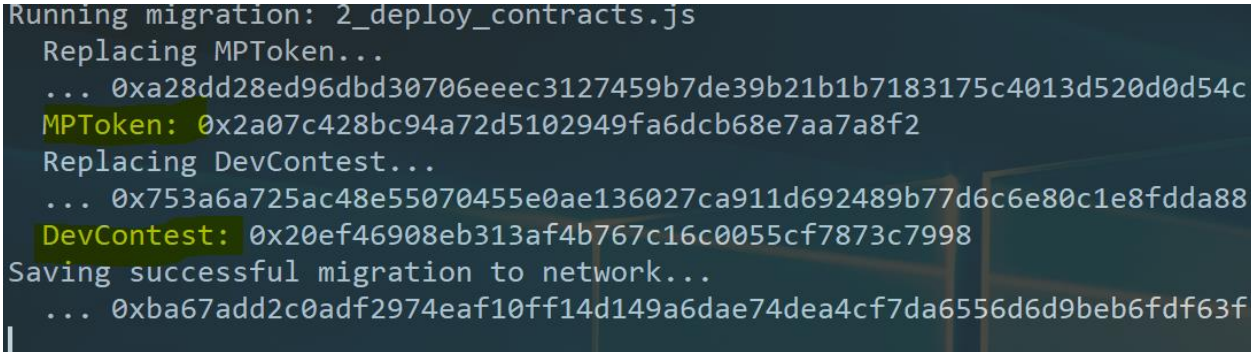


```
λ truffle migrate --network rinkeby --reset
```

In this case we deployed the contract to the rinkeby network.

Since this is all separate from the frontend these files don't necessarily need to be in your project, as the frontend will access the contracts directly from the blockchain.

Take note of the following addresses next to the highlighted contract names:

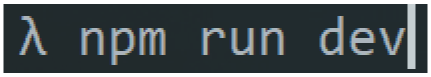


```
Running migration: 2_deploy_contracts.js
  Replacing MPToken...
  ... 0xa28dd28ed96dbd30706eeec3127459b7de39b21b1b7183175c4013d520d0d54c
  MPToken: 0x2a07c428bc94a72d5102949fa6dcb68e7aa7a8f2
  Replacing DevContest...
  ... 0x753a6a725ac48e55070455e0ae136027ca911d692489b77d6c6e80c1e8fdda88
  DevContest: 0x20ef46908eb313af4b767c16c0055cf7873c7998
Saving successful migration to network...
  ... 0xba67add2c0adf2974eaf10ff14d149a6dae74dea4cf7da6556d6d9beb6fdf63f
```

5. These will need to be set in **src/js/GlobalVars.js** where CONTEST_ADDRESS corresponds to DevContest and TOKEN_ADDRESS to MPToken.

Finally, after setting the addresses the UI is ready to work with the contract.

6. Check it out by running it on your localhost via (within the **src** folder):



```
λ npm run dev
```

Log into metamask from the account that deployed the contract to see the owner view from the UI. Any other account with MPToken will now also be able to interact and participate in the contest.