

How to Interact with the ERC-20 Smart Contract

No comments



In this tutorial we are going to run the ERC-20 smart contract deployment on the Rinkeby chain and see the actual token created in our Metamask wallet. After you create and deploy your ERC-20 smart contract, you will be able to see that this token is added on the Rinkeby Etherscan, and by adding the address and specifications of the token, you can add it to your wallet and also you will be able to do the same on Uniswap afterward to be able to swap your token with other tokens.

ERC-20 Smart Contract

In our last related article on [how to create an ERC-20 token](#), we learned how to create a token with the ERC-20 standard. Moreover, we showed how to run it via Ganache-CLI. In this tutorial, we are going to run the very same deployment on the Rinkeby network, and see the actual token created in our Metamask wallet. After you create and deploy your ERC-20 smart contract, you will be able to see that this token is added to the Rinkeby Etherscan. Then, by adding the address and specifications of the token, you can add it to your wallet. Also, you will be able to do the same on Uniswap afterward to be able to swap your token with other tokens (notice that swapping your token is only possible on

Mainnet but not on the Rinkeby test network).

Deploying on Rinkeby Testnet

To begin with the deployment of our ERC-20 token, type the following command in the terminal:

```
brownie run scripts/1_deploy_token.py --network rinkeby
```

Result:

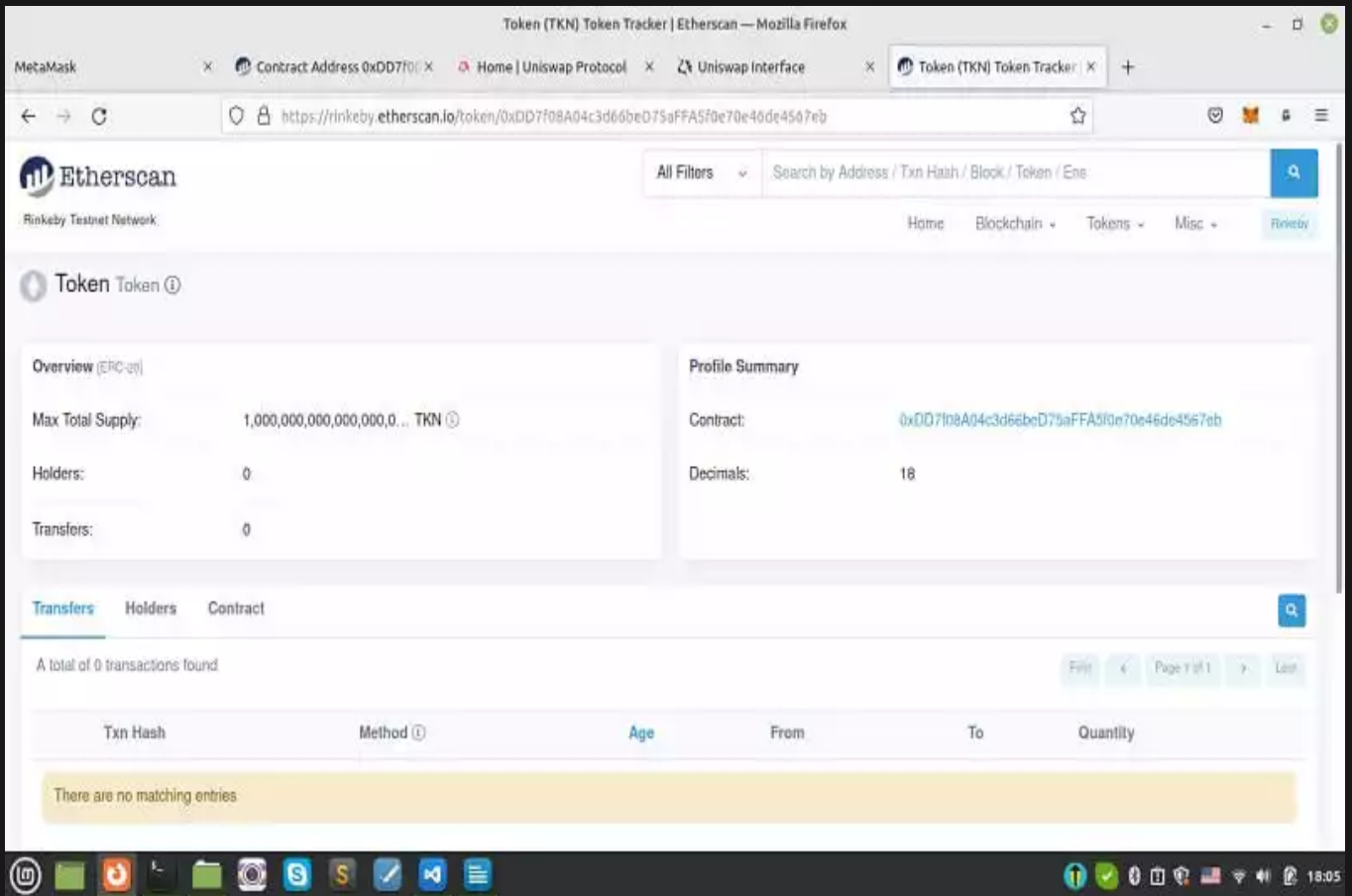
```
Brownie v1.18.1 - Python development framework for EthereumCompiling
contracts... Solc version: 0.6.12 Optimizer: Enabled Runs: 200 EVM
Version: Istanbul Generating build data... - TokenERC20 -
tokenRecipientCompiling contracts... Solc version: 0.8.13 Optimizer:
Enabled Runs: 200 EVM Version: Istanbul Generating build data... -
OpenZeppelin/openzeppelin-contracts@4.3.2/ERC20 -
OpenZeppelin/openzeppelin-contracts@4.3.2/IERC20 -
OpenZeppelin/openzeppelin-contracts@4.3.2/IERC20Metadata -
OpenZeppelin/openzeppelin-contracts@4.3.2/Context -
OurTokenMytokenProject is the active project.Running
'scripts/1_deploy_token.py::main'... Transaction sent:
0xa85a8d1595d290657f726f1533d105c2d2b6604e051409dc0c2e4d9a74bf547e Gas
price: 1.000000019 gwei Gas limit: 745852 Nonce: 62
TokenERC20.constructor confirmed Block: 10468341 Gas used: 678048
(90.91%) TokenERC20 deployed at:
0xDD7f08A04c3d66beD75aFFA5f0e70e46de4567eb
```

Finding the token on Rinkeby Etherscan:

By copying and pasting the `0xDD7f08A04c3d66beD75aFFA5f0e70e46de4567eb` into the Rinkeby Etherscan, you will be able to see that we have indeed created a token called Token:

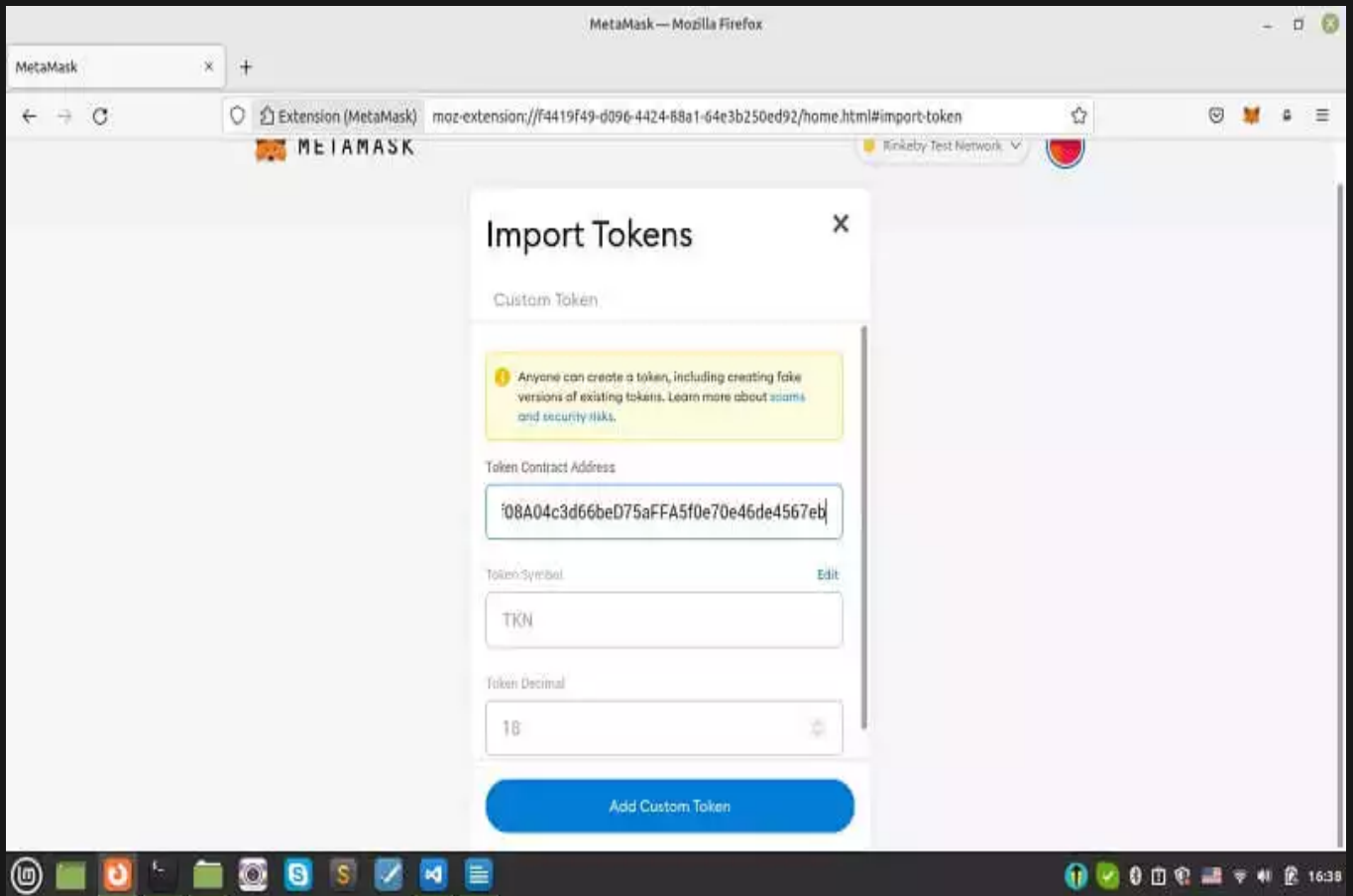
Adding the token to the Metamask:

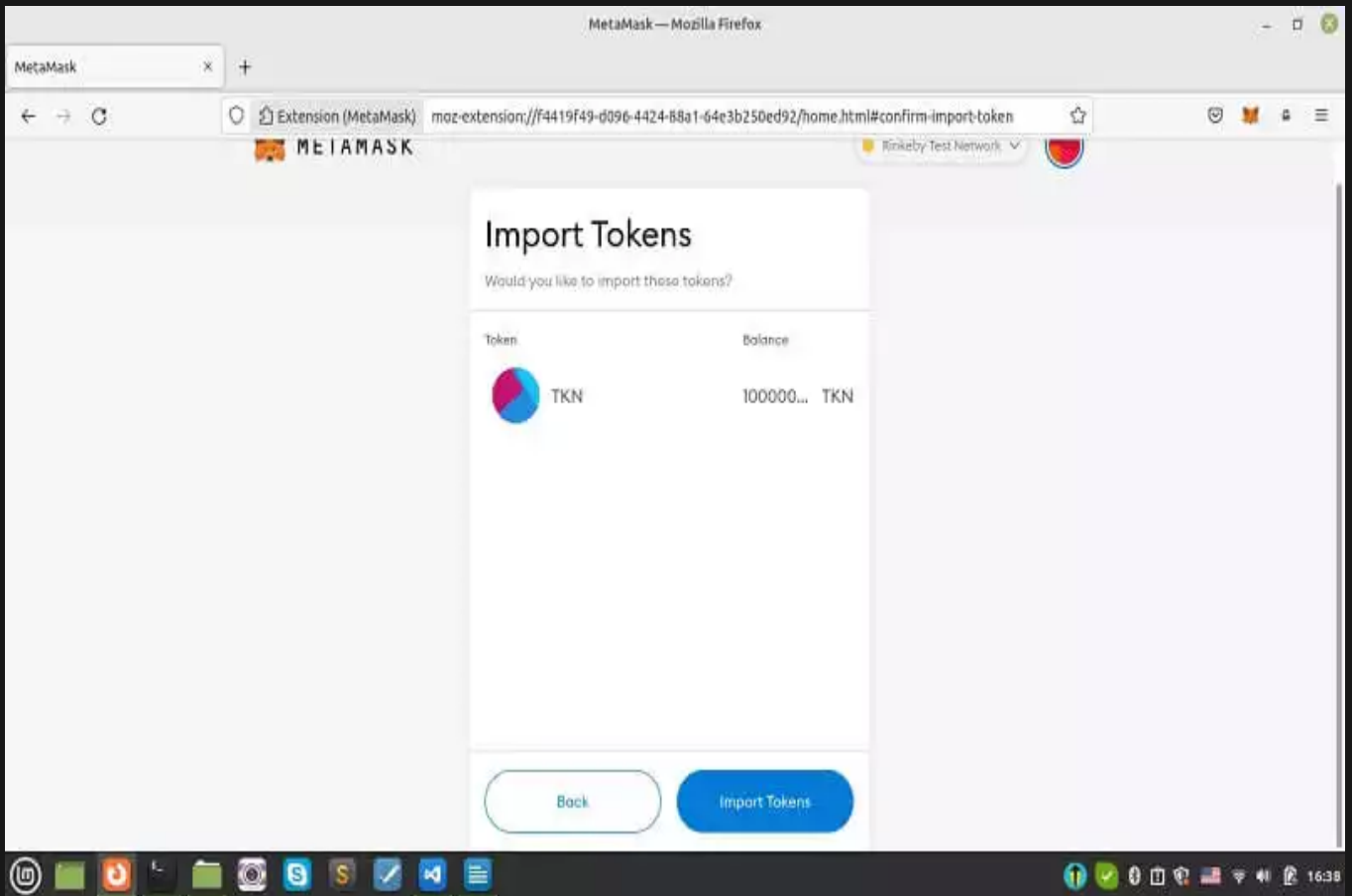
Now, let's head over to our Metamask, and in the Rinkeby network, import our new token by entering the address of the contract deployment:



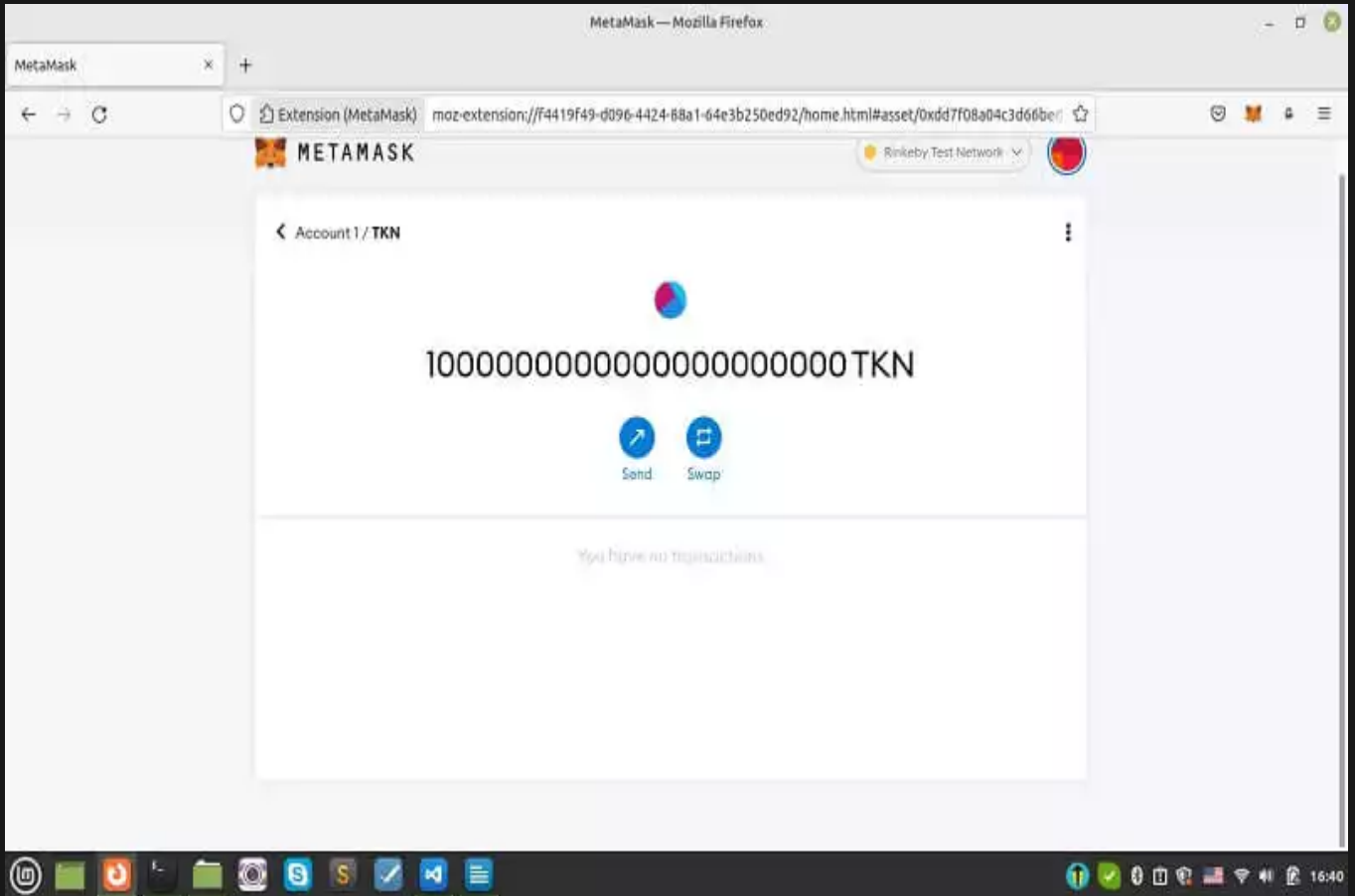
Adding the token to the Metamask:

Now, let's head over to our Metamask, and in the Rinkeby network, import our new token by entering the address of the contract deployment:



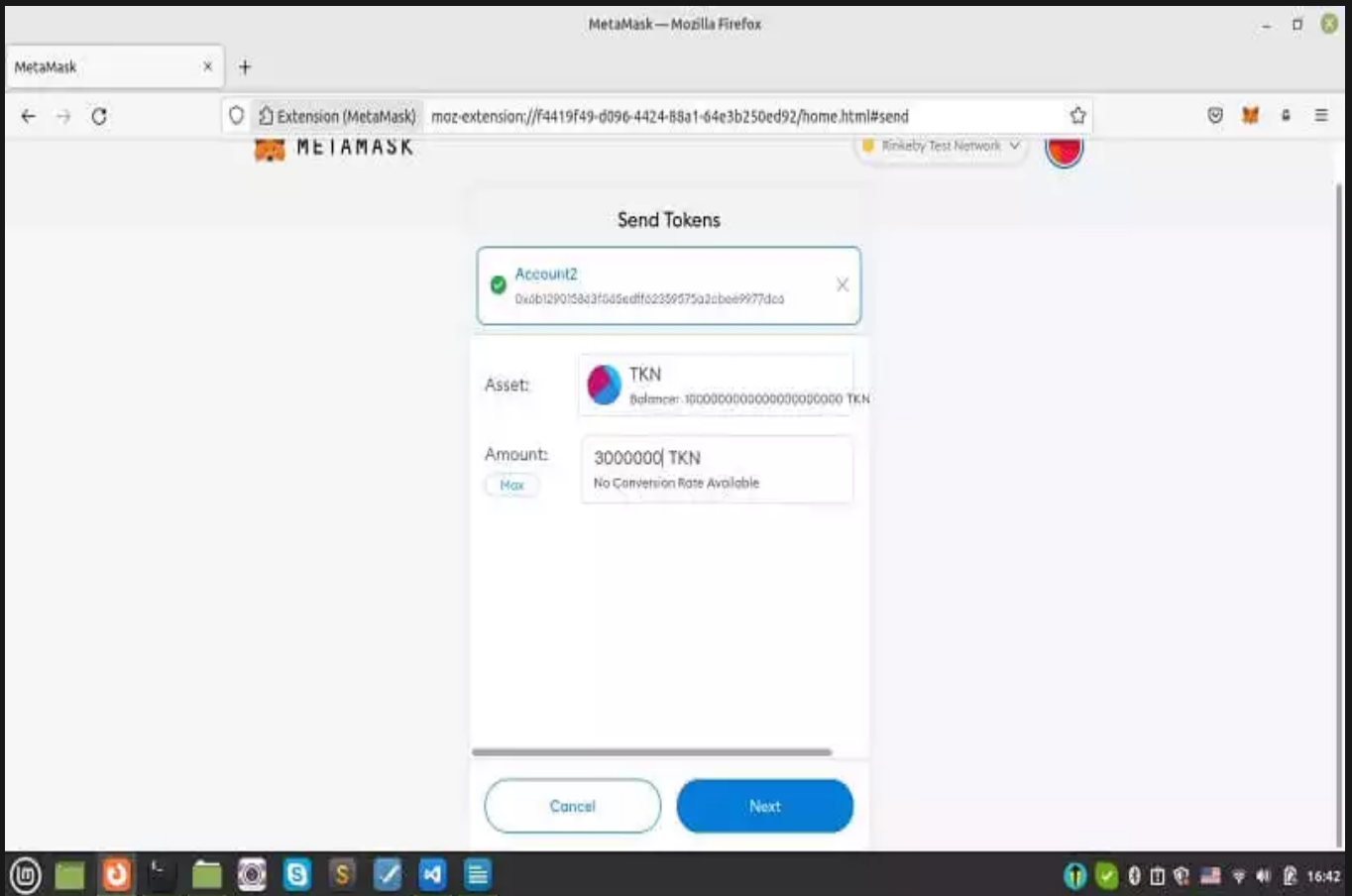


And you will see that our initial supply which was 1000 has been multiplied by 10 to power our decimals (which was 18). As a result, you can see 21 zeros after one as the total supply (All of which are in our account).



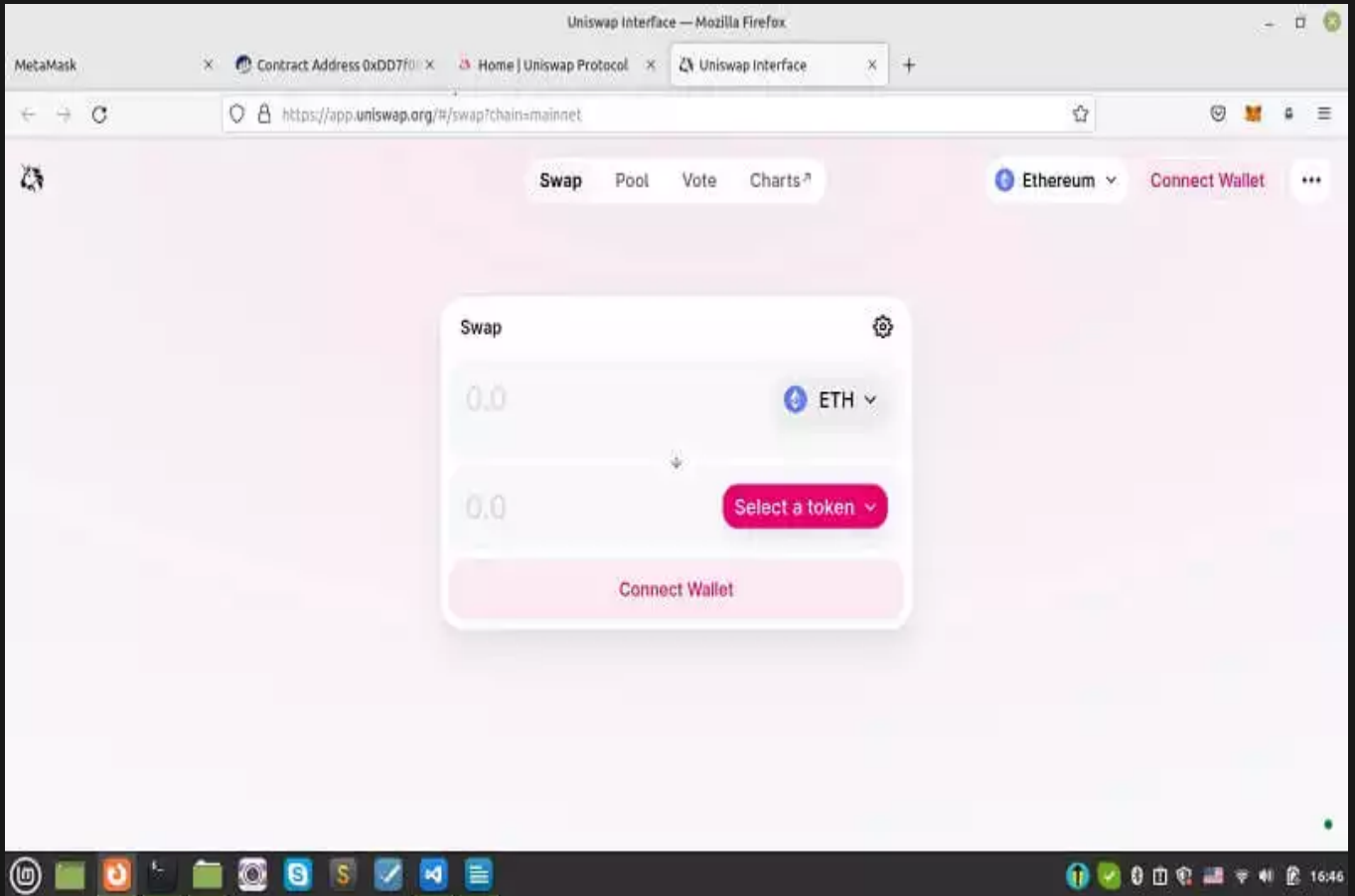
Transferring the tokens:

Now, we can send these tokens to other accounts. For example, here we transfer our TKN token to account 2.

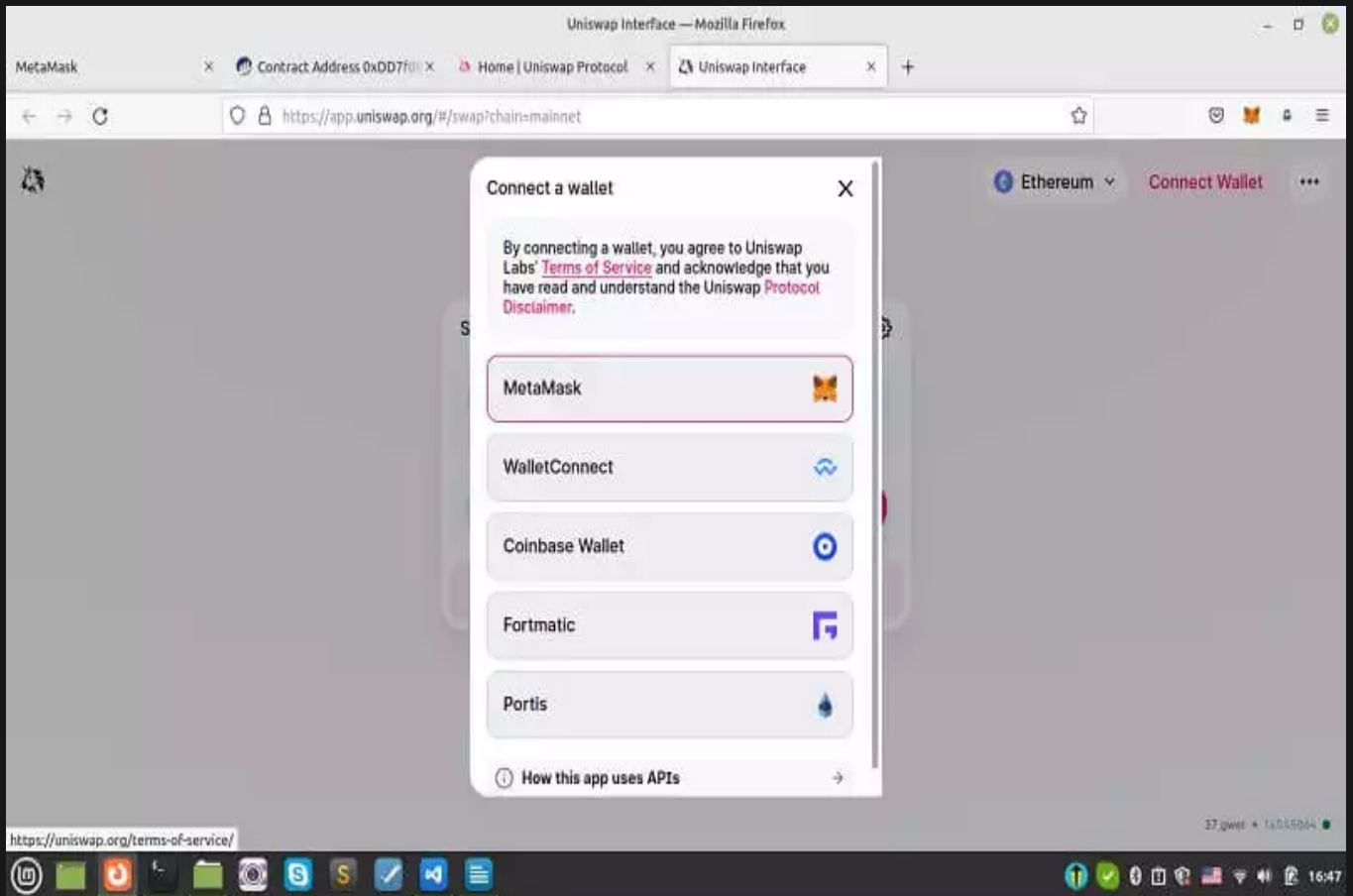


Importing the token in Uniswap:

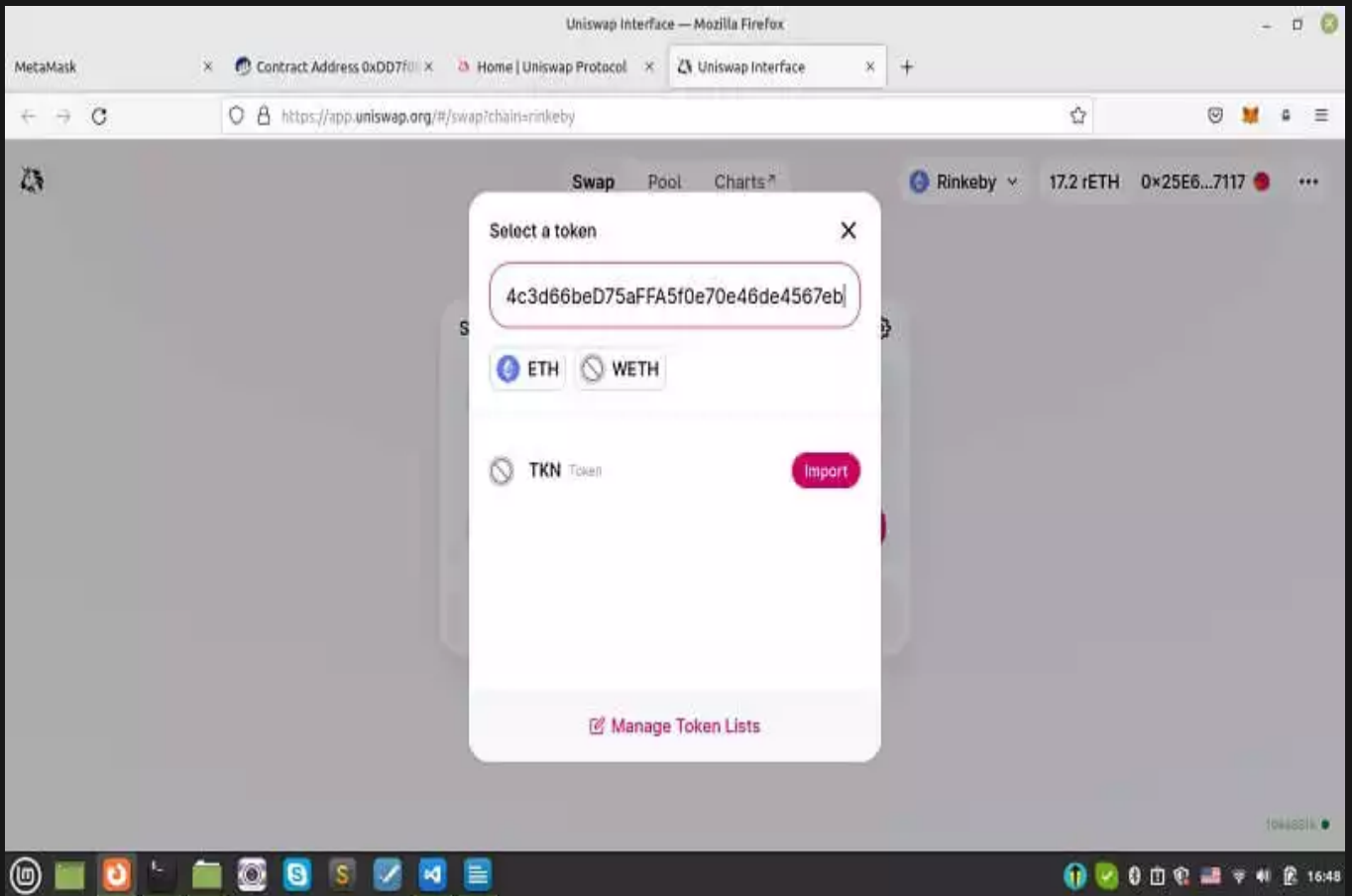
We can also go to Uniswap and see if we can add our token to it.



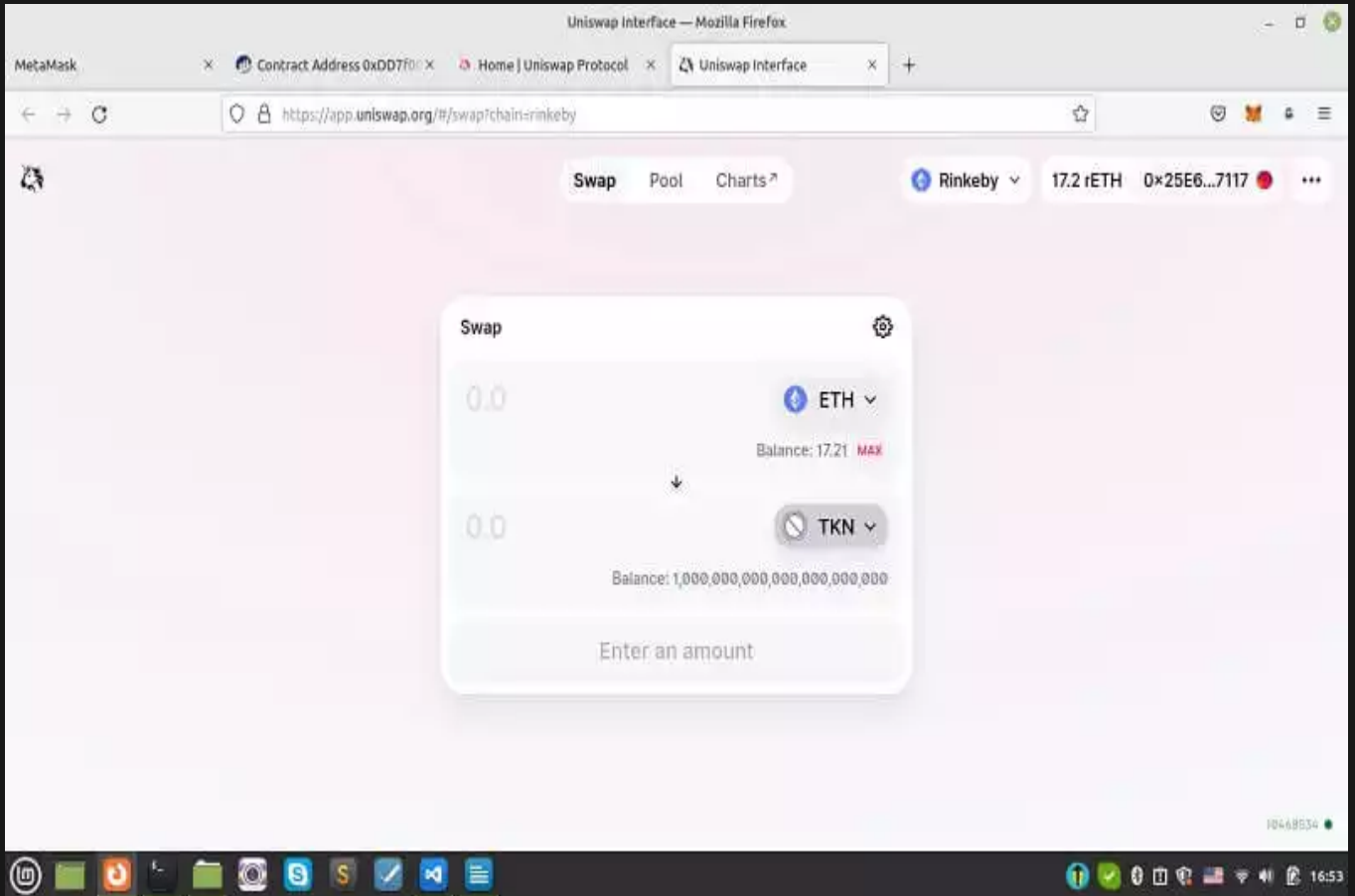
let's first connect to our Metamask wallet:



Then, press the **Select A Token** button. And in order to add your token in Uniswap, you should first enter your address in the box and you should be able to find the TKN token. Press import:



And we have our token in Uniswap!



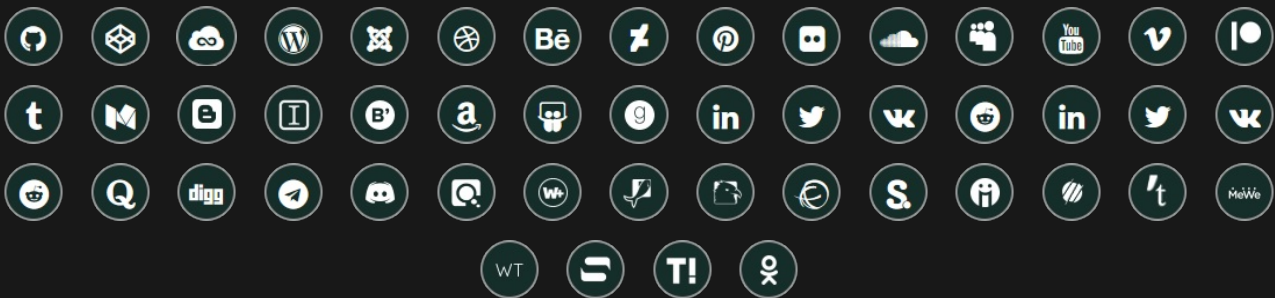
Last Thought on ERC-20 Smart Contract

In this article, we have managed to run the deployment of the ERC-20 smart contract on the Rinkeby chain test network. Then, we copied the address of the deployed contract in the Rinkeby Etherscan and found our created token there. Afterward, we entered the specifications of the token in the add token section of the Metamask wallet and added the created token to it. Finally, we imported our token into the Uniswap Rinkeby Testnet.

Join Arashtad Community

Follow Arashtad on Social Media

We provide variety of content, products, services, tools, tutorials, etc. Each social profile according to its features and purpose can cover only one or few parts of our updates. We can not upload our videos on SoundCloud or provide our eBooks on Youtube. So, for not missing any high quality original content that we provide on various social networks, make sure you follow us on as many social networks as you're active in. You can find out Arashtad's profiles on different social media services.



Get Even Closer!

Did you know that only one universal Arashtad account makes you able to log into all Arashtad network at once? Creating an Arashtad account is free. Why not to try it? Also, we have regular updates on our newsletter and feed entries. Use all these beneficial free features to get more involved with the community and enjoy the many products, services, tools, tutorials, etc. that we provide frequently.

[SIGN UP](#)[NEWSLETTER](#)[RSS FEED](#)