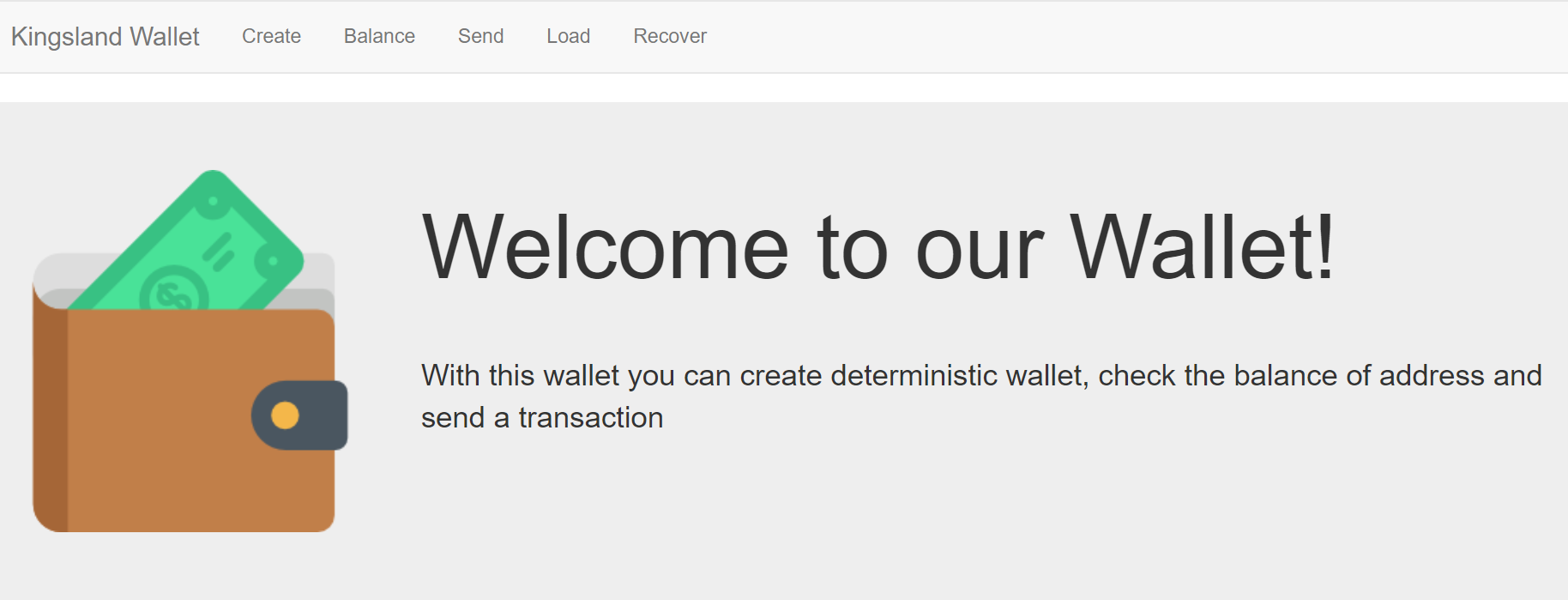
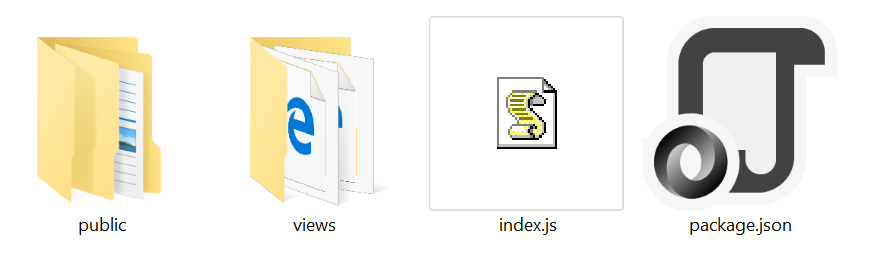
# Exercises: Create Ethereum Wallet with JS and Ethers

In this Exercise we are going to be learning how to create **Ethereum** **wallet** with **JavaScript** and **ethers.js** libraries. During the exercise you will install several **npm** packages, write the source code for different functionalities and in the end, you will **send** and **receive** ether coins with your wallet.



## Download the Code Template from our Resources

Extract the files from the provided Resources and you should have something like this:



Index.js holds all wallet endpoints and Todo comments that you need to implement.

## Install Packages

We need to install several packages for this exercise:

* **express** – Javascript web framework.
* **ejs** – Template engine.
* **ethers** – Library for interacting with the Ethereum blockchain.

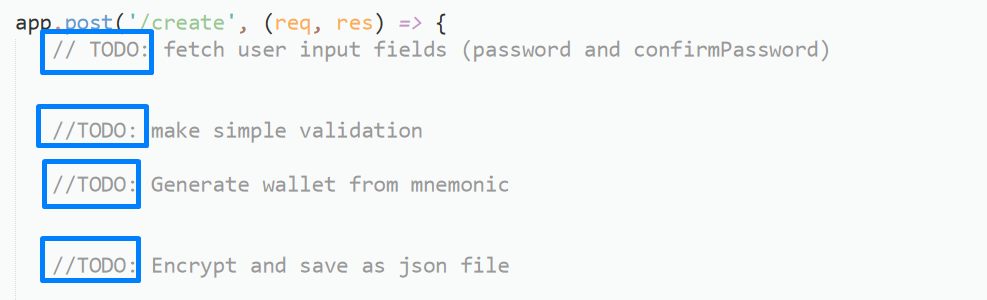
You can install them with:

npm install

## Start Implementing Todo’s

We have 5 endpoints **create, load, send, balance** and **recover** every endpoint hasa **Todo** and you need to write there.

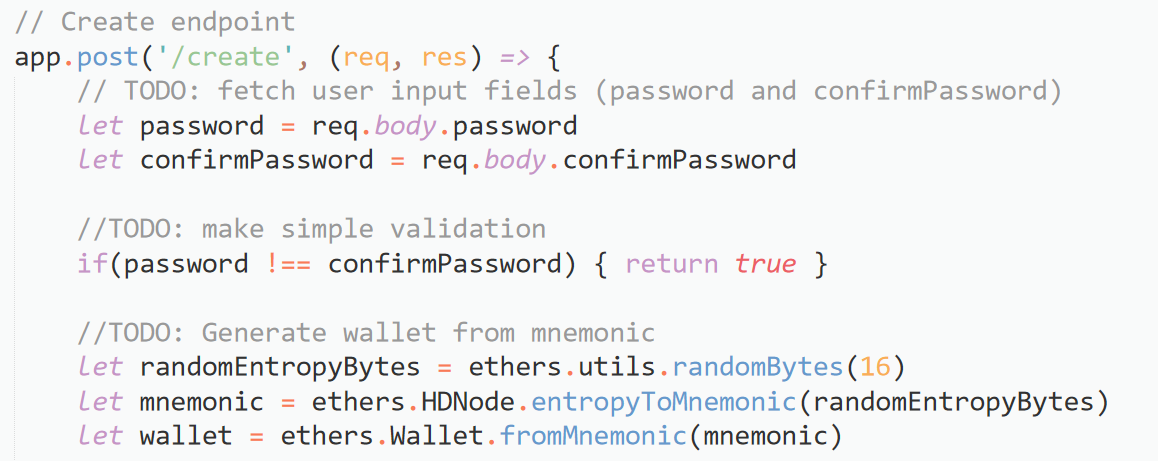
Example:



1. Add Constants

Here you only need to fill the missing information in the constant network which we will use in the exercise. In this case it is **ropsten** Testnet with **infura.io**. We will use another **constant** for the working directory where we will store **encrypted** wallet files. 

1. Implement create 1ich will generate and save wallet in our wallets directory.





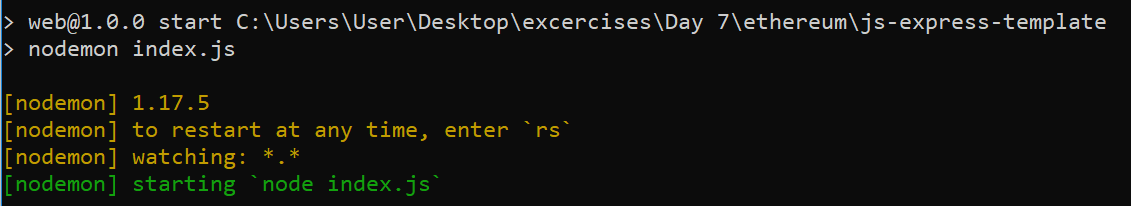


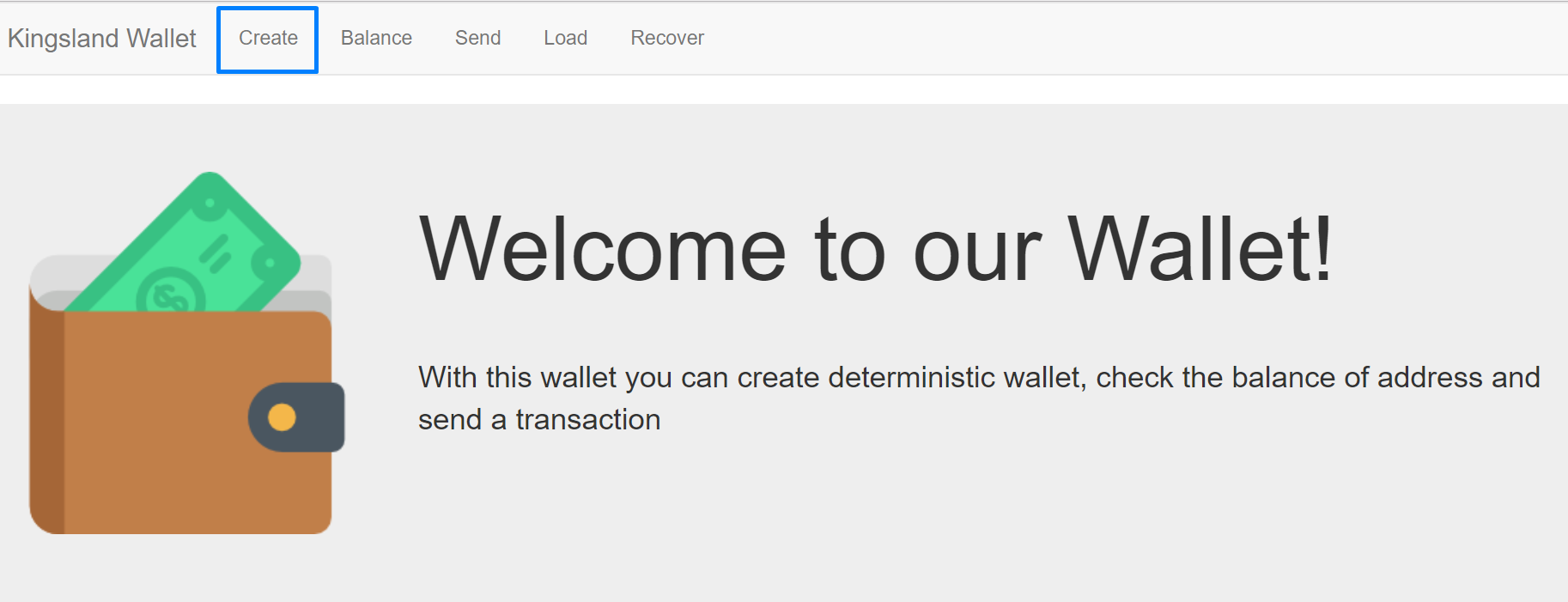
### Test create wallet method

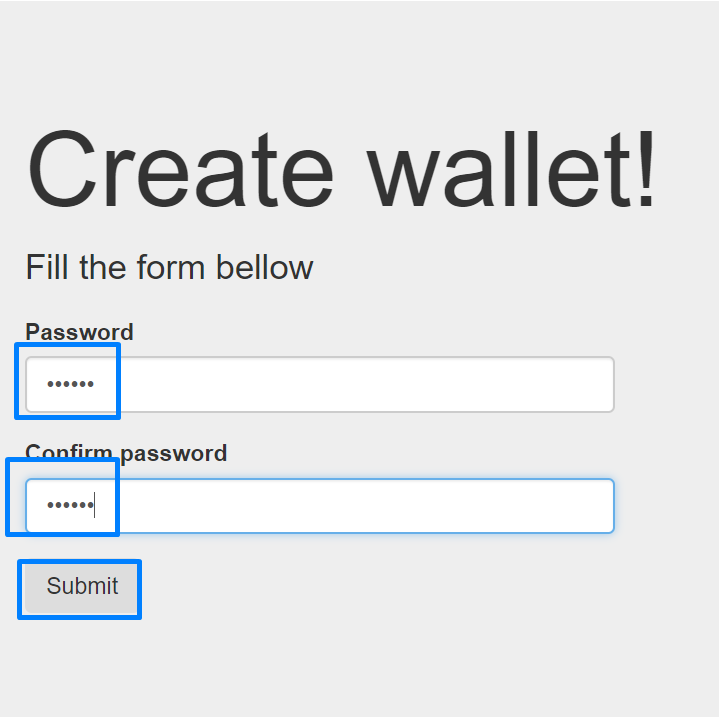
To start the project, write in cmd:

npm start

Output:



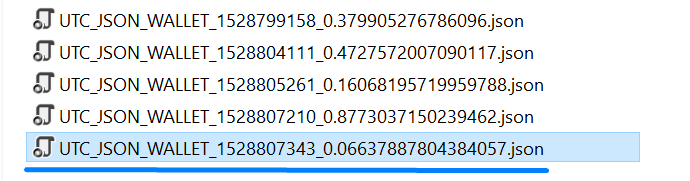
Now our application is running on [localhost:300](http://localhost:3000)0  
Let’s make a new wallet. Fill the form with password and confirm your password.



You will receive your mnemonic and the name of the encrypted wallet.



If we go to our **wallet directory** we will find a file for your wallet.



1. Implement Load Wallet Endpoint

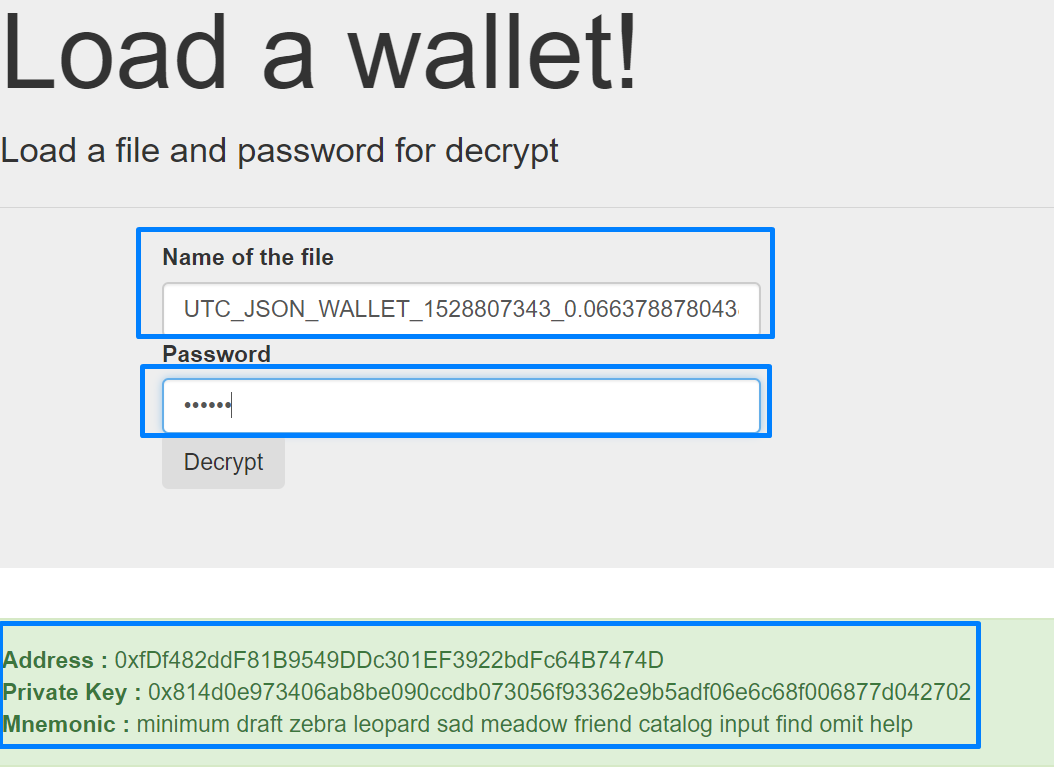
Here is the load endpoint which will read a file and **decrypt** it.





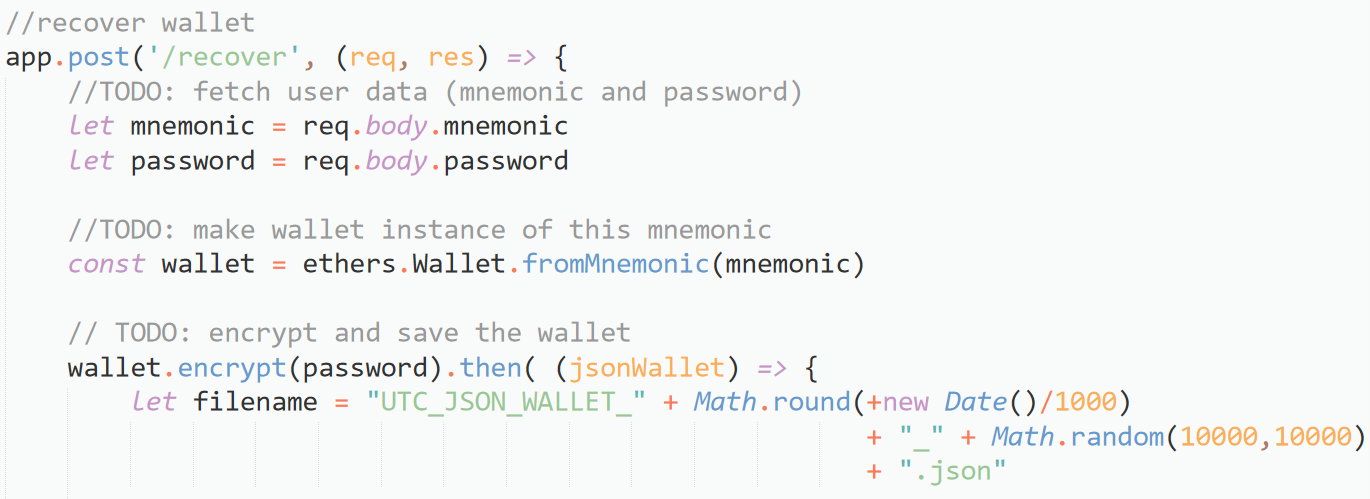
### Test load wallet method

Go to /**load** endpoint, fill your wallet filename and password and submit the form.



1. Implement Recover Wallet

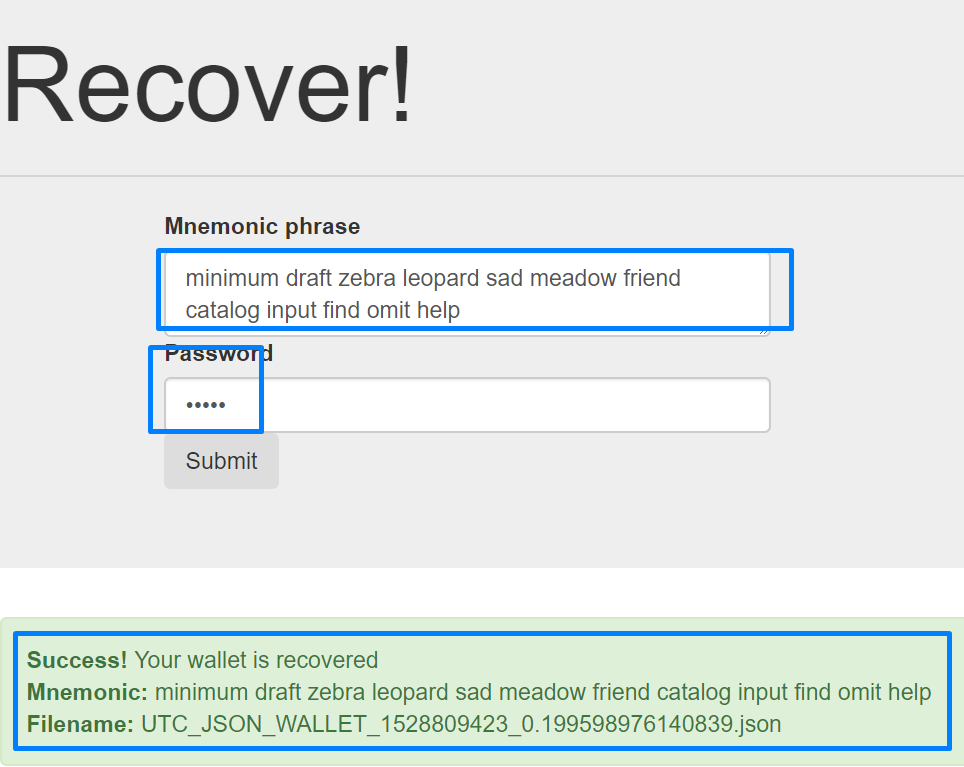
With this method the users will generate new **json** file with their **mnemonic** phrase





### Test recover wallet

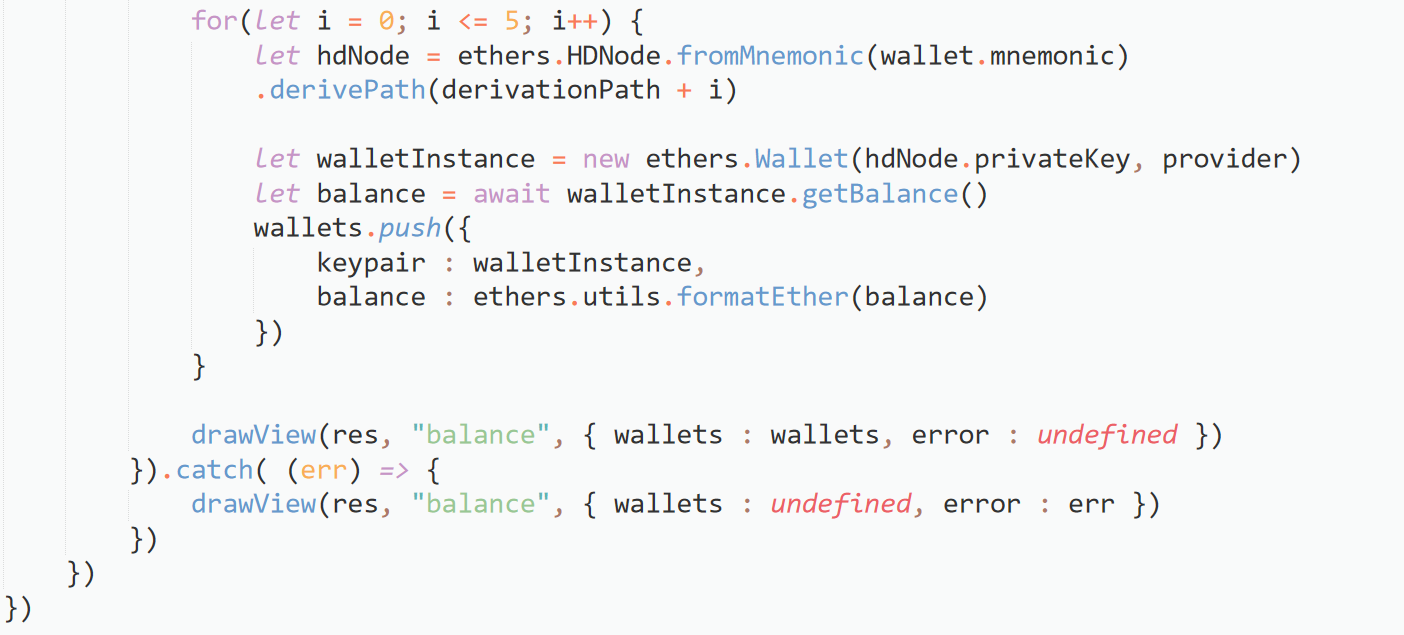
Now let’s assume that we want to use our wallet on the new computer. There isn’t exist saved json file and we must recover our wallet from mnemonic phrase. Do you remember your mnemonic words? Did you hide them in the save place? Now take them, you will need to use these words. Go to /**recover** endpoint. Then write the **words** and type the **new password** and press [**Submit**]. Wallet will be successfully recovered.



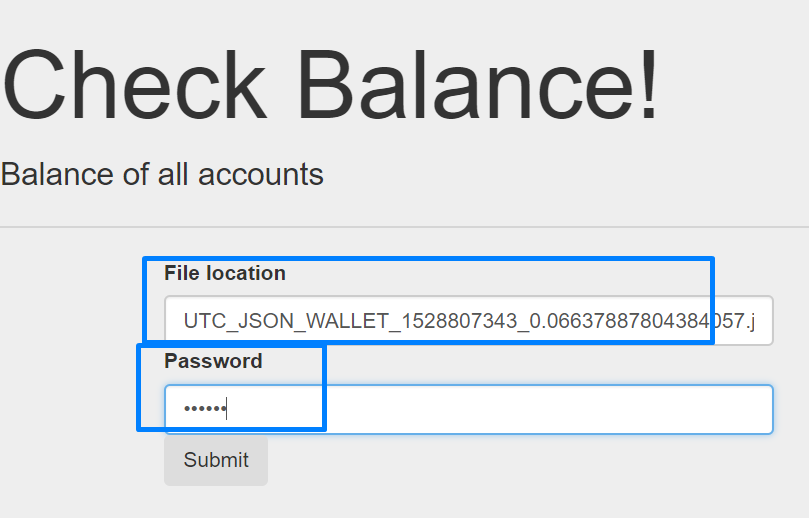
1. Implement Balance Endpoint

In this method we will derive 5 wallets and we will get their balance.

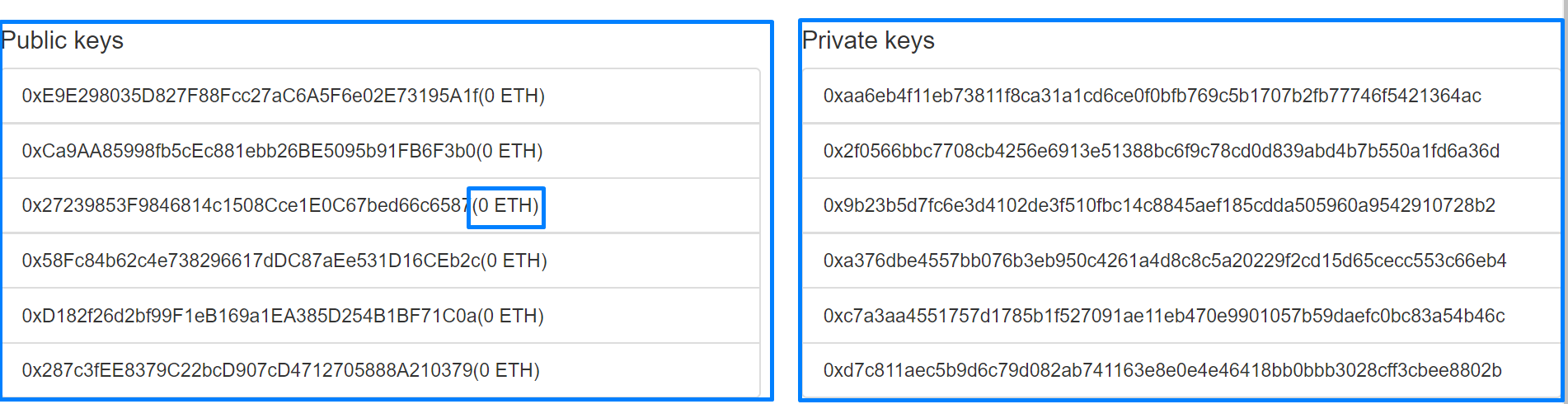




### Test balance method

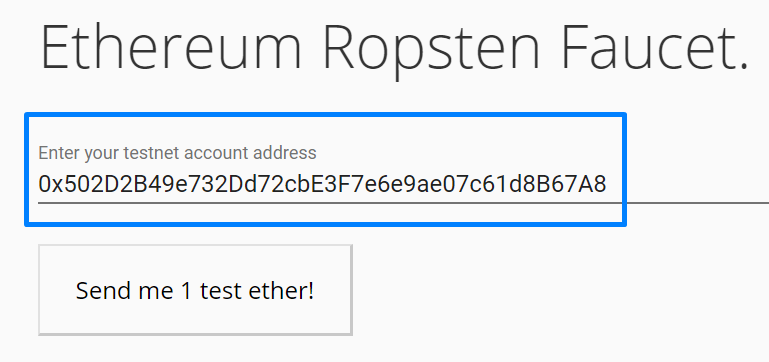
1. Let’s see the balance of our wallet

You will receive 5 accounts (address and private key) without money

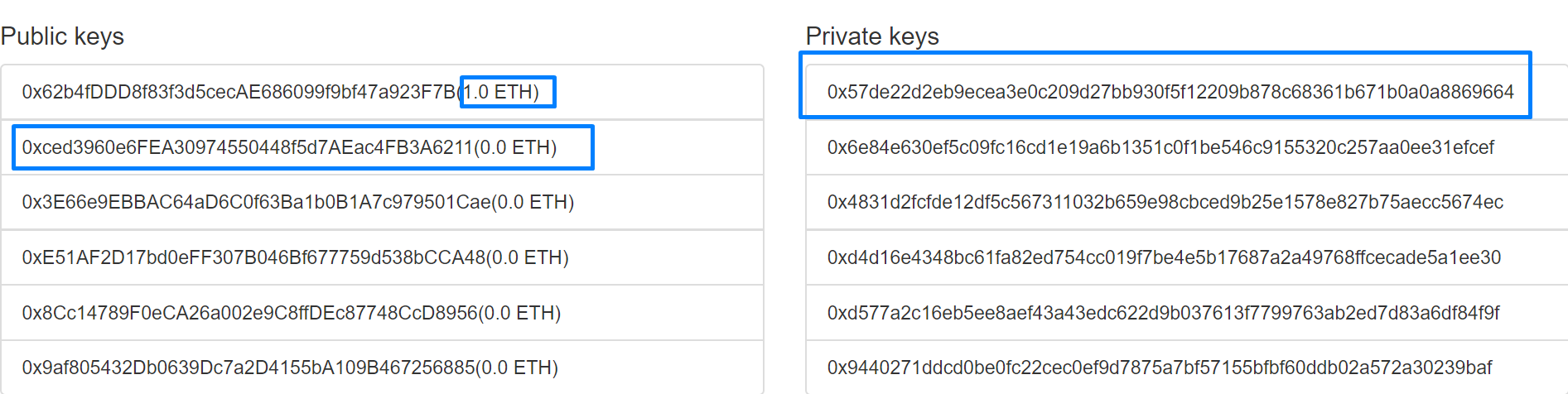


Request money from the faucet

Choose one of the addresses and copy it. Go to <http://faucet.ropsten.be:3001/> , paste your address and wait 3-5 minutes. You can track your transaction status in etherscan.



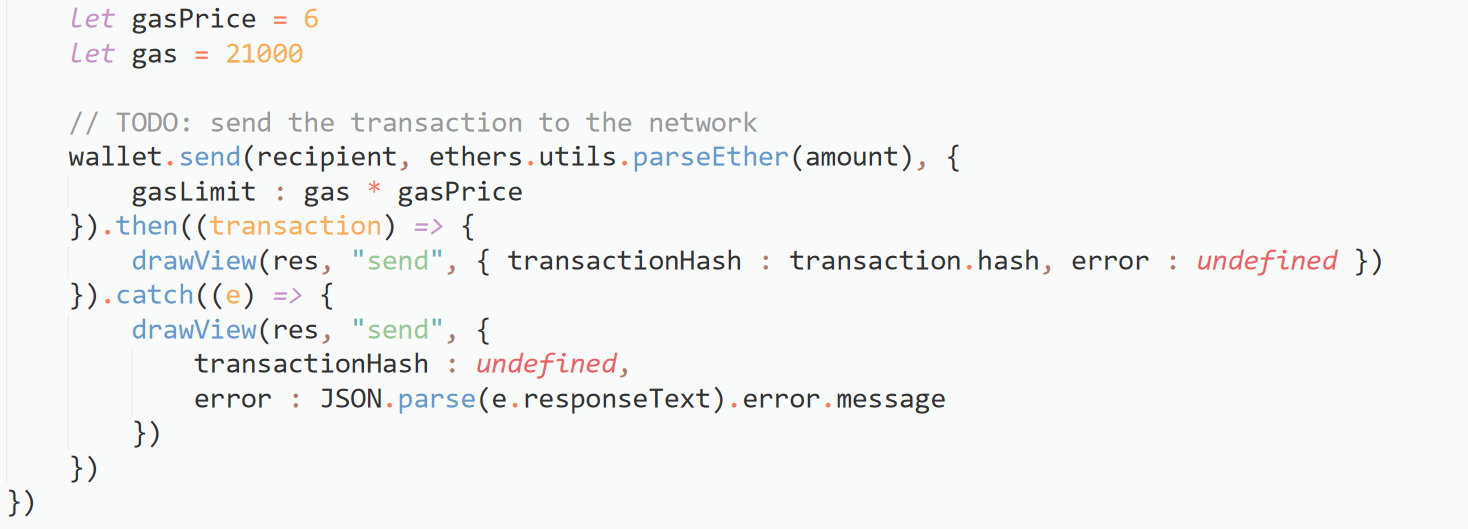
If we check our balance now, we will find 1 ETH in the chosen account. Copy the address and private key of the sender for the next step.



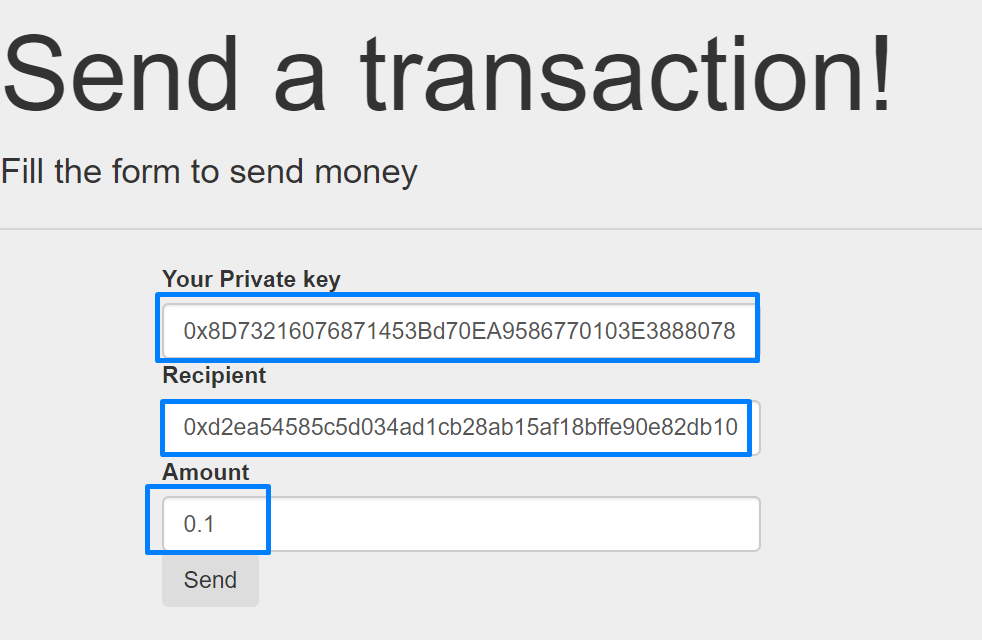
1. Send Transaction Endpoint

Now we will **sign** and **broadcast** transaction to the ropsten network.

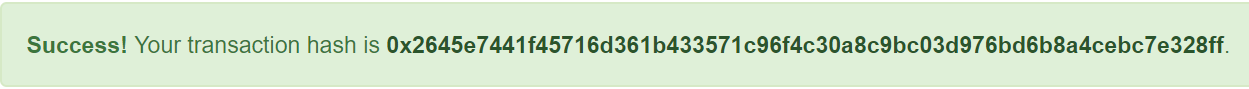


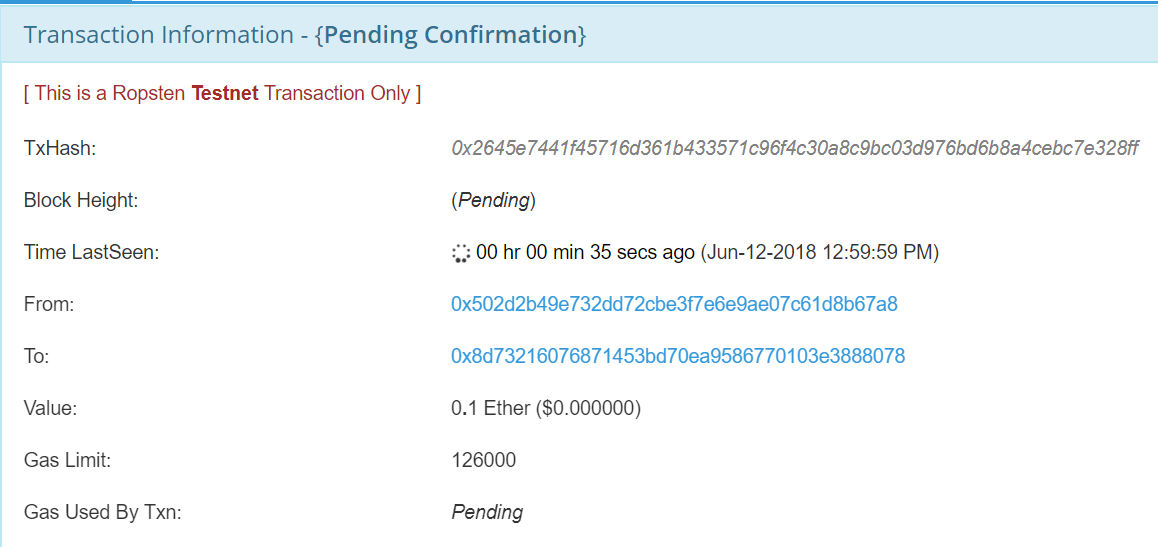


### Test send transaction method

Let’s transfer money between accounts. Copy another address from the balance page as well as the private key of the address that has some ETH.

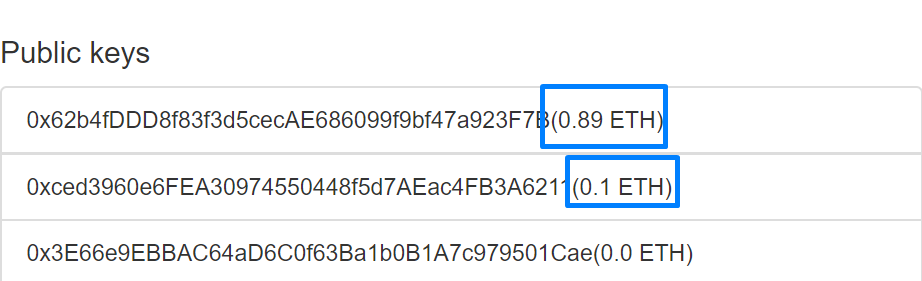
We will receive **Transaction hash** and we can track our transaction in etherscan.





### Check Balance

Let’s **check the balance** again. The second address was **received 1 ETH**. The total balance has lower than 1 ether because of the **fees** payed for the transaction.



# What to Submit?

Create a **zip file** (e.g. your-name-create-ethereum-wallet-with-js-exercise.zip) holding the source code files and screen shots with your experiments.

Submit your zip file as **homework** at the course Web site.