

Images haven't loaded yet. Please exit printing, wait for images to load, and try to print again.

## Transaction via NodeJS Backend



Ahmet Sayarlioglu

Follow

Jun 7, 2018 · 4 min read



# ETHEREUM

*“Note: In this tutorial I use web3js 1.0 version”*

Hello guys, I will explain how to send transaction from a NodeJS backend. I will use rinkeby testnet and will create a router, ,also add some node modules and using infura http provider to complete the tutorial

We need web3js,express and ethereumjs-tx modules to perform the transaction. After that you need to take your own api key from Infura by simply signing up to the infura

### START

First of all you need to create a empty folder for this project to make it easier to access it. After creating the folder you need to open a command shell at that folder. I am assuming that you have npm installed in your computer. npm is distributed by nodejs so if you install nodejs you get npm with it. If you do not have npm download nodejs from this site. After installing npm you need to call.

```
npm init
```

```
this utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (test)
version: (1.0.0)
description:
entry point: (index.js) -
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Users\Sulejir1\Desktop\Test\package.json:
{
  "name": "test",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this ok? (yes) yes
```

PowerShell window after finishing th npm init process

If you want to keep default settings including default name which is the folder name, you can skip all lines by clicking enter. At the end of this process you will have a package.json file.

After initiating the project, inside the folder you need to create a js file with the name that you want which is gonna be our backend

## GETTING MODULES

In my code you can see that there are module requirements, we simply include them into the project by downloading via npm. Their npm calls are;

```
npm install web3 --save
```

```
npm install express --save
```

```
npm install ethereumjs-tx --save
```

You need to call this lines separately from command line to install them into the project. When you finish the installing process you can see that there is a folder which name is node modules it is the folder that your modules downloaded.

## INFURA

When you finish the downloading modules. You need a infura api key. Api key is a crucial part to finish this tutorial. You can get it by signing up to the Infura. When you get the infura api key you need to change the 7th line (YOUR\_API\_KEY) with your api key.

By the way, infura is very good provider. You can also reach ipfs which is a blockchain storage provider with infura api. You can get more information about ipfs from their website

By getting the infura api key we finished the pre-requests of the code. We can start coding. From now on, I will explain the code line by line.

At the first 3 line, I added the modules to the js file.

web3 is the module that will interact to the ethereum.

Express is required for creating a router.

ethereumjs-tx is required for creating transactions

At the line 5, I create my app by using express module.

After creating app, we need to write the last line of the file which is

```
app.listen(3000, () => console.log('Example app listening on  
port 3000!'))
```

At the line 8, I create web3js object with httpprovider which is infura at the nodejs backend. If you searched for ethereum backend before reaching this tutorial you can see that many of them coded by reactjs and they use metamask as provider. NodeJS backend cannot reach the browsers materials so NodeJS backend cannot use metamask as provider so I did a workaround for it and use infura as provider.

In some functions at web3 you need websocket provider for web3 so you need to change the line 8 with this line when needed.

```
web3js = new web3(new
```

```
web3.providers.WebsocketProvider('wss://mainnet.infura.io/_ws'));
```

At the line 10, I create my router which is a get but it needs to be post logically. I create this tutorial for just showing how sending transaction is done so it does not matter at this point.

Inside the router I need my rinkeby Ethereum address(myAddress), the address that I will send my transaction(toAddress), the Contract address(contractAddress) and contract abi(contractABI). You can reach the contracts' abi by Ethereum wallet or etherscan. I deploy my own contract for this tutorial from ethereum wallet you do not need to deploy any contract. you can simply use one of the existing ones.

Last thing that I need is my private key(privateKey). It is not a safe way but since we do not have any wallet at our frontend (because we do not have frontend :) ) we do things manually at this point. you can not reach your private key from your ethereum wallet so you need to import your rinkeby ethereum account to the MyEtherWallet like sites

At the router after defining our variables, we create raw transaction then we signed it with our private key. After signing the transaction we send it to the rinkeby testnet via web3js.

After finishing the coding we need to test it. At the same folder open a command shell and start the backend. My file name is backend.js so I started it by the code

```
node backend.js
```

Which means you can start your backend with the code

```
node <filename with extension>
```

After starting our backend you need to open your browser and you need to write <http://localhost:3000/sendtx> to the address line

That's it. We created our little backend for ethereum.



Thanks for reaching out this tutorial. I am not native so the grammer is probably problematic, sorry for inconvenience. If there is any problem or if you want to ask something you can reach me via twitter





