Lesson 13: Deploy your production Database / Node

With Heroku and Voltage

Considerations before deploying

- Security
  - The node/app is susceptible to attacks
  - The app has ADMIN rights to your node (your funds could be stolen)
  - The activity of the wallet is not private (unless you update it to be so)
- Cost
  - There is a cost hosting the server
  - There is a cost hosting the database
  - There is a cost hosting a node
- What are your goals?
  - Are you trying to show off the mainnet wallet
  - Does testnet suffice?
  - Does it need to be live and deployed?

What is Heroku?

Heroku is a cloud platform service that allows developers to deploy, manage, and scale applications without a lot of overhead. It abstracts away the complexities of managing servers, infrastructure, and databases, allowing developers to focus purely on the code.

- Heroku is made specifically for deploying servers / databases
- Heroku has consistent performance, it’s easy to setup, and easy to scale.
- Heroku used to have a basic free plan but now you need to pay a small fee for both the server and the postgres instance

HEROKU

What is Heroku?

Platform as a Service (PaaS): Heroku is a PaaS, which means it provides both the environment to run applications and the infrastructure to host them. Think of it like a combination of a playground and the land it's on. You bring the toys (your application), and Heroku provides the place and ensures the ground is always ready for play (auto-maintenance, updates, etc.).

Simple Deployment: Deploying an application to Heroku is often as simple as pushing your code using Git. You can think of it like uploading your photos to a cloud service, but in this case, it's your application's code.

Supports Multiple Languages: Heroku supports several programming languages like Ruby, Java, Python, Node.js, and more. It's like a multi-cuisine restaurant where you can bring recipes (code) from different cuisines (programming languages) and expect them to be cooked (run) perfectly.

Dynos: Heroku uses containers called "dynos" to run applications. Imagine these as individual mini-computers within a big computer. Each one can run your application independently, and you can have many of them to handle more users.

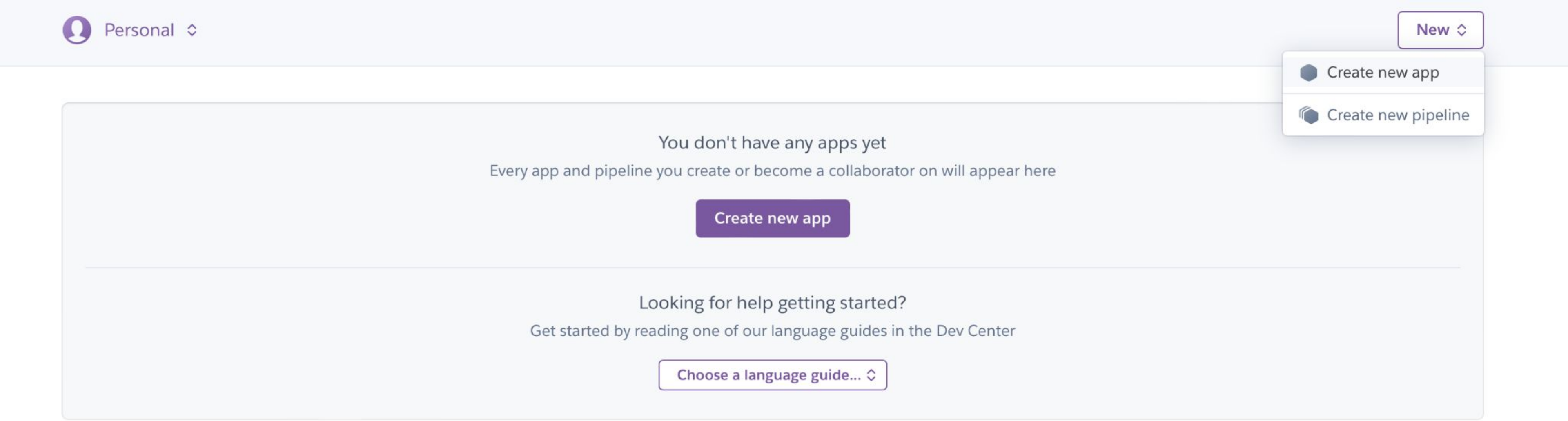
Add-ons: Heroku provides a marketplace of add-ons (services and tools) that you can attach to your applications. It's like adding extra toppings on a pizza. Need a database? Add it! Need email services? Add that too!

Scalability: If your application becomes popular and receives more traffic, Heroku can easily scale up to handle the load. It's like having a magic room that expands when more guests arrive.

Steps to deploying our server / database

1. Create a new Heroku 'App'
2. Connect our backend repo
3. Add a web 'Dyno'
4. Add Postgres Buildpack
5. Setup secret ENV variables
6. Deploy server
7. Test calling the root endpoint
8. Test creating an admin user
9. Test getting all users

Creating our Heroku app



You don't have any apps yet

Every app and pipeline you create or become a collaborator on will appear here

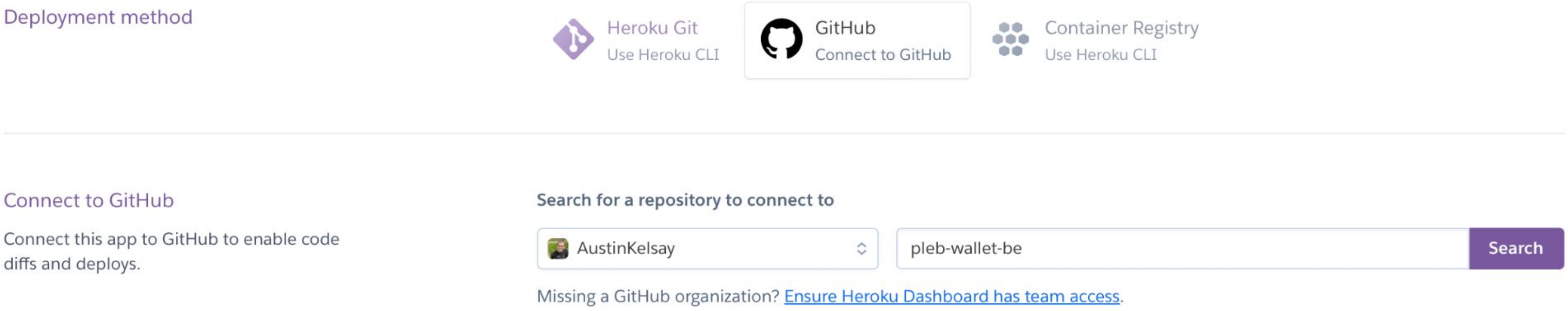
Looking for help getting started?

Get started by reading one of our language guides in the Dev Center

Connecting Heroku to our backend code

Go to the ‘Deploy’ tab and select Github as your deployment method and add the pleb-wallet-be as the repo.

Now hit deploy!



Deployment method

Heroku Git
Use Heroku CLI

GitHub
Connect to GitHub

Container Registry
Use Heroku CLI

Connect to GitHub

Connect this app to GitHub to enable code diffs and deploys.

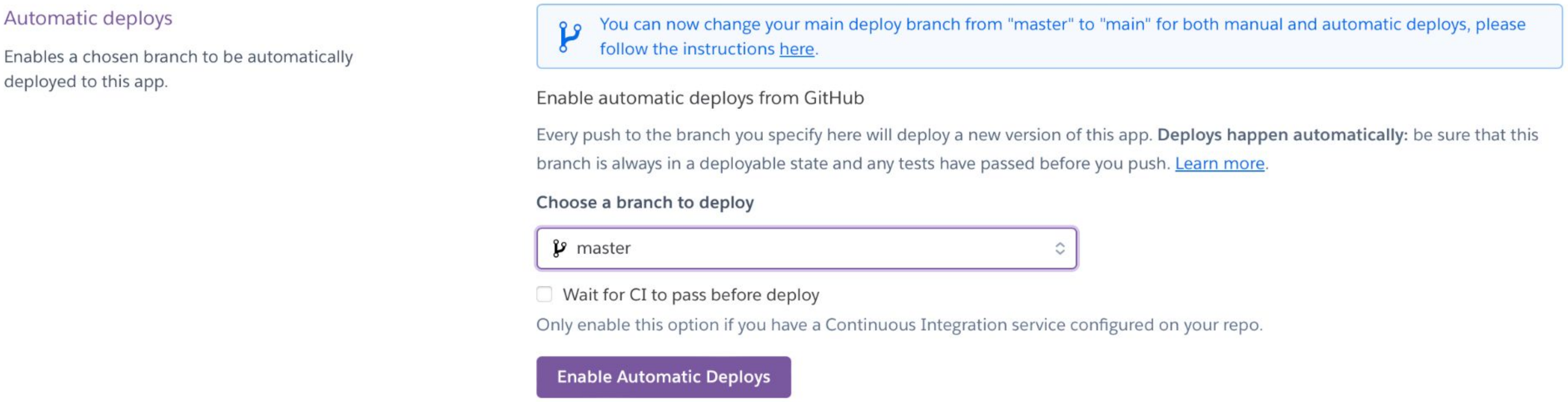
Search for a repository to connect to

AustinKelsay

pleb-wallet-be

Missing a GitHub organization? Ensure Heroku Dashboard has team access.

Deploying our backend code



Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions here.

Enable automatic deploys from GitHub

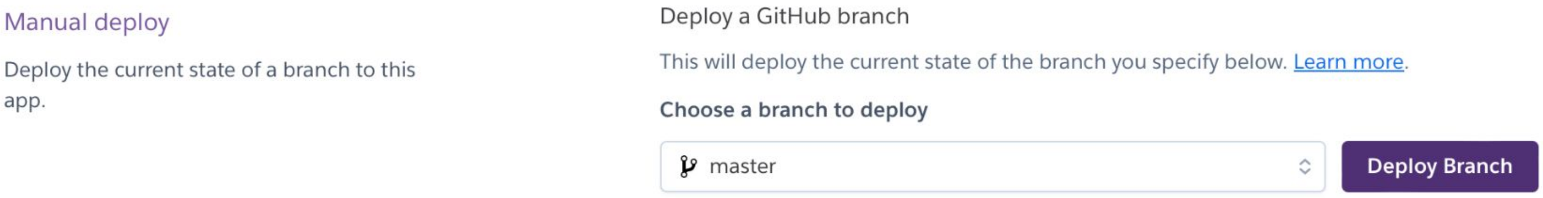
Every push to the branch you specify here will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch is always in a deployable state and any tests have passed before you push. Learn more.

Choose a branch to deploy

master

□ Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.



Manual deploy

Deploy the current state of a branch to this app.

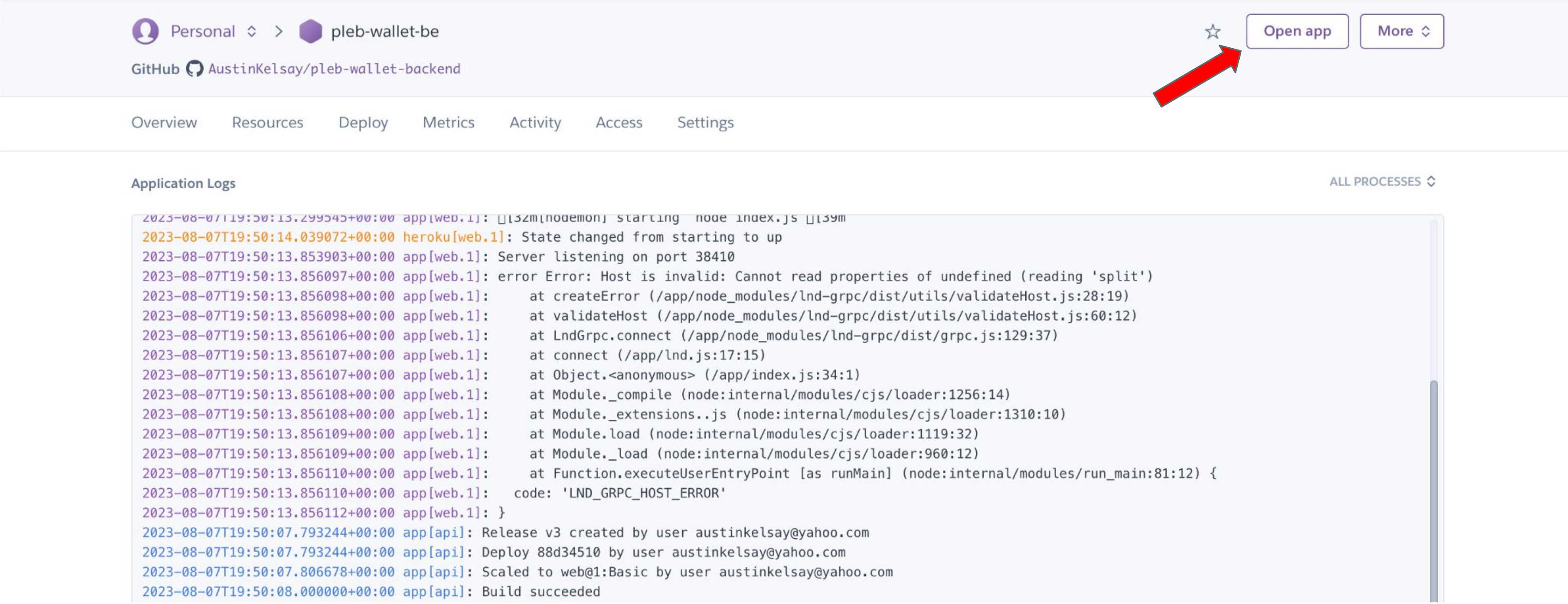
Deploy a GitHub branch

This will deploy the current state of the branch you specify below. Learn more.

Choose a branch to deploy

master

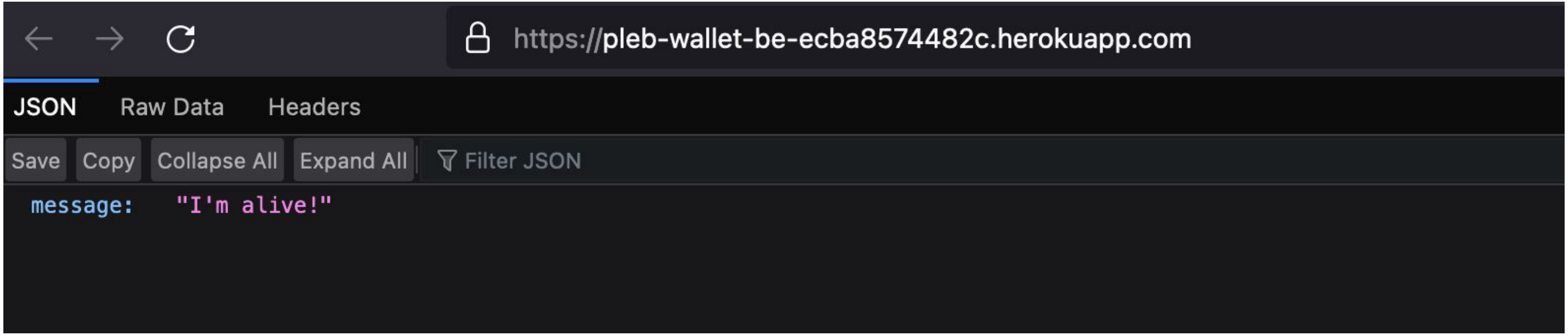
Watching the logs and testing the root endpoint



Application Logs

```
2023-08-07T19:50:13.299545+00:00 app[web.1]: □[32m[nodemon] starting `node index.js` □[39m
2023-08-07T19:50:14.039072+00:00 heroku[web.1]: State changed from starting to up
2023-08-07T19:50:13.853903+00:00 app[web.1]: Server listening on port 38410
2023-08-07T19:50:13.856097+00:00 app[web.1]: error Error: Host is invalid: Cannot read properties of undefined (reading 'split')
2023-08-07T19:50:13.856098+00:00 app[web.1]:     at createError (/app/node_modules/lnd-grpc/dist/utils/validateHost.js:28:19)
2023-08-07T19:50:13.856098+00:00 app[web.1]:     at validateHost (/app/node_modules/lnd-grpc/dist/utils/validateHost.js:60:12)
2023-08-07T19:50:13.856106+00:00 app[web.1]:     at LndGrpc.connect (/app/node_modules/lnd-grpc/dist/grpc.js:129:37)
2023-08-07T19:50:13.856107+00:00 app[web.1]:     at connect (/app/lnd.js:17:15)
2023-08-07T19:50:13.856107+00:00 app[web.1]:     at Object.<anonymous> (/app/index.js:34:1)
2023-08-07T19:50:13.856108+00:00 app[web.1]:     at Module._compile (node:internal/modules/cjs/loader:1256:14)
2023-08-07T19:50:13.856108+00:00 app[web.1]:     at Module._extensions..js (node:internal/modules/cjs/loader:1310:10)
2023-08-07T19:50:13.856109+00:00 app[web.1]:     at Module.load (node:internal/modules/cjs/loader:1119:32)
2023-08-07T19:50:13.856109+00:00 app[web.1]:     at Module._load (node:internal/modules/cjs/loader:960:12)
2023-08-07T19:50:13.856110+00:00 app[web.1]:     at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:81:12) {
2023-08-07T19:50:13.856110+00:00 app[web.1]:   code: 'LND_GRPC_HOST_ERROR'
2023-08-07T19:50:13.856112+00:00 app[web.1]: }
2023-08-07T19:50:07.793244+00:00 app[api]: Release v3 created by user austinkelsay@yahoo.com
2023-08-07T19:50:07.793244+00:00 app[api]: Deploy 88d34510 by user austinkelsay@yahoo.com
2023-08-07T19:50:07.806678+00:00 app[api]: Scaled to web@1:Basic by user austinkelsay@yahoo.com
2023-08-07T19:50:08.000000+00:00 app[api]: Build succeeded
```

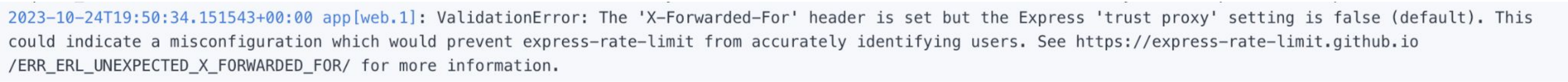
Watching the logs and testing the root endpoint



```
message:   "I'm alive!"
```

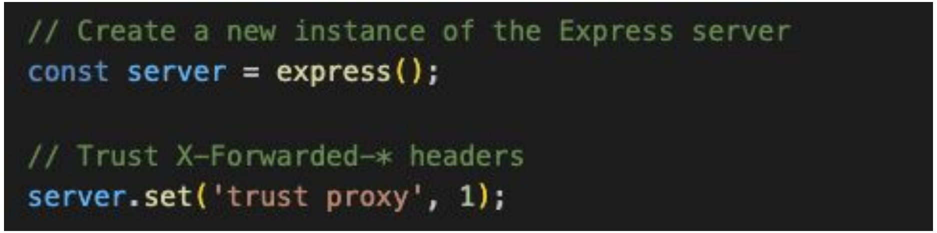
Express rate limit error

If you get this error:



```
2023-10-24T19:50:34.151543+00:00 app[web.1]: ValidationError: The 'X-Forwarded-For' header is set but the Express 'trust proxy' setting is false (default). This
could indicate a misconfiguration which would prevent express-rate-limit from accurately identifying users. See https://express-rate-limit.github.io
/ERR_ERL_UNEXPECTED_X_FORWARDED_FOR/ for more information.
```

Update index.js like so:



```
// Create a new instance of the Express server
const server = express();

// Trust X-Forwarded-* headers
server.set('trust proxy', 1);
```

The error message you're seeing is due to the fact that you're deploying your application behind a reverse proxy (like many cloud hosting solutions do), and `express-rate-limit` is warning you that it might not be getting the correct client IP addresses.

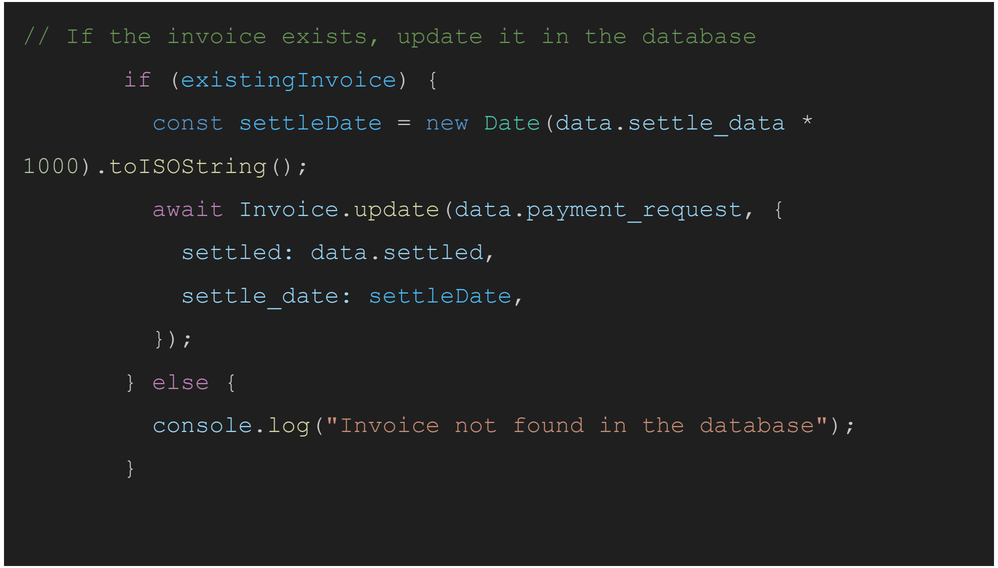
The above tells Express to trust the first proxy it connects to. If your app is behind multiple proxies, you can set `'trust proxy'` to the number of proxy servers or `'loopback'` to trust all proxies.

Invoice datetime error

While we’re already fixing bugs let’s look into this one I got when testing the deployment: error: update "invoices" set "settled" = $1, "settle_date" = $2 where "payment_request" = $3 returning * - date/time field value out of range: "1698189002"

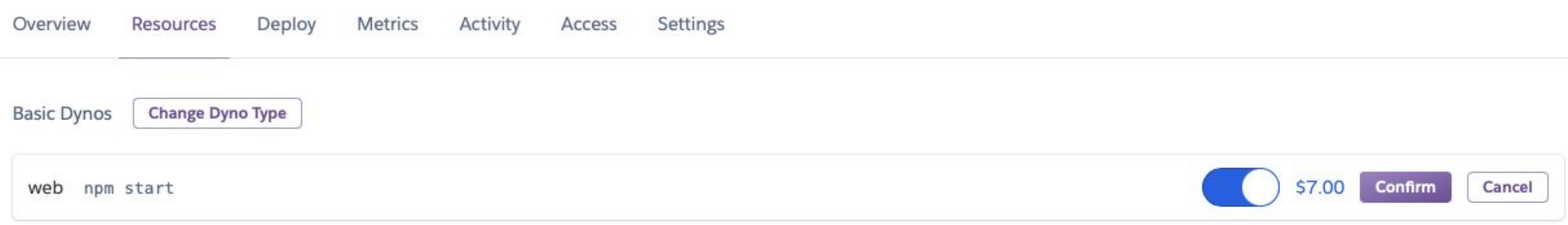
Our db schema is expecting a longer ‘UNIX’ timestamp:

Fix in lnd.js:



```
// If the invoice exists, update it in the database
       if (existingInvoice) {
         const settleDate = new Date(data.settle_data *
1000).toISOString();
         await Invoice.update(data.payment_request, {
           settled: data.settled,
           settle_date: settleDate,
         });
       } else {
         console.log("Invoice not found in the database");
       }
```

Enabling a basic web 'Dyno'



```
web  npm start
```

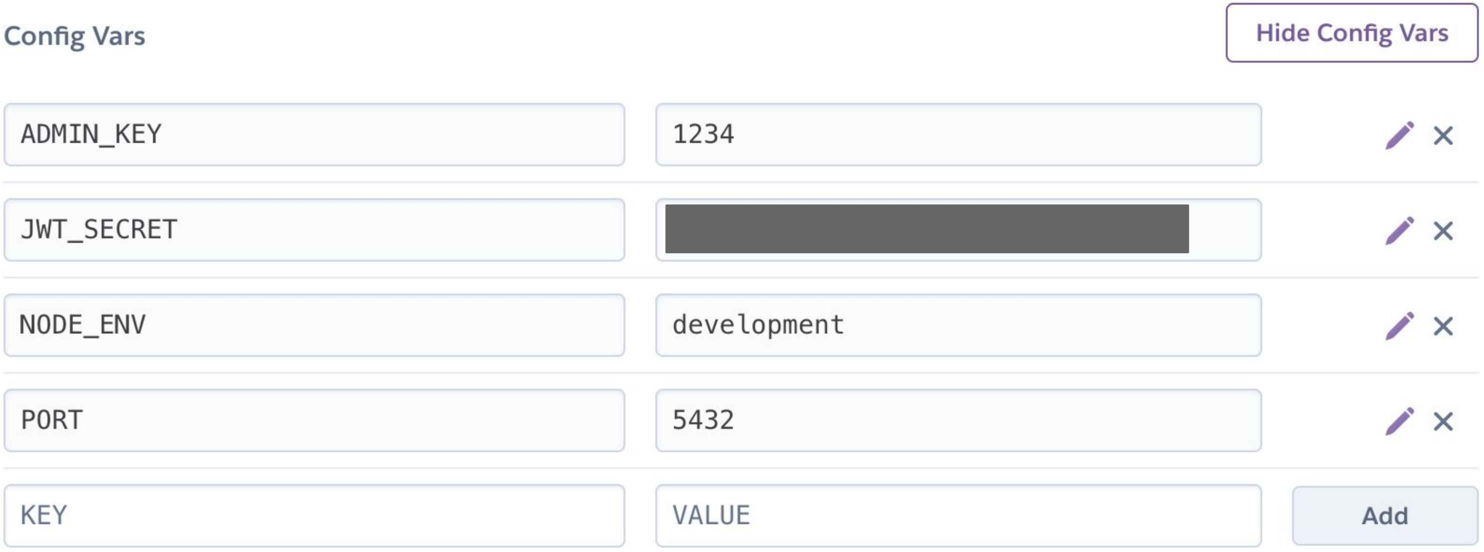
What are Dynos on Heroku? https://www.heroku.com/dynos

Put simply Dynos are the isolated instances of your server that run on hardware in the cloud

Adding ENV secrets

Click “Reveal Config Vars” and add new secure secrets for both ADMIN_KEY and JWT_SECRET

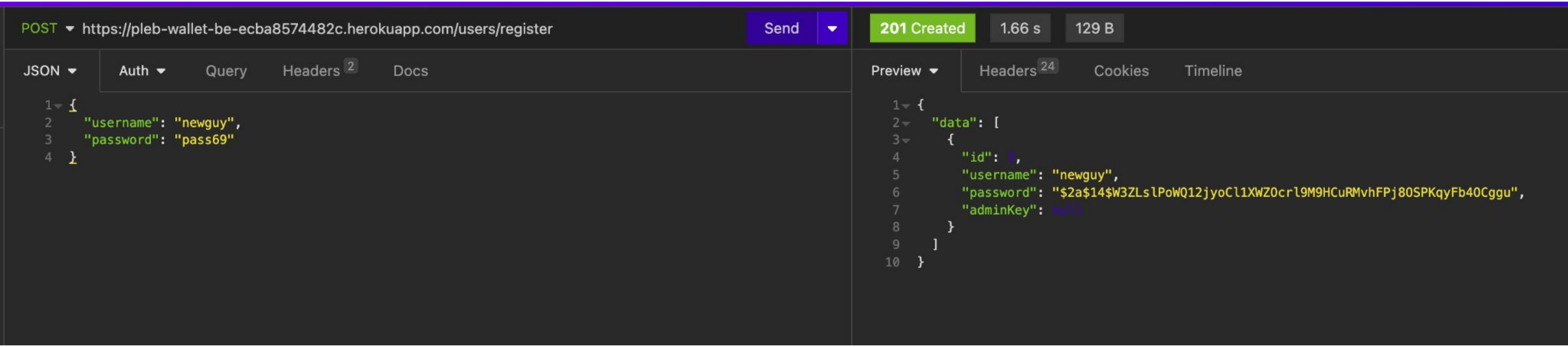
Later we will add more config Vars for our Lightning node once we deploy it



Config Vars

ADMIN_KEY	1234
JWT_SECRET	[redacted]
NODE_ENV	development
PORT	5432

Create a new user

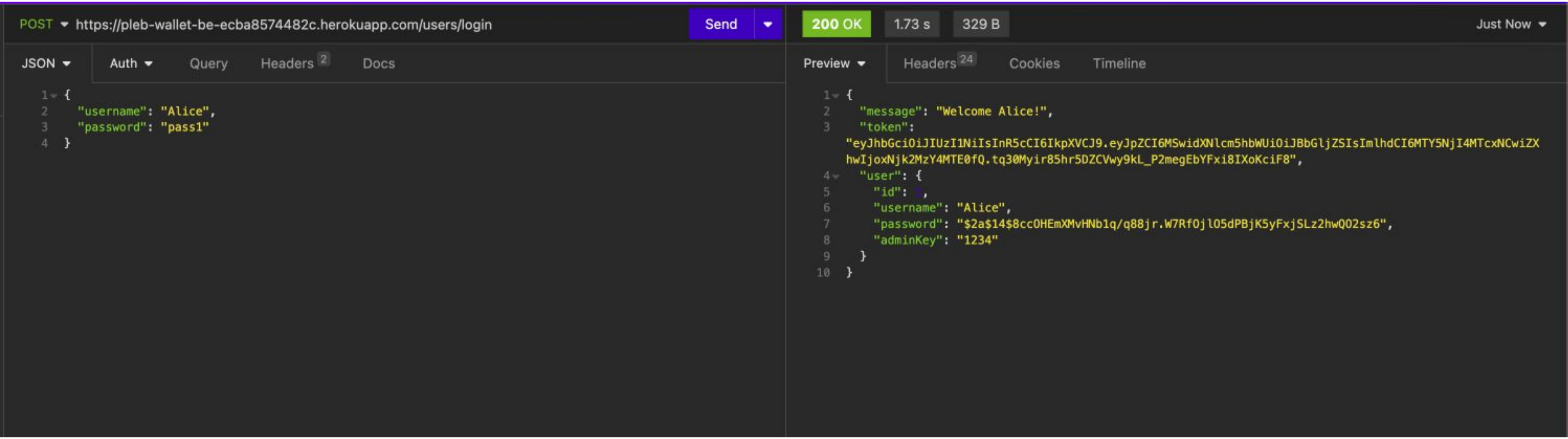


https://pleb-wallet-be-ecba8574482c.herokuapp.com/users/register

```
{
  "username": "newguy",
  "password": "pass69"
}
```

```
{
  "data": [
    {
      "id": 3,
      "username": "newguy",
      "password": "$2a$14$W3ZLslPoWQ12jyoCl1XWZOcrl9M9HCuRMvhFPj8OSPKqyFb4OCggu",
      "adminKey": null
    }
  ]
}
```

Test the login endpoint with a “seeded” user



POST https://pleb-wallet-be-ecba8574482c.herokuapp.com/users/login

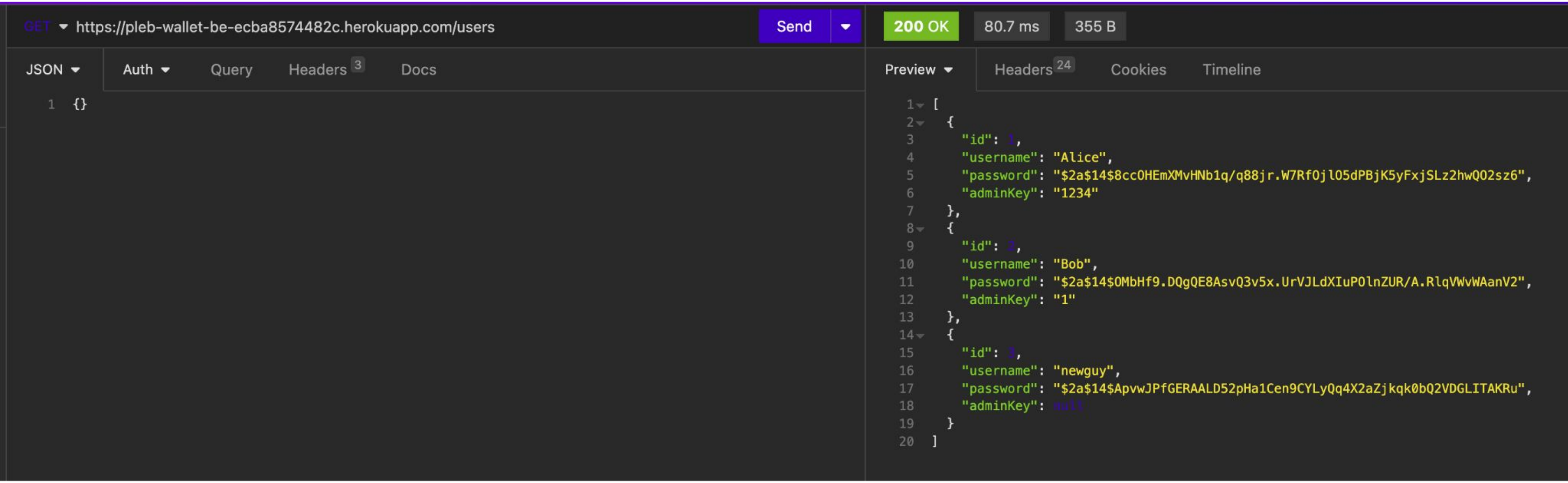
```
{
  "username": "Alice",
  "password": "pass1"
}
```

200 OK 1.73 s 329 B

```
{
  "message": "Welcome Alice!",
  "token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwidXNlcm5hbWUiOiJBbGljZSIsImlhdCI6MTY5NjI4MTcxNCwiZX
hwIjoxNjk2MzY4MTE0fQ.tq30Myir85hr5DZCVwy9kL_P2megEbYFxi8IXoKciF8",
  "user": {
    "id": 1,
    "username": "Alice",
    "password": "$2a$14$8ccOHEmXMvHNb1q/q88jr.W7RfOjlO5dPBjK5yFxjSLz2hwQO2sz6",
    "adminKey": "1234"
  }
}
```

Get all users and see our newly created user

The database is working!



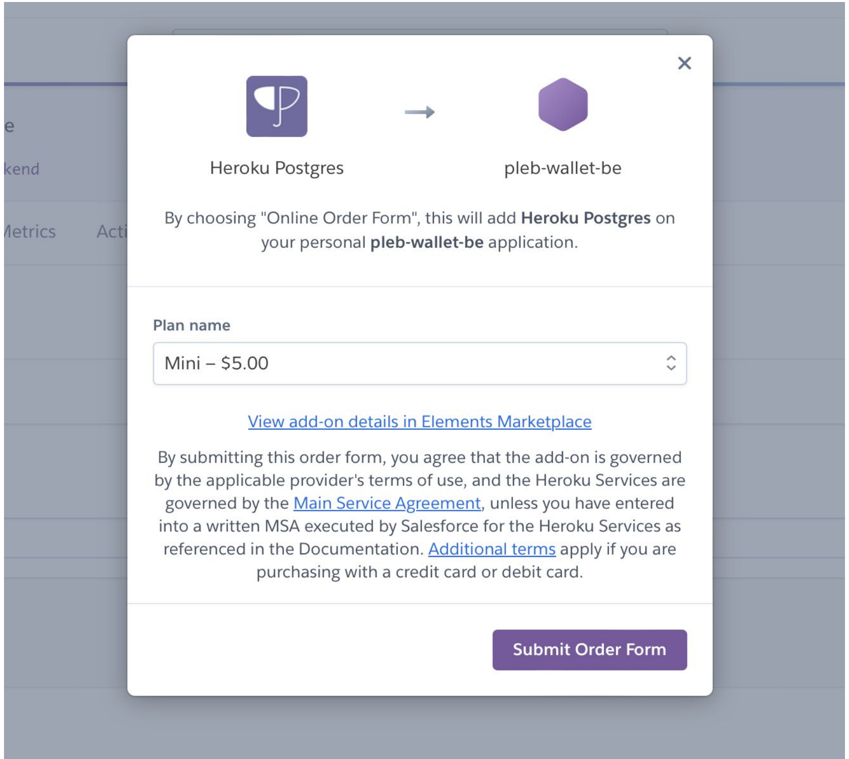
```
{}
```

```
[
  {
    "id": 1,
    "username": "Alice",
    "password": "$2a$14$8ccOHEmXMvHNb1q/q88jr.W7RfOjlO5dPBjK5yFxjSLz2hwQO2sz6",
    "adminKey": "1234"
  },
  {
    "id": 2,
    "username": "Bob",
    "password": "$2a$14$OMbHf9.DQgQE8AsvQ3v5x.UrVJLdXIuPOlnZUR/A.RlqVWvWAanV2",
    "adminKey": "1"
  },
  {
    "id": 3,
    "username": "newguy",
    "password": "$2a$14$ApvwJPfGERAALD52pHa1Cen9CYLyQq4X2aZjkqk0bQ2VDGLITAKRu",
    "adminKey": null
  }
]
```

Adding Postgres Buildpack

Click on “Resources” tab and search for Heroku Postgres

(Sadly Heroku has raised prices so it’s 5$ a month for a pg database)



By choosing "Online Order Form", this will add **Heroku Postgres** on your personal **pleb-wallet-be** application.

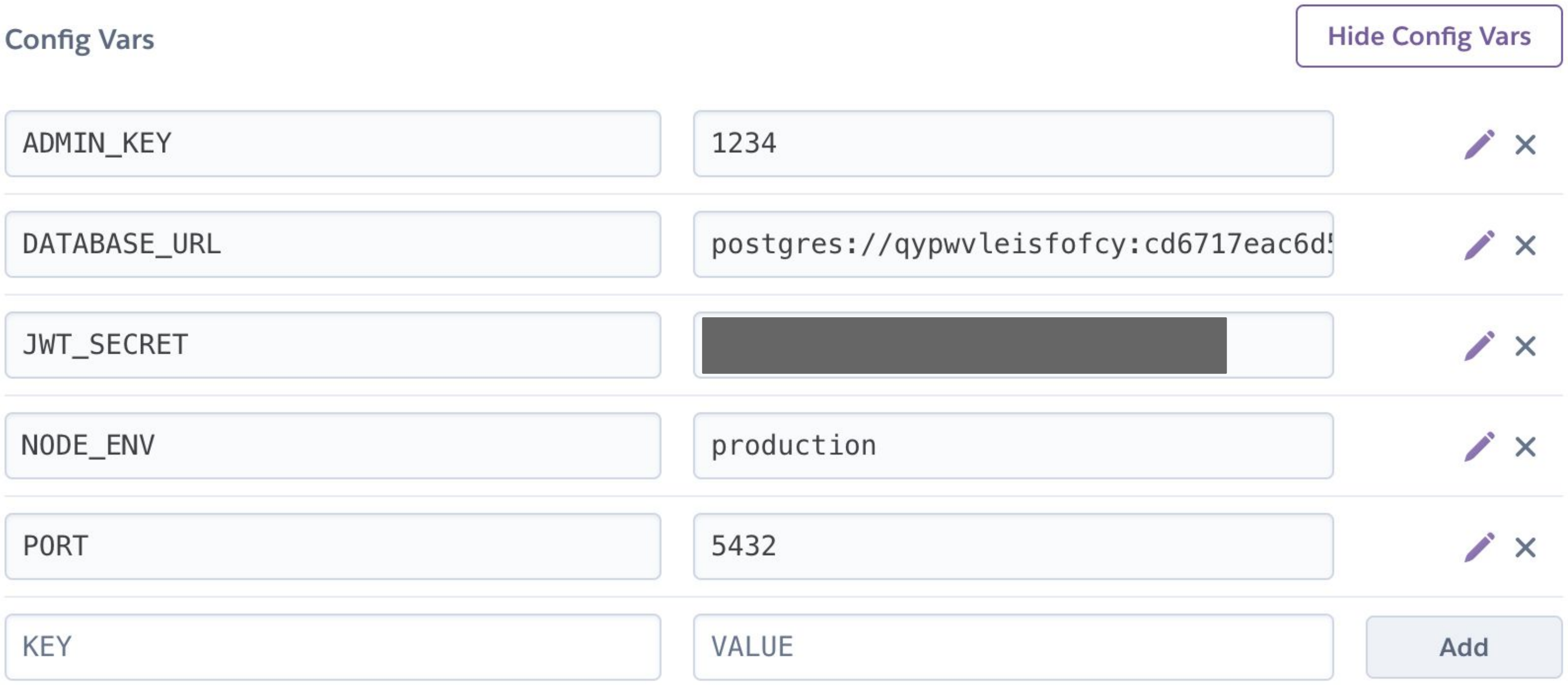
Plan name

Mini – $5.00

View add-on details in Elements Marketplace

By submitting this order form, you agree that the add-on is governed by the applicable provider's terms of use, and the Heroku Services are governed by the Main Service Agreement, unless you have entered into a written MSA executed by Salesforce for the Heroku Services as referenced in the Documentation. Additional terms apply if you are purchasing with a credit card or debit card.

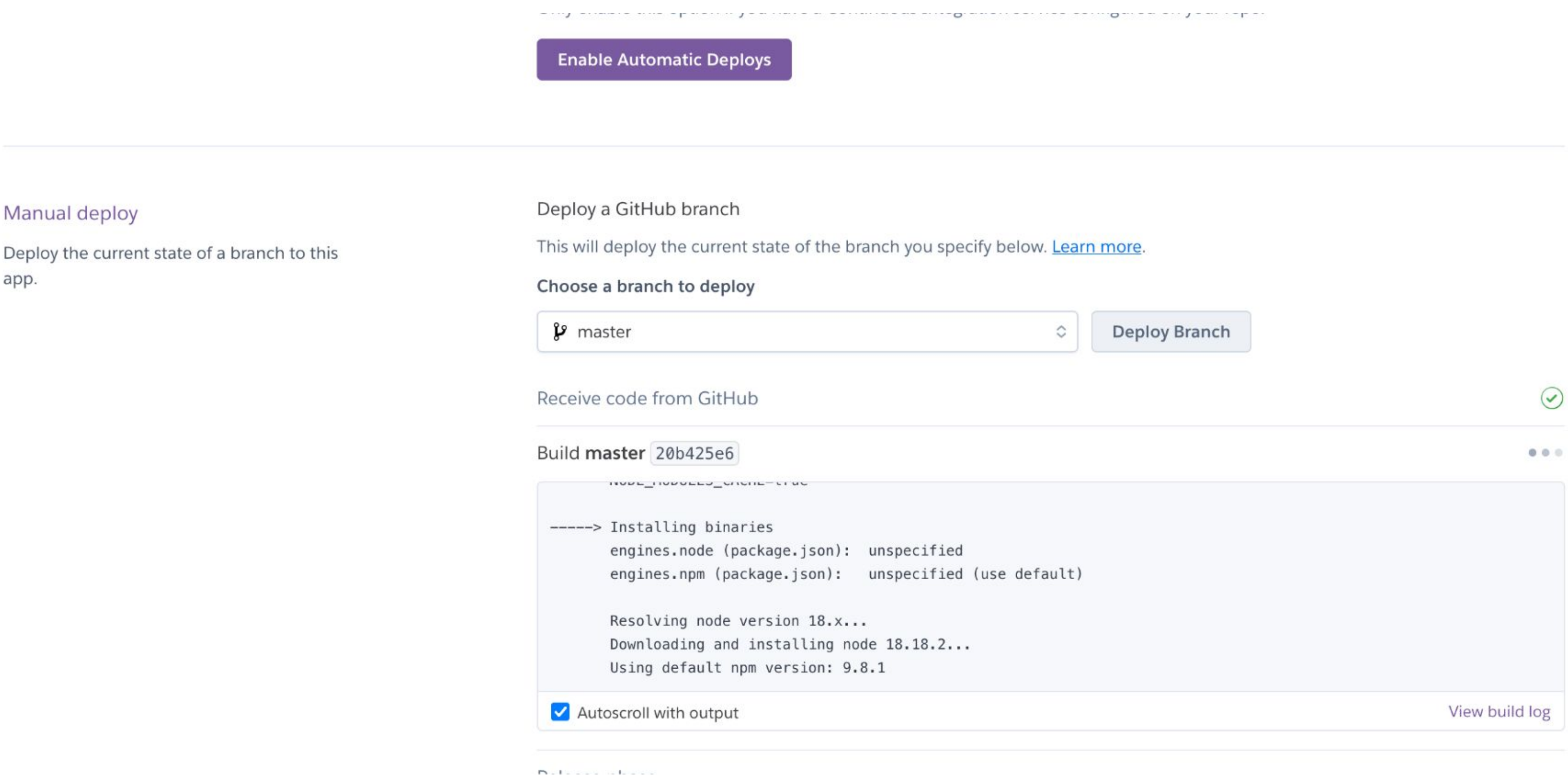
Database url env variable should be added



Config Vars

ADMIN_KEY	1234
DATABASE_URL	postgres://qypwvleisfofcy:cd6717eac6d
JWT_SECRET	[redacted]
NODE_ENV	production
PORT	5432
KEY	VALUE

Redeploy



Only enable this option if you have a continuous integration service configured on your repo.

Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. Learn more.

Choose a branch to deploy

master

Receive code from GitHub ✓

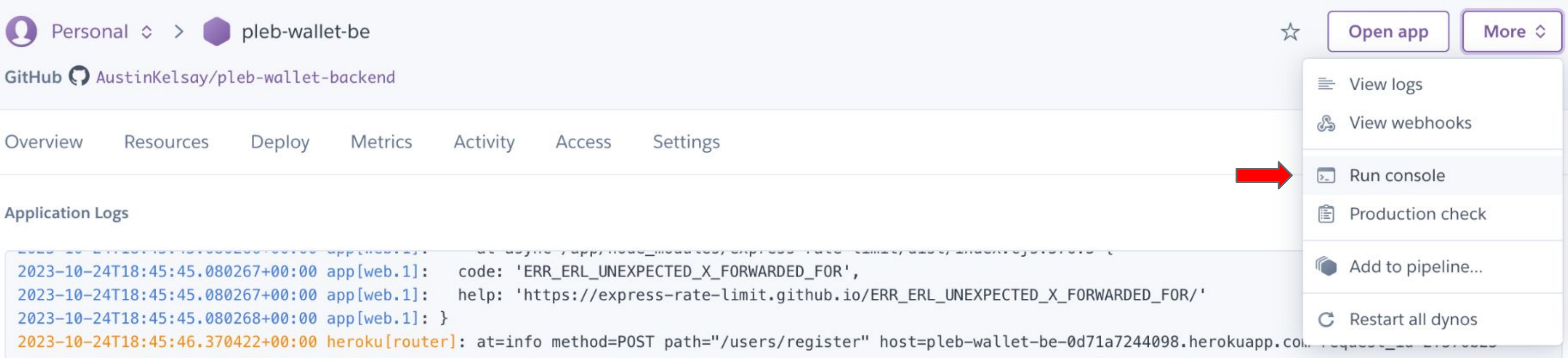
Build master 20b425e6

```
       NODE_MODULES_CACHE=true

-----> Installing binaries
       engines.node (package.json):  unspecified
       engines.npm (package.json):   unspecified (use default)

       Resolving node version 18.x...
       Downloading and installing node 18.18.2...
       Using default npm version: 9.8.1
```

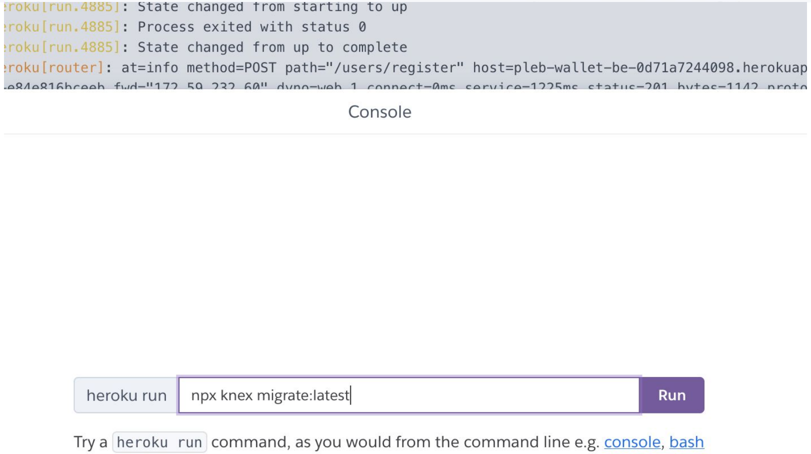
☑ Autoscroll with output

Release phase

Run migration



```
2023-10-24T18:45:45.080267+00:00 app[web.1]:   code: 'ERR_ERL_UNEXPECTED_X_FORWARDED_FOR',
2023-10-24T18:45:45.080267+00:00 app[web.1]:   help: 'https://express-rate-limit.github.io/ERR_ERL_UNEXPECTED_X_FORWARDED_FOR/'
2023-10-24T18:45:45.080268+00:00 app[web.1]: }
2023-10-24T18:45:46.370422+00:00 heroku[router]: at=info method=POST path="/users/register" host=pleb-wallet-be-0d71a7244098.herokuapp.co
```

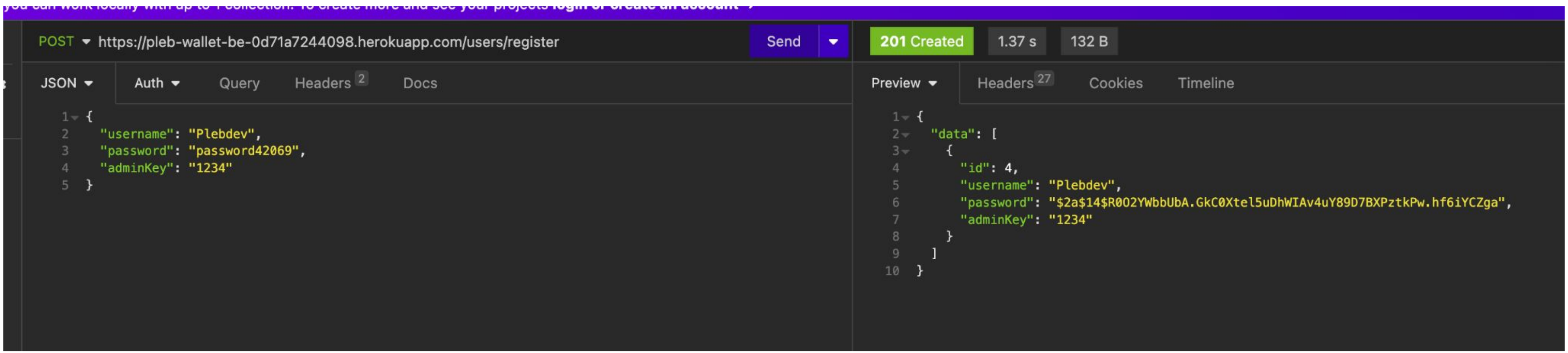


```
eroku[run.4885]: State changed from starting to up
eroku[run.4885]: Process exited with status 0
eroku[run.4885]: State changed from up to complete
eroku[router]: at=info method=POST path="/users/register" host=pleb-wallet-be-0d71a7244098.herokuap
e84e816bceeb fwd="172.59.232.60" dyno=web.1 connect=0ms service=1225ms status=201 bytes=1142 proto
```

```
npx knex migrate:latest
```

Try a `heroku run` command, as you would from the command line e.g. console, bash

Create a admin user in your production db then login and check all users again



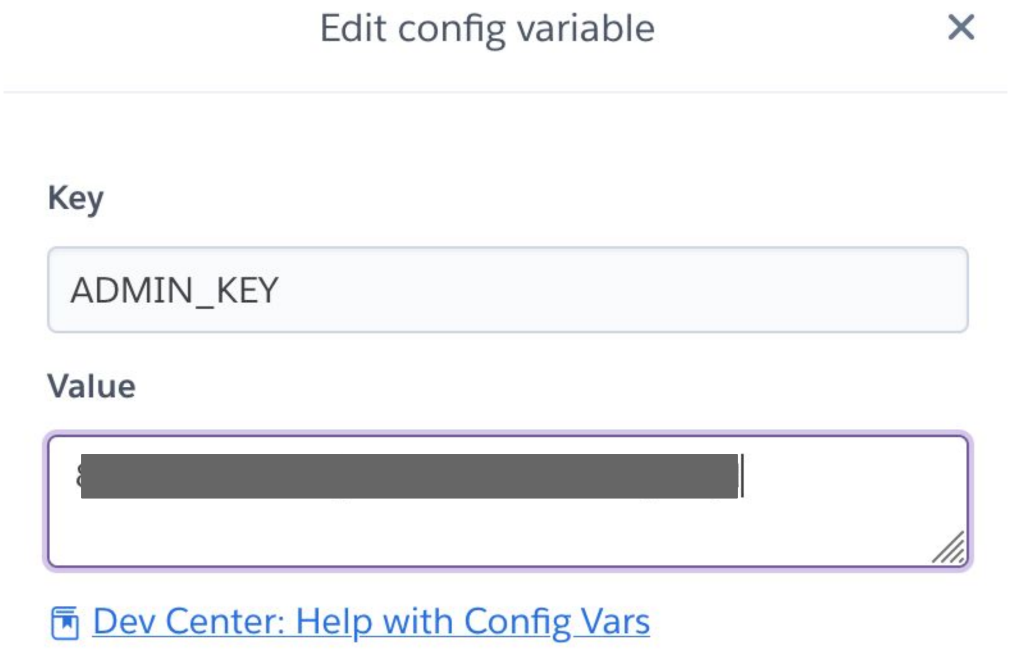
you can work locally with up to 1 collection. To create more and see your projects **login or create an account** →

POST https://pleb-wallet-be-0d71a7244098.herokuapp.com/users/register

```
{
  "username": "Plebdev",
  "password": "password42069",
  "adminKey": "1234"
}
```

```
{
  "data": [
    {
      "id": 4,
      "username": "Plebdev",
      "password": "$2a$14$R0O2YWbbUbA.GkC0Xtel5uDhWIAv4uY89D7BXPztkPw.hf6iYCZga",
      "adminKey": "1234"
    }
  ]
}
```

Now update your adminKey and JWT if you haven't to be secure secrets



Edit config variable

Key

ADMIN_KEY

Value

[redacted]

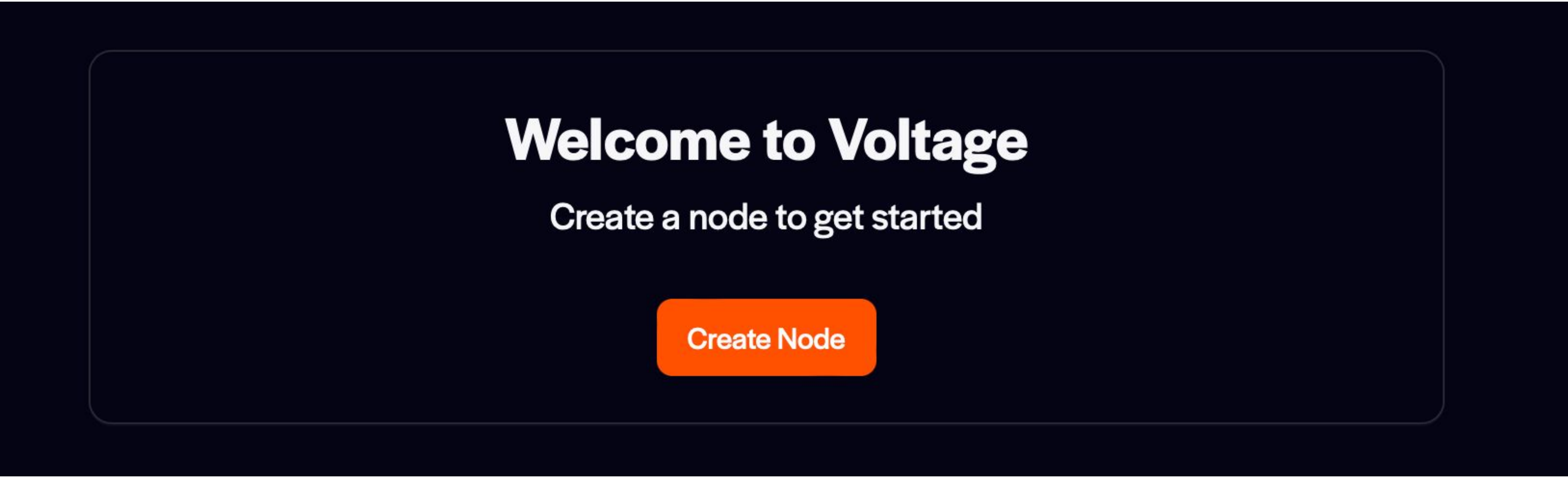
Dev Center: Help with Config Vars

Deploy a Production Lightning node

With Voltage

Create a node

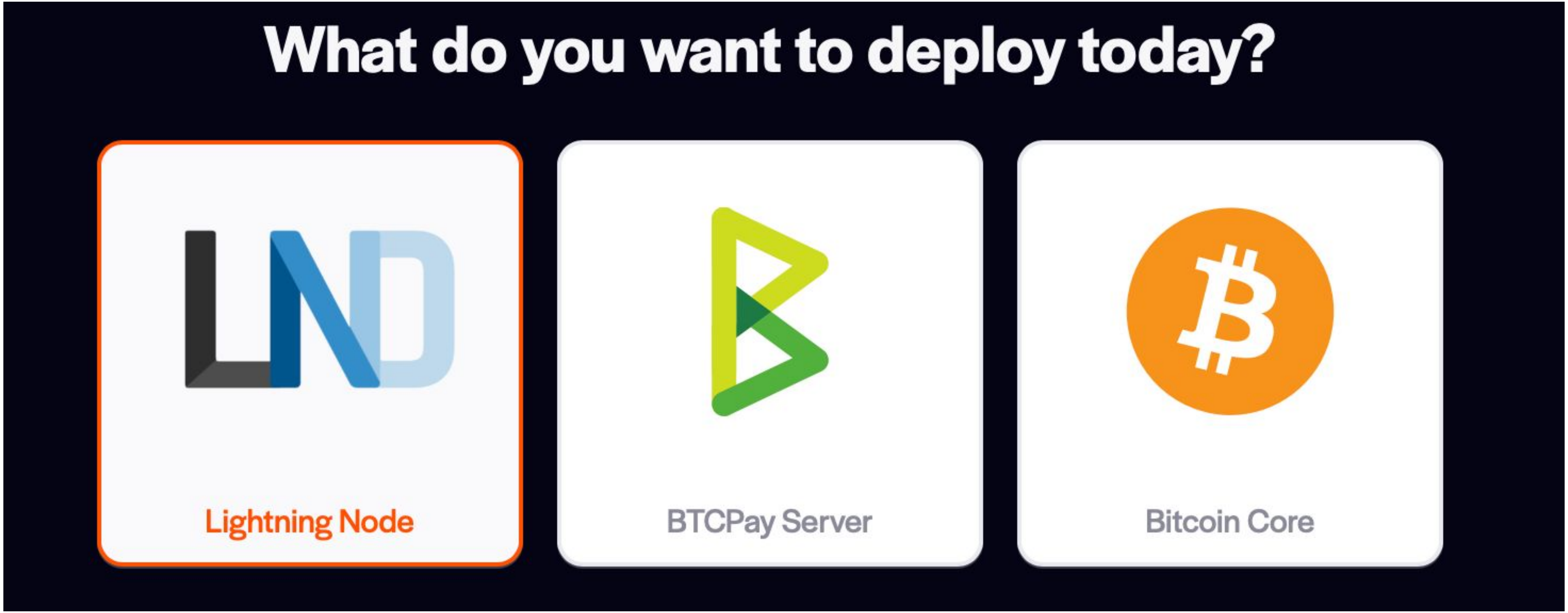
- Visit https://nodes.voltage.cloud
- Click ‘Create Node’



Welcome to Voltage

Create a node to get started

Choose LND



What do you want to deploy today?

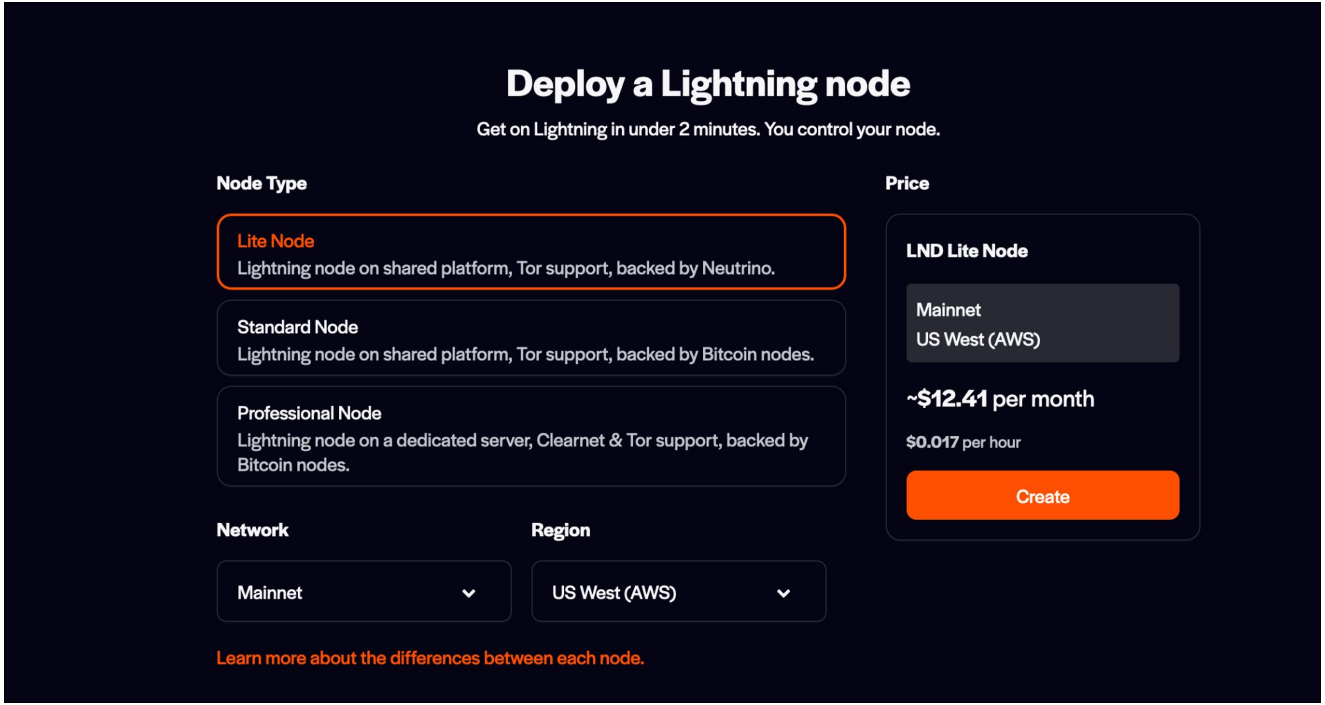
Lightning Node

BTCPay Server

Bitcoin Core

Choose your node type

You can pick a Lite Node to start and use testnet if you want to do further testing but we will use mainnet to start ripping some real invoices



Deploy a Lightning node

Get on Lightning in under 2 minutes. You control your node.

Node Type

● **Lite Node**
Lightning node on shared platform, Tor support, backed by Neutrino.

○ **Standard Node**
Lightning node on shared platform, Tor support, backed by Bitcoin nodes.

○ **Professional Node**
Lightning node on a dedicated server, Clearnet & Tor support, backed by Bitcoin nodes.

Network: Mainnet

Region: US West (AWS)

Learn more about the differences between each node.

Price

LND Lite Node

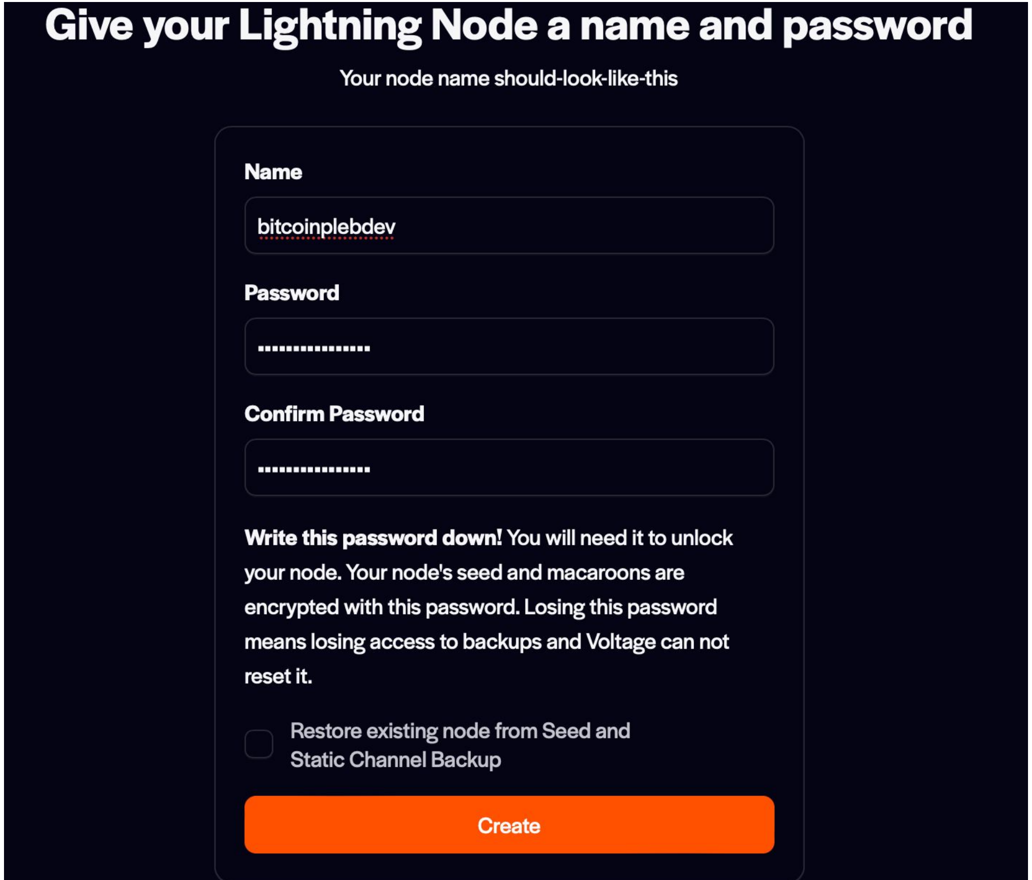
Mainnet
US West (AWS)

~$12.41 per month

$0.017 per hour

Create a username/password for your node

And be sure to write them down!



Give your Lightning Node a name and password

Your node name should-look-like-this

Name

bitcoinplebdev

Password

················

Confirm Password

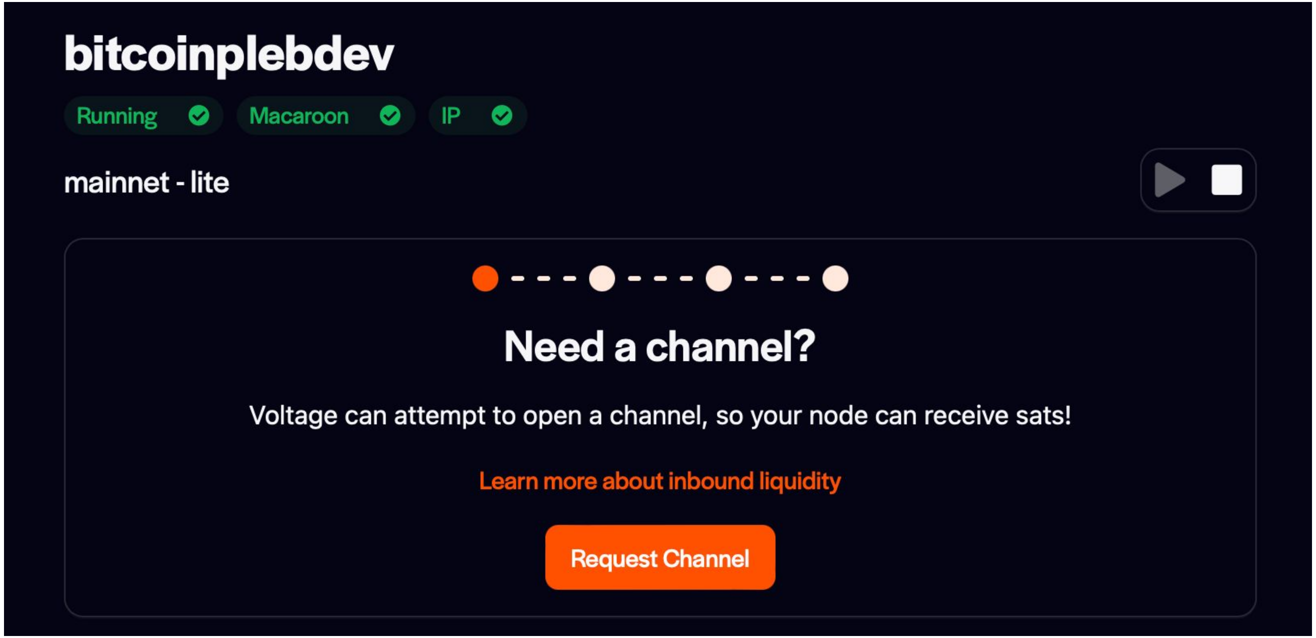
················

Write this password down! You will need it to unlock your node. Your node's seed and macaroons are encrypted with this password. Losing this password means losing access to backups and Voltage can not reset it.

□ Restore existing node from Seed and Static Channel Backup

Get free inbound channel

After your node is done initializing you should see this message on your node's home page to request a free inbound channel! This will allow us to instantly start receiving sats into our pleb-wallet from the wider network.



bitcoinplebdev

Running ✓ Macaroon ✓ IP ✓

mainnet - lite

Need a channel?

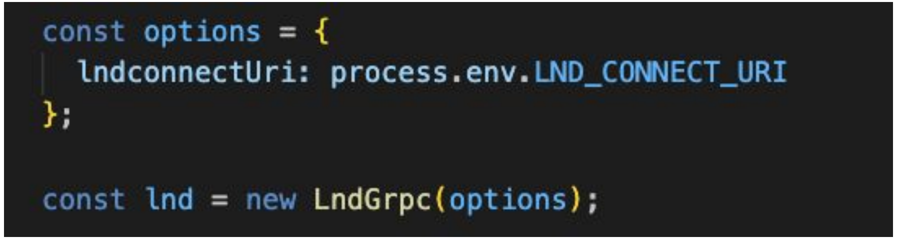
Voltage can attempt to open a channel, so your node can receive sats!

Learn more about inbound liquidity

Update our connection options to use LNDConnect

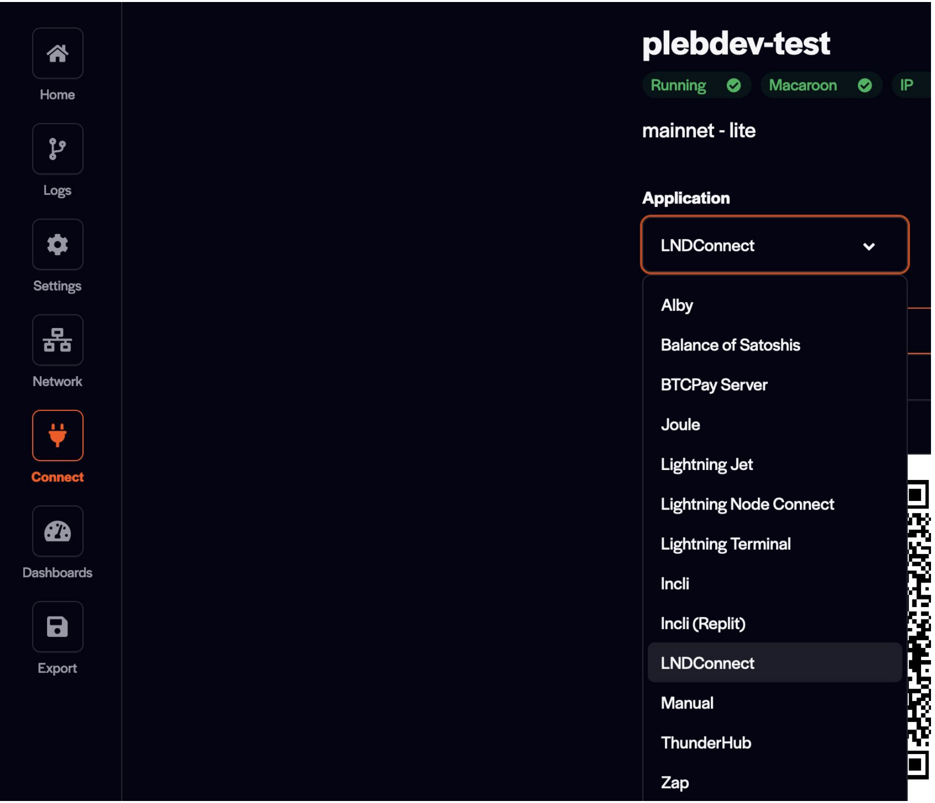
Lightning Node Connect (LNC) is a novel mechanism to create an end-to-end encrypted connection between a Lightning Network node (initially just LND, other implementations might follow) and a web browser.

This will replace our HOST, CERT, and MACAROON env variables and combine all of that information into a single URI

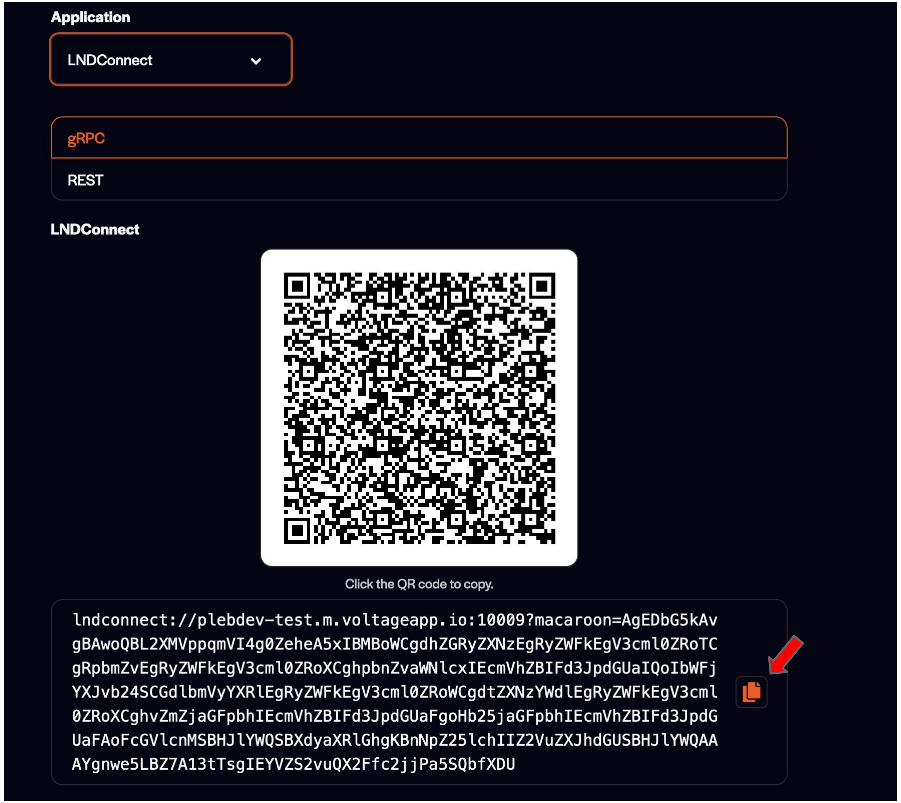


```
const options = {
  lndconnectUri: process.env.LND_CONNECT_URI
};

const lnd = new LndGrpc(options);
```

Grab a new LNDConnect URI from the Voltage connections dropdown



Application: LNDConnect



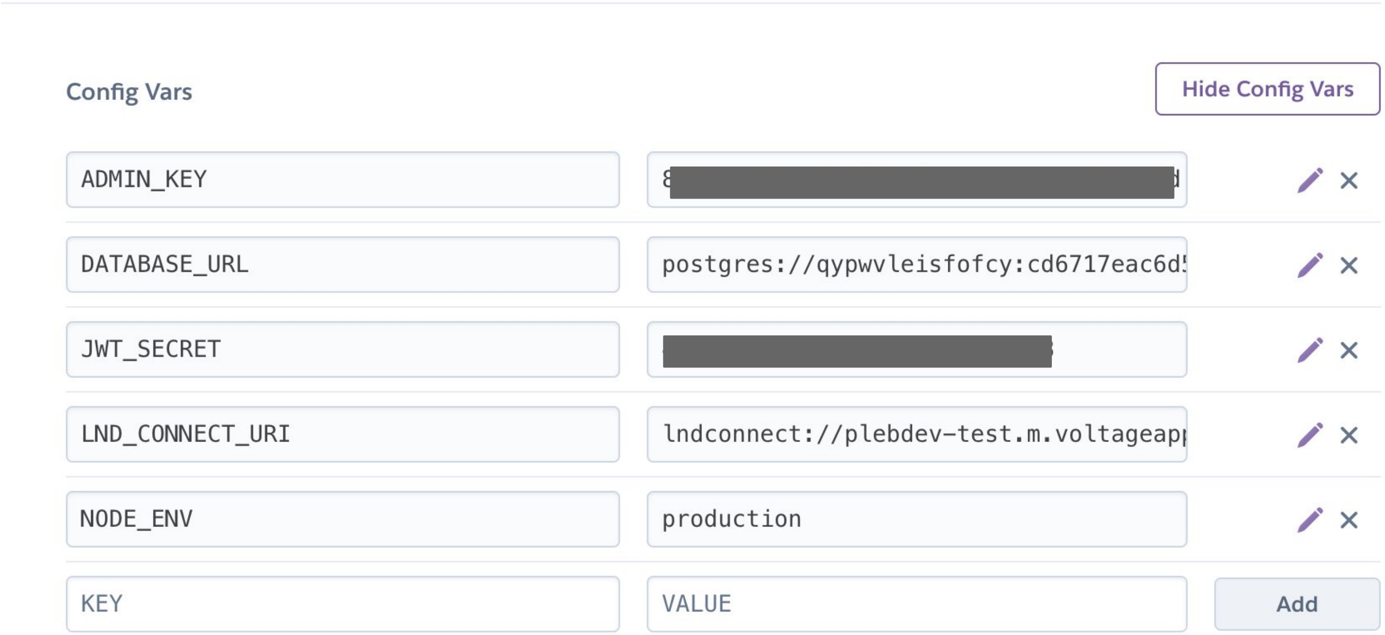
Application: LNDConnect

LNDConnect

```
lndconnect://plebdev-test.m.voltageapp.io:10009?macaroon=AgEDbG5kAv
gBAwoQBL2XMVppqmVI4g0ZeheA5xIBMBoWCgdhZGRyZXNzEgRyZWFkEgV3cml0ZRoTC
gRpbmZvEgRyZWFkEgV3cml0ZRoXCghpbnZvaWNlcxIEcmVhZBIFd3JpdGUaIQoIbWFj
YXJvb24SCGdlbmVyYXRlEgRyZWFkEgV3cml0ZRoWCgdtZXNzYWdlEgRyZWFkEgV3cml
0ZRoXCghvZmZjaGFpbhIEcmVhZBIFd3JpdGUaFgoHb25jaGFpbhIEcmVhZBIFd3JpdG
UaFAoFcGVlcnMSBHJlYWQSBXdyaXRlGhgKBnNpZ25lchIIZ2VuZXJhdGUSBHJlYWQAA
AYgnwe5LBZ7A13tTsgIEYVZS2vuQX2Ffc2jjPa5SQbfXDU
```

Add LND_CONNECT_URI to ENV variables

This should be the full set off all of your ENV variables now

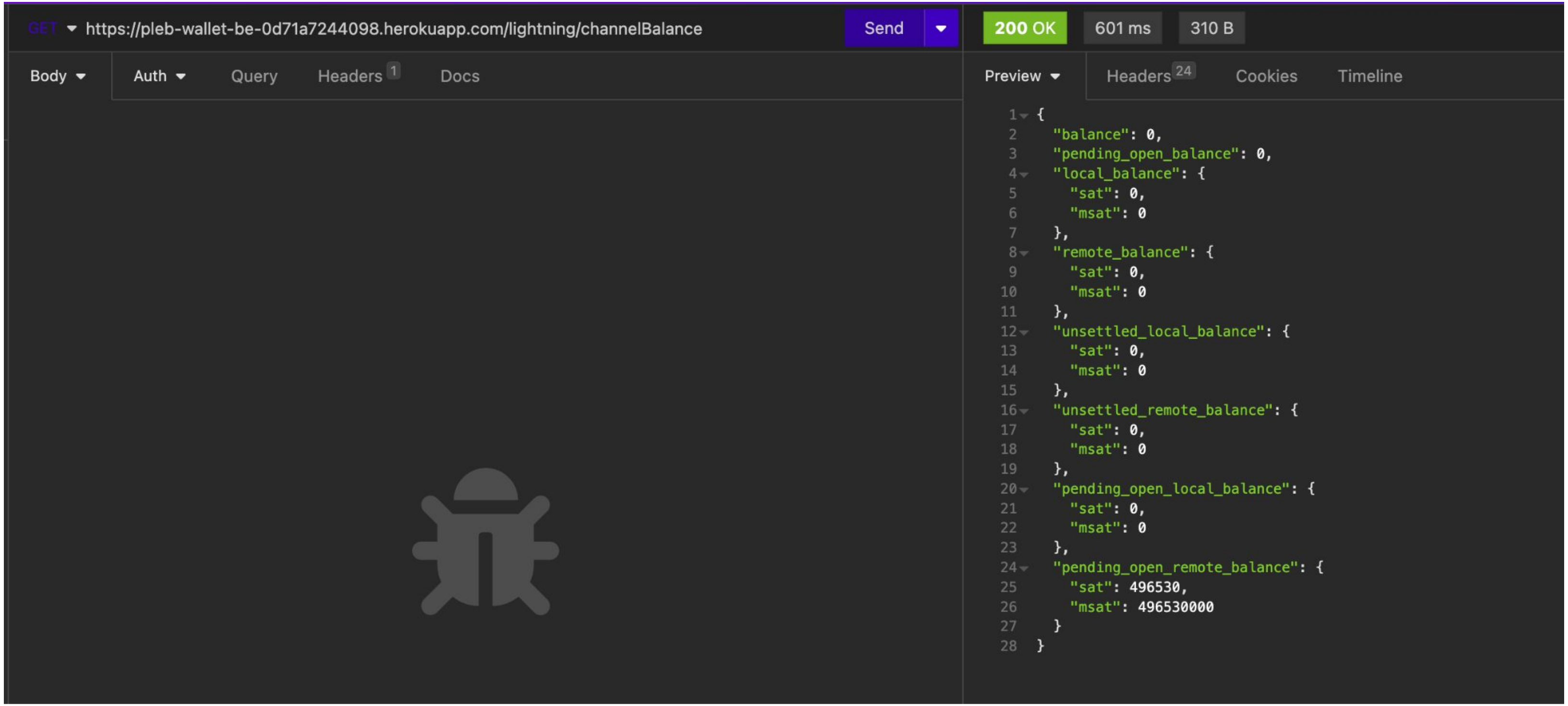


Config Vars

ADMIN_KEY	[redacted]
DATABASE_URL	postgres://qypwvleisfofcy:cd6717eac6d
JWT_SECRET	[redacted]
LND_CONNECT_URI	lndconnect://plebdev-test.m.voltageap
NODE_ENV	production
KEY	VALUE

Redeploy and test a lightning endpoint

After redeploying we should now be able to call our /lightning/channelBalance endpoint



```
{
  "balance": 0,
  "pending_open_balance": 0,
  "local_balance": {
    "sat": 0,
    "msat": 0
  },
  "remote_balance": {
    "sat": 0,
    "msat": 0
  },
  "unsettled_local_balance": {
    "sat": 0,
    "msat": 0
  },
  "unsettled_remote_balance": {
    "sat": 0,
    "msat": 0
  },
  "pending_open_local_balance": {
    "sat": 0,
    "msat": 0
  },
  "pending_open_remote_balance": {
    "sat": 496530,
    "msat": 496530000
  }
}
```

Our production Server / DB / Node are deployed!

Everything is deployed in production and working!

Final step testing with the deployed frontend

- Deploy https://github.com/AustinKelsay/pleb-wallet-frontend to vercel
- Add the REACT_APP_BACKEND_URL env variable on the vercel deployment and put it as the deployed backend
- Go through the full flow testing an admin and non admin user on the deployed frontend