LAB 5 Applications Interface for Accessing the Blockchain

- Introduction
  - In the previous Labs, we had established the structure of blockchain data. This Lab introduces the creation of an API for the subsequent Labs. The API will provide means for transaction, communication or interaction with blockchain and creation of decentralized Ledgers.
- Objective
  - Creation of an API from scratch
- Prerequisite
  - Completion of all previous Labs
- Materials
  Internet source, data and browser.
- Setup, Methods and Procedures
  - Inside the directory dev, create a file api.js
  - Download Express.js (https://www.npmjs.com/package/express)

    npm i express –save
  - create a script vpi.js and type

```
let  express = require('express');
// assign an instance or object of express to our app
let app = express();
// create an endpoint which is just /
app.get('/', function (req, res) {
    // send message to the listening port that will get the
    // alert
    res.send('Hello Veritas World');
    });
// the server is listening to port 3000
app.listen(3000);
```

  - Pre-test of the basic server structure: run the vpi.js on the command line

    node dev/vpi.js
  - and access it from the browser .

    http://localhost:3000
  - Next, create some other endpoints to the blockchain, transaction, and mining

```
const express = require('express');
const app = express();
app.get('/blockchain', function (req, res) {
});
app.post('/transaction', function(req, res) {
});
app.get('/mine', function(req, res) {
```

```
});
app.listen(3000, function(res, req){
        console.log('listening on port 3000... ');
});
```

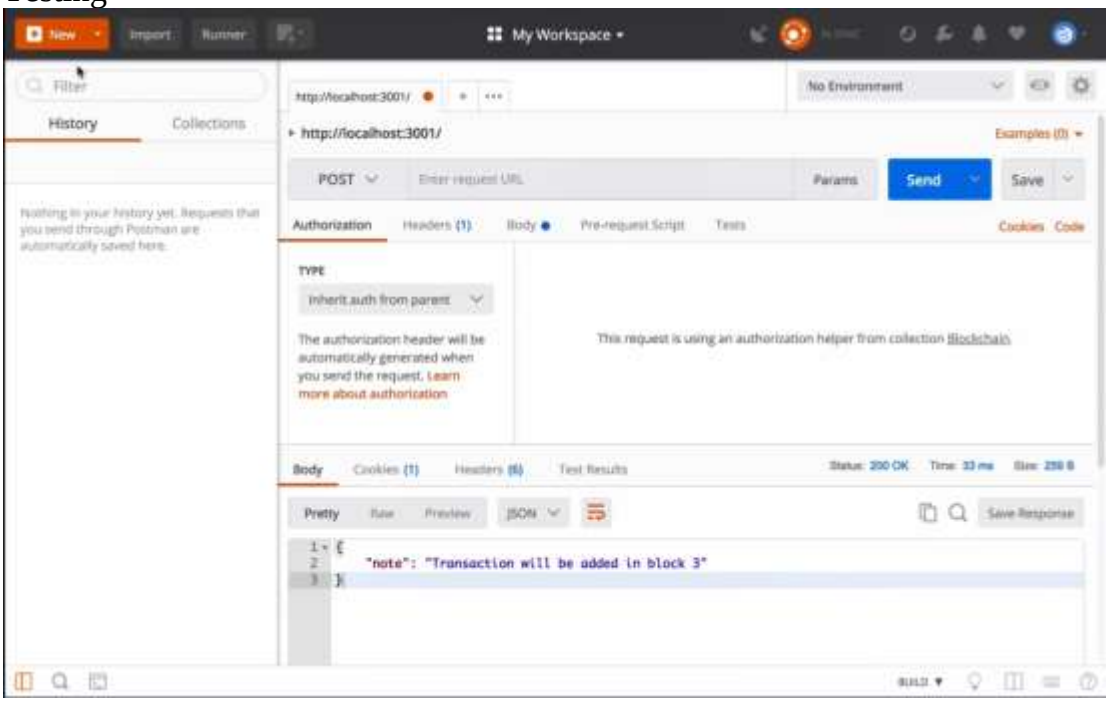
- On the CMD run the vpi.js file

  node dev/vpi.js
- We have to reduce the stress of always running to CMD, by automating the CMD to keep listening to the changes in the scripts, a package known as nodemon helps in this task

  npm i nodemon –save
- To use nodemon, open the package.json script and include the line inside the script tag

  "start": "nodemon --watch dev -e js dev/vpi.js"
- WHERE THE CMD WILL RUN THE VPI.JS WITH NODE ANYTIME START IS RUN

  npm start
- Also to stop running type Control C on your keyboard, likewise to restart without
- To make easy calls, update, send data and retrieve data to and from the server endpoint request to the blockchains, we install a tool called Postman:
- Download Postman @ https://www.getpostman.com and test using the /transaction endpoint

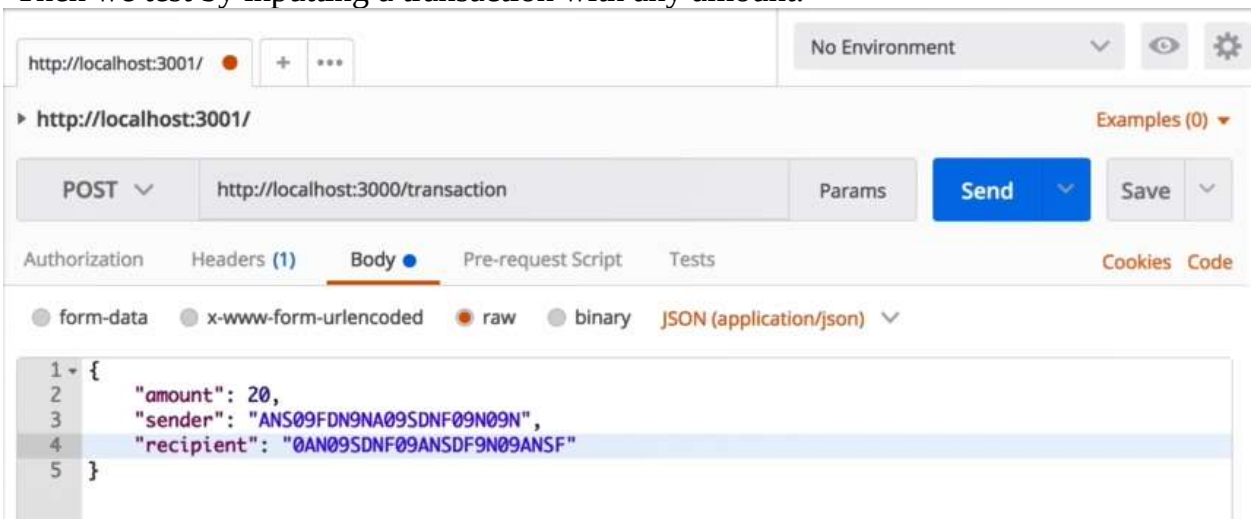
- Testing
- 

- On the Enter url request row enter

http://localhost:3000/transaction

- Update the /transaction endpoint and test

  app.post('/transaction', function(req, res) {
  res.send('It works!!!');
  });
- The on the Postman, click the SEND button.
- We can see the log in the CMD. However, to see the result in the Postman environment, in a more pretty output, we select 'raw', and 'Body', and 'json'. Then we test by inputting a transaction with any amount.



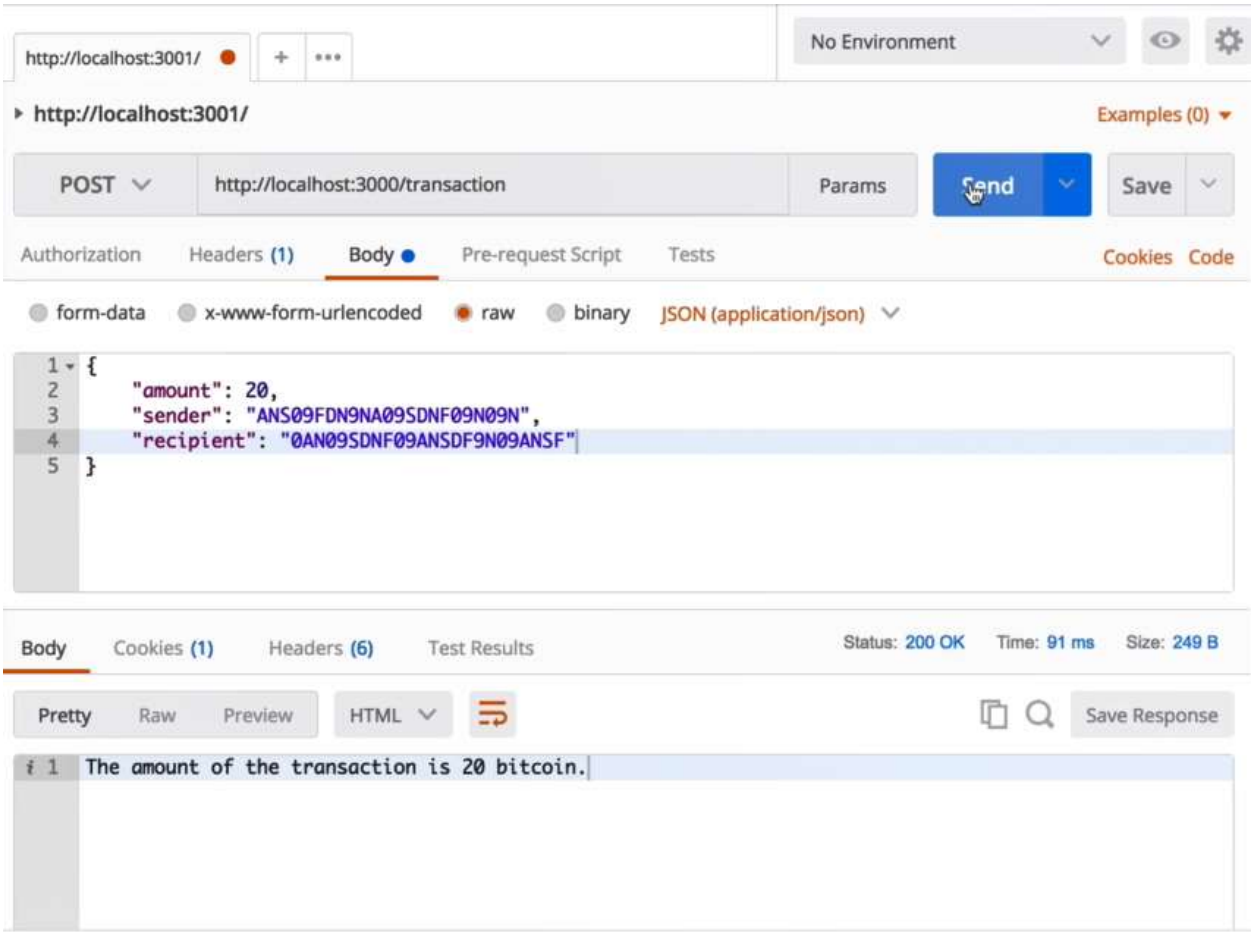
- Update the /transaction endpoint on the vpi.js script, and add data similar to the represented version on the Postman app,

  app.post('/transaction', function(req, res) {
  console.log(req.body);
  res.send(`The amount of the transaction is
  ${req.body.amount}
  veroin.`);
  });
- Next to get a pretty formatted data on the browser, install a body-parser package as follows

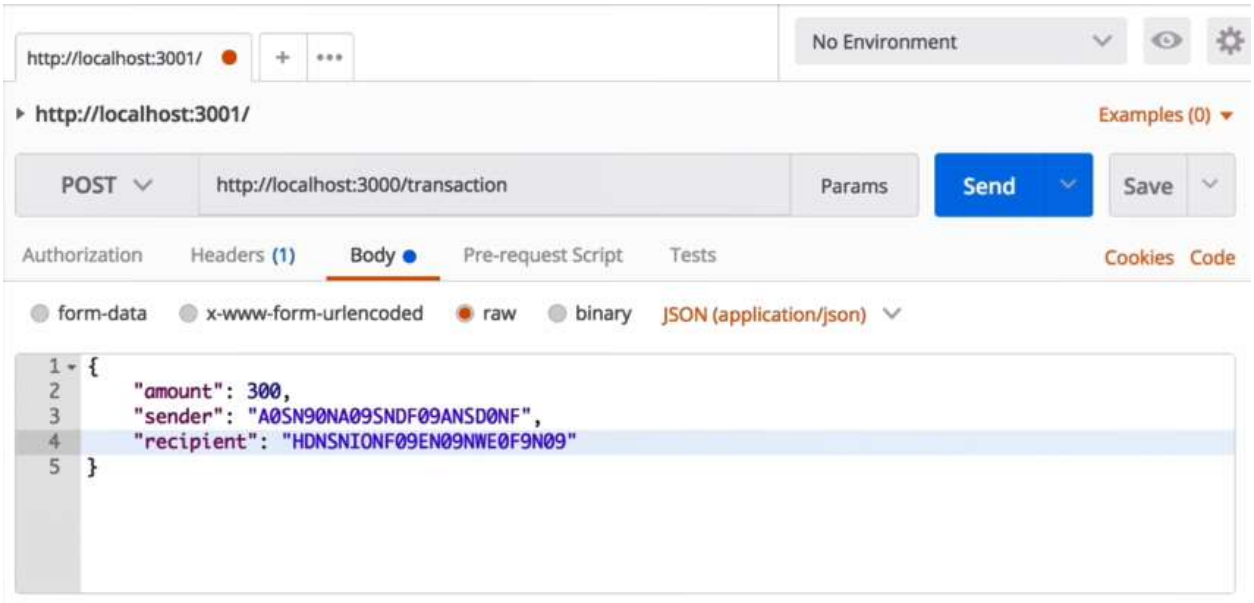
  npm i body-parser –save
- Also update the vpi.js with the body-parser instance

const bodyParser = require('body-parser');
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: false }));

- Now, on the Postman click Send realize the value of amount from the sender



- On the browser, let us test the blockchain endpoint by typing
  localhost:3000/blockchain
- Building a transaction endpoint in the vpi.js and code
  app.post('/transaction', function(req, res) {
  const blockIndex =
  veroin.createNewTransaction(req.body.amount,
  req.body.sender, req.body.recipient)
  res.json({ note:`Transaction will be added in block ${blockIndex}.`});
  });
- In the Postman, click send



- Subsequently, check for the solution on the browser using *http://localhost:3000/blockchain*

- Add more transactions and test

- Troubleshooting
  - Ensure check for typo, and install all tools as required
- Conclusion
  - We have built the API, BLOCKCHAIN ENDPOINT and transaction. The test using Postman was successful and next is to start mining of the blocks.
- Report, Evaluation and Confirmation

LABORATORY REPORT

Course Code: _____
Course Title: _____
Date of Practical: _____/_____/_____ Time of Practical: _____
Lab. Session: _____
Lab Session Title: _____

Practical Work Done:

_____

Experience gained

Problems Encountered:

LABORATORY EXERCISE /ASSIGNMENT EVALUATION

Note: Grades are out of 6, based on the Lecturer's / Lab. Technologist's evaluation of your demo / goal / task / understanding, divided as follows.

0: Not done		2: Late Complete		4: Complete	
1: Incomplete		3: Needs improvement		5: Well Done	

CONFIRMATION

Signature of the Instructor / Lab. Technologist:		Date: _____/_____/_____