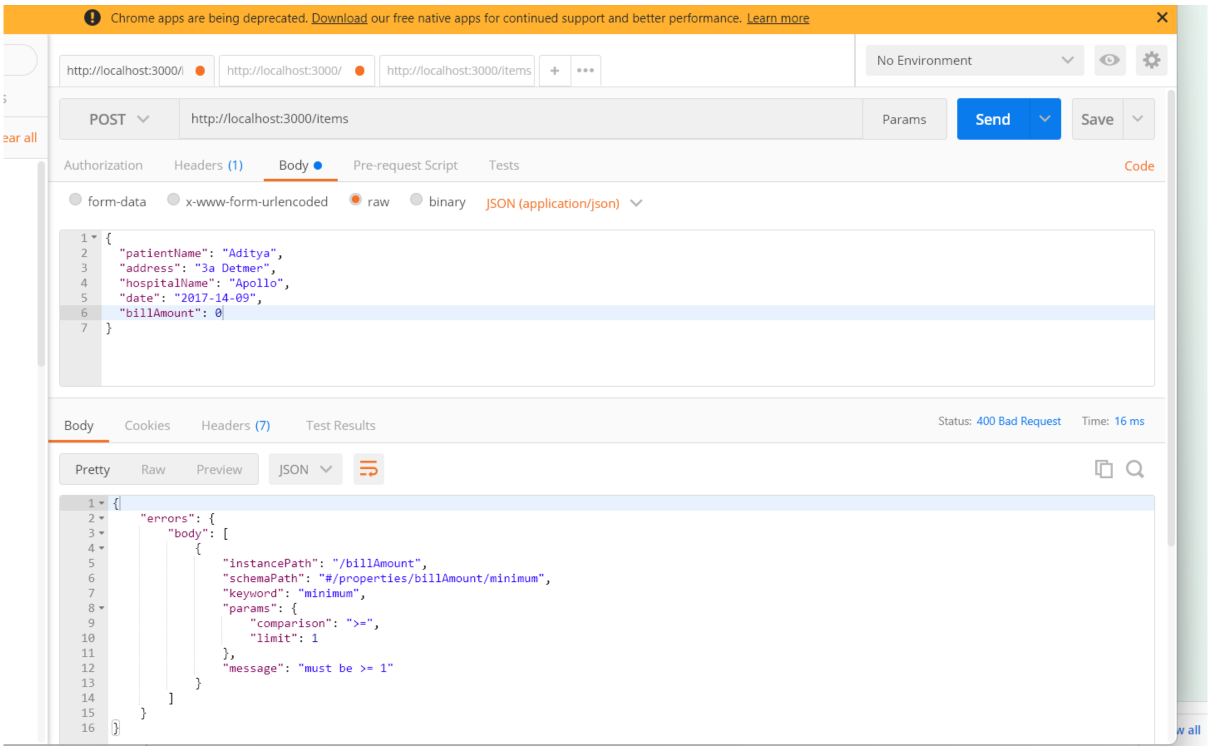
Request Validation: (using `express-json-validator-middleware`)

I've added two Mandatory fields (`"patientName"`, `"hospitalName"`) and minimum length of string as 1.

Request Validation : Bill amount to be > 0.



POST http://localhost:3000/items

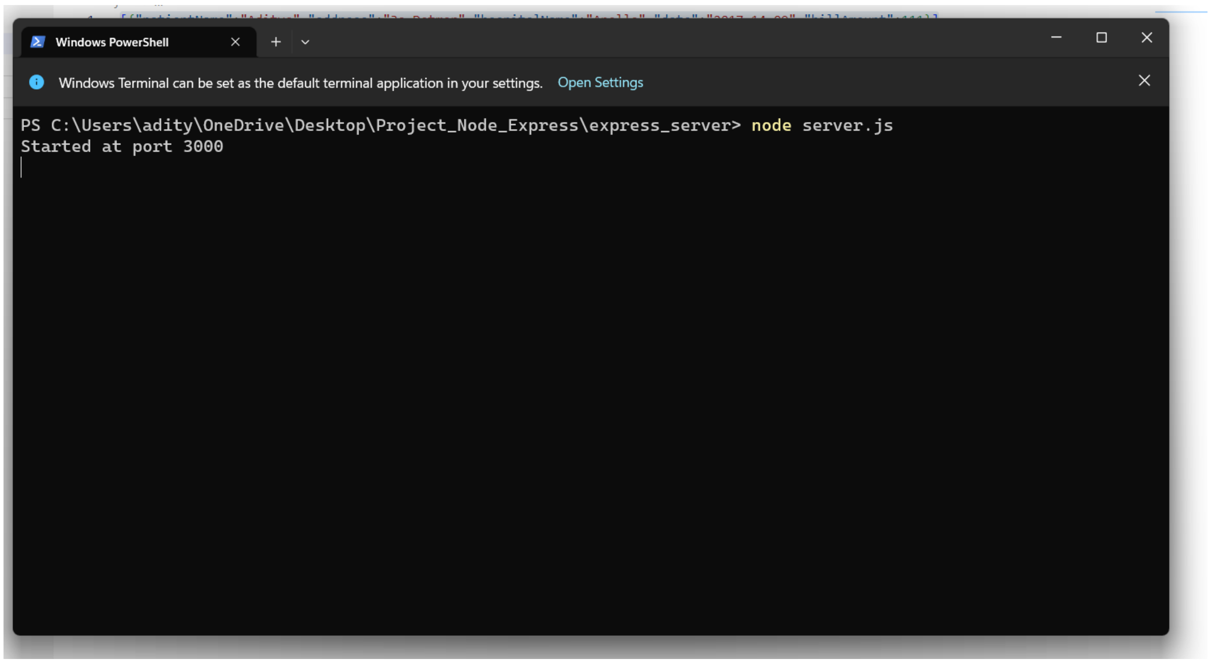
Body: raw, JSON (application/json)

```
{
  "patientName": "Aditya",
  "address": "3a Detmer",
  "hospitalName": "Apollo",
  "date": "2017-14-09",
  "billAmount": 0
}
```

Status: 400 Bad Request   Time: 16 ms

```
{
    "errors": {
        "body": [
            {
                "instancePath": "/billAmount",
                "schemaPath": "#/properties/billAmount/minimum",
                "keyword": "minimum",
                "params": {
                    "comparison": ">=",
                    "limit": 1
                },
                "message": "must be >= 1"
            }
        ]
    }
}
```

Started Server: (node server.js)

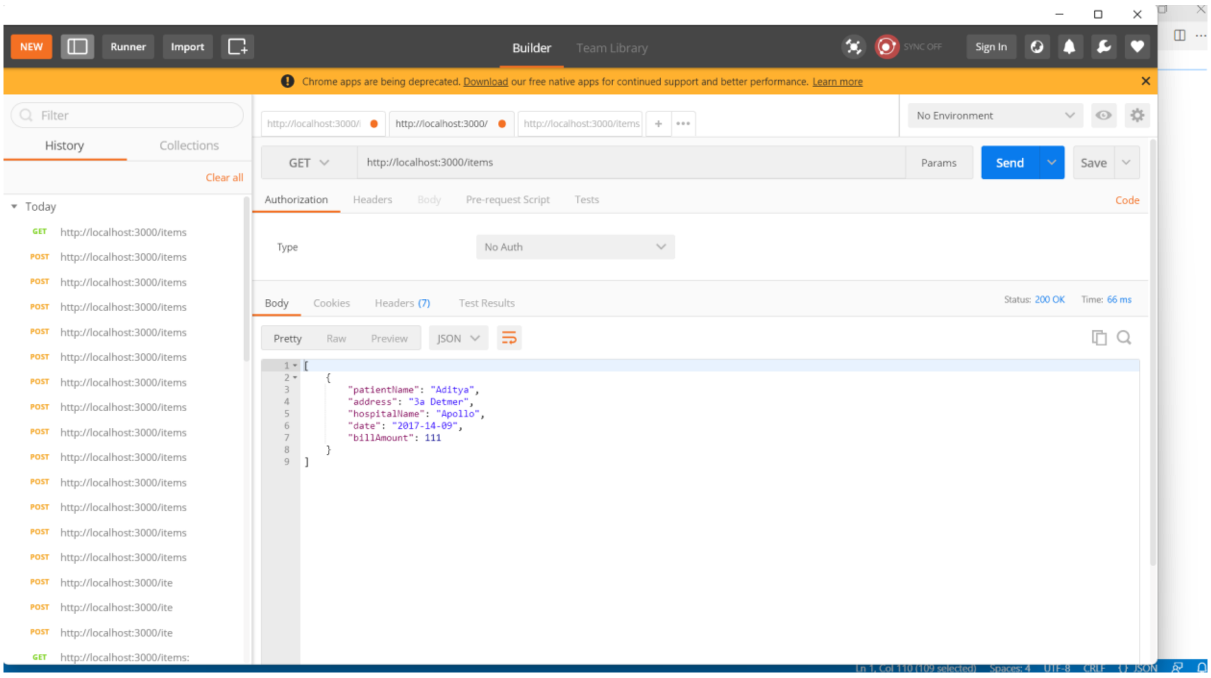


Windows Terminal can be set as the default terminal application in your settings. Open Settings

```
PS C:\Users\adity\OneDrive\Desktop\Project_Node_Express\express_server> node server.js
Started at port 3000
```

Get Endpoint:

http://localhost:3000/items


Chrome apps are being deprecated. Download our free native apps for continued support and better performance. Learn more

GET http://localhost:3000/items

Authorization — Type: No Auth

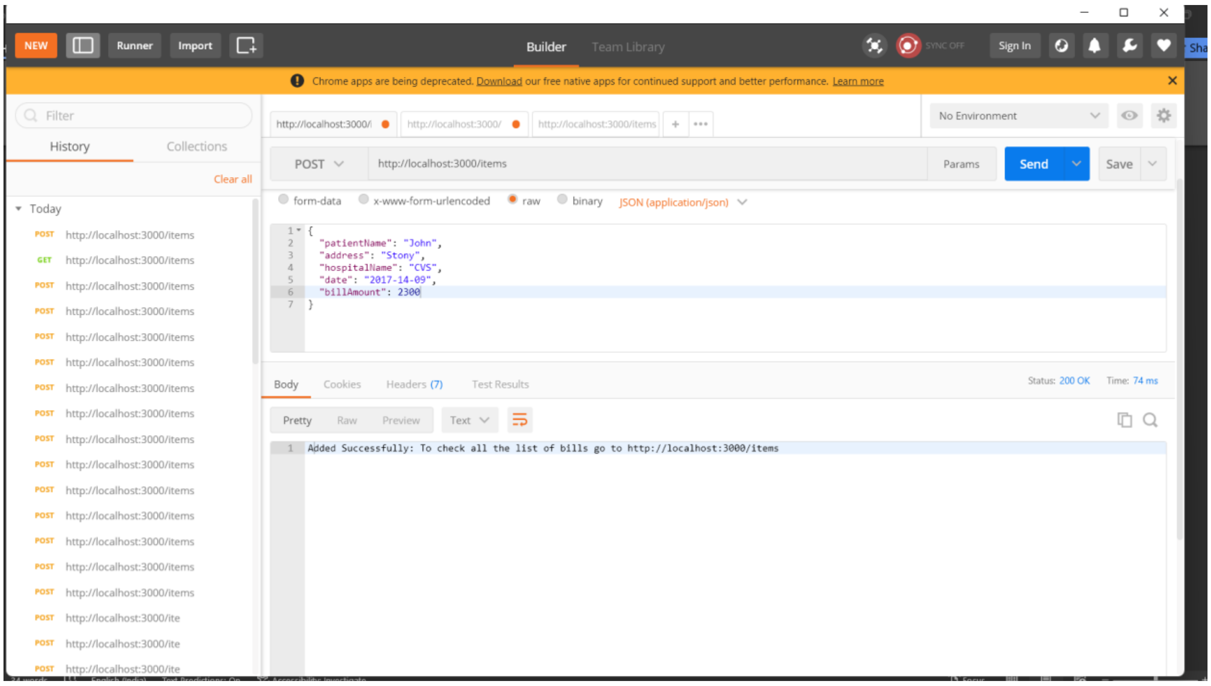
Body — Status: 200 OK Time: 66 ms

Pretty JSON

```
[
    {
        "patientName": "Aditya",
        "address": "3a Detmer",
        "hospitalName": "Apollo",
        "date": "2017-14-09",
        "billAmount": 111
    }
]
```

Post Endpoint:

http://localhost:3000/items -> request body -> json of bill information to be passes


Chrome apps are being deprecated. Download our free native apps for continued support and better performance. Learn more

POST http://localhost:3000/items

raw — JSON (application/json)

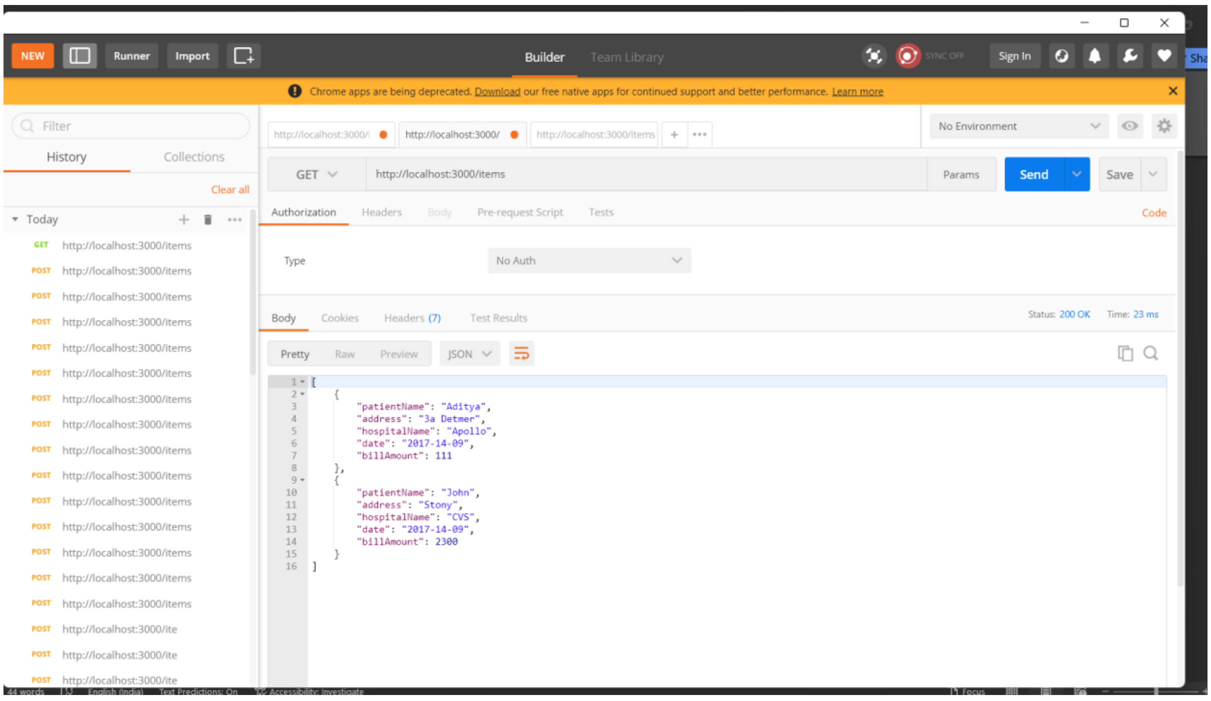
```
{
  "patientName": "John",
  "address": "Stony",
  "hospitalName": "CVS",
  "date": "2017-14-09",
  "billAmount": 2300
}
```

Body — Status: 200 OK Time: 74 ms

Pretty Text

```
Added Successfully: To check all the list of bills go to http://localhost:3000/items
```

Again, Get Endpoint To see the above saved bill information:



GET http://localhost:3000/items

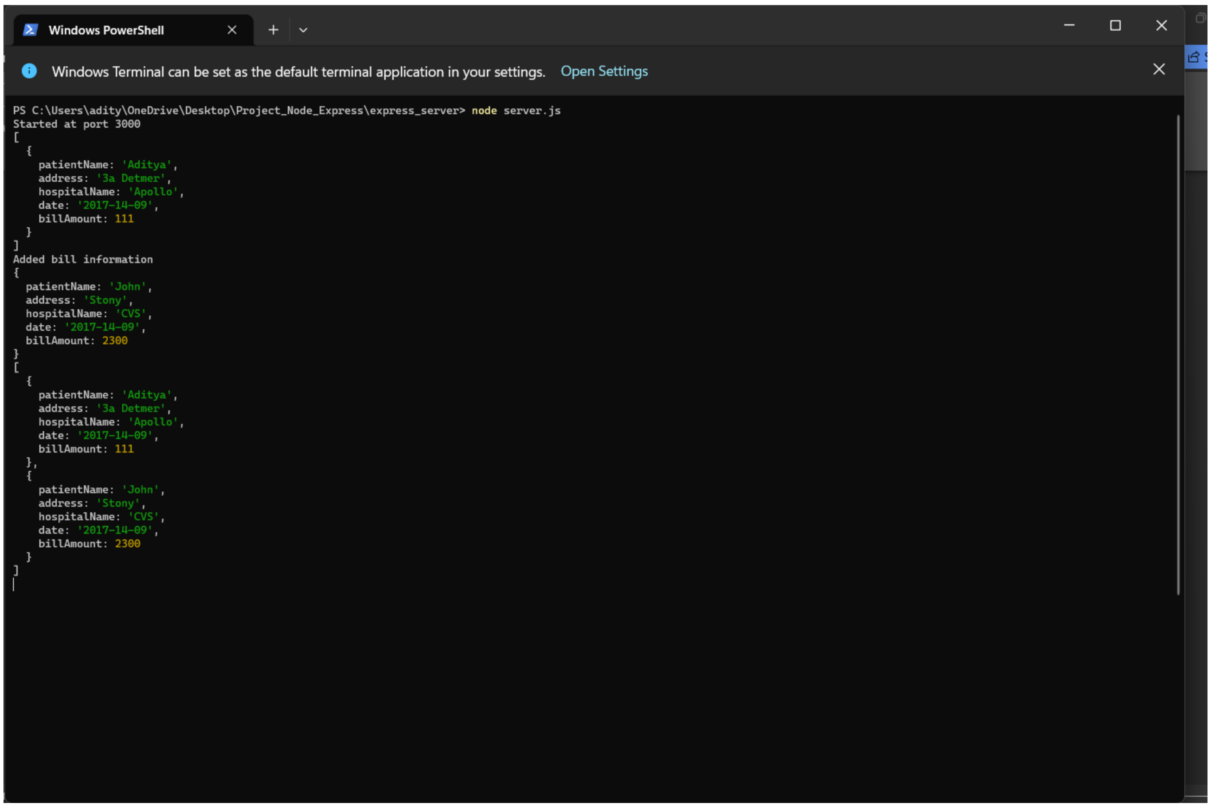
Type: No Auth

Status: 200 OK Time: 23 ms

```
GET   http://localhost:3000/items
POST  http://localhost:3000/items
POST  http://localhost:3000/items
POST  http://localhost:3000/items
POST  http://localhost:3000/items
POST  http://localhost:3000/items
POST  http://localhost:3000/items
POST  http://localhost:3000/items
POST  http://localhost:3000/items
POST  http://localhost:3000/items
POST  http://localhost:3000/items
POST  http://localhost:3000/items
POST  http://localhost:3000/items
POST  http://localhost:3000/items
POST  http://localhost:3000/items
POST  http://localhost:3000/ite
POST  http://localhost:3000/ite
POST  http://localhost:3000/ite
```

```json
[
    {
        "patientName": "Aditya",
        "address": "3a Detmer",
        "hospitalName": "Apollo",
        "date": "2017-14-09",
        "billAmount": 111
    },
    {
        "patientName": "John",
        "address": "Stony",
        "hospitalName": "CVS",
        "date": "2017-14-09",
        "billAmount": 2300
    }
]
```

Terminal with all 3 calls (First GET, then POST and then GET):


Windows PowerShell
Windows Terminal can be set as the default terminal application in your settings.
Open Settings

```
PS C:\Users\adity\OneDrive\Desktop\Project_Node_Express\express_server> node server.js
Started at port 3000
[
  {
    patientName: 'Aditya',
    address: '3a Detmer',
    hospitalName: 'Apollo',
    date: '2017-14-09',
    billAmount: 111
  }
]
Added bill information
{
  patientName: 'John',
  address: 'Stony',
  hospitalName: 'CVS',
  date: '2017-14-09',
  billAmount: 2300
}
[
  {
    patientName: 'Aditya',
    address: '3a Detmer',
    hospitalName: 'Apollo',
    date: '2017-14-09',
    billAmount: 111
  },
  {
    patientName: 'John',
    address: 'Stony',
    hospitalName: 'CVS',
    date: '2017-14-09',
    billAmount: 2300
  }
]
```

Simple Logic:

Created one file at the same path “bills.json”, storing information of all the bills,

For GET API call, it reads all the list of bills from the file

For POST API call, it updates the file by adding one more item in the bills.json files.

Source Code:

File - > Server.js:

```
var express = require("express");
var app = express();
var bodyParser = require("body-parser");
const fs = require('fs');

app.use(bodyParser.json());


const {
    Validator,
    ValidationError,
} = require("express-json-validator-middleware");

const { validate } = new Validator();

function validationErrorMiddleware(error, request, response, next) {
    if (response.headersSent) {
        return next(error);
    }

    const isValidationError = error instanceof ValidationError;
    if (!isValidationError) {
        return next(error);
    }
    response.status(400).json({
        errors: error.validationErrors,
    });

    next();
}

const userSchema = {
    type: "object",
    required: ["patientName", "hospitalName"],
    properties: {
        patientName: {
            type: "string",
            minLength: 1,
        },
        address: {
            type: "string",
            minLength: 1,
        },
        hospitalName: {
            type: "string",
            minLength: 1,
        },
        date: {
            type: "string",
            minLength: 1,
        },
        billAmount: {
            type: "integer",
            minimum: 1,
        },
    },
};

app.listen(3000, function(){
    console.log("Started at port 3000")
});

app.post('/items', validate({ body: userSchema }), function (req, res, next){
    res.setHeader('Content-Type', 'application/json');
    console.log("Added bill information")
    console.log(req.body);
    res.send("Added Successfully: To check all the list of bills go to
http://localhost:3000/items ");

    var item = req.body;
    var data = fs.readFileSync('bills.json', 'utf8');
    var list = (data.length) ? JSON.parse(data): [];
    if (list instanceof Array) list.push(item)
    else list = [item]
    fs.writeFileSync('bills.json', JSON.stringify(list));
    next();
    });

app.use(validationErrorMiddleware);

app.get("/items", function(req,res){
    res.setHeader('Content-Type', 'application/json');
    fs.readFile('bills.json', (err, data) => {
        if (err) throw err;
        let bills = JSON.parse(data);
        console.log(bills);
        res.send(bills);

    });
})
```

File -> Package.Json

```
{
  "name": "express_server",
  "version": "1.0.0",
  "description": "demo",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Aditya",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.20.1",
    "express": "^4.18.2",
    "express-json-validator-middleware": "^3.0.1"
  }
}
```

File -> bills.json

```
[{"patientName":"Aditya","address":"3a
Detmer","hospitalName":"Apollo","date":"2017-14-
09","billAmount":111},{"patientName":"John","address":"Stony","hospitalName":"
CVS","date":"2017-14-09","billAmount":2300}]
```