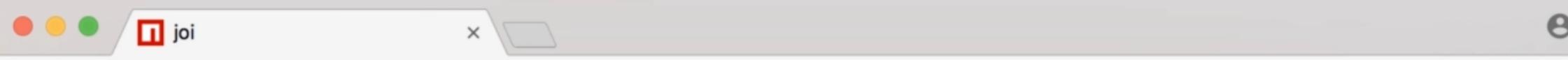


Input Validation



Mosh Hamedani

programmingwithmosh.com



_apps For quick access, place your bookmarks here on the bookmarks bar. [Import bookmarks now...](#)

Nil Packet Minimizer

npm Enterprise features pricing documentation support



find packages



sign up or log in



We need your input. Help make JavaScript better: [Take the 2017 JavaScript Ecosystem survey »](#)

joi public



joi



npm install joi

how? learn more



marsup published 2 we...

13.1.0 is the latest of 178 re...

github.com/hapijs/joi



index.js - Applications - Visual Studio Code

File Edit Selection View Go Run Terminal Help



EXPLORER

OPEN EDITORS

JS index.js 11. Input Validation

APPLICATIONS

- > 01. Introduction
- > 02. RESTful Services
- > 03. Introducing Express
- > 04. Building Your First Web Server
- > 05. Nodemon
- > 06. Environment Variables
- > 07. Route Parameters
- > 08. Handling HTTP GET Requests
- > 09. Handling HTTP POST Requests
- > 10. Calling Endpoints Using Postman
- > 11. Input Validation
- > express-demo
 - { } package.json
 - JS index.js
 - npm-debug.log
- > 12. Handling HTTP PUT Requests
- > 13. Handling HTTP Delete Requests
- > 14. Project- Build the Genres API
- > 15. Building RESTful APIs with Express Recap

JS index.js

11. Input Validation > JS index.js > ...

```
1  const express = require('express');
2  const app = express();
3  app.use(express.json());
4  // express. json() is a method inbuilt in express to recognize the incoming Request Object as a JSON Object.
5  // This method is called as a middleware in your application using the code: app.
6  const courses=[
7      {id: 1, name: 'course1'},
8      {id: 2, name: 'course2'},
9      {id: 3, name: 'course3'}
10 ];
11 app.get('/',(req, res)=>{
12     res.send('Hello World !!! Its corona-time');
13 });
14 app.get('/api/courses',(req, res)=>{
15     res.send(courses);
16 });
Complexity is 4 Everything is cool!
17 app.post('/api/courses', (req, res) =>{ █
18     if(!req.body.name || req.body.name.length < 3){
19         // 400 Bad Request
20         res.status(400).send(' Name is required and should be minimum 3 characters.');
21         return;
22     }
23     const course = {
24         id: courses.length + 1,
25         name: req.body.name
26     };
27     courses.push(course);
28     res.send(course);
29 });
Complexity is 3 Everything is cool!
30 app.get('/api/courses/:id',(req, res)=>{ █
31     const course = courses.find(c => c.id === parseInt(req.params.id));
32     if(!course) res.status(404).send('The course with the given ID was not found.');
33     res.send(course);
34 });
//PORT
35 const port = process.env.PORT || 3000;//Initialize PORT in the terminal : export PORT=3333
36 app.listen(port,() => console.log(`Listening to port ${port} ....`));
37 // To add environment variable use :
38 // Ubuntu : export PORT=3333
39 // Windows : SET PORT=3333
```

> OUTLINE

git master* ↻ ⌂ ⌂ ⌂ ⌂ ⌂ javascript | ✓ index.js

Ln 40, Col 29 Spaces: 4 UTF-8 LF Babel JavaScript JavaScript Standard Style

index.js - Applications - Visual Studio Code

The screenshot shows a code editor with the following interface elements:

- File Edit Selection View Go Run Terminal**: The top navigation bar.
- EXPLORER**: A sidebar pane on the left containing:
 - OPEN EDITORS**:
 - `index.js` 11. Input Validation (marked with a red X)
 - APPLICATIONS**:
 - > 01. Introduction
 - > 02. RESTful Services
 - > 03. Introducing Express
 - > 04. Building Your First Web Server
 - > 05. Nodemon
 - > 06. Environment Variables
 - > 07. Route Parameters
 - > 08. Handling HTTP GET Requests
 - > 09. Handling HTTP POST Requests
 - > 10. Calling Endpoints Using Postman
 - 11. Input Validation** (selected, indicated by a grey border)
 - express-demo**:
 - { } package.json
 - `index.js`
 - npm-debug.log
 - > 12. Handling HTTP PUT Requests
 - > 13. Handling HTTP Delete Requests
 - > 14. Project- Build the Genres API
 - > 15. Building RESTful APIs with Express Recap

OUTLINE: A bottom pane showing the outline of the current file, `index.js`.

```
dandy@My-Ubuntu:~/Keep/Node.js/04. Building RESTful API_s Using Express/Applications/11. Input Validation/express-demo$ sudo npm init -yes
[sudo] password for dandy:
Wrote to /home/dandy/Keep/Node.js/04. Building RESTful API_s Using Express/Applications/11. Input Validation/express-demo/package.json:

{
  "name": "express-demo",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

dandy@My-Ubuntu:~/Keep/Node.js/04. Building RESTful API_s Using Express/Applications/11. Input Validation/express-demo$ sudo npm i joi
npm WARN deprecated joi@14.3.1: This module has moved and is now available at @hapi/joi. Please update your dependencies as this version is no longer maintained and may contain bugs and security issues.
npm WARN deprecated topo@3.0.3: This module has moved and is now available at @hapi/topo. Please update your dependencies as this version is no longer maintained and may contain bugs and security issues.
npm WARN deprecated hoek@6.1.3: This module has moved and is now available at @hapi/hoek. Please update your dependencies as this version is no longer maintained and may contain bugs and security issues.
express-demo@1.0.0 /home/dandy/Keep/Node.js/04. Building RESTful API_s Using Express/Applications/11. Input Validation/express-demo
└── joi@14.3.1  extraneous

npm WARN express-demo@1.0.0 No description
npm WARN express-demo@1.0.0 No repository field.
dandy@My-Ubuntu:~/Keep/Node.js/04. Building RESTful API_s Using Express/Applications/11. Input Validation/express-demo$ sudo npm i joi@13.1.0
npm WARN deprecated joi@13.1.0: This version has been deprecated in accordance with the hapi support policy (hapi.im/support). Please upgrade to the latest version to get the best features, bug fixes, and security patches. If you are unable to upgrade at this time, paid support is available for older versions (hapi.im/commercial).
npm WARN deprecated topo@3.0.3: This module has moved and is now available at @hapi/topo. Please update your dependencies as this version is no longer maintained and may contain bugs and security issues.
npm WARN deprecated hoek@5.0.4: This version has been deprecated in accordance with the hapi support policy (hapi.im/support). Please upgrade to the latest version to get the best features, bug fixes, and security patches. If you are unable to upgrade at this time, paid support is available for older versions (hapi.im/commercial).
npm WARN deprecated hoek@6.1.3: This module has moved and is now available at @hapi/hoek. Please update your dependencies as this version is no longer maintained and may contain bugs and security issues.
express-demo@1.0.0 /home/dandy/Keep/Node.js/04. Building RESTful API_s Using Express/Applications/11. Input Validation/express-demo
└── joi@13.1.0  extraneous

npm WARN express-demo@1.0.0 No description
npm WARN express-demo@1.0.0 No repository field.
dandy@My-Ubuntu:~/Keep/Node.js/04. Building RESTful API_s Using Express/Applications/11. Input Validation/express-demo$
```

master* ② ✘ 0 ⚠ 0 ✓ javascript | ✓ index.js

Ln 40, Col 29 Spaces: 4 UTF-8 LF Babel JavaScript JavaScript Standard Style

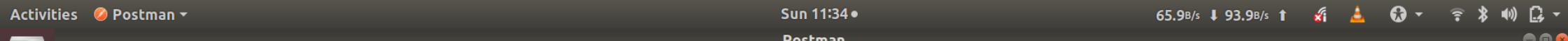
index.js - Applications - Visual Studio Code

```
11. Input Validation > JS index.js > ...
1   const Joi = require('joi'); // It returns a class that why following nomenclature for class.
2   const express = require('express');
3   const app = express();
4   app.use(express.json());
5   // express.json() is a method inbuilt in express to recognize the incoming Request Object as a JSON Object.
6   // This method is called as a middleware in your application using the code: app.
7   const courses=[
8       {id: 1, name: 'course1'},
9       {id: 2, name: 'course2'},
10      {id: 3, name: 'course3'}
11  ];
12  app.get('/',(req, res)=>{
13      res.send('Hello World !!! Its corona-time');
14  });
15  app.get('/api/courses',(req, res)=>{
16      res.send(courses);
17  });
Complexity is 4 Everything is cool!
18  app.post('/api/courses', (req, res) =>{ █
19      const schema = {
20          name: Joi.string().min(3).required()
21      };
22      const result = Joi.validate(req.body, schema);
23      if(!req.body.name || req.body.name.length < 3){
24          // 400 Bad Request
25          res.status(400).send('Name is required and should be minimum 3 characters.');
26          return;
27      }
28      const course = {
29          id: courses.length + 1,
30          name: req.body.name
31      };
32      courses.push(course);
33      res.send(course);
34  });
Complexity is 3 Everything is cool!
35  app.get('/api/courses/:id',(req, res)=>{ █
36      const course = courses.find(c => c.id === parseInt(req.params.id));
37      if(!course) res.status(404).send('The course with the given ID was not found.');
38      res.send(course);
39  });
//PORT
40  const port = process.env.PORT || 3000;//Initialize PORT in the terminal : export PORT=3333
41  app.listen(port,() => console.log(`Listening to port ${port} ....`));
42  // To add environment variable use :
43  // Ubuntu : export PORT=3333
44  // Windows : SET PORT=3333
```

index.js - Applications - Visual Studio Code

```
1 Input Validation > JS index.js > ...
2   const Joi = require('joi'); // It returns a class that why following nomenclature for class.
3   const express = require('express');
4   const app = express();
5   app.use(express.json());
6   // express.json() is a method inbuilt in express to recognize the incoming Request Object as a JSON Object.
7   // This method is called as a middleware in your application using the code: app.
8   const courses=[
9     {id: 1, name: 'course1'},
10    {id: 2, name: 'course2'},
11    {id: 3, name: 'course3'}
12  ];
13  app.get('/',(req, res)=>{
14    res.send('Hello World !!! Its corona-time');
15  });
16  app.get('/api/courses',(req, res)=>{
17    res.send(courses);
18  });
19 Complexity is 4 Everything is cool!
20 app.post('/api/courses', (req, res) =>{ █
21   const schema = {
22     name: Joi.string().min(3).required()
23   };
24   const result = Joi.validate(req.body, schema);
25   console.log(result);
26   if(!req.body.name || req.body.name.length < 3){
27     // 400 Bad Request
28     res.status(400).send('Name is required and should be minimum 3 characters.');
29     return;
30   }
31   const course = {
32     id: courses.length + 1,
33     name: req.body.name
34   };
35   courses.push(course);
36   res.send(course);
37 });
38 Complexity is 3 Everything is cool!
39 app.get('/api/courses/:id',(req, res)=>{ █
40   const course = courses.find(c => c.id === parseInt(req.params.id));
41   if(!course) res.status(404).send('The course with the given ID was not found.');
42   res.send(course);
43 });
44 //PORT
45 const port = process.env.PORT || 3000;//Initialize PORT in the terminal : export PORT=3333
46 app.listen(port,() => console.log(`Listening to port ${port} ....`));
47 // To add environment variable use :
48 // Ubuntu : export PORT=3333
49 // Windows : SET PORT=3333
50 // To Kill the activity on the port 3000 :sudo kill -9 $(sudo lsof -t -i:9001)
```

```
DEBUG CONSOLE PROBLEMS 1 OUTPUT TERMINAL  
dandy@My-Ubuntu:~/Keep/Node.js/04. Building RESTful API's Using Express/Applications/11. Input Validation$ export PORT=3333  
dandy@My-Ubuntu:~/Keep/Node.js/04. Building RESTful API's Using Express/Applications/11. Input Validation$ nodemon index.js  
[nodemon] 2.0.2  
[nodemon] to restart at any time, enter `rs`  
[nodemon] watching dir(s): **.*  
[nodemon] watching extensions: js,mjs,json  
[nodemon] starting `node index.js'  
Listening on port 3333 ....
```



Activities Postman

Sun 11:34 •

Postman

File Edit View Help

New Import Runner

My Workspace Invite

Filter

History Collections APIs

+ New API

Untitled Request

POST http://localhost:3333/api/courses

Send Save

No Environment

Comments 0

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

Beautify

No APIs yet

APIs define related collections and environments under a consistent schema.

Create an API

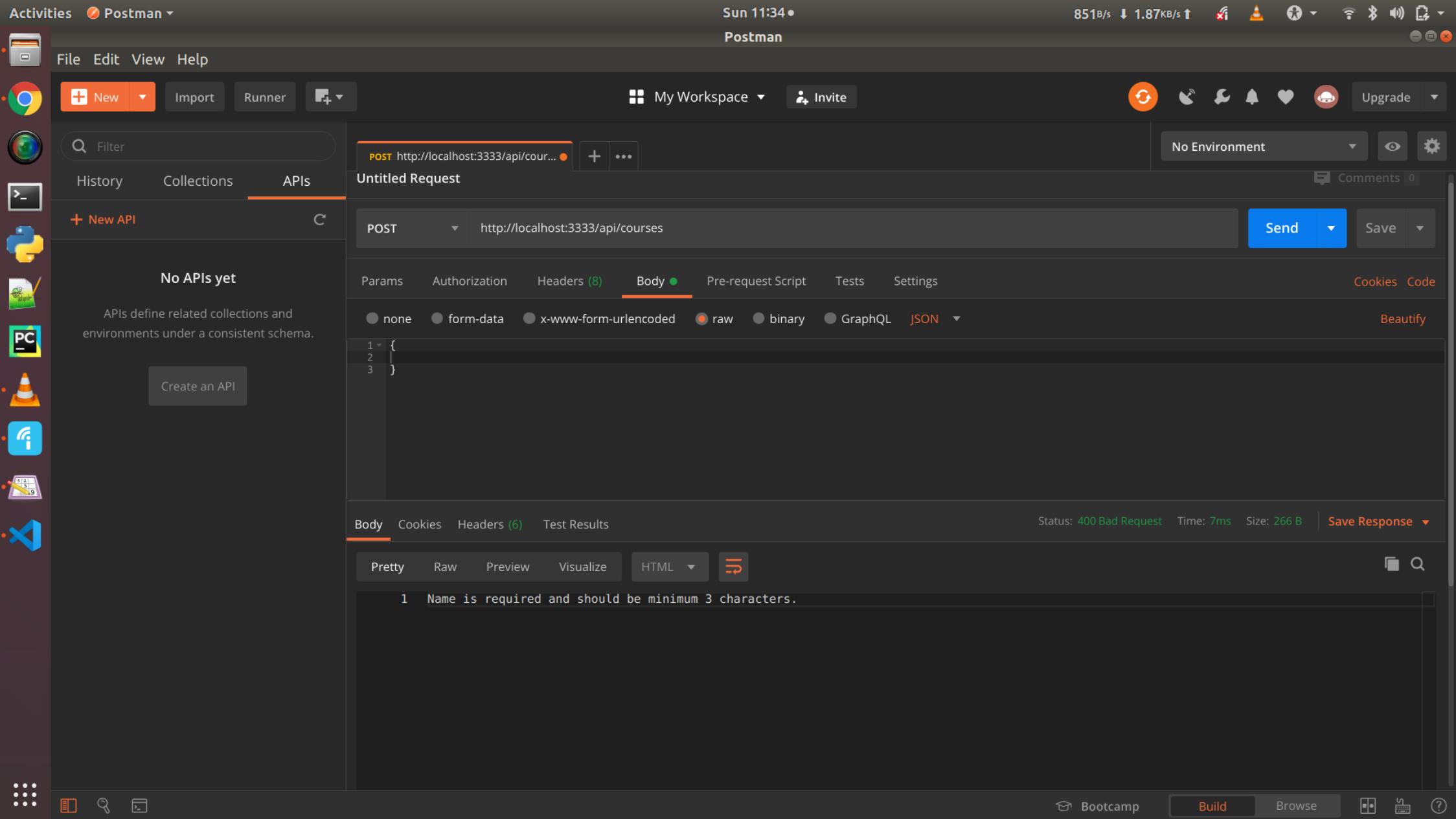
Body Cookies Headers (6) Test Results Status: 200 OK Time: 21ms Size: 243 B Save Response

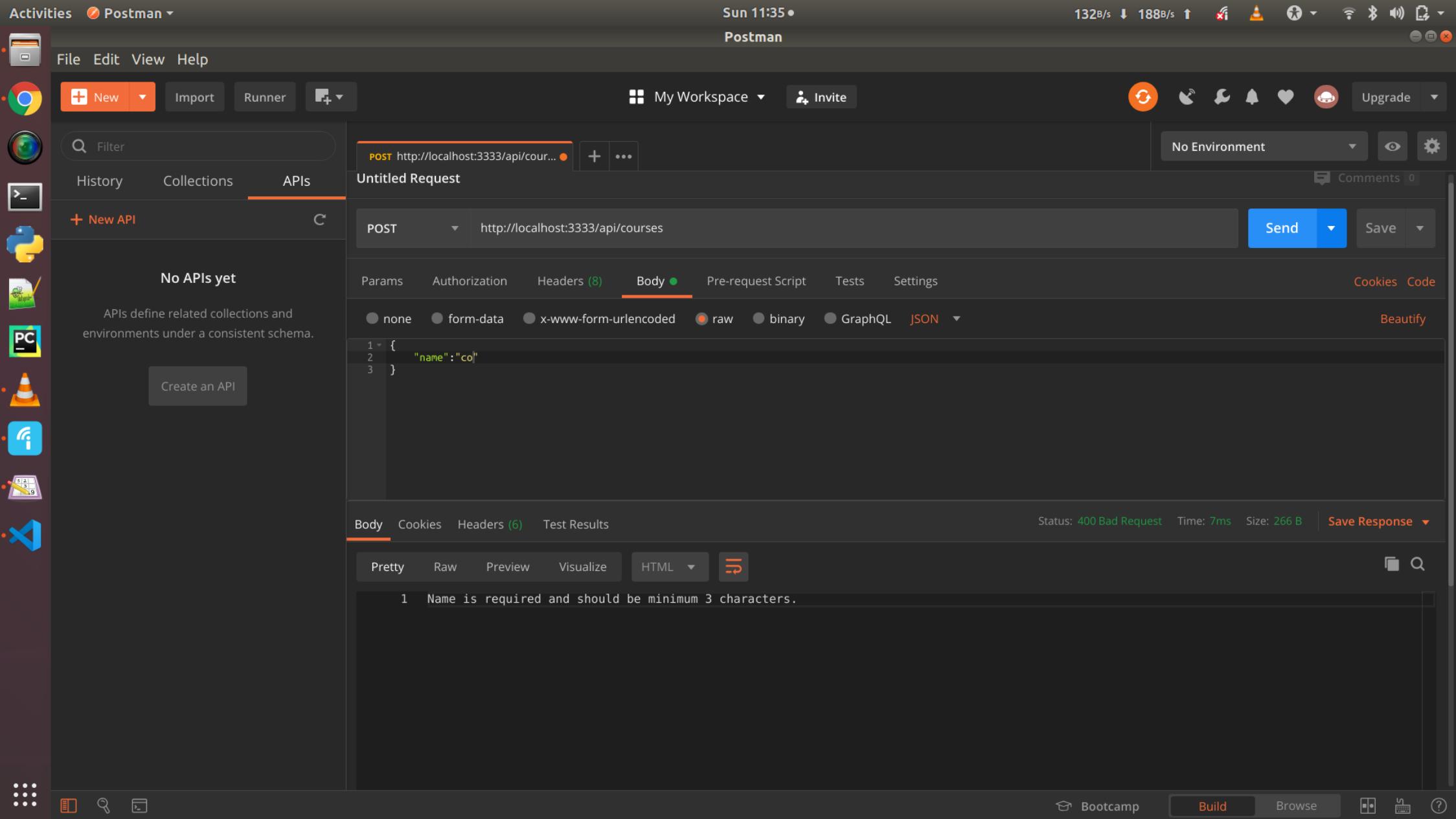
Pretty Raw Preview Visualize JSON

```
1 {  
2   "id": 5,  
3   "name": "course-nodejs"  
4 }
```

Bootcamp Build Browse

The screenshot shows the Postman application interface. On the left, there's a sidebar with various icons for different tools like a browser, terminal, and code editor. The main workspace is titled 'Untitled Request' and shows a POST request to 'http://localhost:3333/api/courses'. The 'Body' tab is selected, displaying a JSON payload: { "id": 5, "name": "course-nodejs" }. Below the body, the response is shown with a status of 200 OK, a response time of 21ms, and a size of 243 B. The response body is also displayed in pretty-printed JSON format. The top right corner shows network activity: 65.9B/s down and 93.9B/s up. The bottom right corner has buttons for 'Bootcamp', 'Build', and 'Browse'.



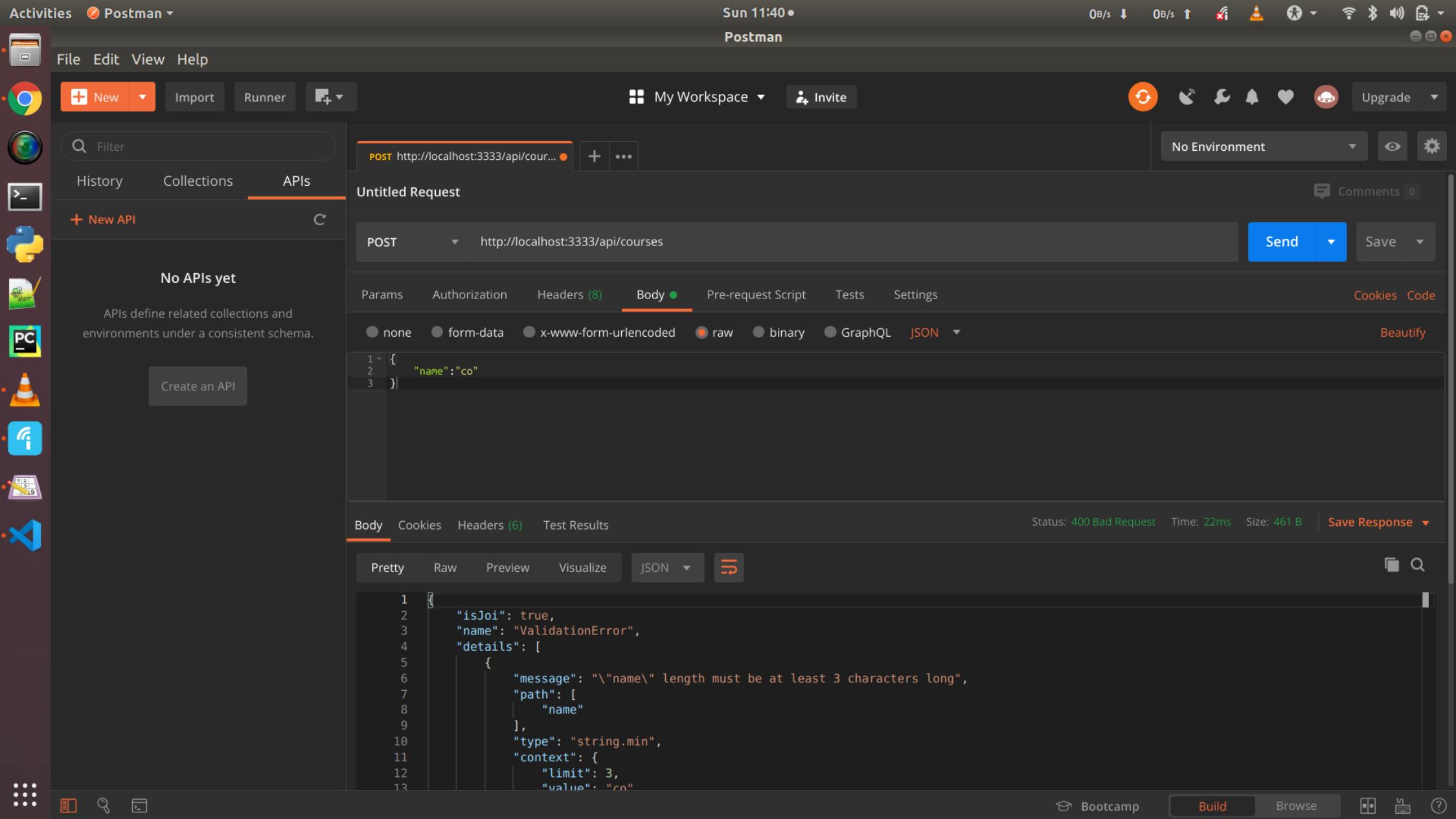


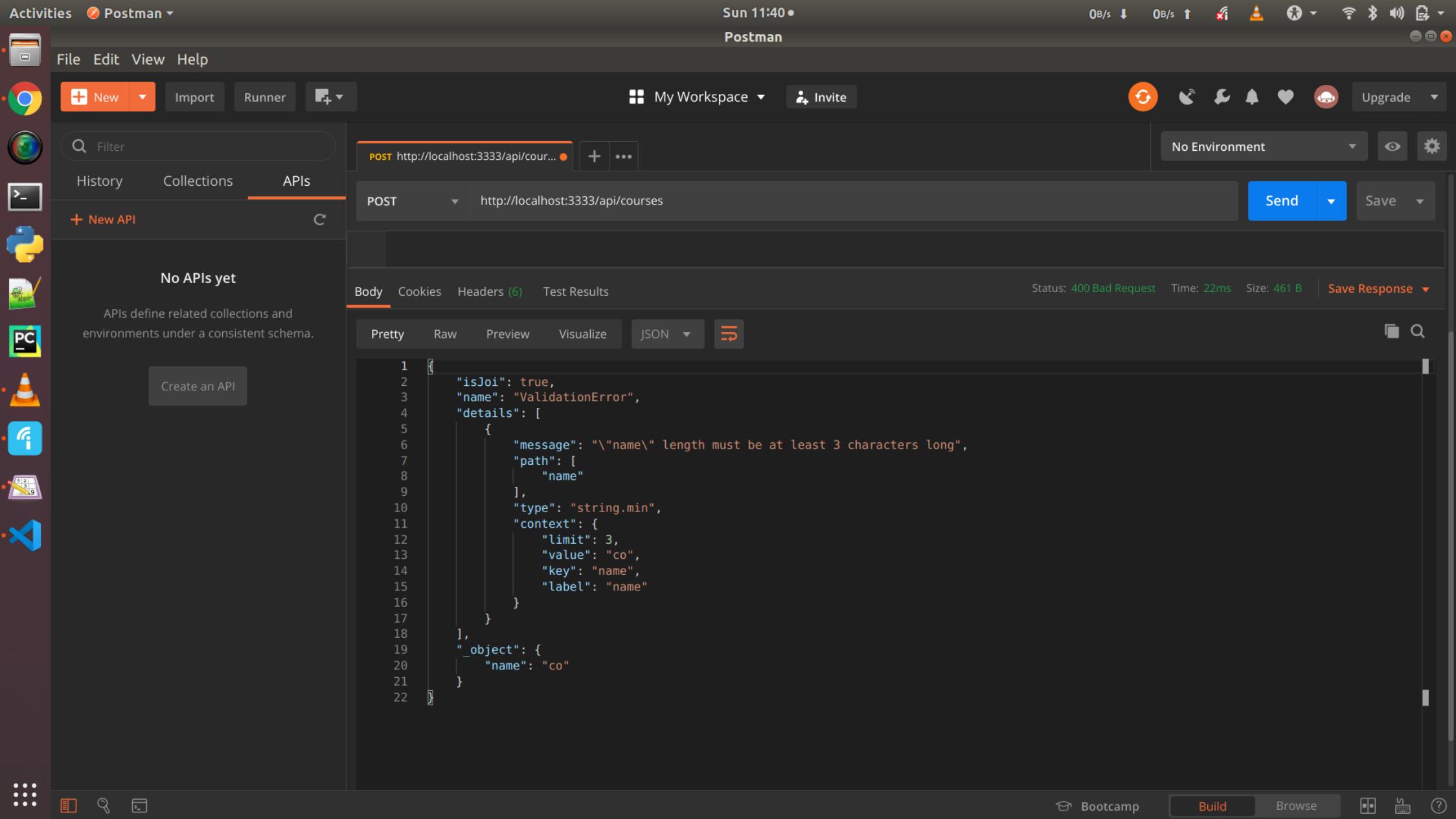
index.js - Applications - Visual Studio Code

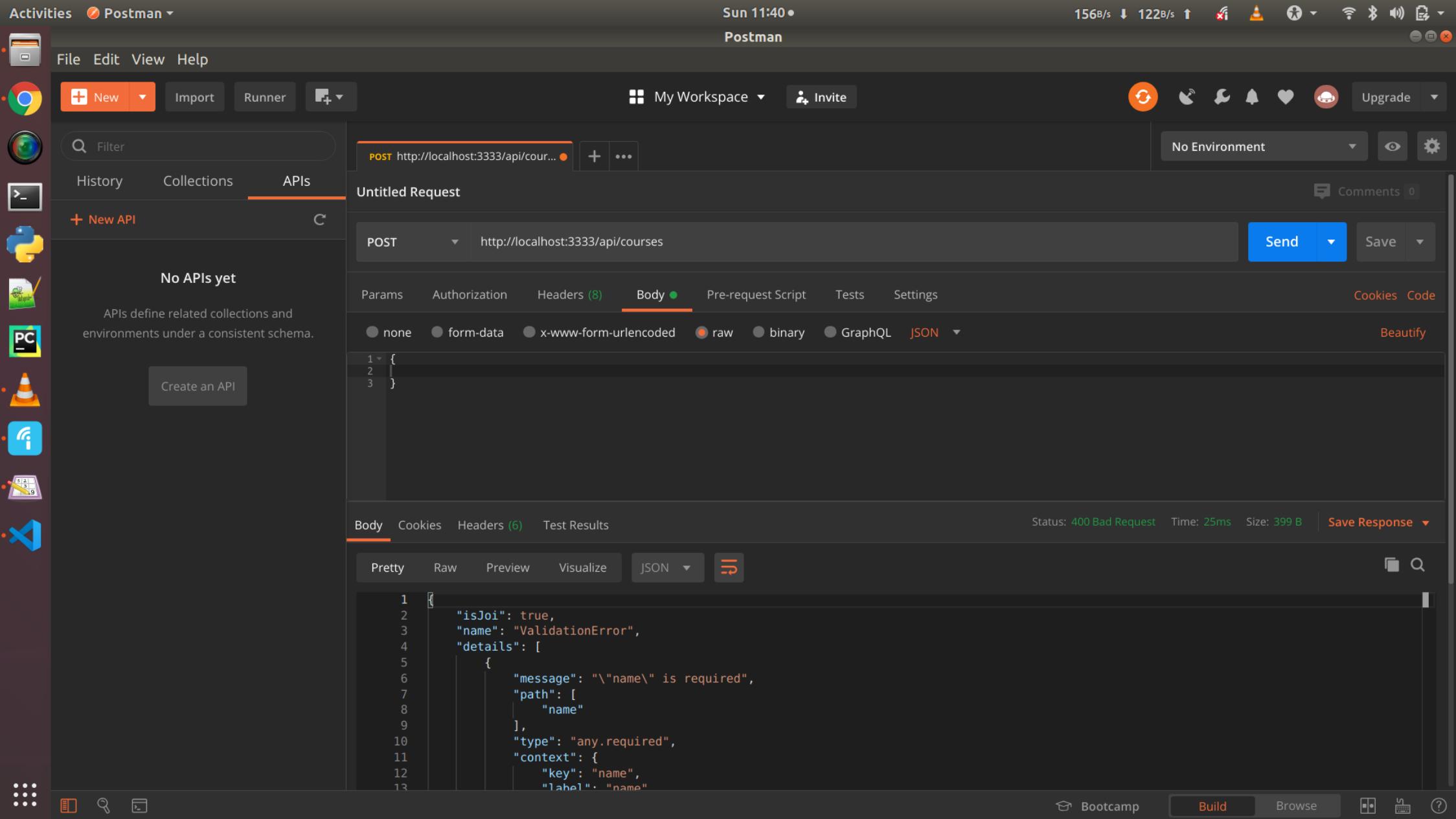
```
dandy@My-Ubuntu:~/Keep/Node.js/04. Building RESTful API_s Using Express/Applications/11. Input Validation$ export PORT=3333
dandy@My-Ubuntu:~/Keep/Node.js/04. Building RESTful API_s Using Express/Applications/11. Input Validation$ nodemon index.js
[nodemon] 2.0.2
[nodemon] to restart at any time, enter `rs`
[nodemon] watching dir(s): ***!
[nodemon] watching extensions: js,mjs,json
[nodemon] starting node index.js
Listening to port 3333 ....
{
  error: null,
  value: { name: 'course-nodejs' },
  then: [Function: then],
  catch: [Function: catch]
}
{ error:
  {
    ValidationError: child "name" fails because ["name" is required]
    at Object.exports.process (/home/dandy/node_modules/joi/lib/errors.js:203:19)
    at internals.Object._validateWithOptions (/home/dandy/node_modules/joi/lib/types/any/index.js:764:31)
    at module.exports.internals.Any.root.validate (/home/dandy/node_modules/joi/lib/index.js:147:23)
    at app.post (/home/dandy/Keep/Node.js/04. Building RESTful API_s Using Express/Applications/11. Input Validation/index.js:22:24)
    at Layer.handle [as handle_request] (/home/dandy/node_modules/express/lib/router/layer.js:95:5)
    at next (/home/dandy/node_modules/express/lib/router/route.js:137:13)
    at Route.dispatch (/home/dandy/node_modules/express/lib/router/route.js:112:3)
    at Layer.handle [as handle_request] (/home/dandy/node_modules/express/lib/router/layer.js:95:5)
    at /home/dandy/node_modules/express/lib/router/index.js:281:22
    at Function.process_params (/home/dandy/node_modules/express/lib/router/index.js:335:12)
      isJoi: true,
      name: 'ValidationError',
      details: [ [Object] ],
      _object: {},
      annotate: [Function] ,
    value: {},
    then: [Function: then],
    catch: [Function: catch]
  }
  error:
  {
    ValidationError: child "name" fails because ["name" length must be at least 3 characters long]
    at Object.exports.process (/home/dandy/node_modules/joi/lib/errors.js:203:19)
    at internals.Object._validateWithOptions (/home/dandy/node_modules/joi/lib/types/any/index.js:764:31)
    at module.exports.internals.Any.root.validate (/home/dandy/node_modules/joi/lib/index.js:147:23)
    at app.post (/home/dandy/Keep/Node.js/04. Building RESTful API_s Using Express/Applications/11. Input Validation/index.js:22:24)
    at Layer.handle [as handle_request] (/home/dandy/node_modules/express/lib/router/layer.js:95:5)
    at next (/home/dandy/node_modules/express/lib/router/route.js:137:13)
    at Route.dispatch (/home/dandy/node_modules/express/lib/router/route.js:112:3)
    at Layer.handle [as handle_request] (/home/dandy/node_modules/express/lib/router/layer.js:95:5)
    at /home/dandy/node_modules/express/lib/router/index.js:281:22
    at Function.process_params (/home/dandy/node_modules/express/lib/router/index.js:335:12)
      isJoi: true,
      name: 'ValidationError',
      details: [ [Object] ],
      _object: { name: 'co' },
      annotate: [Function] ,
    value: { name: 'co' },
    then: [Function: then],
    catch: [Function: catch]
  }
}
```

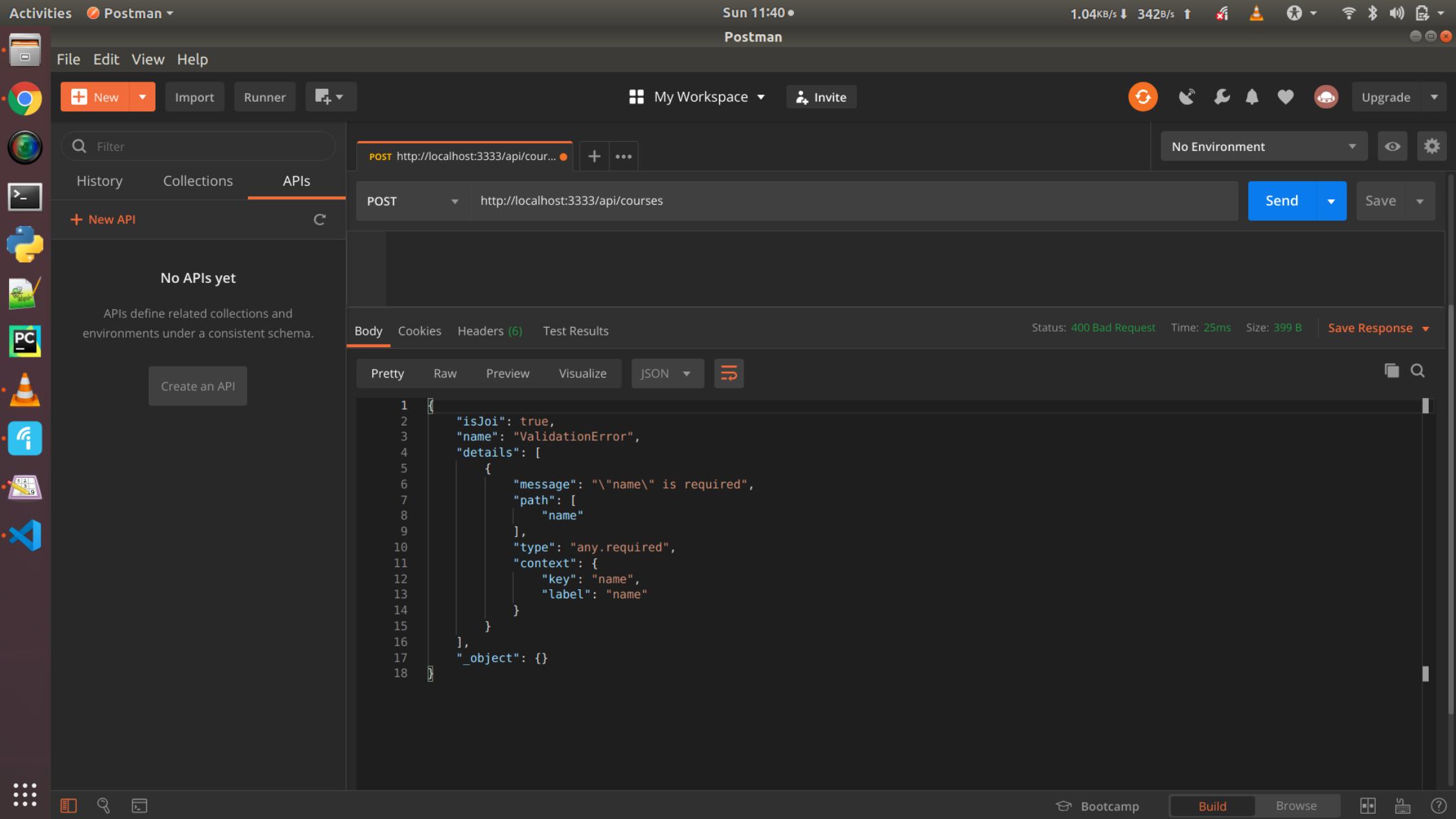
index.js - Applications - Visual Studio Code

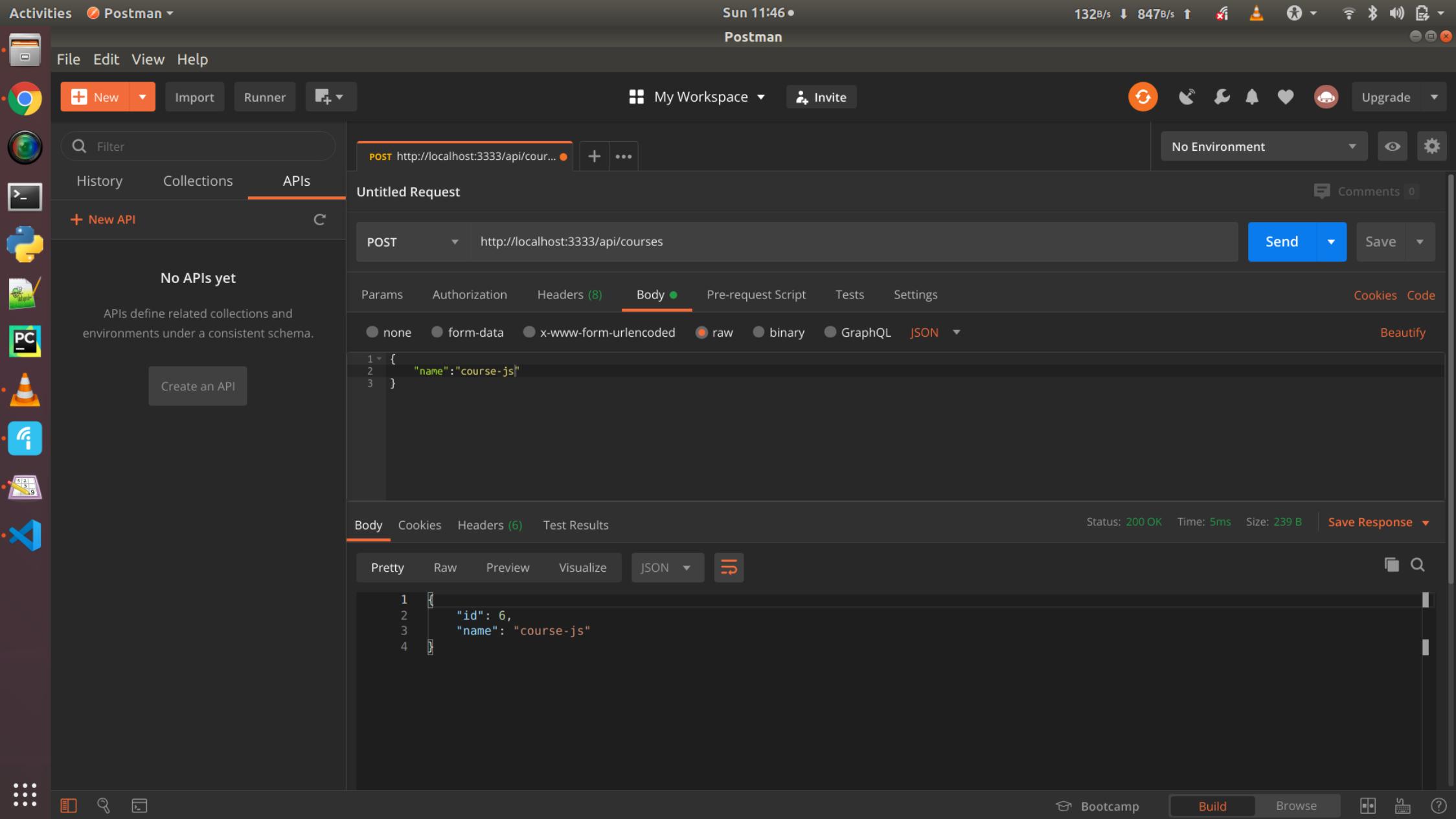
index.js - Applications - Visual Studio Code



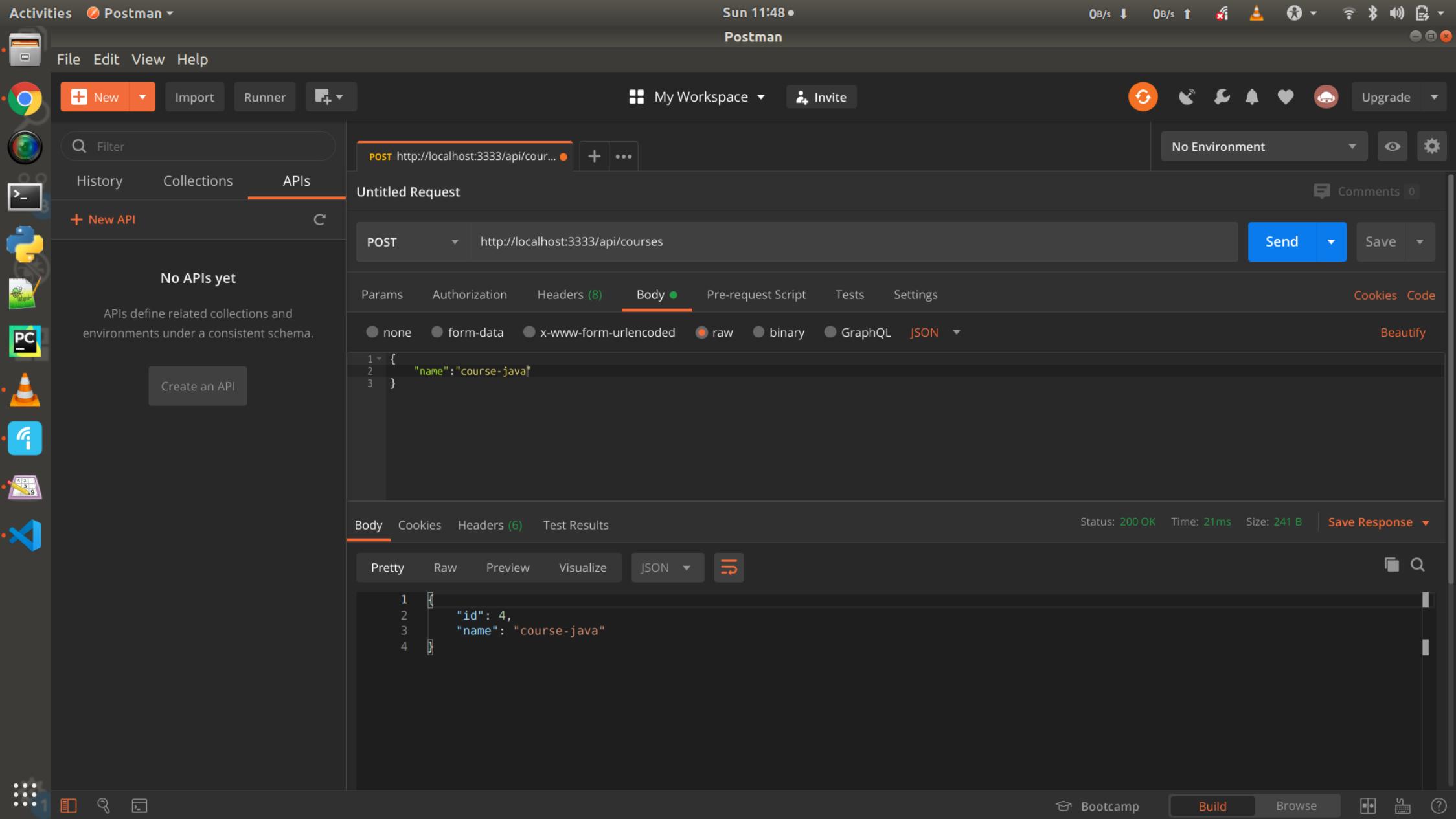








index.js - Applications - Visual Studio Code



index.js - Applications - Visual Studio Code