Back-End Development

Exploring

#5 Loopback Module

Purwadhika
Startup and Coding School

Loopback

LoopBack is a highly-extensible, open-source Node.js framework that enables you to create dynamic end-to-end REST APIs with little or no coding, access databases and also Incorporate model relationships and access controls for complex APIs (*http://loopback.io*).

- **Installation**

```
$ npm install -g loopback-cli
```

- **Check version**

```
$ npx loopback-cli -v
```

Purwadhika
Startup and Coding School

Create App

- On terminal, type:

  $ npx loopback-cli

- *What's the name of your application?*
  lin_loopback
- *Enter name of the directory to contain the project:*
  lin_loopback
- *Which version of LoopBack would you like to use?*
  3.x (current)
- *What kind of application do you have in mind?*
  api-server

- Go to project directory:

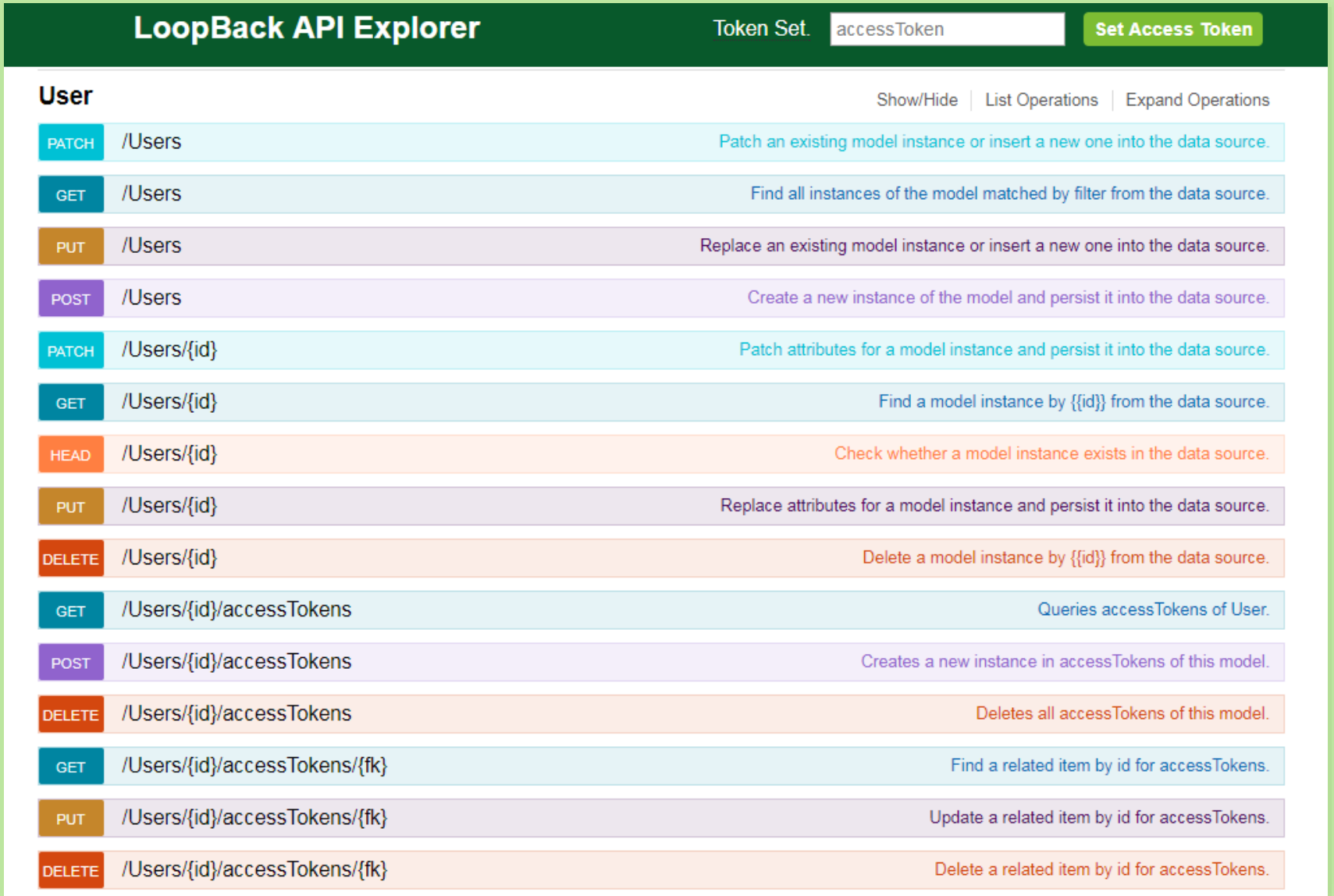
  $ cd lin_loopback

- Run the application:

  $ node .

- *Web server listening at: http://localhost:3000*
- *Browse your REST API at http://localhost:3000/explorer*

http://localhost:3000/explorer



Set App

- By default, we must set token for accessing loopback API. To use its API without any authentication, do these steps:

  Go to ***projectDir****/server/boot/authentication.js*
  Comment this line: *server.enableAuth();*

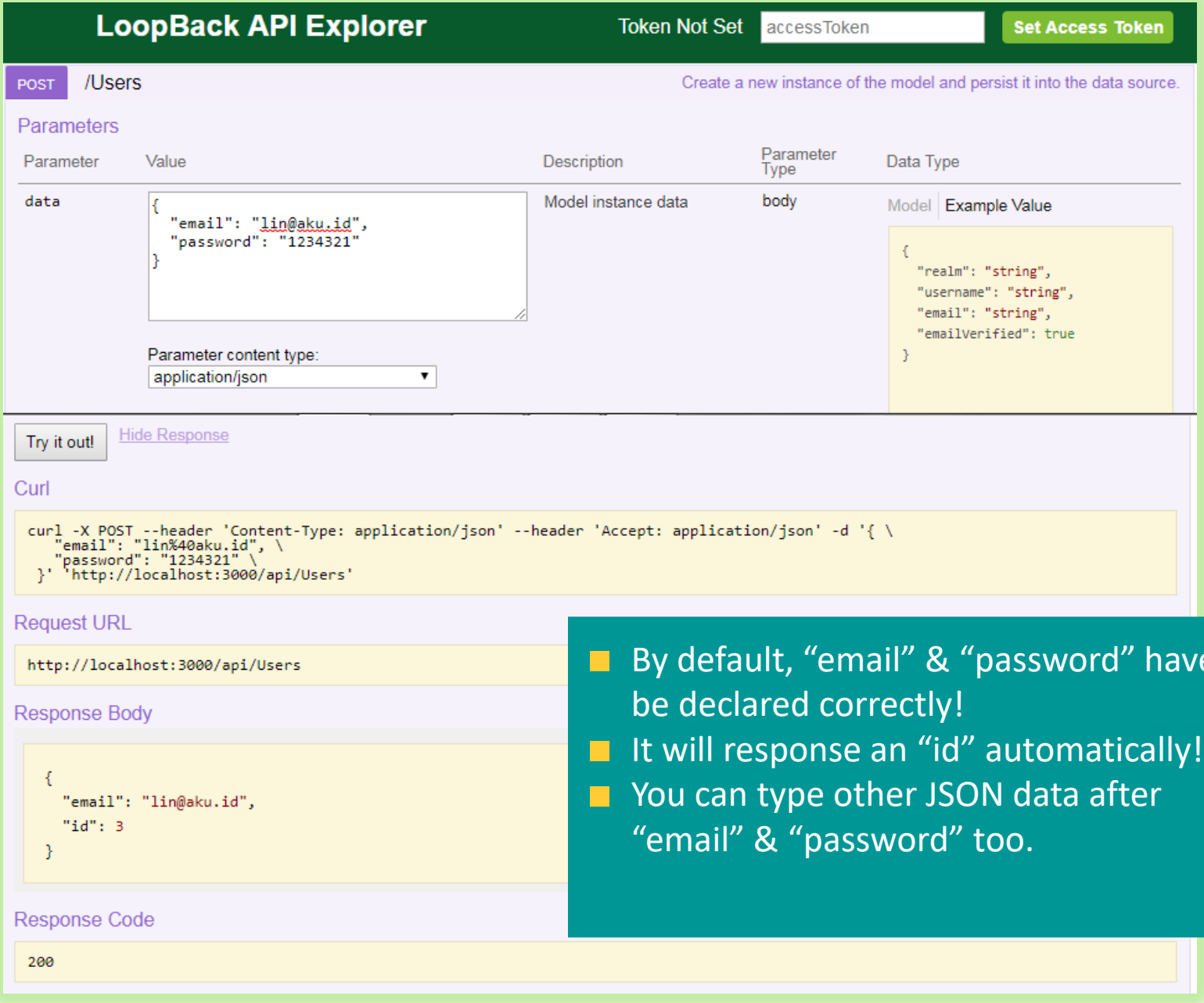
- Try to ***POST*** at ***POST/Users!***

```
{
    "email" : "lintang@purwadhika.com",
    "password" : "123456"
}
```

- Try to post custom data, then ***GET*** at ***GET/Users!***

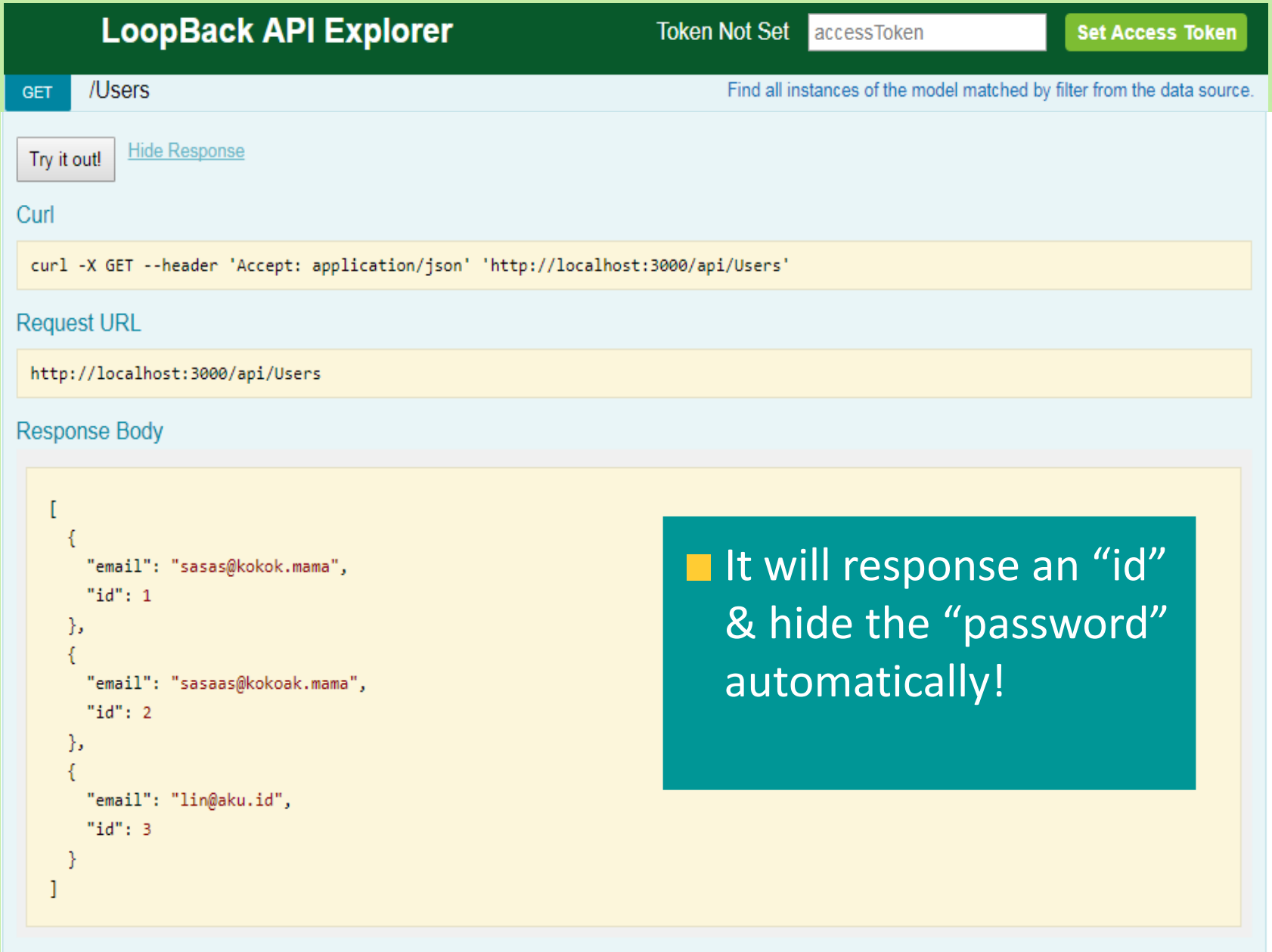
Purwadhika
Startup and Coding School

POST



- By default, “email” & “password” have to be declared correctly!
- It will response an “id” automatically!
- You can type other JSON data after “email” & “password” too.

GET



- It will response an “id” & hide the “password” automatically!

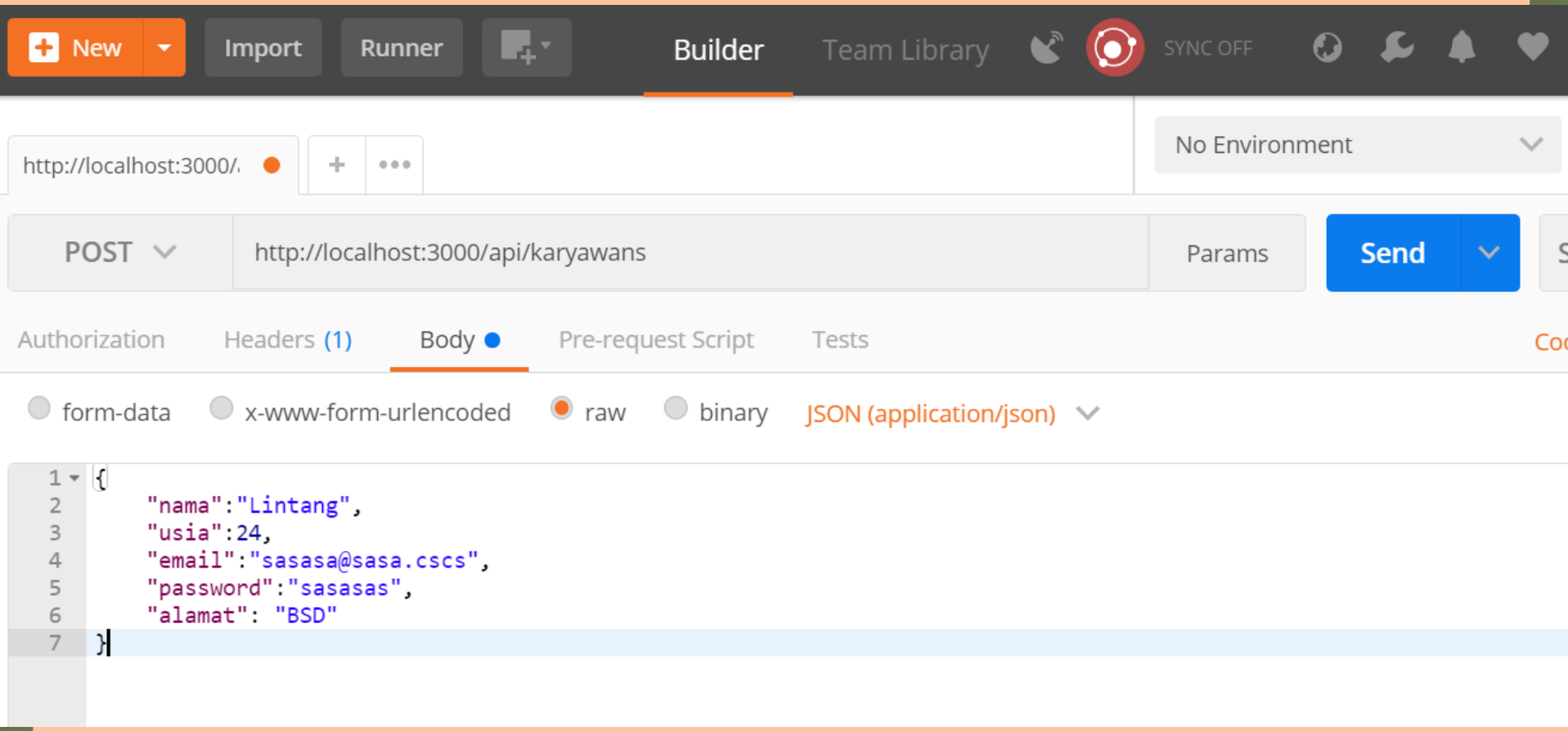
POSTMAN

- A powerful GUI platform to make API development faster & easier, from building API requests through testing, documentation & sharing.
- Postman has features for every API developer: request building, tests & pre-request scripts, variables, environments & request descriptions, designed to work seamlessly together.
- Download from *getpostman.com*

Try to POST & GET with POSTMAN!

Purwadhika
Startup and Coding School

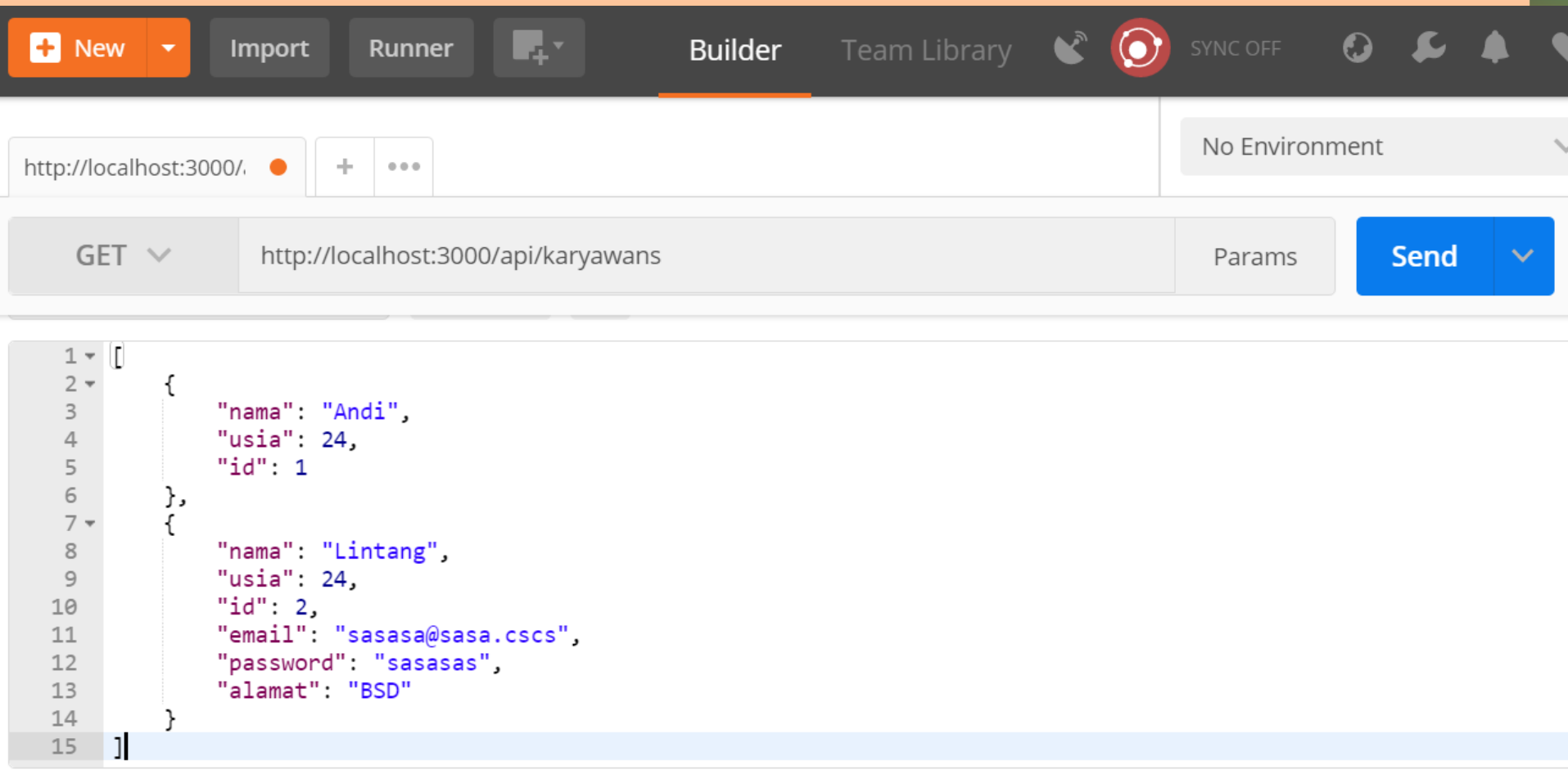
POST with POSTMAN



Set Header Request to
Content-Type: application/json

Purwadhika
Startup and Coding School

GET with POSTMAN



Purwadhika
Startup and Coding School

Create Model

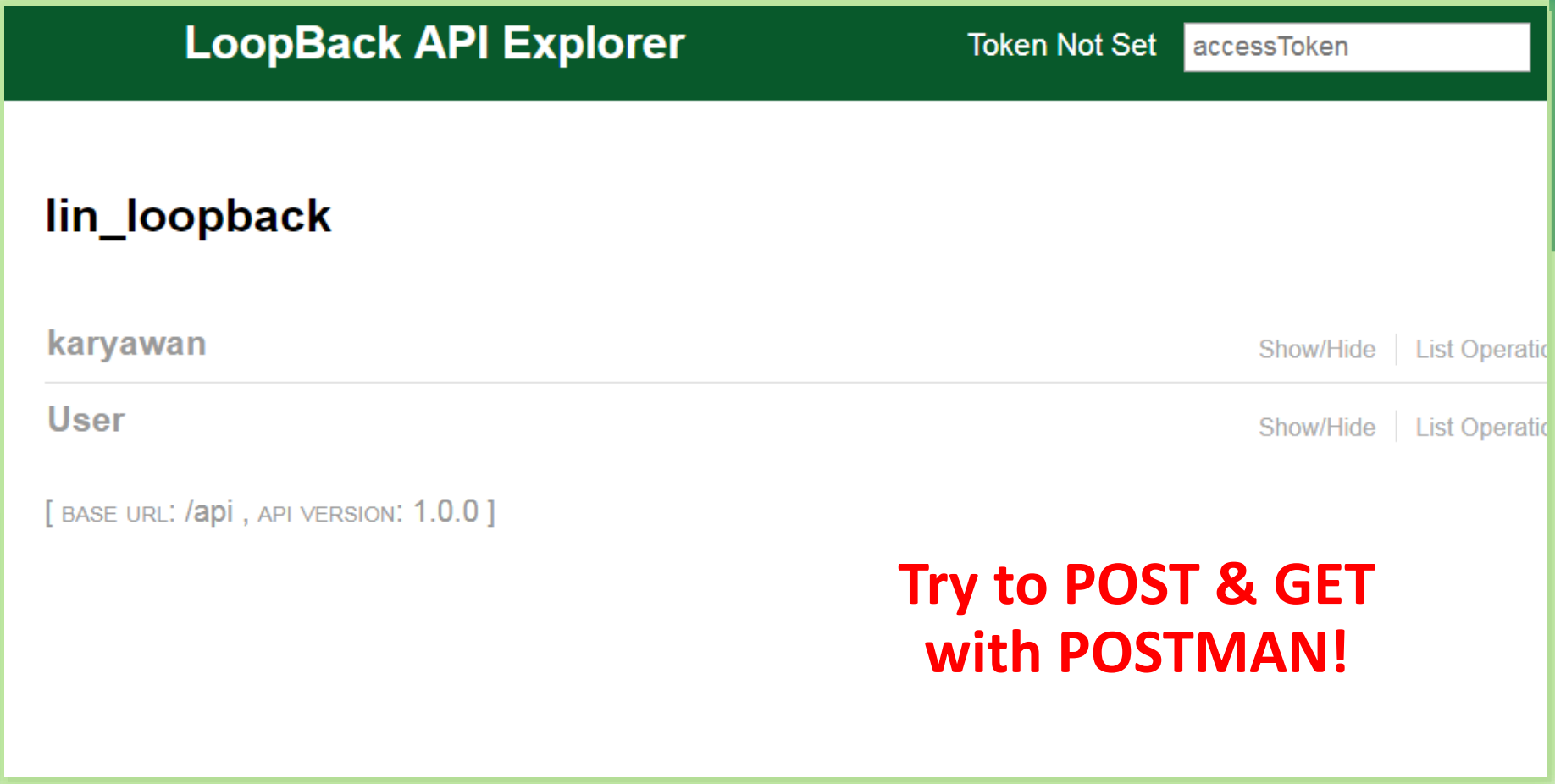
- On directory project, type:
  $ npx loopback-cli model

```
➢ [?] Enter the model name: karyawan
➢ [?] Select the data-source to attach person to: db
➢ [?] Select model`s base class (PersistedModel)
➢ [?] Expose person via the REST API? Yes
➢ [?] Custom plural form (used to build REST URL):
➢ [?] Common model or server only? Common
Let's add some person properties now.

Enter an empty property name when done.
➢ [?] Property name: id
➢ [?] Property type: (Use arrow keys) number
➢ [?] Required? (y/N) y
```

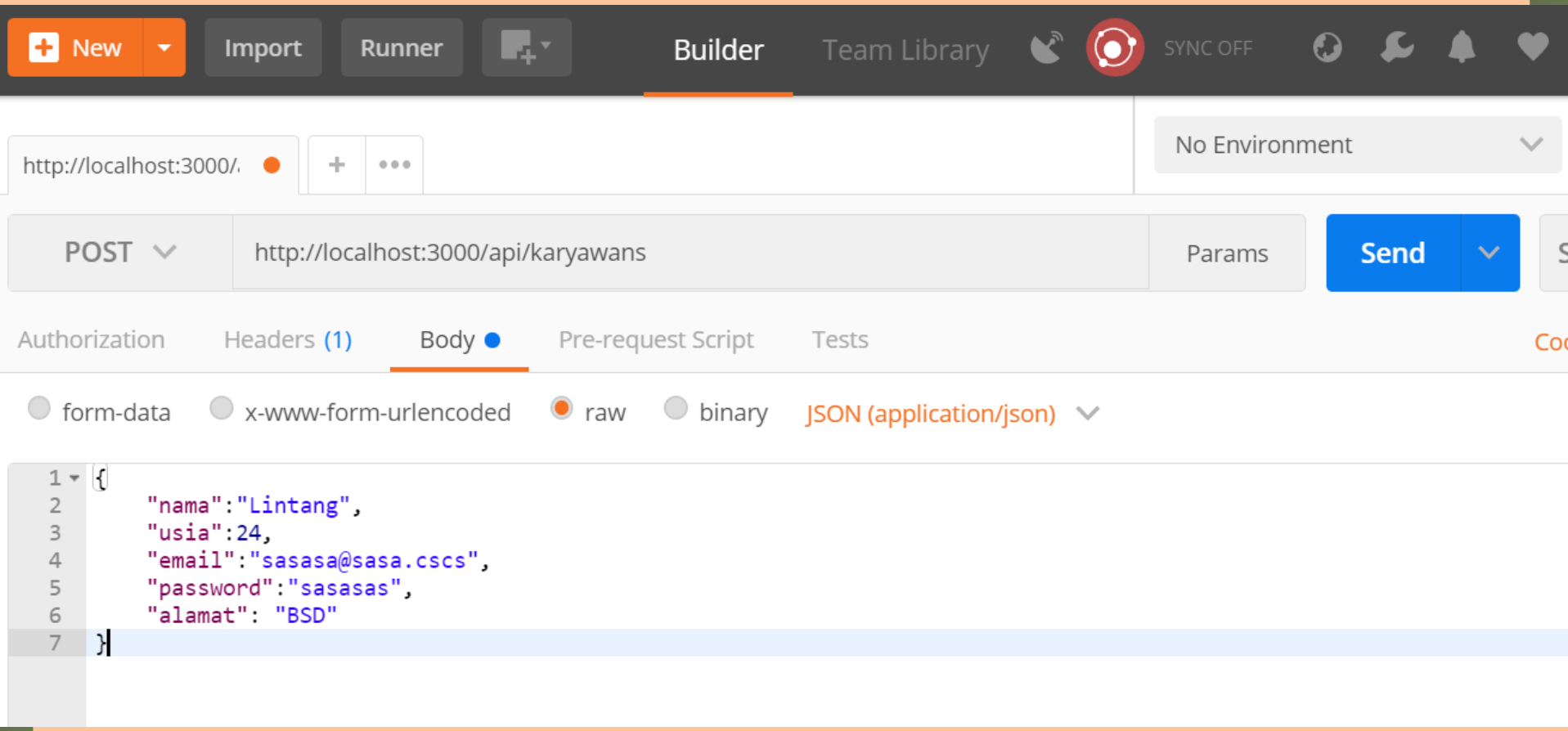
- Property ***"id"*** is a must! After done, try to run its app via terminal: **$ node .**

Create Model



Purwadhika
Startup and Coding School

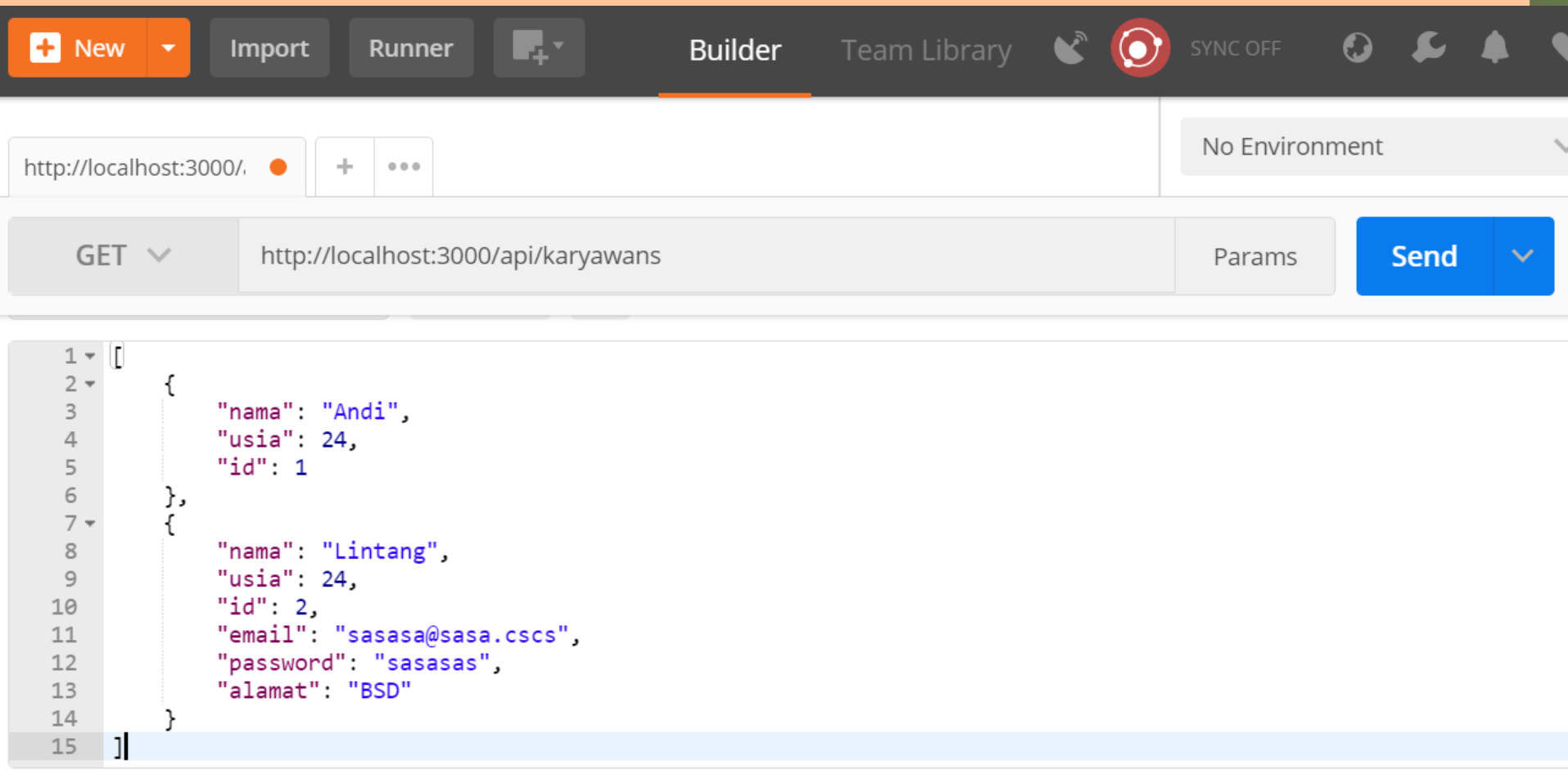
POST with POSTMAN



Set Header Request to
Content-Type: application/json

Purwadhika
Startup and Coding School

GET with POSTMAN



Purwadhika
Startup and Coding School

Back-End Development

Exploring

#5 Loopback Module

Purwadhika
Startup and Coding School