Front-End Development

JSON-Server

Create a fake REST API

Purwadhika
Startup and Coding School

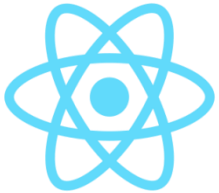
API

- ***API*** *(Application Programming Interface)* is a software intermediary which allows applications to talk to each other.

- When you use an application on your mobile phone, the application connects to the Internet and sends data to a server. The server then retrieves that data, interprets it, performs the necessary actions and sends it back to your phone. The application then interprets that data and presents to you with the information that you wanted, in a readable way. This is what an API is - all of this happens via API.

API

Purwadhika
Startup and Coding School



JSON-Server

- ***JSON-Server*** is created for web front-end developers who need a quick back-end for prototyping and mocking.
- With this package, developers can get a full fake REST API with ***zero coding*** in less than ***30 seconds***!
- More information click: ***github.com/typicode/json-server***

Purwadhika
Startup and Coding School

Create Fake REST API with JSON-Server #1

- Create a project folder, for eg ***lin_fake_api***
- Create a json file on project folder, for eg ***db.json***

```
{
 "karyawan":[
  {"id":1,"nama":"Andi","usia":25,"kota":"Jakarta"},
  {"id":2,"nama":"Budi","usia":27,"kota":"Yogya"},
  {"id":3,"nama":"Caca","usia":25,"kota":"Bandung"},
  {"id":4,"nama":"Dedi","usia":31,"kota":"Jakarta"},
  {"id":5,"nama":"Euis","usia":22,"kota":"Bandung"}
 ]
}
```

Purwadhika
Startup and Coding School

Create Fake REST API with JSON-Server #2

- On project folder, type:

  $ npm init

- Install json-server:

  $ npm install –g json-server --save

- Activate json-server:

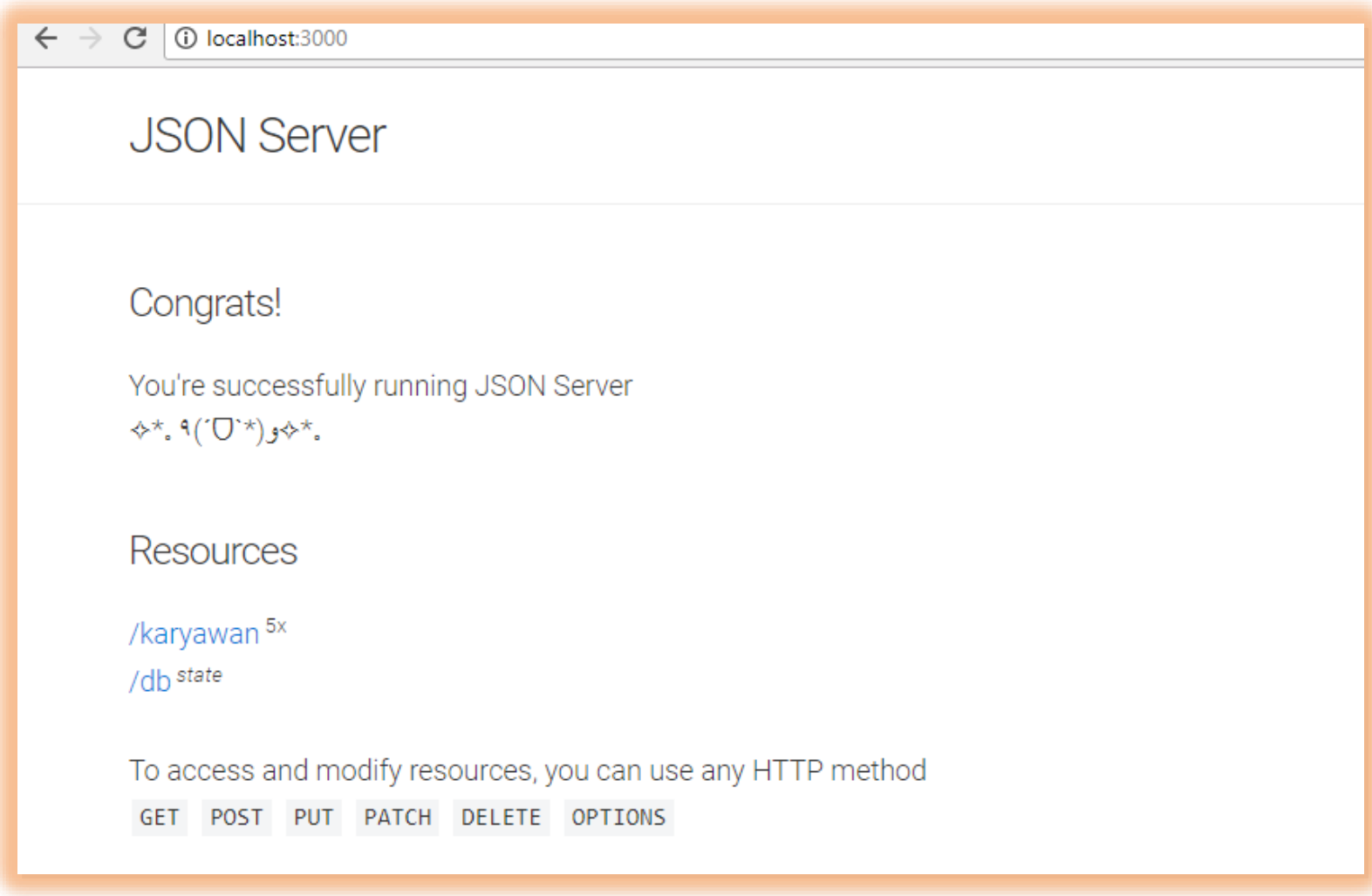
  $ json-server --watch db.json

- Done, see on browser

  http://localhost:3000

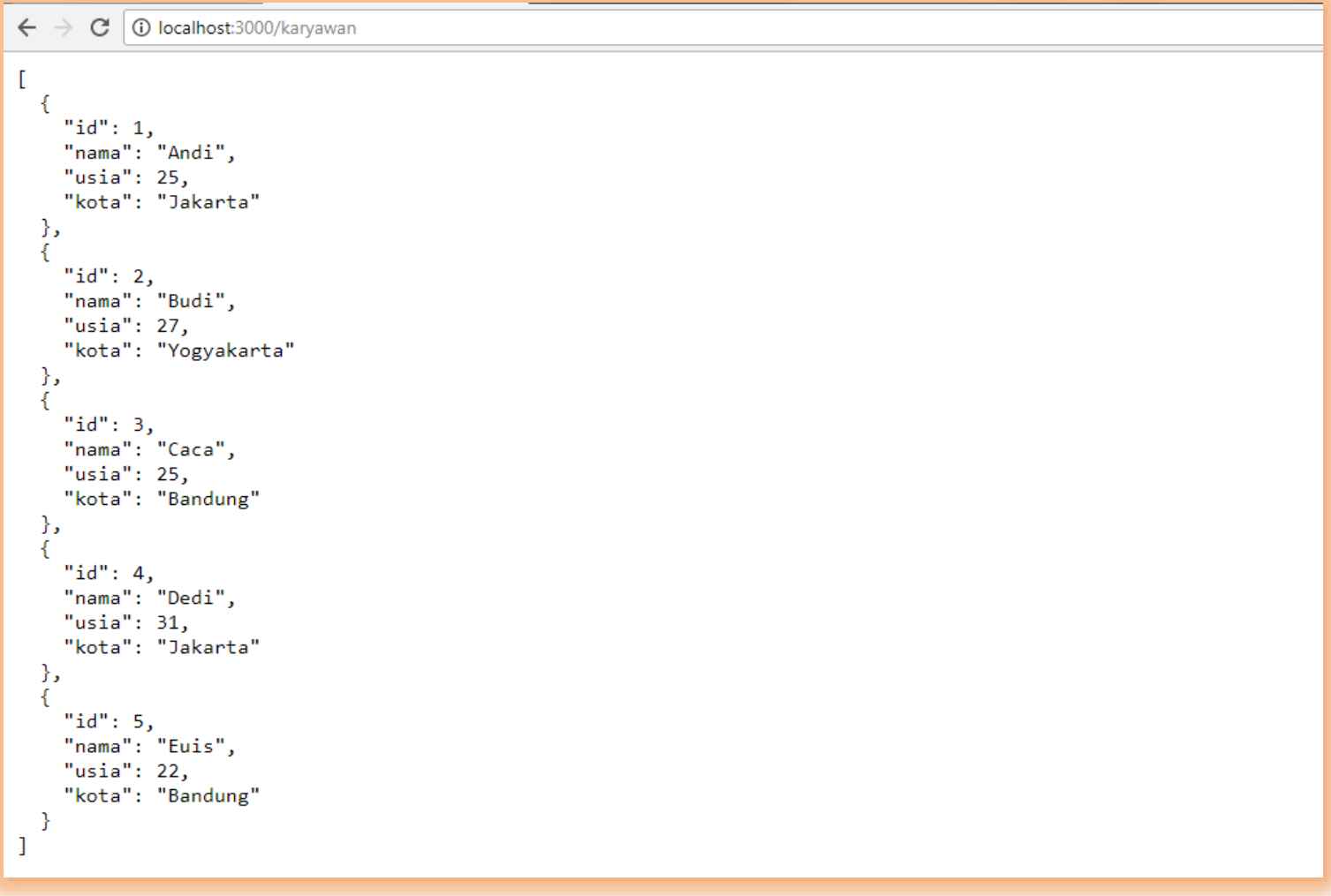
Purwadhika
Startup and Coding School

http://localhost:3000



Purwadhika
Startup and Coding School

http://localhost:3000/karyawan



Purwadhika
Startup and Coding School

GET on Postman

localhost:3000/db
localhost:3000/karyawan

localhost:3000/karyawan/3
localhost:3000/karyawan?id=3
localhost:3000/karyawan?usia=25

localhost:3000/karyawan?_limit=3
localhost:3000/karyawan?_sort=nama&_order=desc
localhost:3000/karyawan?usia_gte=26
localhost:3000/karyawan?usia_gte=23&usia_lte=27

localhost:3000/karyawan?q=Jakarta

POST on Postman

- Set POST to http://localhost:3000/karyawan
- Set Headers Key: Content-Type & value: app/json
- Insert data json on Body raw & send it, for eg:

```
{
   "id":6,
   "nama" : "Fifi",
   "usia" : 28,
   "kota" : "Medan"
}
```

- Data on db.json is updated!

Purwadhika
Startup and Coding School

DELETE on Postman

- Set DELETE to http://localhost:3000/karyawan/5 to delete data karyawan number 5.

UPDATE on Postman

- Set PATCH to http://localhost:3000/karyawan/2 to update data karyawan number 2.
- Set Headers Key: Content-Type & value: app/json. Insert data json on Body raw & send it, for eg:

```
{
    "kota" : "Solo"
}
```

Purwadhika
Startup and Coding School

Try to connect React Project to Fake REST API!

Purwadhika
Startup and Coding School

Front-End Development

JSON-Server

Create a fake REST API

Purwadhika
Startup and Coding School