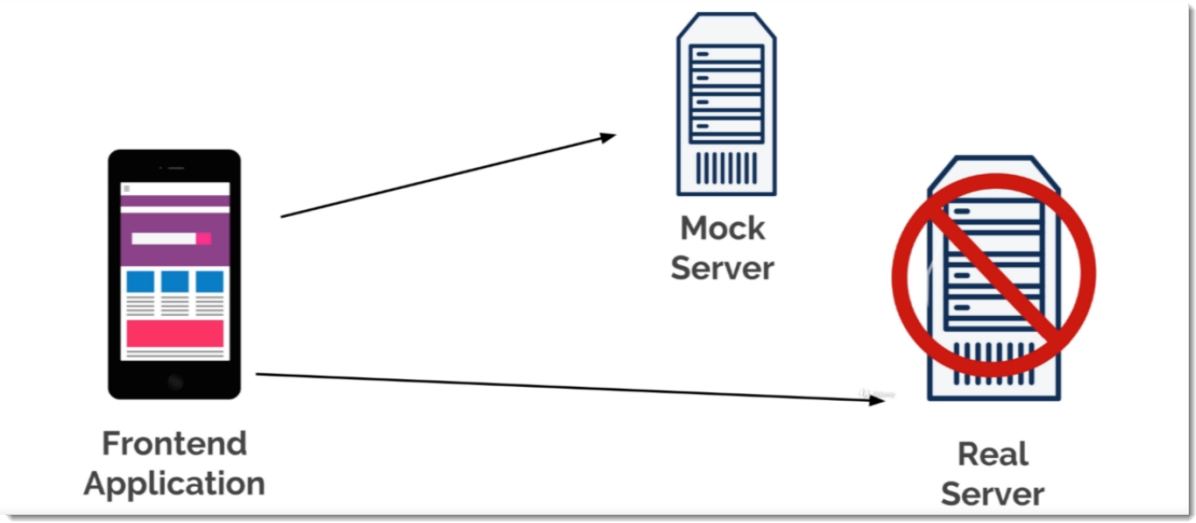
Mock Server

- Author : Dipanjal Maitra
- Version : 0.0.1



What is Mock Server ?

```
A mock API server or mock server imitates a real API server by providing realistic
mock API responses to requests. They can be on your local machine or the public
Internet. Responses can be static or dynamic, and simulate the data the real API would
return, matching the schema with data types, objects, and arrays.
```

Here we are going to use **json-server** to Mock our APIs.

> **NOTE** You must have **NodeJS** and **npm** installed

JSON Server

***json-server** is a Node Module that you can use to create demo rest json webservice in less than a minute. All you need is a JSON file for sample data.*

Installation

```
npm install -g json-server
```

Configuration

- First you need to put your JSONArray formatted data into a file

click to expand mock db

```
{
  "posts": [
    { "id": 1, "title": "json-server", "author": "typicode" }
  ],
  "comments": [
    { "id": 1, "body": "some comment", "postId": 1 }
  ],
  "profile": { "name": "typicode" }
}
```

NOTE	You must provide **JSON ARRAY** as a dataset.

- Good to have a configuration file

click to expand config

```
{
  "port": "3080",
  "host": "172.16.229.133",
  "watch": "/home/user/remittance/stub_server/language-db.json"
}
```

NOTE	The configuration file name must be **json-server.json**, should be placed in the execution directory, or the **json-server** installed location. for CentOS 8, the location is `/usr/local/bin` but it may vary depending on OS you are using.

Execution

```
json-server --watch your_db.json --host your_host_address
```

Example

```
json-server --watch db.json --host 172.16.229.133
```

Run in Background

***Well, We have successfully launched our json mock server but now things** are getting a bit tricky here. The* **json-server** *will run as long as the terminal is open, because the application is running in the foreground CLI thread. There are several ways to run an application as a background process* **nohup** *is one of them. But, there are some* `cons` *of nohup it's hard to monitor the process status, the occupied resources, heap size and so on. Here PM2 has come into the picture.*

What is PM2 ?

***A Production level process manager for Node.js apps with a built-in load** balancer.*

Table 1. PM2 vs nohup

Feature	PM2	nohup
Graceful/rolling restarts	YES	NO
OS startup script support	YES	NO
Default Process Monitor	YES	NO
Default Load Balancer	YES	NO

PM2 Installation

```
npm install pm2 -g
```

Execution With PM2

With PM2, now we are going to run our** **json-server** **as a** **process

Table 2. Essential PM2 Commands

Description	Command
Start Json Server	`pm2 start json-server -- language-db.json`

Monitor all processes	`pm2 monit`
Stop specific process id	`pm2 stop 0`
Restart specific process id	`pm2 restart 0`
Save processes list to respawn at reboot	`pm2 save`

References

- [1] Json Server Official Documentation
- [2] PM2 Official Documentation
- [3] PM2 Cheatsheet