

# Fetch Rewards Coding Exercise - Backend Software Engineering

Documented by:

Akash Kumar Shetty

[akumarshetty@hawk.iit.edu](mailto:akumarshetty@hawk.iit.edu)

Designed a web service that accepts http requests and returns responses validating if the entered word is a “Pyramid Word” or not.

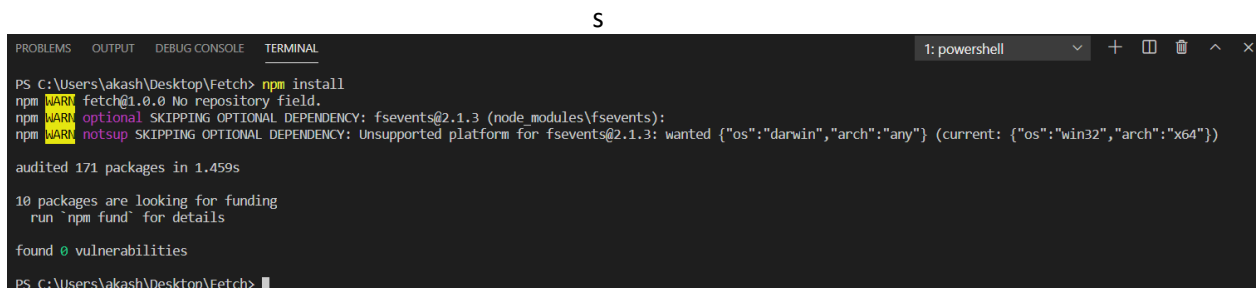
I have used Express to implement this task. I have prior experience using spring boot for similar tasks but find node.js more flexible and seamless for me to use. I can complete this task using spring boot if it’s required.

Please make sure you have the following services to execute this code:

1. npm and node.js [Downloading and installing node.js and npm](#)
2. Postman [Postman installation](#)

After installing Node.js and Postman, please follow the following steps:

- Clone/download the project file.
- Open the file with code editor like Visual Studio Code.
- Type “npm install” in the terminal to install the necessary dependencies.

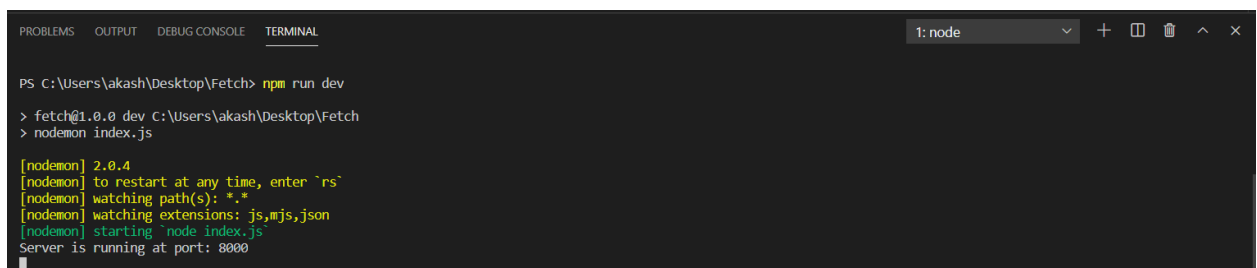


```
PS C:\Users\akash\Desktop\Fetch> npm install
npm WARN fetch@1.0.0 No repository field.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.3 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

audited 171 packages in 1.459s
10 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities

PS C:\Users\akash\Desktop\Fetch>
```

- Use “npm run dev” to execute the code.

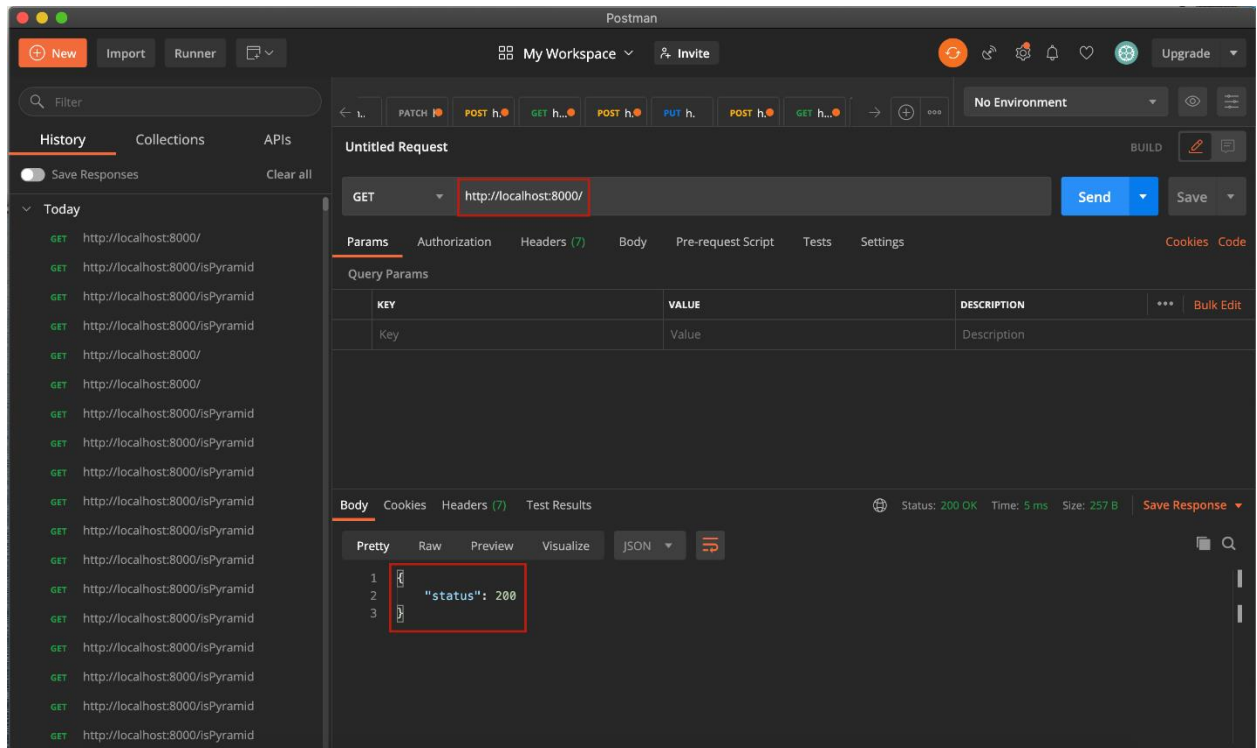


```
PS C:\Users\akash\Desktop\Fetch> npm run dev
> fetch@1.0.0 dev C:\Users\akash\Desktop\Fetch
> nodemon index.js

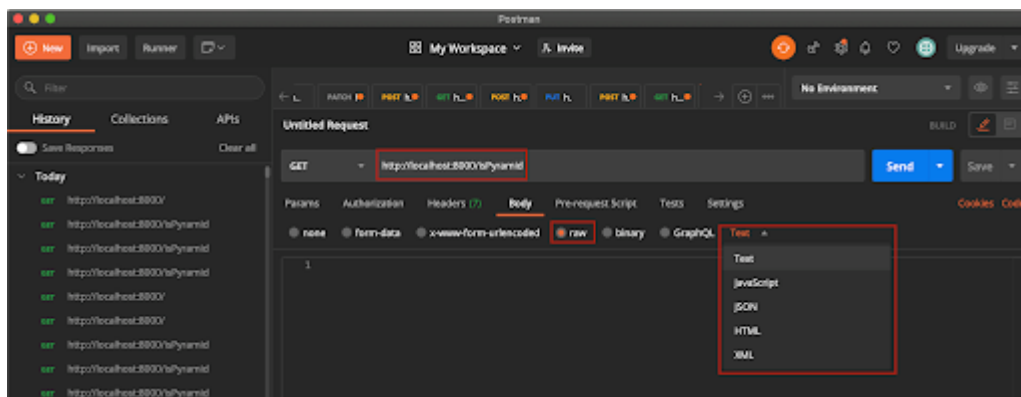
[nodemon] 2.0.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting node index.js
Server is running at port: 8000
```

Once the server has started, please go to Postman to check the service.

- Type <http://localhost:8000/> and check if we get “status: 200” for successful establishment of connection.



- Then type <http://localhost:8000/isPyramid>. Select “raw” and “text” for the input type and provide a text input. This can be implemented using json as well, but I have chosen text input as it is even simpler for the user to test.



- Give any word of your choice to check if that is a pyramid word or not. The following screenshots shows different testcases.

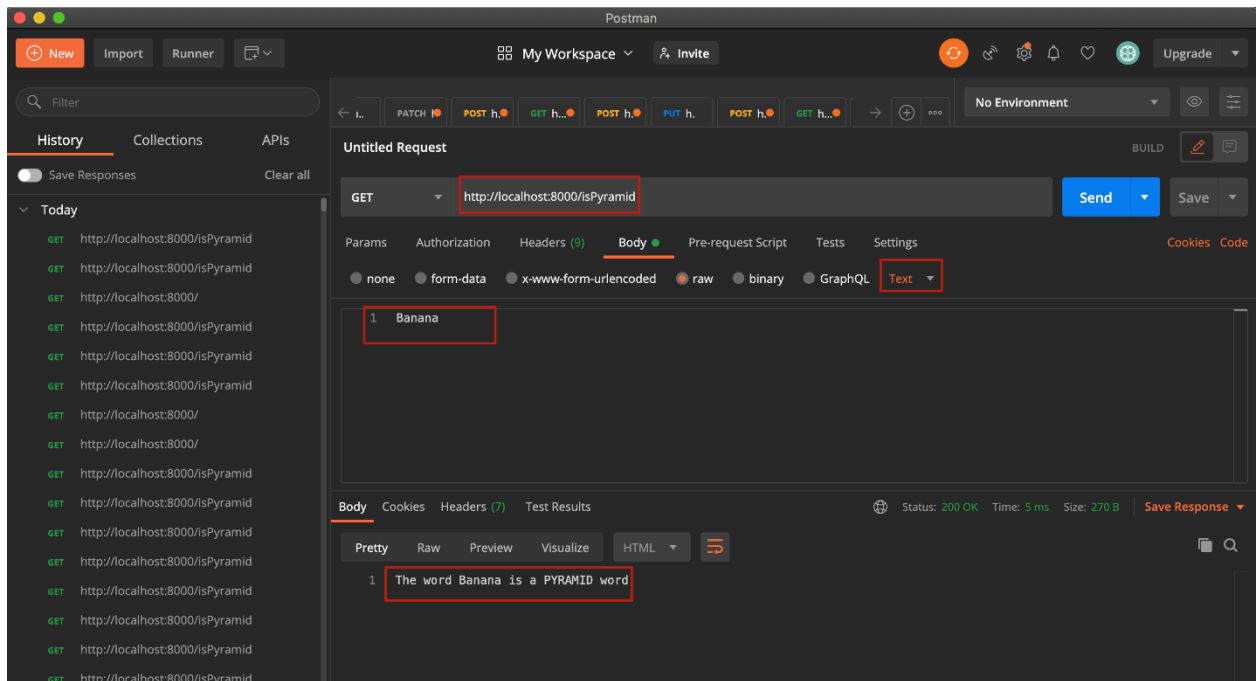


Figure 1: “Banana” is a pyramid word.

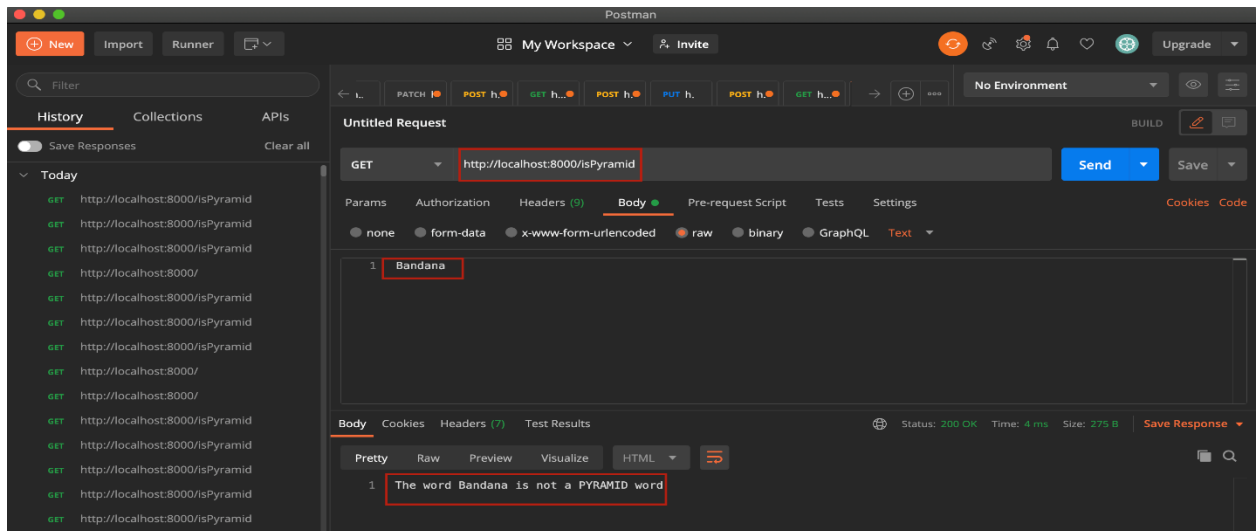


Figure 2: Bandana is not a pyramid word.

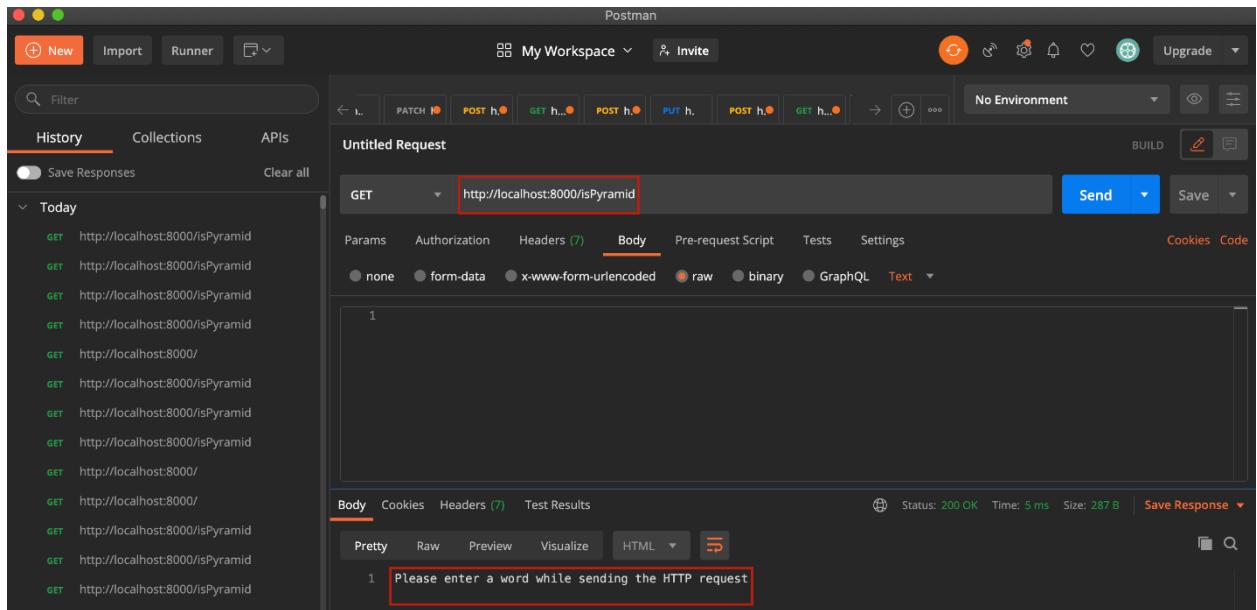


Figure 3: When no input is given.

When there is no input given, there is a message displayed asking the user to enter a text while requesting for a response. This condition is handled using a **catch block**.

Please feel free to contact me if any necessary changes are to be made or if any problem encountered during the execution.