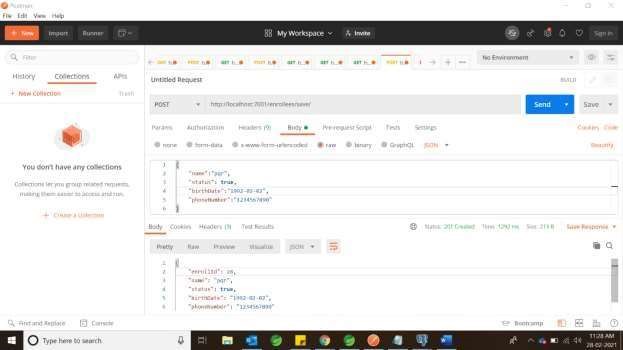
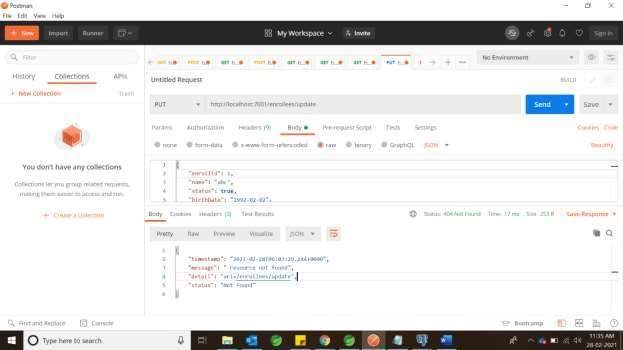
* **Follow API Proper Standards** (Database is PostgreSQL)

First access the request payload and response with json and proper status code



Handling the exceptions or error with error codes



* **Must be using Spring boot framework, not custom**

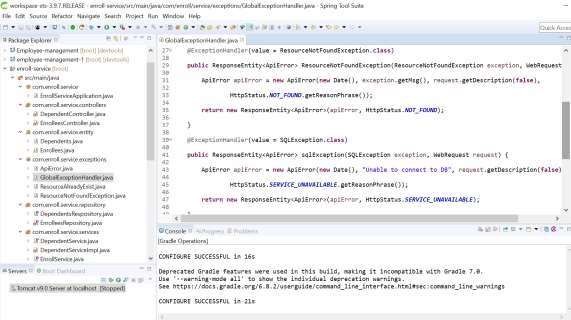
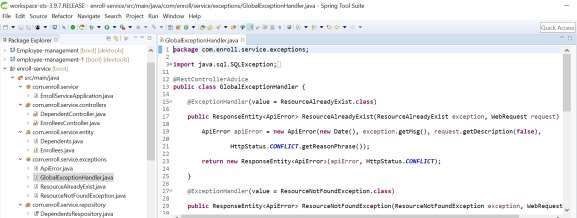
-Used spring boot

* **Provide readme (not just in one place)**

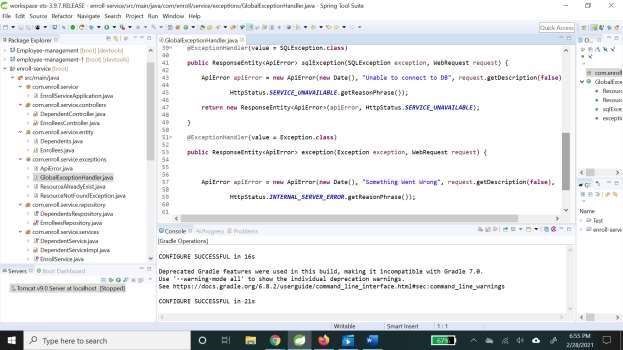
-Provide required documentation in this word doc to explain to showcase everything clearly!

* **Exceptional handling**

-The code handles exception handling and Database handling correctly here are the screenshots: it does throw error messages correctly for all possible use cases, I have used PostgreSQL as the database:

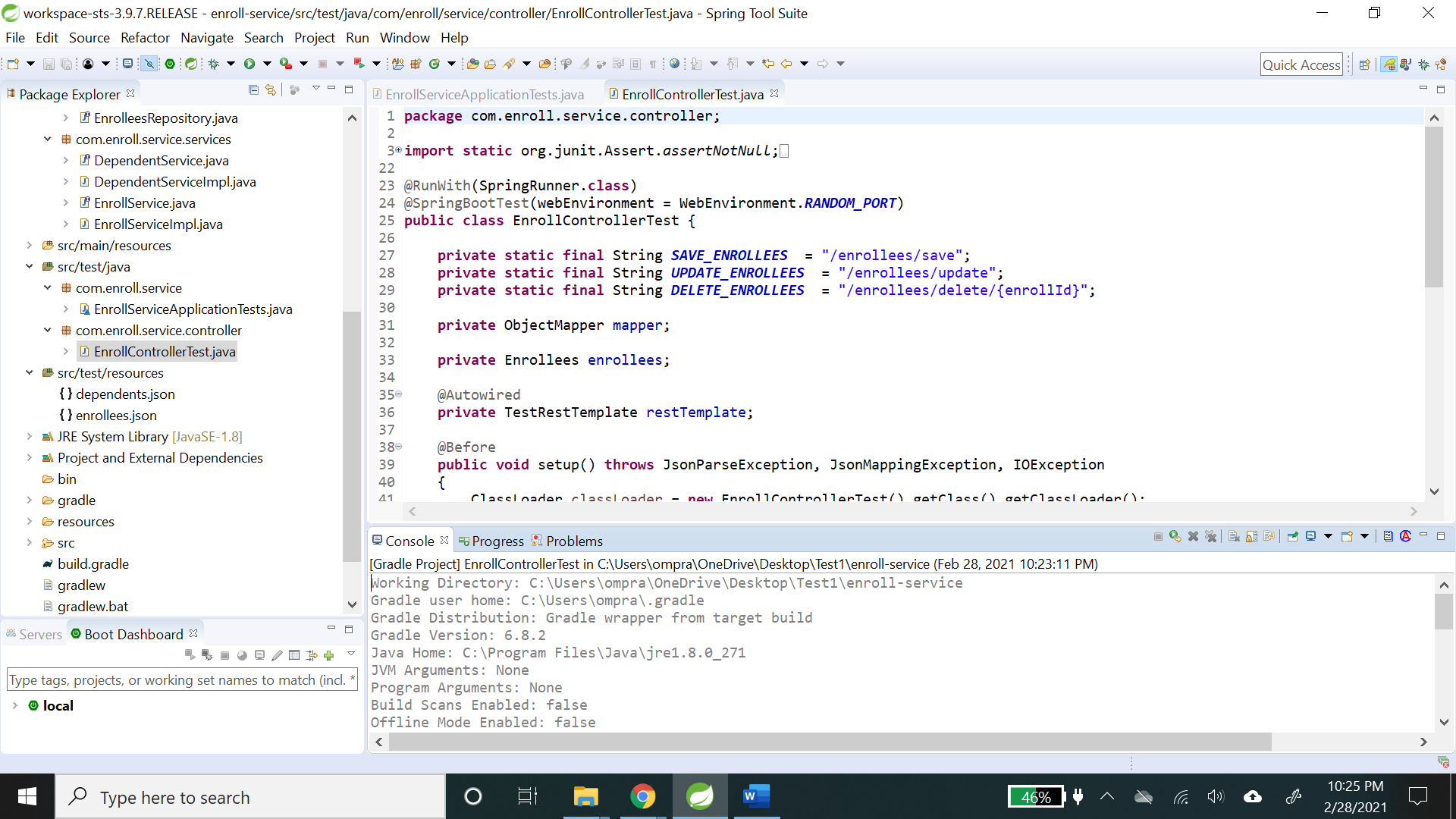


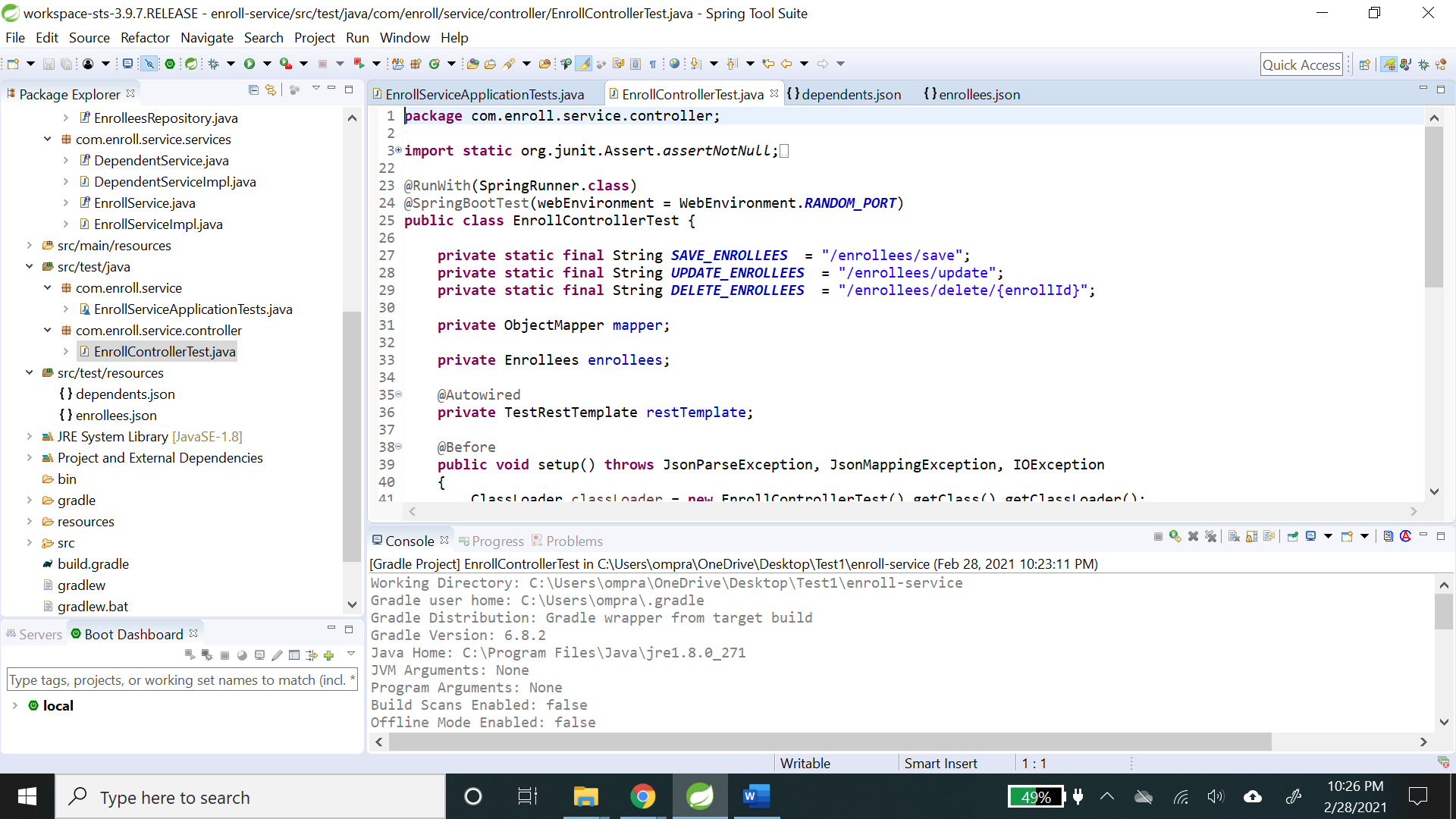
Handles SQL exception as below:



* **Include Unit Test cases**

Junit test cases for service.controller which has 85 percent coverage(under the folder test)





Along with JSON files!

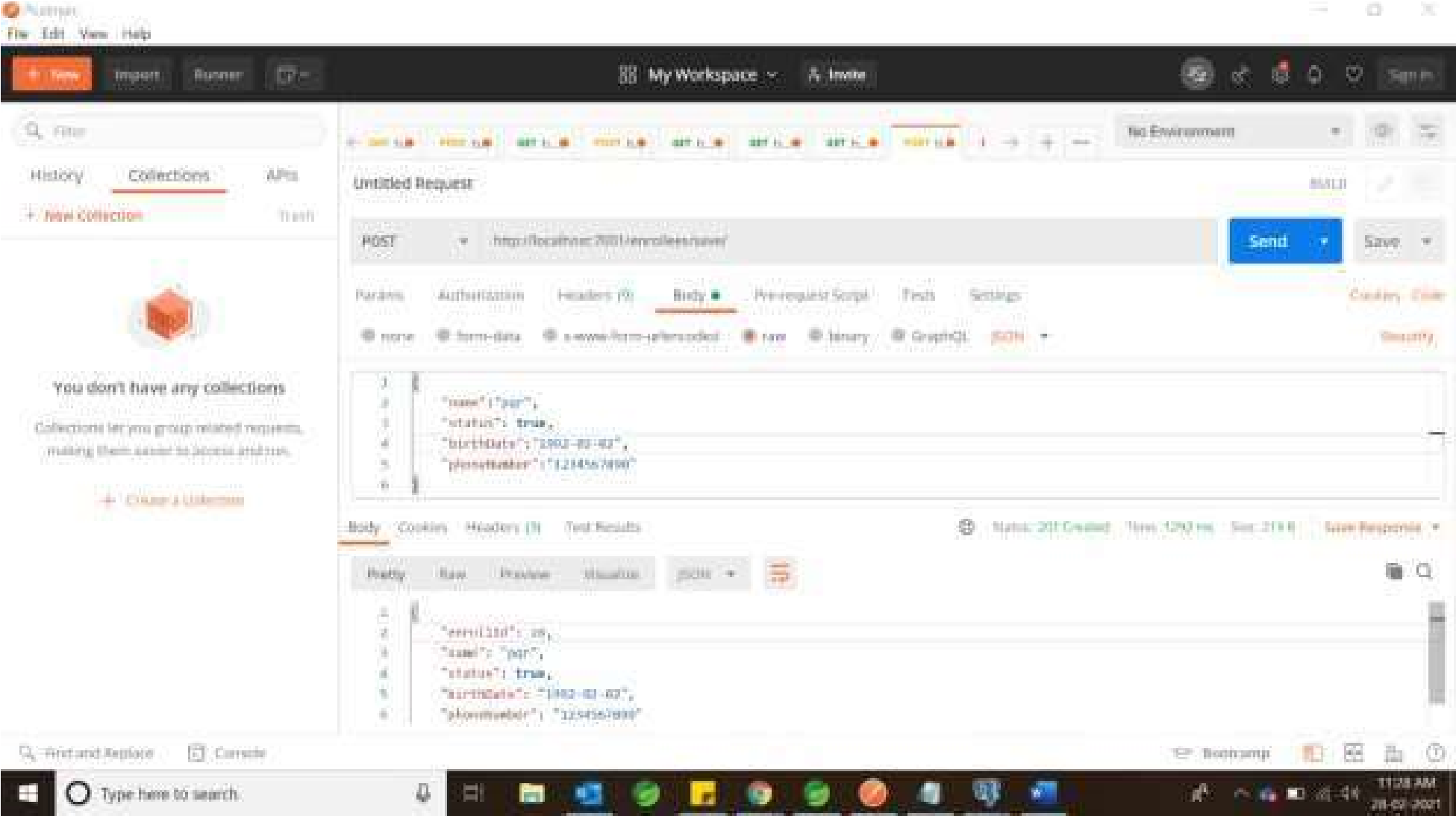
* **Clean Code and Proper Code Standards**

-

# The challenge

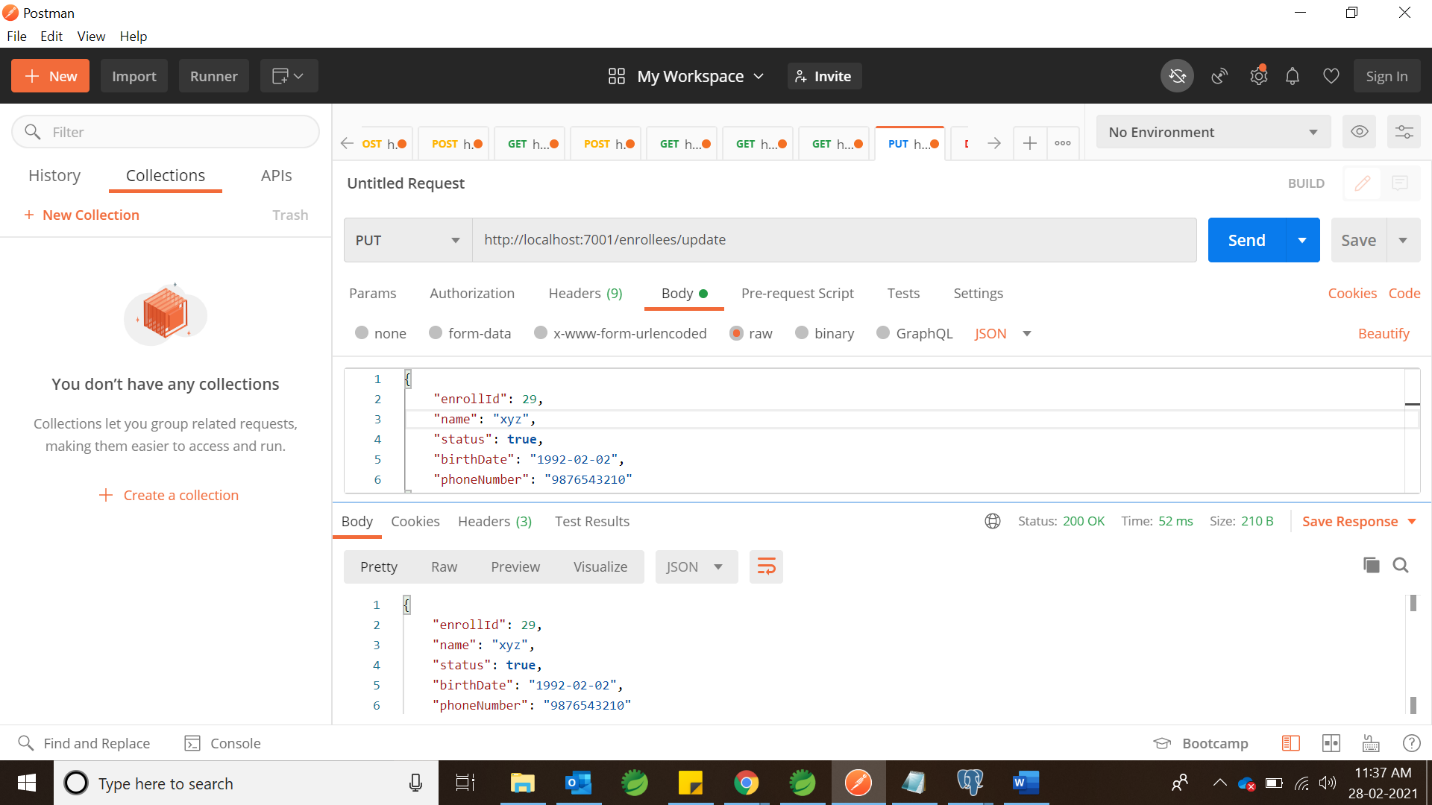
Using Spring Boot or Go, and your database of choice (PostgreSQL, MySQL, MongoDB -- any you'd like), develop a microservice for tracking the status of enrollees in a health care program.

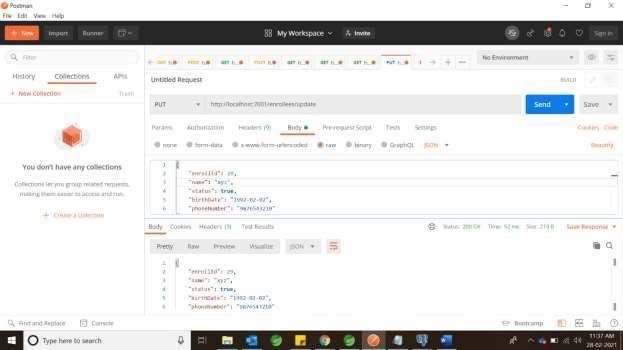
* Enrollees must have an id, name, and activation status (true or false), and a birth date
* Enrollees may have a phone number (although they do not have to supply this)
* Enrollees may have zero or more dependents
* Each of an enrollee's dependents must have an id, name, and birth date The application we will be building will need to be able to do these things:
* Add a new enrollee



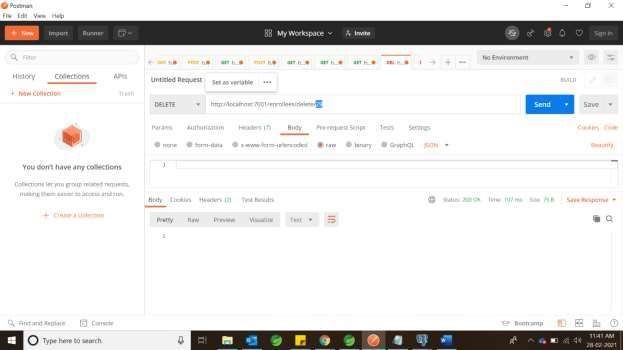
The above data has all the required fields

* Modify an existing enrollee

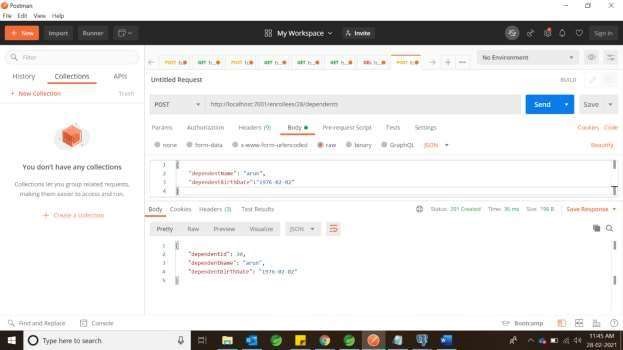




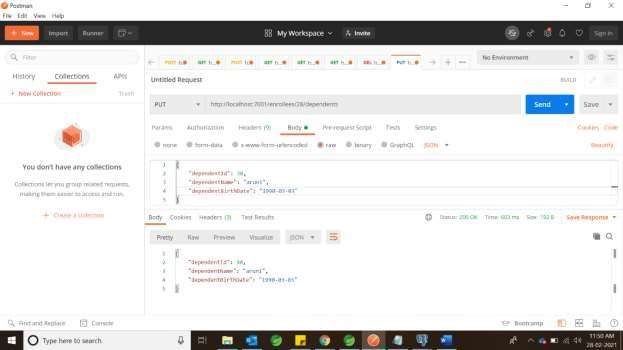
* Remove an enrollee entirely



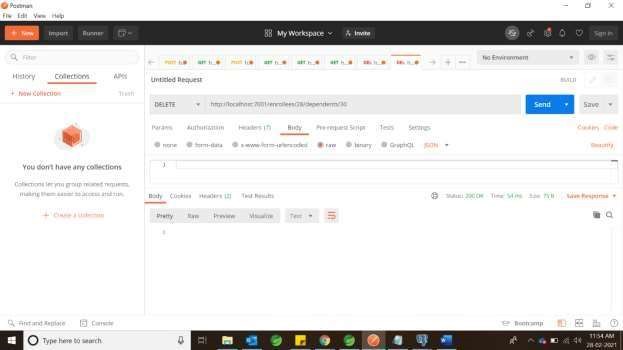
* Add dependents to an enrollee



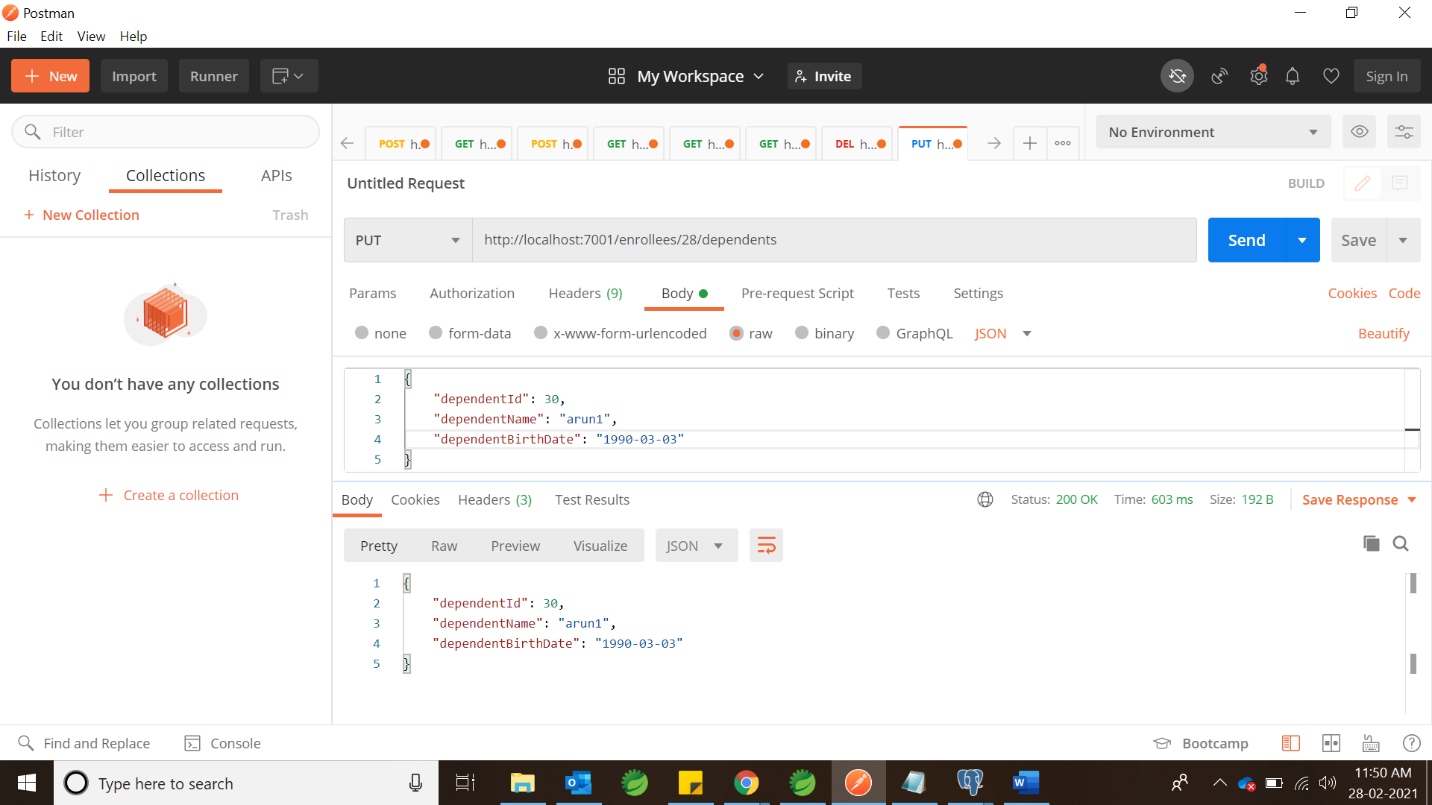
* Modify existing dependents



* Remove dependents from an enrollee



- Modify existing dependents



- Remove dependents from an enrollee

