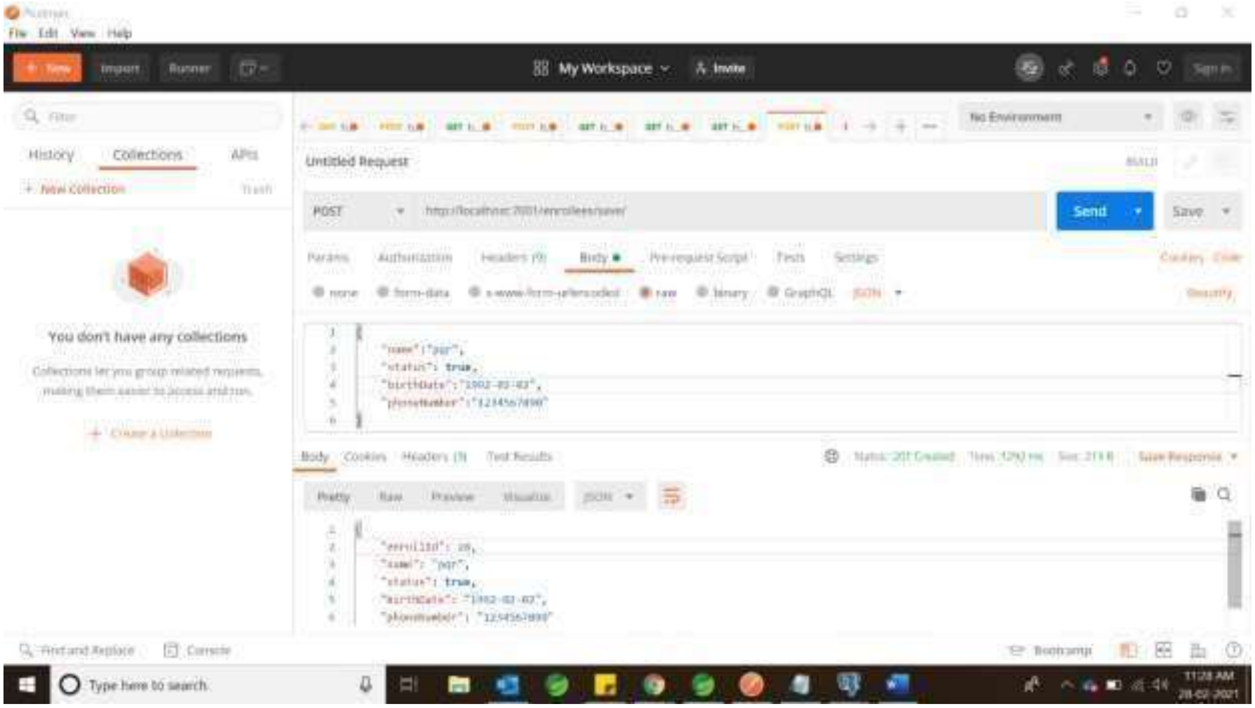
- **Follow API Proper Standards** (Database is PostgreSQL)

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



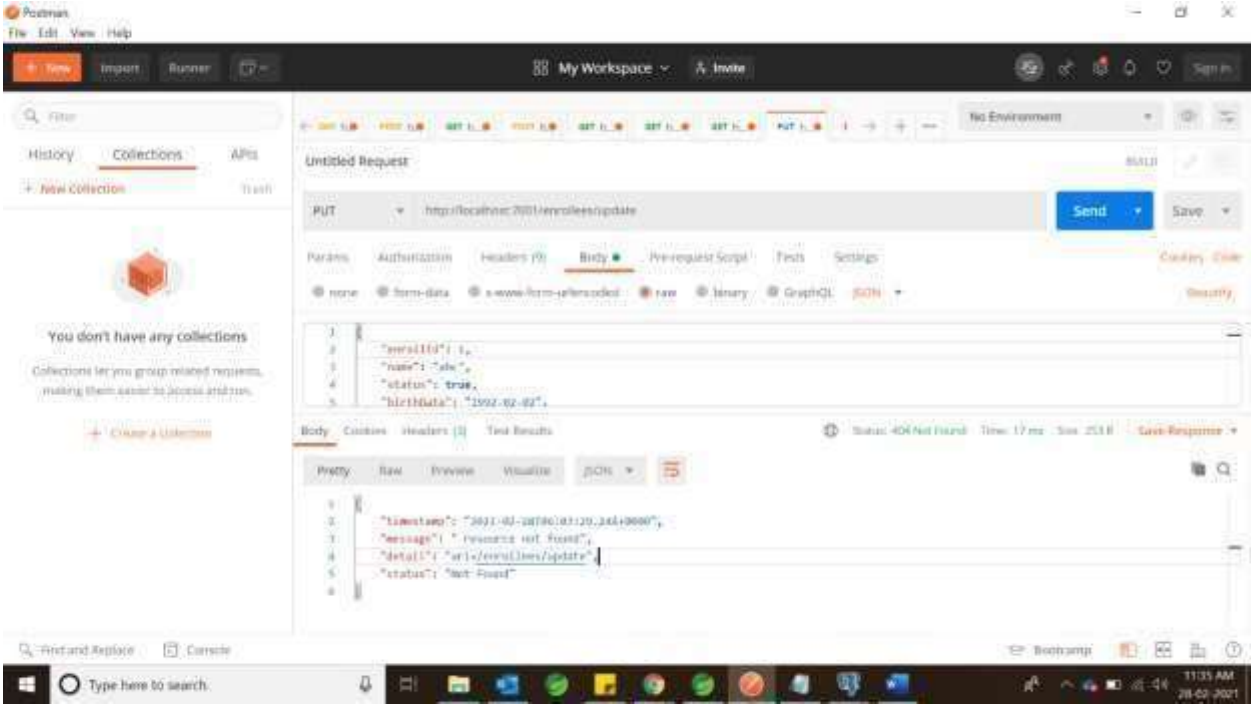
POST http://localhost:7001/enrollees/save/

```
{
    "name":"pqr",
    "status": true,
    "birthDate":"1992-02-02",
    "phoneNumber":"1234567890"
}
```

Status: 201 Created Time: 1292 ms Size: 213 B

```
{
    "enrollId": 28,
    "name": "pqr",
    "status": true,
    "birthDate": "1992-02-02",
    "phoneNumber": "1234567890"
```

Handling the exceptions or error with error codes



PUT http://localhost:7001/enrollees/update

```
{
    "enrollId": 1,
    "name": "abc",
    "status": true,
    "birthDate": "1992-02-02",
```

Status: 404 Not Found Time: 17 ms Size: 253 B

```
{
    "timestamp": "2021-02-28T06:03:20.244+0000",
    "message": " resource not found",
    "detail": "uri=/enrollees/update",
    "status": "Not Found"
}
```

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

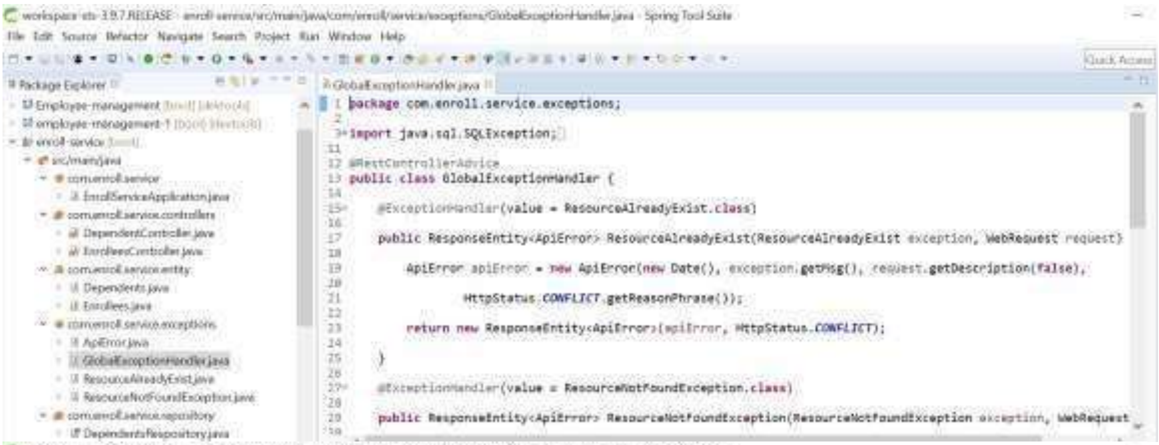
-Used spring boot

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

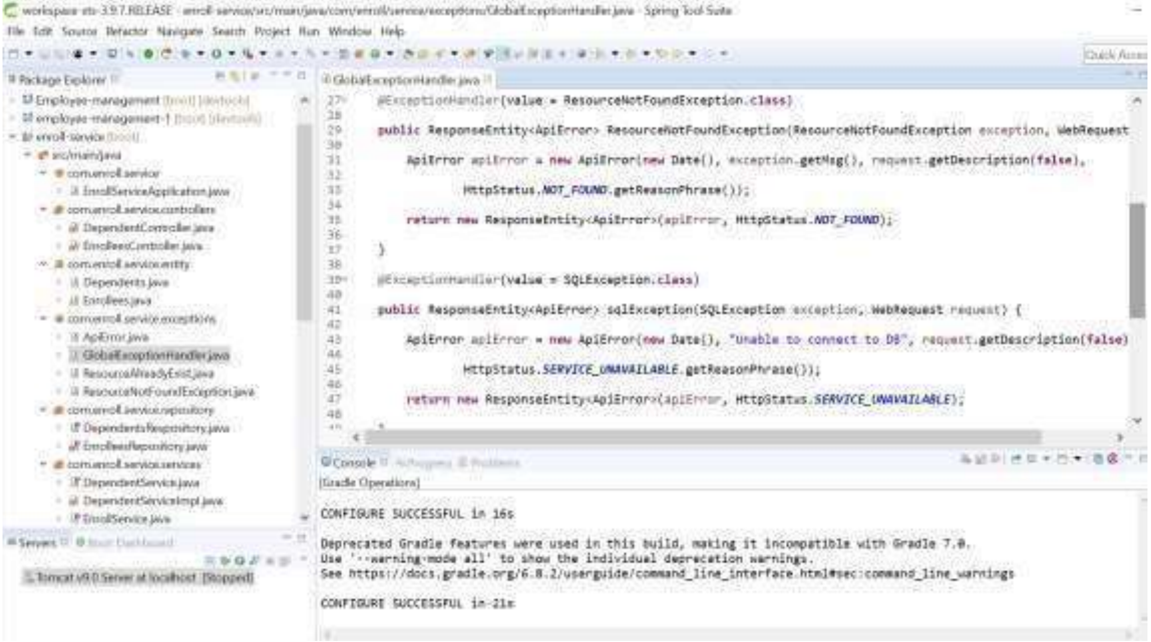
-Provide required documentation in this word doc to explain to showcase everything clearly!

- **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:


workspace-sts-3.9.7.RELEASE - enroll-service/src/main/java/com/enroll/service/exceptions/GlobalExceptionHandler.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
Employee-management [boot] [devtools]
employee-management-1 [boot] [devtools]
enroll-service [boot]
src/main/java
com.enroll.service
EnrollServiceApplication.java
com.enroll.service.controllers
DependentController.java
EnrolleesController.java
com.enroll.service.entity
Dependents.java
Enrollees.java
com.enroll.service.exceptions
ApiError.java
GlobalExceptionHandler.java
ResourceAlreadyExist.java
ResourceNotFoundException.java
com.enroll.service.repository
DependentsRespository.java
GlobalExceptionHandler.java

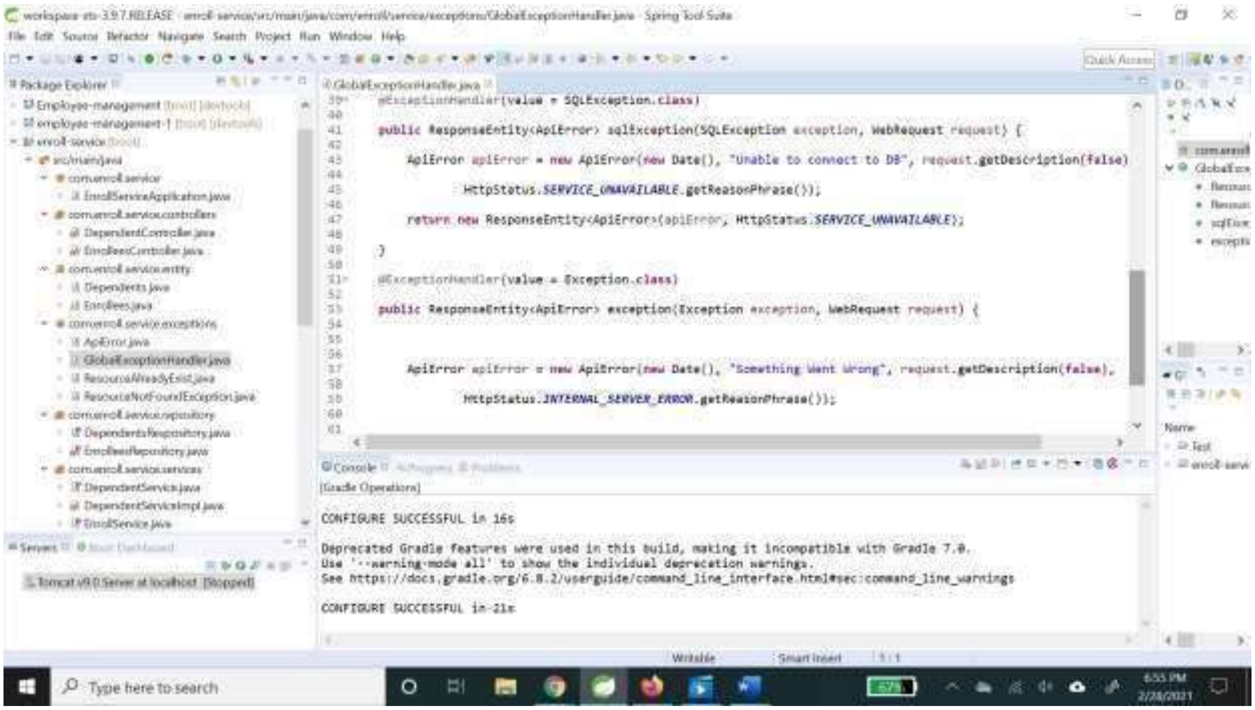
```
package com.enroll.service.exceptions;

import java.sql.SQLException;

@RestControllerAdvice
public class GlobalExceptionHandler {

    @ExceptionHandler(value = ResourceAlreadyExist.class)

    public ResponseEntity<ApiError> ResourceAlreadyExist(ResourceAlreadyExist exception, WebRequest request)

        ApiError apiError = new ApiError(new Date(), exception.getMsg(), request.getDescription(false),

                HttpStatus.CONFLICT.getReasonPhrase());

        return new ResponseEntity<ApiError>(apiError, HttpStatus.CONFLICT);

    }

    @ExceptionHandler(value = ResourceNotFoundException.class)

    public ResponseEntity<ApiError> ResourceNotFoundException(ResourceNotFoundException exception, WebRequest
```


workspace-sts-3.9.7.RELEASE - enroll-service/src/main/java/com/enroll/service/exceptions/GlobalExceptionHandler.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
Employee-management [boot] [devtools]
employee-management-1 [boot] [devtools]
enroll-service [boot]
src/main/java
com.enroll.service
EnrollServiceApplication.java
com.enroll.service.controllers
DependentController.java
EnrolleesController.java
com.enroll.service.entity
Dependents.java
Enrollees.java
com.enroll.service.exceptions
ApiError.java
GlobalExceptionHandler.java
ResourceAlreadyExist.java
ResourceNotFoundException.java
com.enroll.service.repository
DependentsRespository.java
EnrolleesRepository.java
com.enroll.service.services
DependentService.java
DependentServiceImpl.java
EnrollService.java
Servers
Boot Dashboard
Tomcat v9.0 Server at localhost [Stopped]
GlobalExceptionHandler.java
Console
Progress
Problems
[Gradle Operations]

```
    @ExceptionHandler(value = ResourceNotFoundException.class)

    public ResponseEntity<ApiError> ResourceNotFoundException(ResourceNotFoundException exception, WebRequest

        ApiError apiError = new ApiError(new Date(), exception.getMsg(), request.getDescription(false),

                HttpStatus.NOT_FOUND.getReasonPhrase());

        return new ResponseEntity<ApiError>(apiError, HttpStatus.NOT_FOUND);

    }

    @ExceptionHandler(value = SQLException.class)

    public ResponseEntity<ApiError> sqlException(SQLException exception, WebRequest request) {

        ApiError apiError = new ApiError(new Date(), "Unable to connect to DB", request.getDescription(false)

                HttpStatus.SERVICE_UNAVAILABLE.getReasonPhrase());

        return new ResponseEntity<ApiError>(apiError, HttpStatus.SERVICE_UNAVAILABLE);
```

```
CONFIGURE SUCCESSFUL in 16s

Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/6.8.2/userguide/command_line_interface.html#sec:command_line_warnings

CONFIGURE SUCCESSFUL in 21s
```

Handles SQL exception as below:

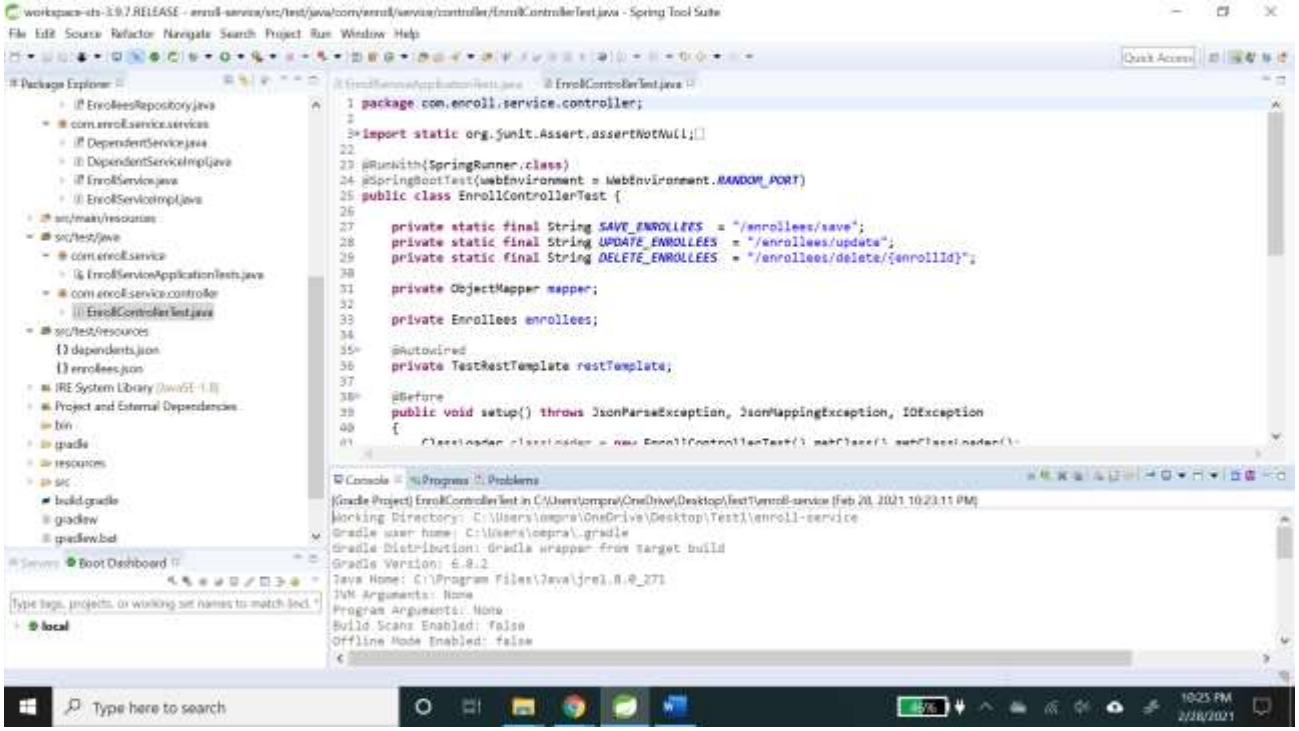

workspace-sts-3.9.7.RELEASE - enroll-service/src/main/java/com/enroll/service/exceptions/GlobalExceptionHandler.java - Spring Tool Suite
GlobalExceptionHandler.java
Console

```java
    @ExceptionHandler(value = SQLException.class)

    public ResponseEntity<ApiError> sqlException(SQLException exception, WebRequest request) {

        ApiError apiError = new ApiError(new Date(), "Unable to connect to DB", request.getDescription(false)

                HttpStatus.SERVICE_UNAVAILABLE.getReasonPhrase());

        return new ResponseEntity<ApiError>(apiError, HttpStatus.SERVICE_UNAVAILABLE);

    }

    @ExceptionHandler(value = Exception.class)

    public ResponseEntity<ApiError> exception(Exception exception, WebRequest request) {



        ApiError apiError = new ApiError(new Date(), "Something Went Wrong", request.getDescription(false),

                HttpStatus.INTERNAL_SERVER_ERROR.getReasonPhrase());
```

```
[Gradle Operations]

CONFIGURE SUCCESSFUL in 16s

Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/6.8.2/userguide/command_line_interface.html#sec:command_line_warnings

CONFIGURE SUCCESSFUL in 21s
```

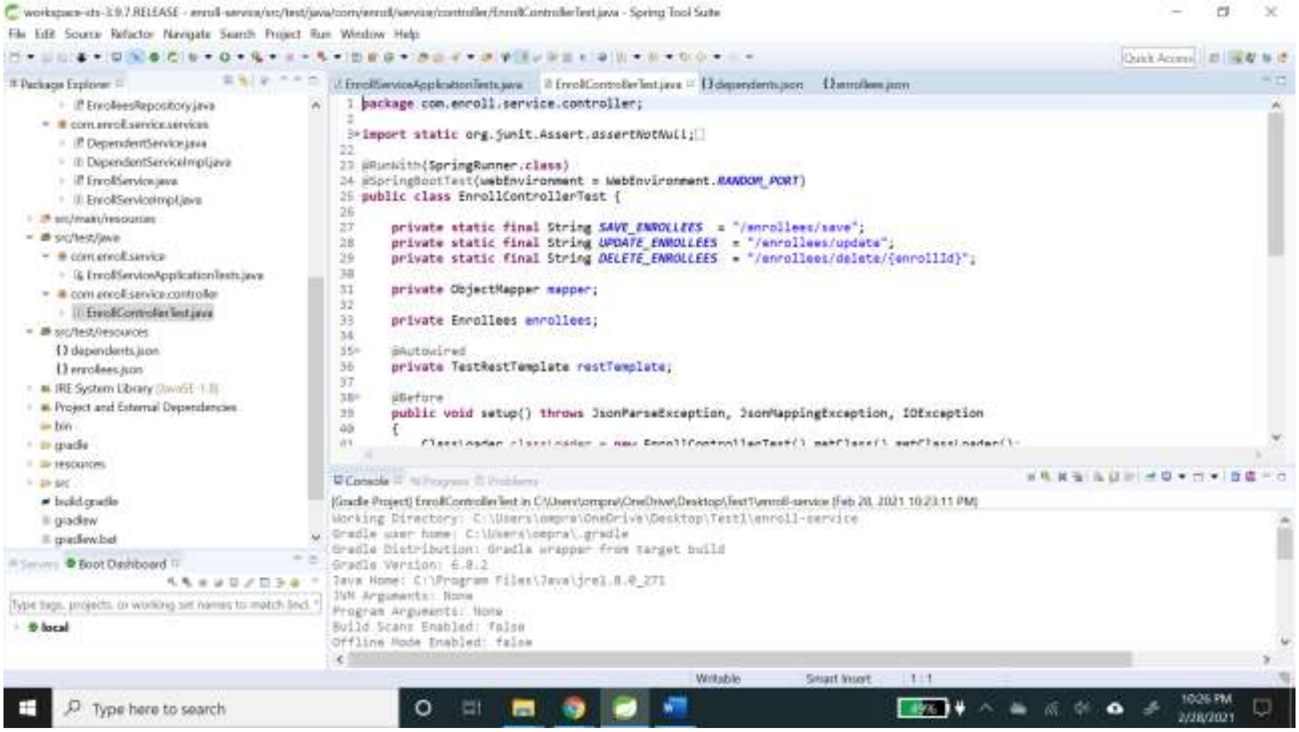
- **Include Unit Test cases**

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


workspace-sts-3.9.7.RELEASE - enroll-service/src/test/java/com/enroll/service/controller/EnrollControllerTest.java - Spring Tool Suite
EnrollControllerTest.java
Console
Progress
Problems

```java
package com.enroll.service.controller;

import static org.junit.Assert.assertNotNull;

@RunWith(SpringRunner.class)
@SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT)
public class EnrollControllerTest {

    private static final String SAVE_ENROLLEES  = "/enrollees/save";
    private static final String UPDATE_ENROLLEES  = "/enrollees/update";
    private static final String DELETE_ENROLLEES  = "/enrollees/delete/{enrollId}";

    private ObjectMapper mapper;

    private Enrollees enrollees;

    @Autowired
    private TestRestTemplate restTemplate;

    @Before
    public void setup() throws JsonParseException, JsonMappingException, IOException
    {
        ClassLoader classLoader = new EnrollControllerTest().getClass().getClassLoader();
```

```
[Gradle Project] EnrollControllerTest in C:\Users\ompra\OneDrive\Desktop\Test1\enroll-service [Feb 28, 2021 10:23:11 PM]
Working Directory: C:\Users\ompra\OneDrive\Desktop\Test1\enroll-service
Gradle user home: C:\Users\ompra\.gradle
Gradle Distribution: Gradle wrapper from target build
Gradle Version: 6.8.2
Java Home: C:\Program Files\Java\jre1.8.0_271
JVM Arguments: None
Program Arguments: None
Build Scans Enabled: false
Offline Mode Enabled: false
```


workspace-sts-3.9.7.RELEASE - enroll-service/src/test/java/com/enroll/service/controller/EnrollControllerTest.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
EnrolleesRepository.java
com.enroll.service.services
DependentService.java
DependentServiceImpl.java
EnrollService.java
EnrollServiceImpl.java
src/main/resources
src/test/java
com.enroll.service
EnrollServiceApplicationTests.java
com.enroll.service.controller
EnrollControllerTest.java
src/test/resources
dependents.json
enrollees.json
JRE System Library [JavaSE-1.8]
Project and External Dependencies
bin
gradle
resources
src
build.gradle
gradlew
gradlew.bat
Servers
Boot Dashboard
Type tags, projects, or working set names to match (incl. *
local
EnrollServiceApplicationTests.java
EnrollControllerTest.java
dependents.json
enrollees.json
Console
Progress
Problems
Writable
Smart Insert
1 : 1
Type here to search
10:26 PM
2/28/2021

```
package com.enroll.service.controller;

import static org.junit.Assert.assertNotNull;

@RunWith(SpringRunner.class)
@SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT)
public class EnrollControllerTest {

    private static final String SAVE_ENROLLEES  = "/enrollees/save";
    private static final String UPDATE_ENROLLEES  = "/enrollees/update";
    private static final String DELETE_ENROLLEES  = "/enrollees/delete/{enrollId}";

    private ObjectMapper mapper;

    private Enrollees enrollees;

    @Autowired
    private TestRestTemplate restTemplate;

    @Before
    public void setup() throws JsonParseException, JsonMappingException, IOException
    {
        ClassLoader classLoader = new EnrollControllerTest().getClass().getClassLoader();
```

```
[Gradle Project] EnrollControllerTest in C:\Users\ompra\OneDrive\Desktop\Test1\enroll-service (Feb 28, 2021 10:23:11 PM)
Working Directory: C:\Users\ompra\OneDrive\Desktop\Test1\enroll-service
Gradle user home: C:\Users\ompra\.gradle
Gradle Distribution: Gradle wrapper from target build
Gradle Version: 6.8.2
Java Home: C:\Program Files\Java\jre1.8.0_271
JVM Arguments: None
Program Arguments: None
Build Scans Enabled: false
Offline Mode Enabled: false
```

Along with JSON files!

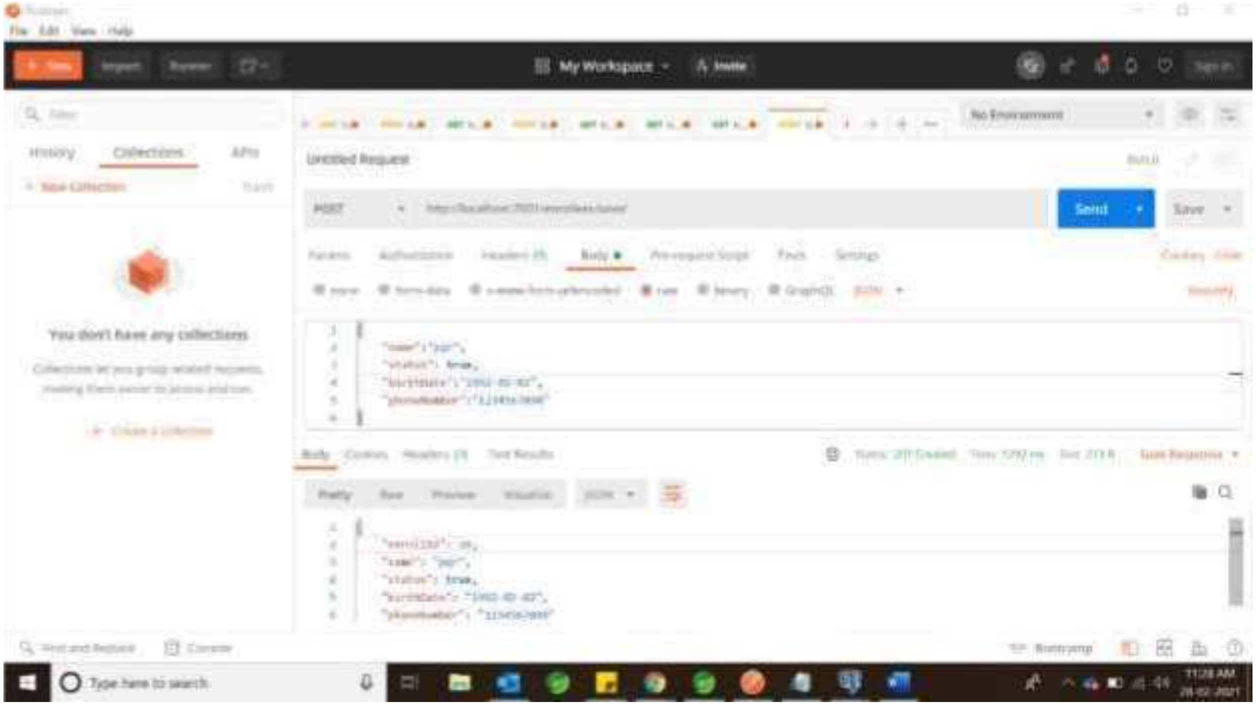
- **Clean Code and Proper Code Standards**

-

# 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



POST http://localhost:8081/enrollees/save/

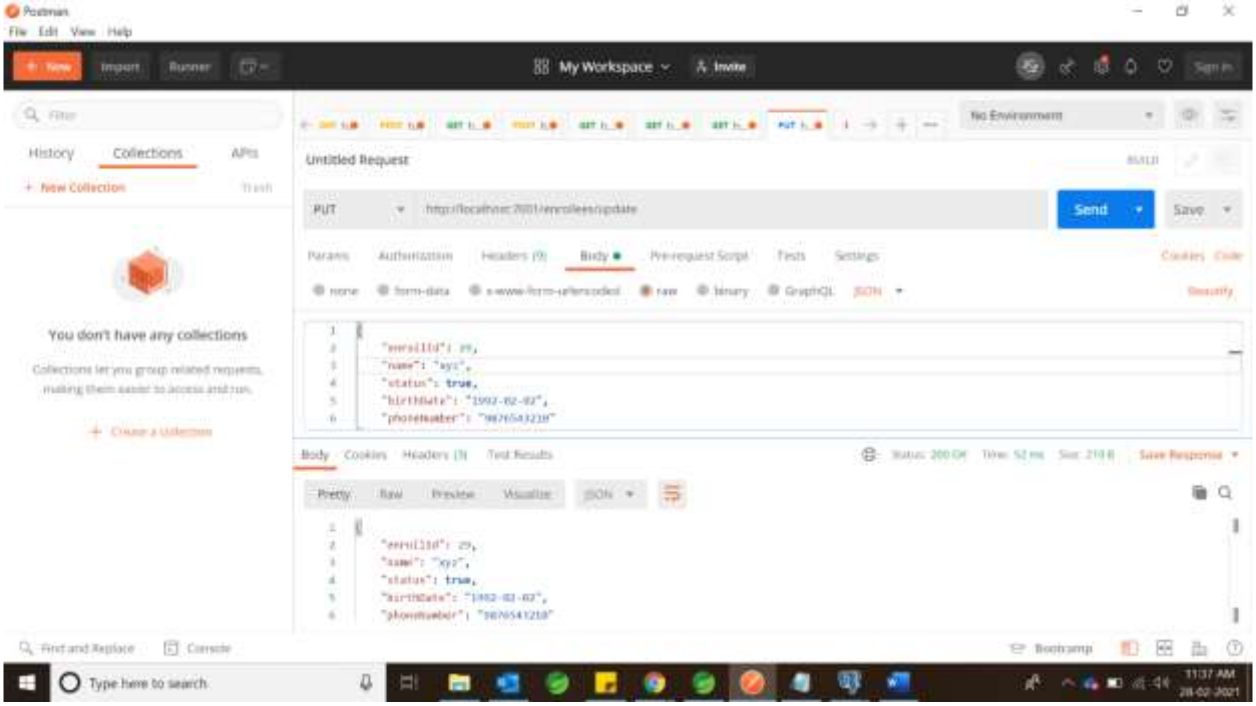
```
{
    "name":"pqr",
    "status": true,
    "birthDate":"1992-02-02",
    "phoneNumber":"1234567890"
}
```

Status: 201 Created Time: 1292 ms Size: 213 B

```
{
    "enrolleeId": 20,
    "name": "pqr",
    "status": true,
    "birthDate": "1992-02-02",
    "phoneNumber": "1234567890"
```

The above data has all the required fields

- Modify an existing enrollee



Untitled Request

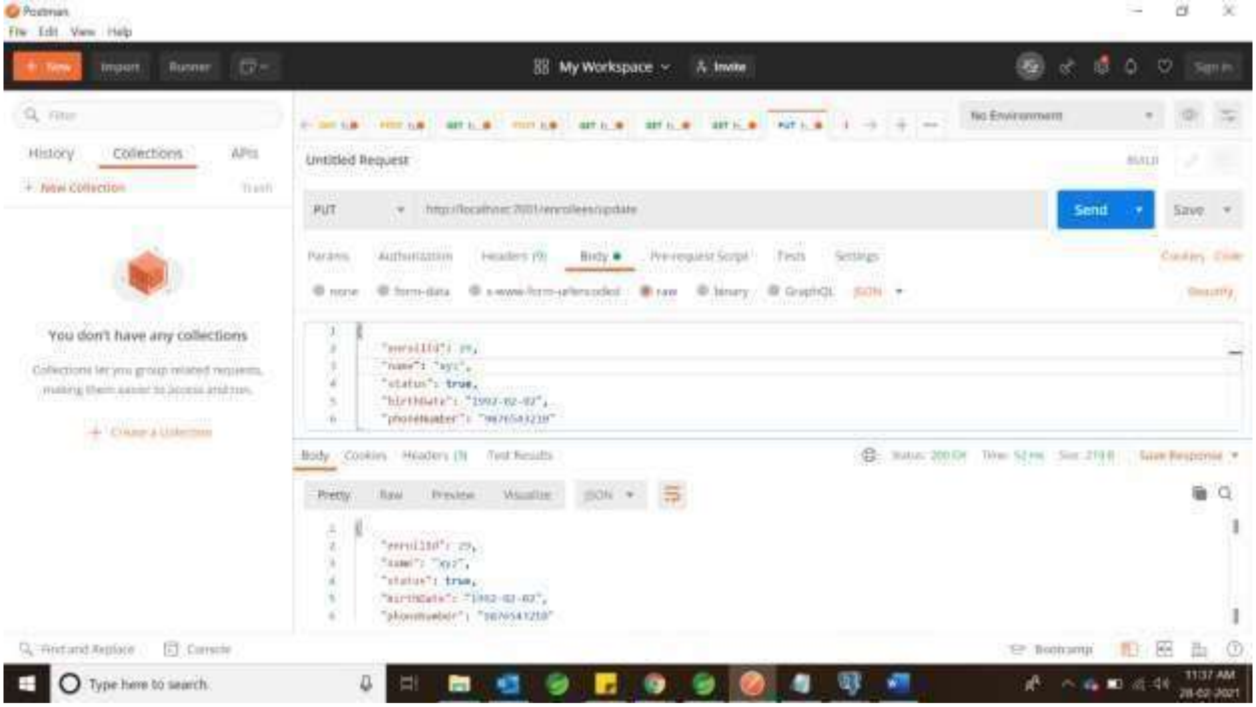
PUT http://localhost:7001/enrollees/update

Body: raw JSON

```
{
    "enrollId": 29,
    "name": "xyz",
    "status": true,
    "birthDate": "1992-02-02",
    "phoneNumber": "9876543210"
```

Status: 200 OK Time: 52 ms Size: 219 B

```
{
    "enrollId": 29,
    "name": "xyz",
    "status": true,
    "birthDate": "1992-02-02",
    "phoneNumber": "9876543210"
```



Untitled Request

PUT http://localhost:7001/enrollees/update

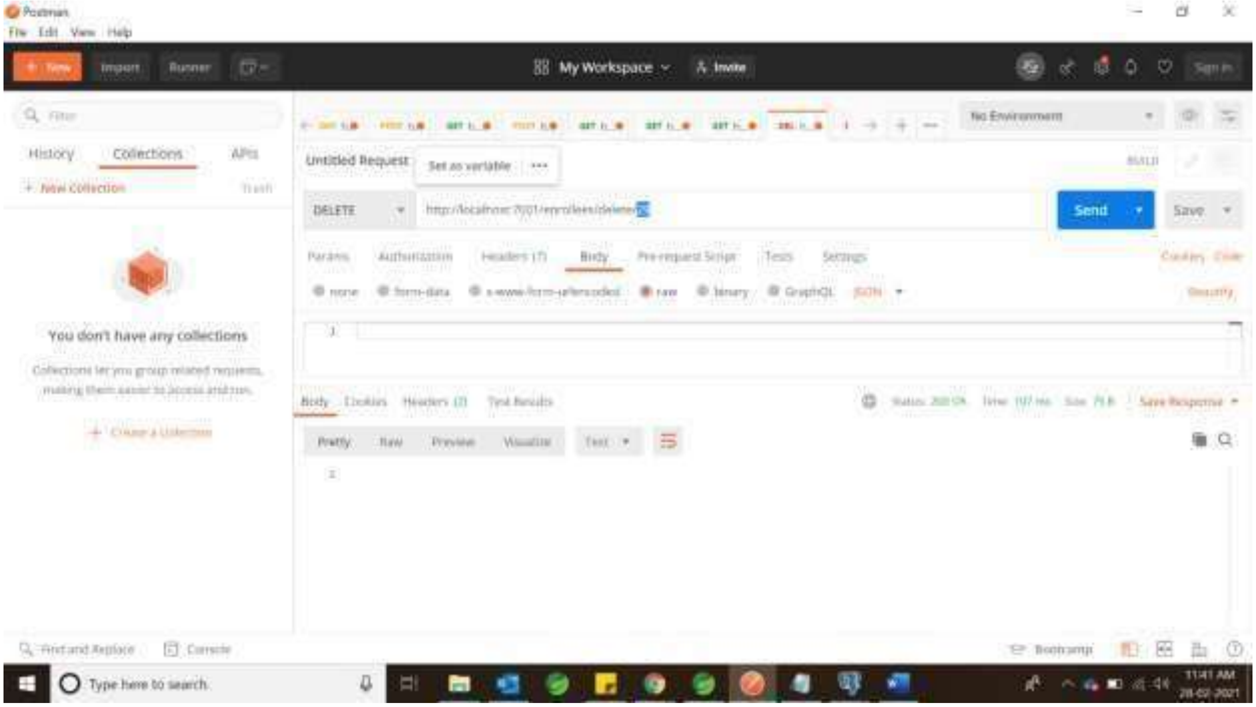
Body: raw JSON

```
{
    "enrollId": 29,
    "name": "xyz",
    "status": true,
    "birthDate": "1992-02-02",
    "phoneNumber": "9876543210"
```

Status: 200 OK Time: 52 ms Size: 219 B

```
{
    "enrollId": 29,
    "name": "xyz",
    "status": true,
    "birthDate": "1992-02-02",
    "phoneNumber": "9876543210"
```

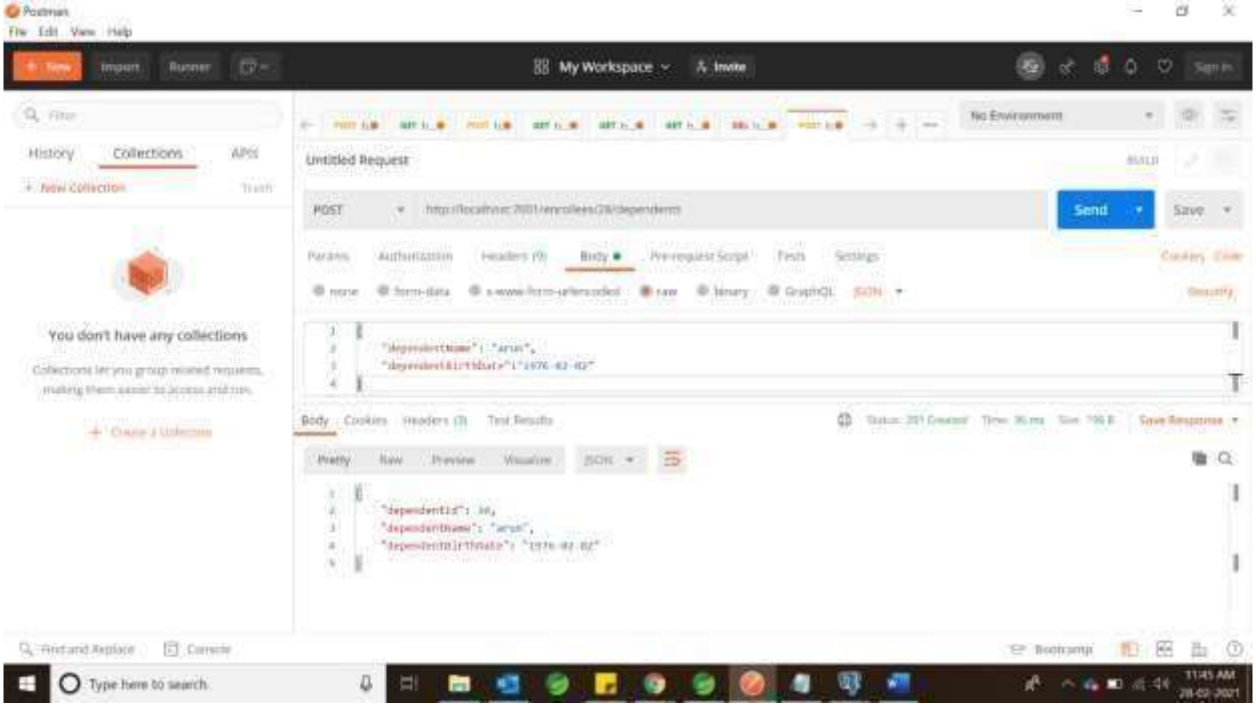
- Remove an enrollee entirely



DELETE http://localhost:7001/enrollees/delete/[illegible]

Status: 200 OK Time: 107 ms Size: 75 B

- Add dependents to an enrollee



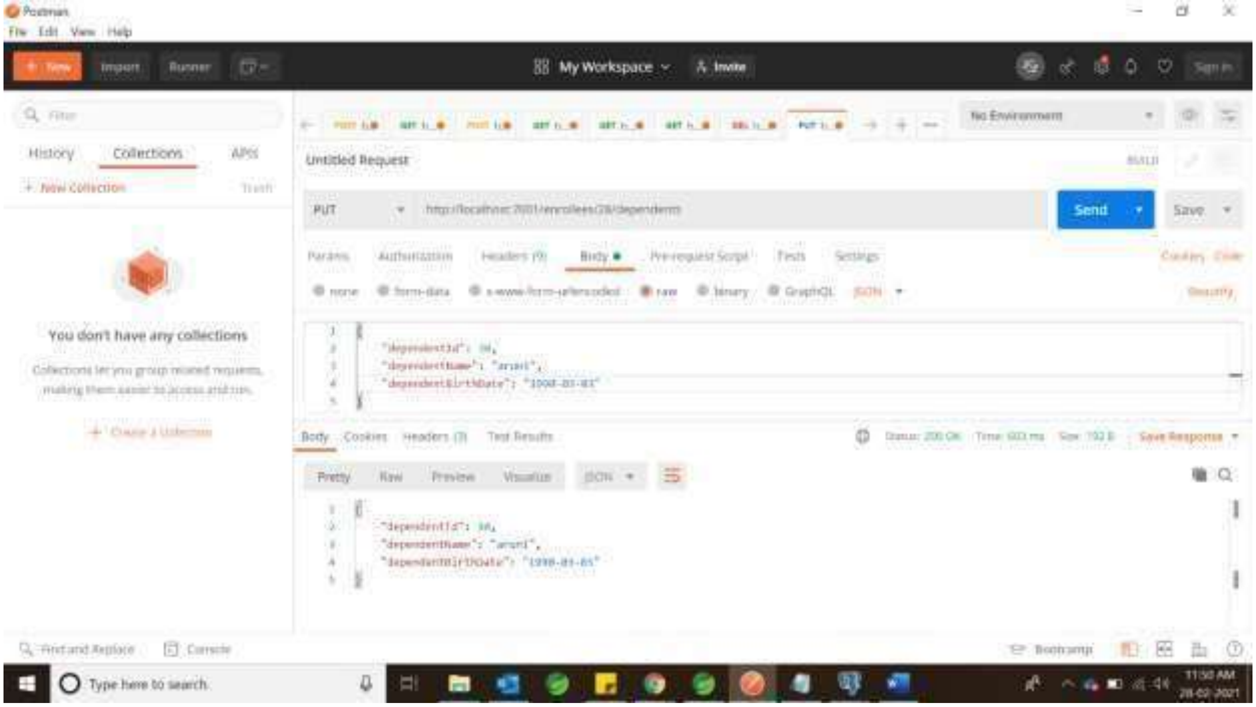
POST http://localhost:7001/enrollees/28/dependents

```
{
    "dependentName": "arun",
    "dependentBirthDate":"1976-02-02"
}
```

Status: 201 Created Time: 36 ms Size: 106 B

```
{
    "dependentId": 30,
    "dependentName": "arun",
    "dependentBirthDate": "1976-02-02"
}
```

- Modify existing dependents



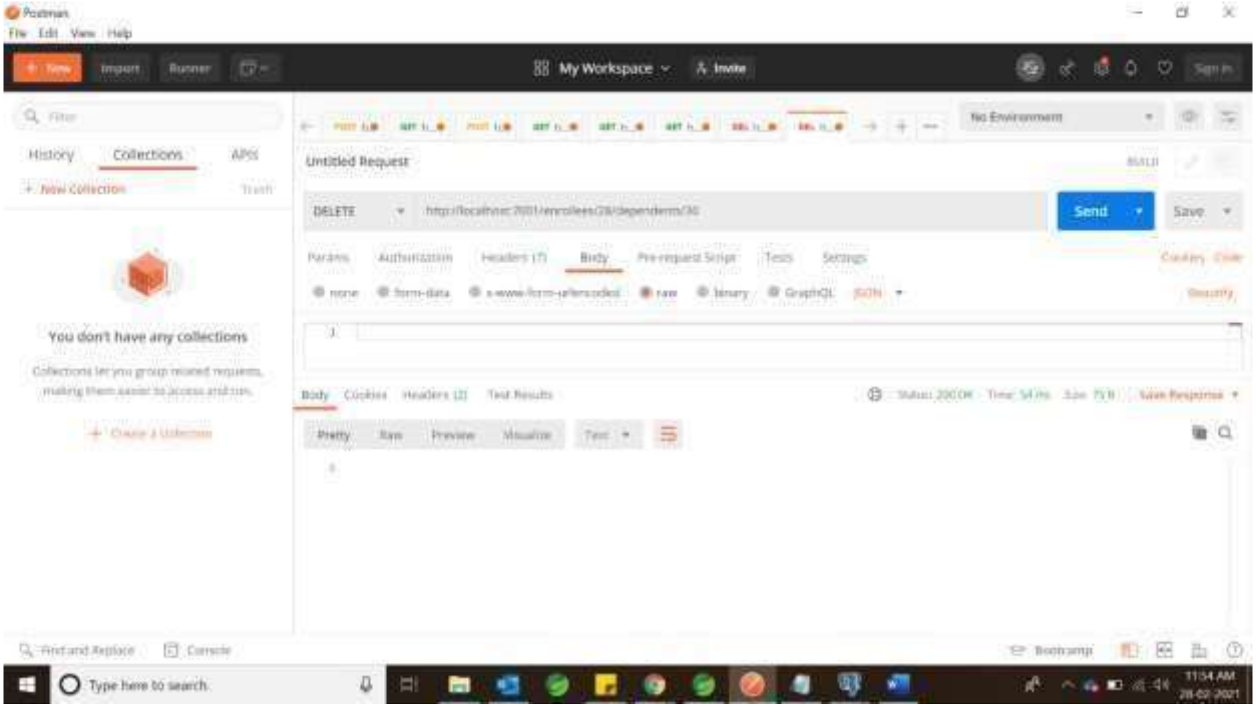
PUT http://localhost:2001/enrollees/28/dependents

```
{
    "dependentId": 38,
    "dependentName": "arun1",
    "dependentBirthDate": "1998-03-03"
}
```

Status: 200 OK Time: 603 ms Size: 192 B

```
{
    "dependentId": 38,
    "dependentName": "arun1",
    "dependentBirthDate": "1998-03-05"
}
```

- Remove dependents from an enrollee

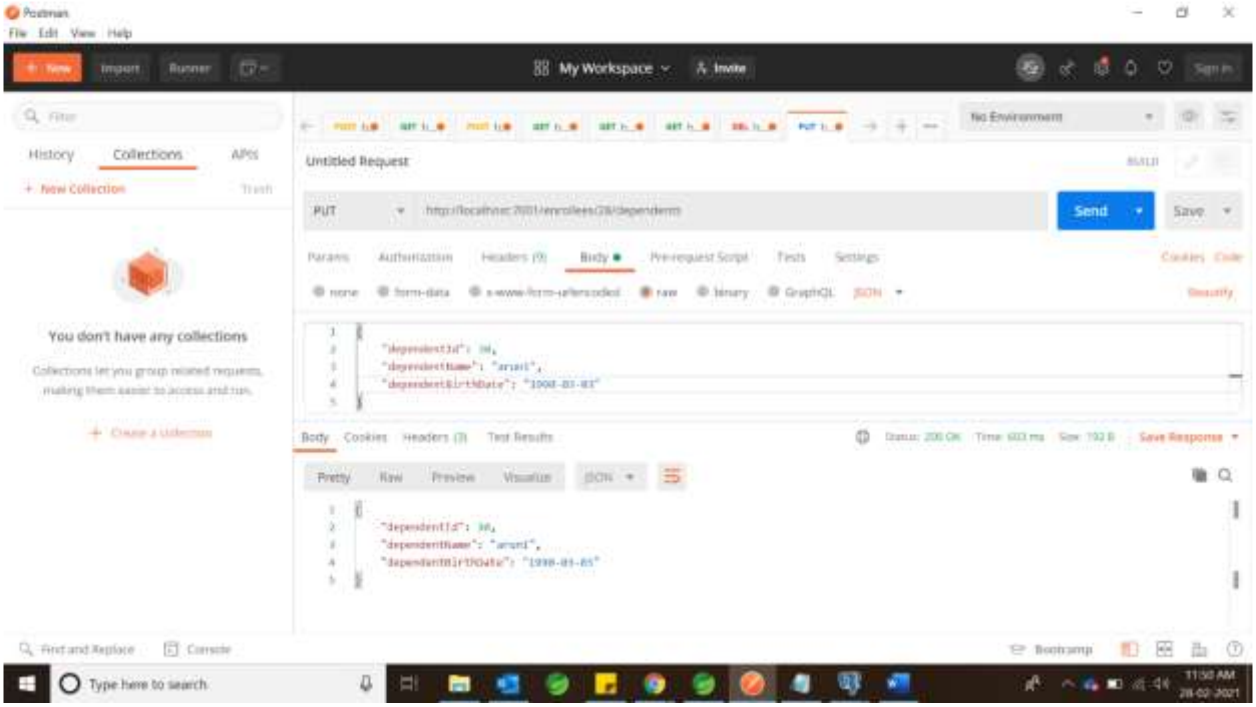

Untitled Request

DELETE http://localhost:7001/enrollees/28/dependents/30

Body: raw JSON

Status: 200 OK Time: 54 ms Size: 75 B

- Modify existing dependents


Untitled Request

PUT http://localhost:7001/enrollees/28/dependents

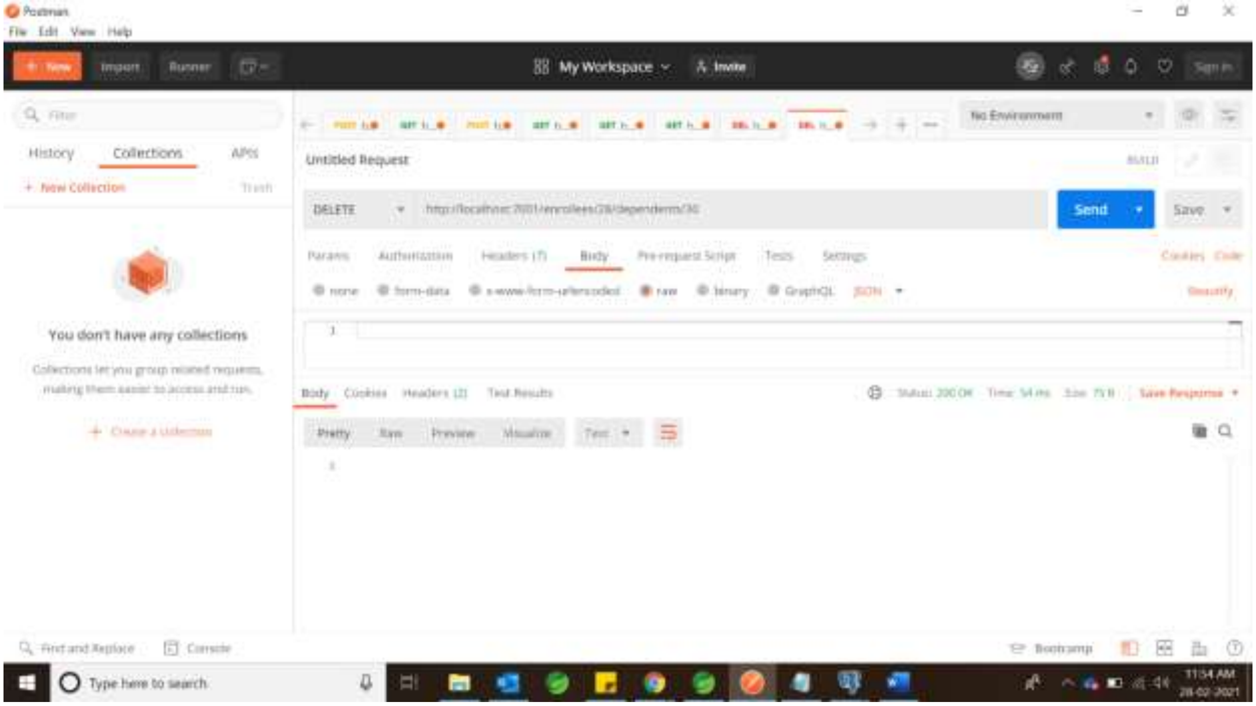
Body: raw JSON

```
{
    "dependentId": 30,
    "dependentName": "aruni",
    "dependentBirthDate": "1998-03-03"
}
```

Status: 200 OK Time: 603 ms Size: 192 B

```
{
    "dependentId": 30,
    "dependentName": "aruni",
    "dependentBirthDate": "1998-03-05"
}
```

- Remove dependents from an enrollee



Untitled Request

DELETE http://localhost:7001/enrollees/28/dependents/30

Status: 200 OK Time: 54 ms Size: 75 B

11:54 AM
28-02-2021