

# WEB PROGRAMMING

## CRUD API - Ruby On Rails

---

Submitted by: Group 3

- Soujanya Namburi
  - Sreelakshmi K
  - Ajay P Sakthikumar
- 

[Click here for the code](#)

### STEPS:

1. Create the application

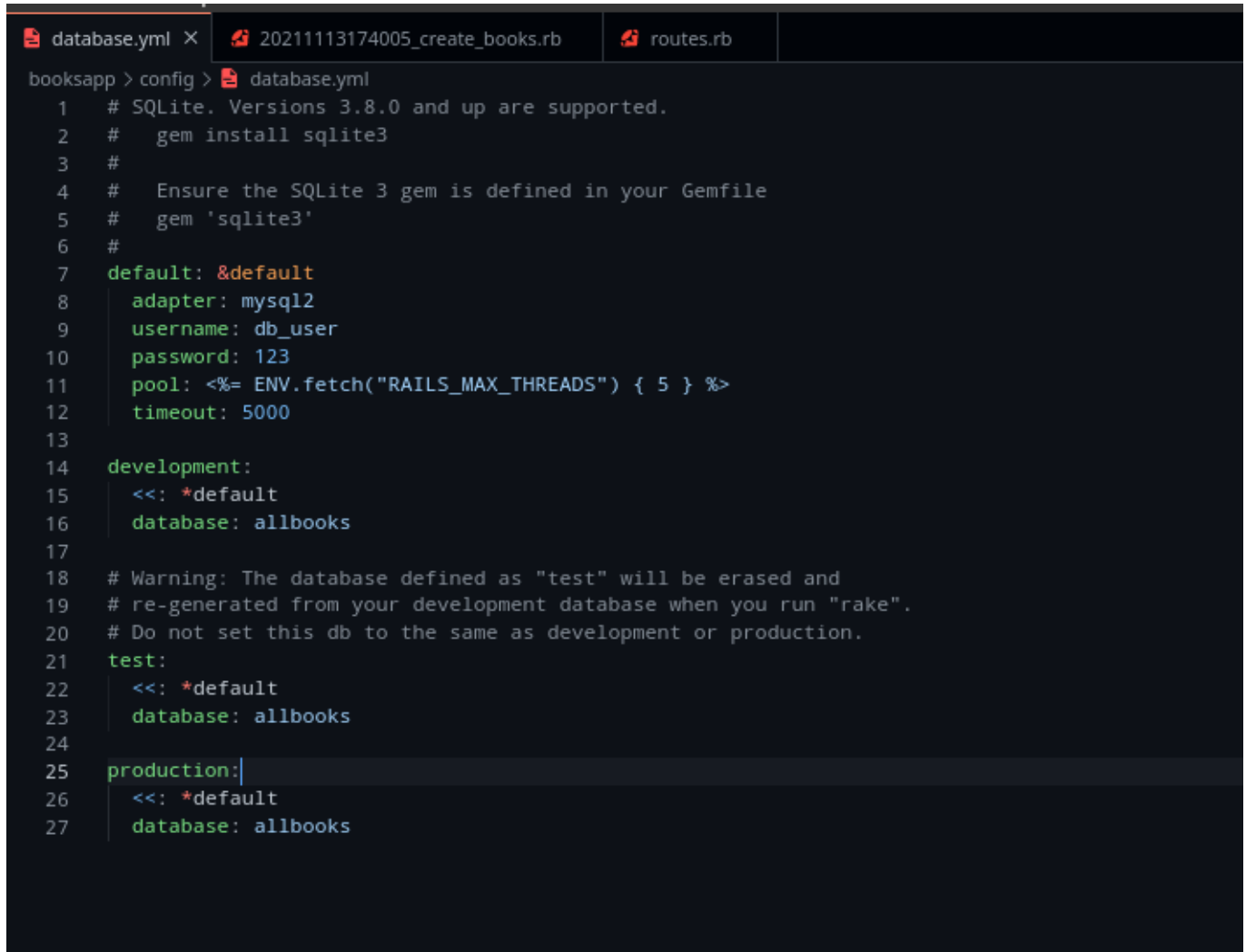
```
rails new booksapp --api
cd booksapp
```

2. Set up the database:

We are using mysql2 instead of sqlite3, so make the following change in the Gemfile.

```
# gem 'sqlite3'
gem 'mysql2'
```

Make the following changes to database.yml file as well



```
booksapp > config > database.yml
1  # SQLite. Versions 3.8.0 and up are supported.
2  #   gem install sqlite3
3  #
4  # Ensure the SQLite 3 gem is defined in your Gemfile
5  #   gem 'sqlite3'
6  #
7  default: &default
8    adapter: mysql2
9    username: db_user
10   password: 123
11   pool: <%= ENV.fetch("RAILS_MAX_THREADS") { 5 } %>
12   timeout: 5000
13
14  development:
15    <<: *default
16    database: allbooks
17
18  # Warning: The database defined as "test" will be erased and
19  # re-generated from your development database when you run "rake".
20  # Do not set this db to the same as development or production.
21  test:
22    <<: *default
23    database: allbooks
24
25  production:
26    <<: *default
27    database: allbooks
```

3. We use the MVC (Model View Controller) Architecture here. We create a model Book with the given attributes.

```
Book[B_id, B_title, Author, Publisher, Year]
```

PS. we will have another id element in the database. That is used to perform the CRUD operations. B\_id is a different attribute

Run the following command to create a model:

```
rails g model Book B_title Author Year Publisher
```

The following files will be created on running the command:

- a. app/models/books.rb
- b. db/migrate/[date\_time]\_create\_books.rb

This is the final schema: (In db/migrate/schema.rb)

```
ActiveRecord::Schema.define(version: 2021_11_13_174005) do

  create_table "books", charset: "utf8mb4", collation: "utf8mb4_unicode_ci", force: :cascade do |t|
    t.float "B_id"
    t.string "B_title"
    t.string "Author"
    t.string "Publisher"
    t.date "Year"
    t.datetime "created_at", precision: 6, null: false
    t.datetime "updated_at", precision: 6, null: false
  end
end
```

- Now that we created the database with appropriate columns, we migrate it to the actual database using the following command:

```
rails db:migrate
```

- We could also initiate values by editing the db/migrate/seeds.rb file. We have used an external library to make dummy values on which we could work on.

```
5.times do
  Book.create({
    B_id: Faker::Number.number(digits: 4),
    B_title: Faker::Book.title,
    Author: Faker::Book.author,
    Publisher: Faker::Book.publisher,
    Year: Faker::Date.between(from: '1900-09-23', to: '2014-09-25'),
  })
end
```

This can be initialised using the following command:

```
rails db:seed
```

- Make an api folder in the app/controllers folder with the following contents: v1/books\_controller.rb

```
module Api
  module V1
    class BooksController < ApplicationController
      def index
        books = Book.order('created_at DESC');
        render json: {status: 'SUCCESS', message:'Loaded books',
data:books},status: :ok
      end
    end
  end
end
```

```
      def show
        book = Book.find(params[:id])
        render json: {status: 'SUCCESS', message: 'Loaded book',
data:book},status: :ok
      end

      def create
        book = Book.new(book_params)

        if book.save
          render json: {status: 'SUCCESS', message: 'Saved book',
data:book},status: :ok
        else
          render json: {status: 'ERROR', message: 'book not saved',
data:book.errors},status: :unprocessable_entity
        end
      end

      def destroy
        book = Book.find(params[:id])
        book.destroy
        render json: {status: 'SUCCESS', message: 'Deleted book',
data:book},status: :ok
      end

      def update
        book = Book.find(params[:id])
        if book.update(book_params)
          render json: {status: 'SUCCESS', message: 'Updated book',
data:book},status: :ok
        else
          render json: {status: 'ERROR', message: 'book not updated',
data:book.errors},status: :unprocessable_entity
        end
      end

      private

      def book_params
        params.permit(:B_id, :B_title, :Author, :Publisher, :Year)
      end
    end
  end
end
```

Define the above methods

7. Edit the routes.rb file to update the versions. The versions are made to made the API more scalable, config/routes.rb

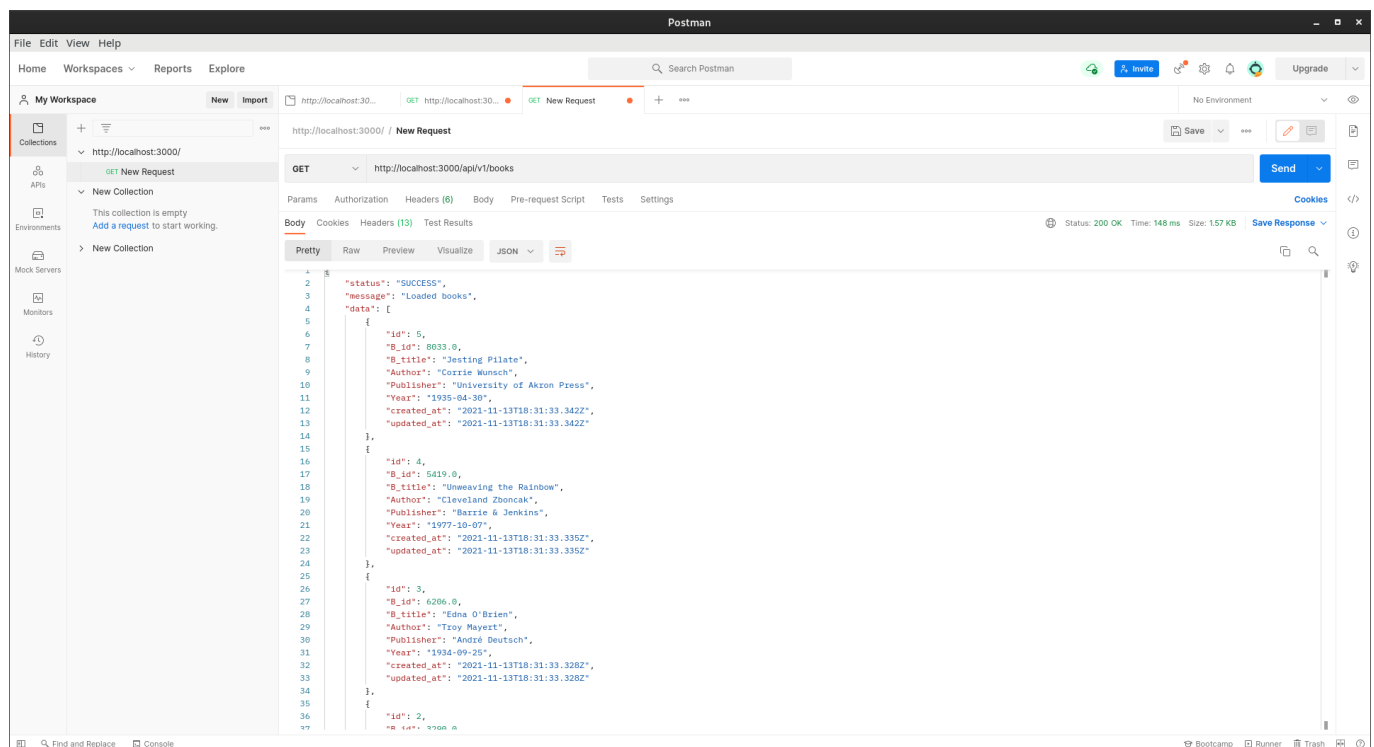
```
Rails.application.routes.draw do
  namespace 'api' do
    namespace 'v1' do
      resources :books
    end
  end
end
```

8. The API is ready. Run the app using the following command:

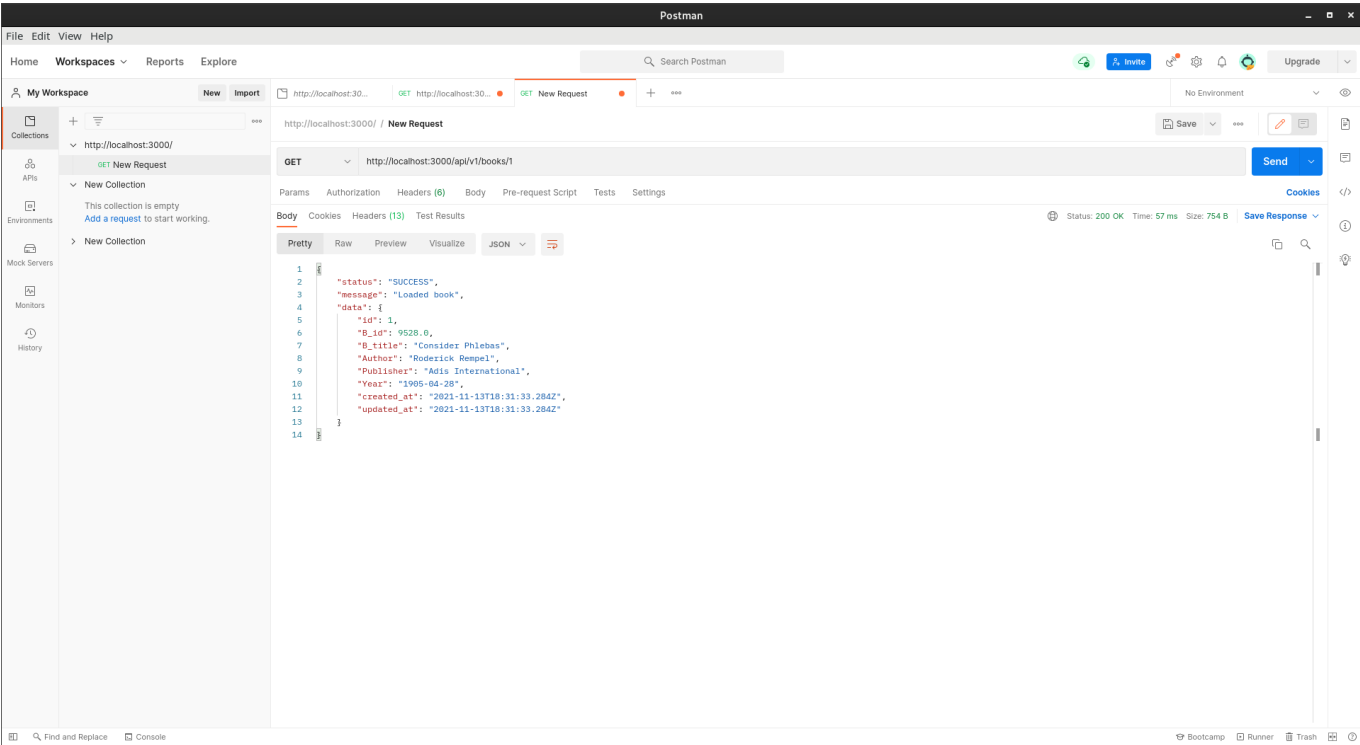
```
rails s
```

9. Give the below requests using POSTMAN or any other services

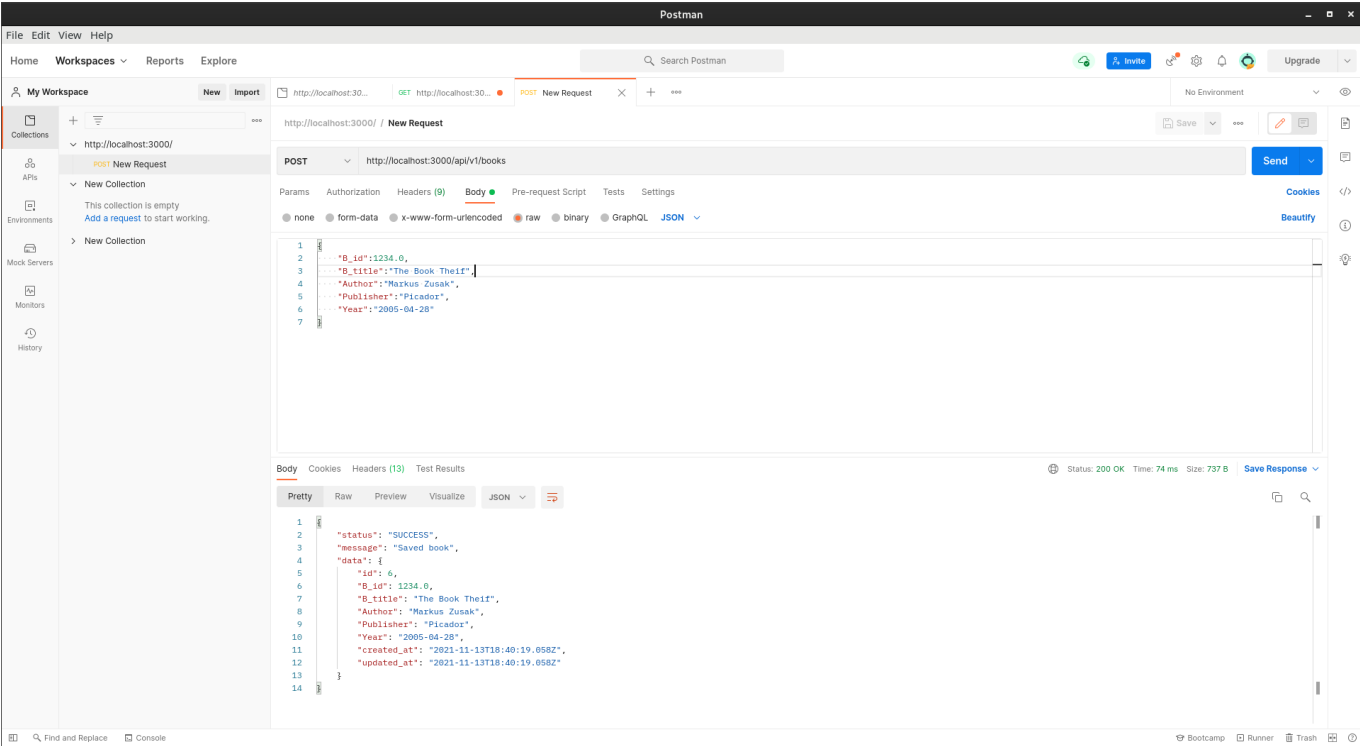
## GET ALL



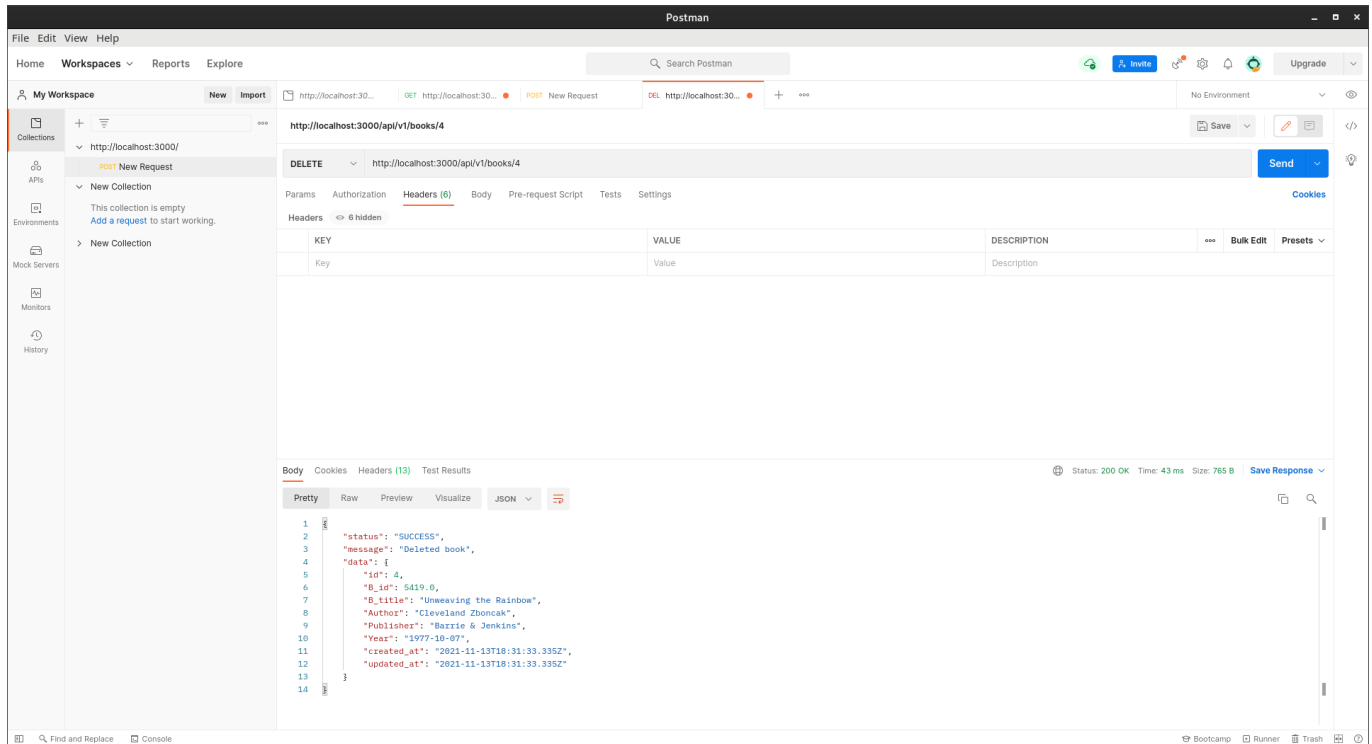
## GET with certain ID



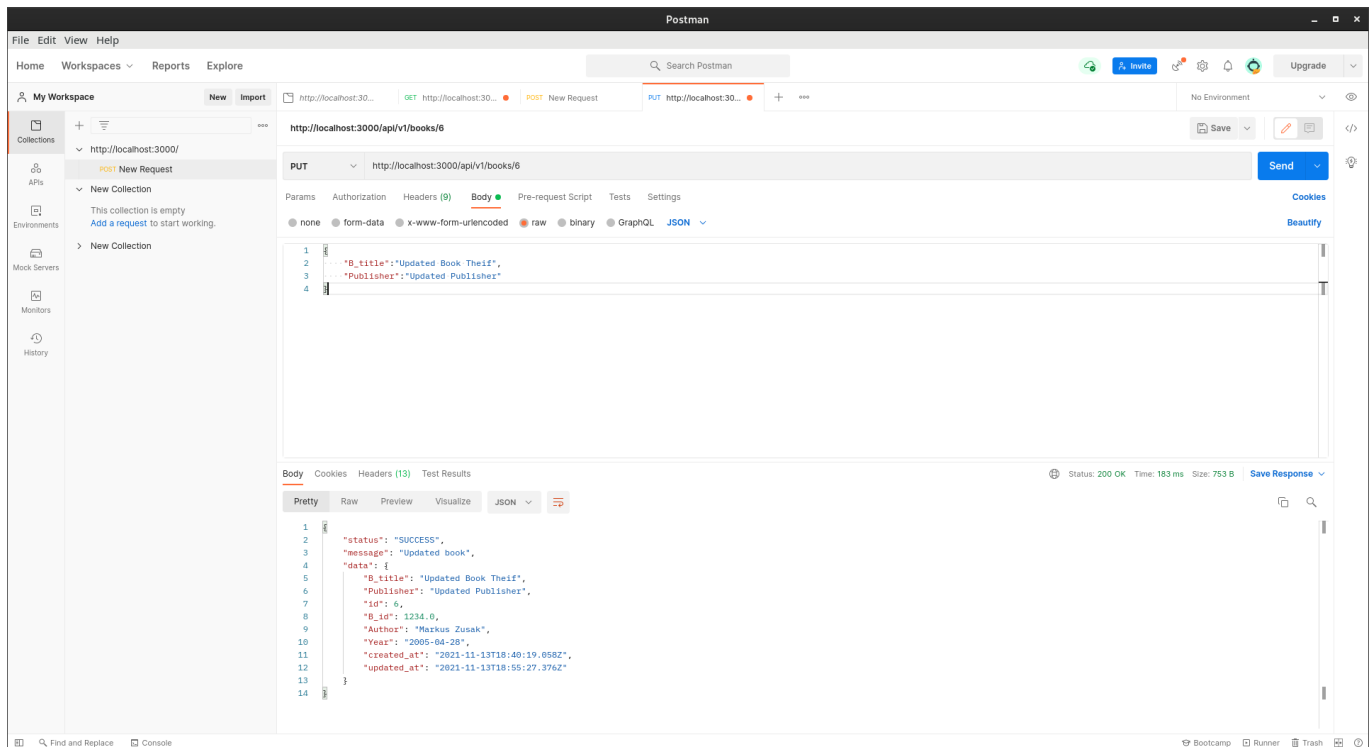
POST



DELETE



## UPDATE



## Individual Contributions

1. Ajay P Sakthikumar - POST, DELETE
2. Soujanya Namburi - PUT, Routes, Database
3. Sreelakshmi K - Initialized DB with Faker, GET, GET with id

Testing - All together