

**Mohamed Ahmed Hashish**

**Blog Application Documentation**

Using Ruby on Rails

## Introduction

A blog application developed using Ruby on Rails (Rails 7.2.1 and ruby 3.3.4) on Ubuntu 22.04.5 LTS

## Main gems used

- Bcrypt: to make sure that passwords are hashed in the database
- Jwt: to use JWT authentication throughout the application
- Sidekiq and redis: to schedule and perform post deletion, 24 hours after creation time.

## Models:

- User:
  - id, name, email, password, image.
  - Email is unique
  - Images can possibly be represented as Base64 encoded
- Post:
  - id, title, body, user\_id, tags, comments
- Comment:
  - id, user\_id, body
- Tag:
  - id, name

## Relationships and DB tables:

- Tables:
  - Users
  - Posts
  - Comments
  - Tags
  - Posts\_tags (many-to-many relationship)

## Endpoints:

- **Post** "user/create": to be able to create new users, and a token is returned to authenticate later while using all other endpoints
- **Get** "user/login": to login with the email & password combination, a token is returned to access all other functionalities that need authentication
- **Post** "post/new": to create a new post which belongs to the author(user)
- **Put** "post/update": to let the author update the title and/or the body of his post, and it must be done by the author
- **Put** "post/tags": to update the tags of specific posts, and it must be done by the author
- **Delete** "post": to delete a specific post, and it must be done by the author
- **Post** "comment/create": any author is capable of adding any comment on an existing post

- **Delete** "comment": the author of any comment can delete his/her own comment
- **Put** "comment": to update the text of a specific comment, and it must be done by the author

## **Testing & Tools**

- Postman: mainly used to manually test the endpoints of the API
- A test file of 6 test cases is used to test the Users Controller using RSpec

## **To run the application**

sudo docker-compose up