Mini Blogging Platform - Code Snippets

Code Snippets

User Model

```ruby
# app/models/user.rb
class User < ApplicationRecord
  devise :database_authenticatable, :registerable, :recoverable,
     :validatable
  has_many :posts, dependent: :destroy
  has_many :comments, dependent: :destroy
end
```

Post Model

```ruby
# app/models/post.rb
class Post < ApplicationRecord
  extend FriendlyId
  friendly_id :title, use: :slugged
  belongs_to :user
  has_many :comments, dependent: :destroy
  enum status: { draft: 0, published: 1 }
  validates :title, :body, presence: true
end
```

Comment Model

```ruby
# app/models/comment.rb
class Comment < ApplicationRecord
  belongs_to :user
  belongs_to :post
  validates :body, presence: true
end
```

Posts Controller

```ruby
# app/controllers/posts_controller.rb
class PostsController < ApplicationController
  before_action :authenticate_user!, except: [:index, :show]
  before_action :set_post, only: [:show, :edit, :update, :destroy]

  def index
    @posts = Post.published
    @posts = @posts.where('title␣ILIKE␣?␣OR␣body␣ILIKE␣?', "%#{params[:search]}%", "%#{params[:search]}%") if params[:search]
    @posts = @posts.where(user: current_user) if params[:own_posts] && current_user
    @posts = @posts.where(created_at: params[:start_date]..params[:end_date]) if params[:start_date] && params[:end_date]
  end

  def create
    @post = current_user.posts.build(post_params)
    if @post.save && @post.published?
      NotificationJob.perform_later(@post)
      redirect_to @post, notice: 'Post␣published!'
    elsif @post.save
      redirect_to @post, notice: 'Post␣saved␣as␣draft!'
    else
      render :new
    end
  end

  private
  def set_post
    @post = Post.friendly.find(params[:id])
  end
  def post_params
    params.require(:post).permit(:title, :body, :status)
  end
end
```

API Posts Controller

```
# app/controllers/api/posts_controller.rb
module Api
  class PostsController < ApplicationController
    before_action :authenticate_api_user!

    def index
      @posts = Post.published
      render json: @posts
    end

    def show
      @post = Post.friendly.find(params[:id])
      render json: @post
    end

    private
    def authenticate_api_user!
      authenticate_or_request_with_http_token do |token, _options|
        User.find_by(api_token: token)
      end
    end
  end
end
```

Notification Job

```
# app/jobs/notification_job.rb
class NotificationJob < ApplicationJob
  queue_as :default

  def perform(post)
    NotificationMailer.post_published(post).deliver_later
    Notification.create(user: post.user, message: "Post␣’#{post.title}’␣published!")
  end
end
```

Admin Controller

```
# app/controllers/admin_controller.rb
class AdminController < ApplicationController
  before_action :authenticate_admin!

  def index
    @posts = Post.all
    @users = User.all
  end

  def destroy_post
    @post = Post.friendly.find(params[:id])
    @post.destroy
    redirect_to admin_index_path, notice: ’Post␣deleted.’
  end

  private
  def authenticate_admin!
    authenticate_or_request_with_http_basic do |username, password|
      username == ’admin’ && password == ’secret’
    end
  end
end
```

Routes

```ruby
# config/routes.rb
Rails.application.routes.draw do
  devise_for :users
  resources :posts do
    resources :comments, only: [:create, :destroy]
  end
  get '/dashboard', to: 'dashboard#index'
  namespace :api do
    resources :posts, only: [:index, :show]
  end
  get '/admin', to: 'admin#index', as: :admin_index
  delete '/admin/posts/:id', to: 'admin#destroy_post'
  root 'posts#index'
end
```