# Designing API Endpoints

# User

## get user

* Endpiont

GET /api/v1/users

* Request Body : None
* Response Body

{  
 "message" : "Users data retrived successfully",  
 "status" : 200,  
 "data" :[  
 {  
 "id" : 1,  
 "username" : something,  
 "email" : something@something.com  
 }  
 ]  
}

## delete user

* Endpoint

DELETE /api/v1/users/{id}

* Request Body : None
* Response Body :

{  
 "message" : "User account deleted successfully",  
 "status" : 200  
}

# Auth

## register user

* Endpoint

POST /api/v1/auth/register

* Request Body

{  
 "username" : "something",  
 "email" : "something@something.com",  
 "password" : "something"  
}

* Response Body

{  
 "message" : "Register Successfully",  
 "status" : 201  
 "data" : {  
 "id" : 1,  
 "username" : "something",  
 "email" : "something@something.com"  
 }  
}

## login user

* Endpoint

POST /api/v1/user/login

* Request Body

{  
 "email" : "something@something.com",  
 "password" : "something"  
}

* Response Body

{  
 "message" : "Loggedin succuessfully",  
 "status" : 200,  
 "data" : {  
 "id" : 1,  
 "username" : "something",  
 "email" : "something",  
 "tokens" : {  
 "accessToken" : "something...something",  
 "refreshToken" : "something...something",  
 }  
 }  
}

## logout user

* Endpoint

POST /api/v1/auth/logout

* Request Body : None
* Response Body

{  
 "message" : "Logout Successfully",  
 "status" : 200  
}

## get access token

* Endpoint

POST /api/v1/auth/refresh

* Request Body

{  
 "authToken" : "something...something"  
}

* Response Body

{  
 "message" : "Access token generated",  
 "status" : 200,  
 "data" : {  
 "accessToken" : "something..something"  
 }  
}

# Password management

## change password

* Endpoint

POST /api/v1/auth/change-password

* Request Body

{  
 "newPassword" : "something...something"  
}

**Note** : user can be authenicated using token passed in header

* Response Body

{  
 "message" : "Password changed",  
 "status" : 200  
}

## forgot password

* Endpoint

POST /api/v1/auth/forgot-password

* Request Body

{  
 "email" : "something@something.com"  
}

* Response Body

{  
 "message" : "Link sent to your email address",  
 "status" : 200  
}

## reset password

* Endpoint

POST /api/v1/auth/reset-password

* Request Body

{  
 "token" : "token for changing password",  
 "newPassword" "somethingsomething"  
}

* Response Body

{  
 "message" : "Password reset successfully",  
 "status" : 200  
}

# Profile

## get current user profile

* Endpoint

GET /api/v1/prfile/me

* Response Body

{  
 "message" : "user profile retreived successfully",  
 "status" : 200,  
 "data" : {  
 "id" : 1,  
 "username" : "something",  
 "email" : "something@something.com"  
 }  
}

## update current user profile

* Endpont

PATCH /api/v1/profile/me

* Request Body

{  
 "username" : "anything",  
}

* Response Body

{  
 "message" : "User data updated",  
 "status" : 200,  
 "data" : {  
 "id" : 1,  
 "username" : "anything",  
 "email" : "something@something.com",  
 }  
}

## get public user profile

* Endpoint

GET /api/v1/profile/{id}

* Response Body

{  
 "message" : "user data retrived successfully",  
 "status" : 200,  
 "data" : {  
 "id" : 1,  
 "username" : "anything",  
 "email" : "something@something.com"  
 }  
}

# Posts

## list posts

* Endpoint

GET /api/v1/posts

* Response Body

{  
 "message": "Posts fetched successfully",  
 "status": 200,  
 "data": {  
 "posts": [  
 {  
 "id": "1",  
 "title": "My First Tech Blog",  
 "content": "This is the content of the blog post...",  
 "author": {  
 "id": "1",  
 "username": "anything",  
 },  
 },  
 {....},  
 {....},  
 ],  
 "pagination": {  
 "page": 1,  
 "limit": 2,  
 "totalPosts": 12,  
 "totalPages": 6  
 }  
 }  
}

## get post by id

* Endpoint

GET /api/v1/posts/{id}

* Response Body

{  
 "message" : "Post reterived successfully",  
 "status" : 200,  
 "data" : {  
 "id": "1",  
 "title": "My First Tech Blog",  
 "content": "This is the content of the blog post...",  
 "author": {  
 "id": "1",  
 "username": "anything",  
 },  
 }  
}

* Example

GET /api/v1/posts/1

## create post

* Endpoint

POST /api/v1/posts/

* Request Body

{  
 "title" : "new post title",  
 "content" "This is the new post",  
 "author" : 1 // user id as foregin key  
}

* Response Body

{  
 "message" : "Post created",  
 "status" : 201,  
 "data" : {  
 "id": "1",  
 "title": "new post title",  
 "content": "This is the new post",  
 "author": {  
 "id": "1",  
 "username": "anything",  
 },  
 },  
}

## update post

* Endpoint

PATCH /api/v1/posts/{id}

* Request Body

{  
 "title" : "Changed : new post title"  
}

* Response Body

{  
 "message" : "Post updated successfully",  
 "status" : 200,  
 "data" : {  
 "id" : 1,  
 "title" : "Changed : new post title",  
 "content": "This is the new post",  
 "author": {  
 "id": "1",  
 "username": "anything",  
 },   
 }   
}

## delete post

* Endpoint

DELETE /api/v1/posts{id}

* Response Body

{  
 "message" : "Post deleted",  
 "status" : 200  
}

# User Posts

## get logged in users post

* Endpoint

GET /api/v1/posts/me

* Response Body

{  
 "message" : "Post reterived",  
 "status" : 200,  
 data : [  
 {  
 "id" : 1,  
 "title" : "Changed : new post title",  
 "content": "This is the new post",  
 },  
 {......},  
 {......},  
 ]  
}

## Get Posts from a Specific User

* Endpoint

GET /api/v1/post/user/{id}

* Response Body

{  
 "message" : "Post reterived",  
 "status" : 200,  
 data : [  
 {  
 "id" : 1,  
 "title" : "Changed : new post title",  
 "content": "This is the new post",  
 },  
 {......},  
 {......},  
 ]  
}

# Comments

## add comments to a post

* Endpoint

POST /api/v1/posts/{id}/comments

* Request Body

{  
 "content" "This is the new post",  
 "user" : 1 // user id as foregin key  
}

* Response Body

{  
 "message" : "Comment added",  
 "status" : 201,  
 "data" : {  
 "id": "1",  
 "content": "This is the new post",  
 "author": {  
 "id": "1",  
 "username": "anything",  
 },  
 },  
}

## update post

* Endpoint

PATCH /api/v1/posts/{id}/comments

* Request Body

{  
 "content" : "Changed : new post title"  
}

* Response Body

{  
 "message" : "Post updated successfully",  
 "status" : 200,  
 "data" : {  
 "id" : 1,  
 "content": "Changed : new post title",  
 "author": {  
 "id": "1",  
 "username": "anything",  
 },   
 }   
}

## list posts

* Endpoint

GET /api/v1/posts/{id}/comments

* Response Body

{  
 "message": "comments fetched successfully",  
 "status": 200,  
 "data": {  
 "comments": [  
 {  
 "id": "1",  
 "content": "This is the content of the blog post...",  
 "author": {  
 "id": "1",  
 "username": "anything",  
 },  
 },  
 {....},  
 {....},  
 ],  
 }  
}

## delete comment

* Endpoint

DELETE /api/v1/posts{id}/comments

* Response Body

{  
 "message" : "comment deleted",  
 "status" : 200  
}

## Get all comments by a Current User

* Endpoint

GET /api/v1/posts/{id}/comments/me

* Response body

{  
 "message": "comments fetched successfully",  
 "status": 200,  
 "data": {  
 "comments": [  
 {  
 "id": "1",  
 "content": "This is the content of the blog post...",  
 },  
 {....},  
 {....},  
 ],  
 }  
}

**Note** : Authorization is done using token passed in header

# Likes

## Like a Post

* Endpoint

POST /api/v1/posts/{id}/like

* Resquest -> Header : Authentication token
* Response Body

{  
 "message": "Post liked successfully",  
 "status": 201,  
 "data": {  
 "postId": "1",  
 "likedBy": "anything",  
 }  
}

## Unlike a Post

* Endpoint

DELETE /api/v1/posts/{id}/like

* Resquest -> Header : Authentication token
* Response Body

{  
 "message": "Post unl11iked successfully",  
 "status": 200,  
 "data": {  
 "postId": "1",  
 "unlikedBy": "anything",  
 }  
}

## Get total likes for a Post

* Endpoint

GET /api/v1/posts/{id}/like/count

* Response Body

{  
 "message": "Likes count fetched successfully",  
 "status": 200,  
 "data": {  
 "postId": "1",  
 "totalLikes": 152  
 }  
}

# Has Current User Like a Post

* Endpoint

GET /api/v1/posts/{postId}/likes/me/status

* Request -> Header : Authentication token
* Response Body

{  
 "message": "Like status fetched successfully",  
 "status": 200,  
 "data": {  
 "postId": "1",  
 "liked": true  
 }  
}

## Posts liked by Current User

* Endpoint

GET /api/v1/posts/{postId}/likes/me

* Request -> Header : Authentication token
* Response Body

{  
 "message": "Liked posts fetched successfully",  
 "status": 200,  
 "data": [  
 {  
 "postId": "1",  
 "title": "10 Tips for Learning TypeScript",  
 "status" : true  
 },  
 {  
 "postId": "2",  
 "title": "How to Build a REST API with FastAPI",  
 "status" : true  
 }  
 ]  
}