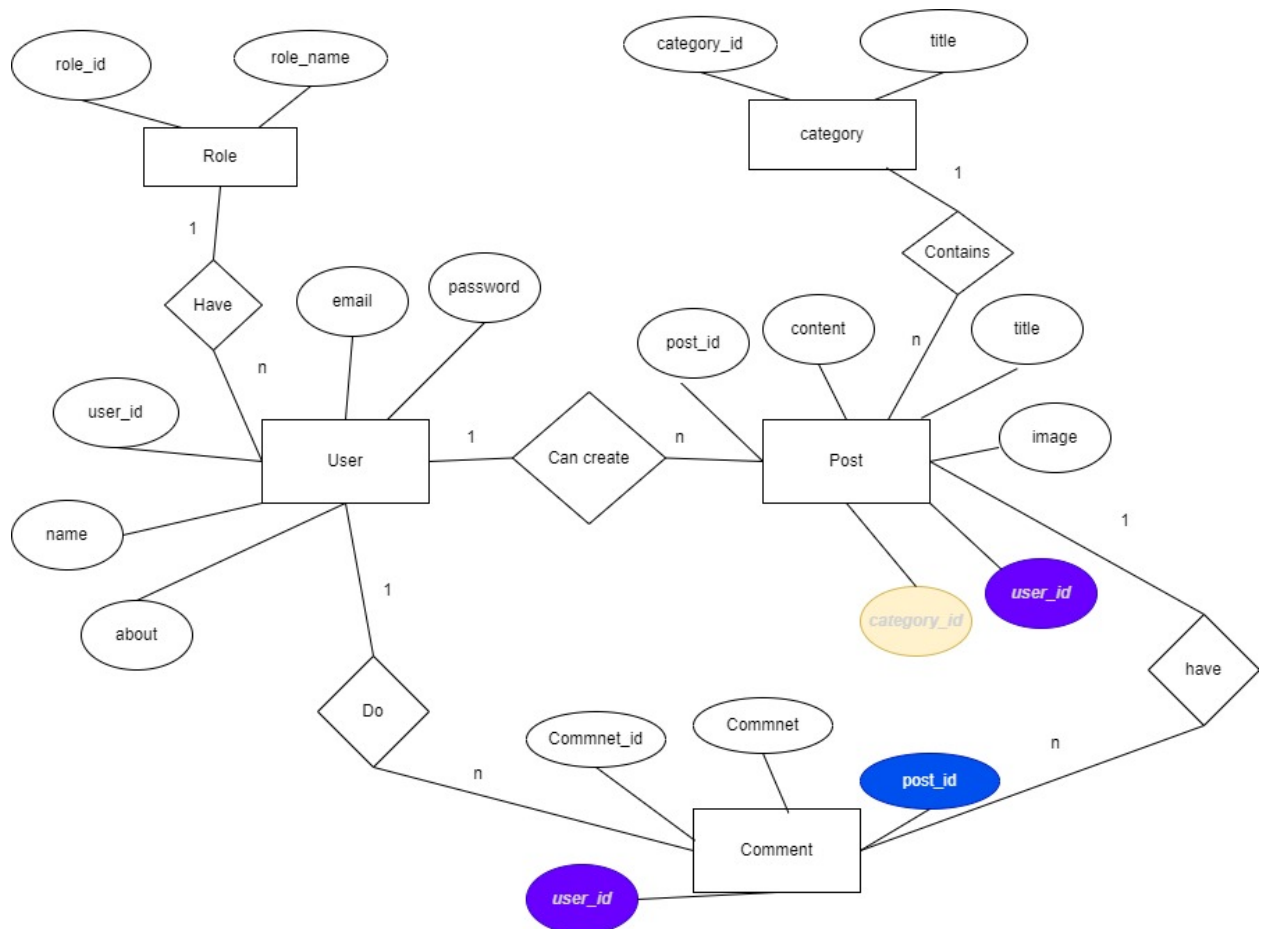


Backend Project of a Blogging Application

By- Shivam Kumar Jha

- Entity Relationship Diagram



Documentation of Apis of Blogging App

Payloads Strcuture

1. UserDto

```
1 {  
2   "name": "",  
3   "email": "",  
4   "password": "",  
5   "about": ""  
6 }
```

2. PostDto

```
{  
  "title": "",  
  "content": ""  
}
```

3. CategoryDto

```
{  
  "title": "",  
  "description": ""  
}
```

4. CommentDto

```
{  
  "content": ""  
}
```

5. JwtRequest

```
{  
  "username": "",  
  "password": ""  
}
```

6. PostResponse

```
List<PostDto> content;  
int pageNumber;  
int pageSize;  
int totalElements;  
int totalPages;  
boolean lastPage;
```

7. JwtResponse

```
String token;  
String username;
```

APIs for Different entities

Open Apis without any token requirements(Public URLs)

s.no	API type	URL	paramters	Body	Ouput
1	POST	http://localhost:9095/auth/login	None	UserDto	UserDto
2	POST	http://localhost:9095/auth/register	None	JWtRequest	JWtReponse

APIs with token requirements

PostController

s.no	API type	URL	paramters	Body	Ouput
1	POST	http://localhost:9095/api/user/{userId}/category/{categoryId}/posts	userId,categoryId	PostDto	PostDto
2	POST	http://localhost:9095/api/post/image/upload/{postId}	image,PostId	None	PostDto
3.	GET	http://localhost:9095/api/user/{userId}/posts	userId	None	List<PostDto>
4	GET	http://localhost:9095/api/category/{categoryId}/posts	CategoryId	None	List<PostDto>
5	GET	http://localhost:9095/api/posts	None	None	PostResponse
6	GET	http://localhost:9095/api/posts/{postId}	postId	None	PostDto
7	GET	http://localhost:9095/api/post/search/{keyword}	keyword	None	List<PostDto>
8	GET	http://localhost:9095/api/post/image/{imageName}	imageName	None	None
9	PUT	http://localhost:9095/api/posts/{postId}	postId	PostDto	PostDto
10	DELETE	http://localhost:9095/api/posts/{postId}	posId	None	ApiResponse

UserController

S.NO	API type	URL	Parameters	Body	Output
1	POST	http://localhost:9095/api/users/	None	UserDto	UserDto
2	PUT	http://localhost:9095/api/users/{userId}	userId	UserDto	UserDto
3	DELETE	http://localhost:9095/api/users/{userId}	userId	None	ApiResponse
4	GET	http://localhost:9095/api/users/{userId}	userId	None	UserDto
5	GET	http://localhost:9095/api/users/	None	None	List<userDto>

CommentController

S.NO	API type	URL	Parameters	Body	Output
1	POST	http://localhost:9095/api/post/{postId}/user/{userId}/comments	None	CommentDto	CommentDto
2	DELETE	http://localhost:9095/api/{commentId}	commentId	None	ApiResponse
3	GET	http://localhost:9095/api/user/{userId}/comments	userId	None	UserDto
4	GET	http://localhost:9095/api/post/{postId}/comments	postId	None	List<CommentDto>

CategoryController

S.NO	API type	URL	Parameters	Body	Output
1	POST	http://localhost:9095/api/categories/	None	CategoryDto	CategoryDto
2	PUT	http://localhost:9095/api/categories/{catId}	catId	CategoryDto	CategoryDto
3	DELETE	http://localhost:9095/api/categories/{catId}	catId	None	ApiResponse
4	GET	http://localhost:9095/api/categories/{catId}	catId	None	CategoryDto
5	GET	http://localhost:9095/api/categories/	None	None	List<CategoryDto>

Steps to use token

1. Use authlogin api with correct username and password
2. You will get a token as a response
3. Pass the token as header for every other Api call with Header = Bearer+” “ +token