# **Blog Project Dökümantasyonu**

**Proje Adı:** Blog Project

**Proje Açıklaması:** Blog Project, kullanıcıların blog yazıları oluşturmasına, görüntülemesine ve yönetmesine olanak tanıyan bir web uygulamasıdır.

## **Teknolojiler**

* Spring Boot
* Spring Data JPA
* PostgreSQL
* MapStruct
* Springdoc OpenAPI

## **Proje Yapısı**

### **Controller'lar**

#### 1. UserController

* **GET /users**: Tüm kullanıcıları getirir.
* **GET /users/{userId}**: Belirli bir kullanıcıyı getirir.
* **POST /users**: Yeni bir kullanıcı oluşturur.
* **PUT /users/{userId}**: Belirli bir kullanıcıyı günceller.
* **DELETE /users/{userId}**: Belirli bir kullanıcıyı siler.

#### 2. PostController

* **GET /posts**: Tüm blog yazılarını getirir.
* **GET /posts/{postId}**: Belirli bir blog yazısını getirir.
* **POST /posts**: Yeni bir blog yazısı oluşturur.
* **PUT /posts/{postId}**: Belirli bir blog yazısını günceller.
* **DELETE /posts/{postId}**: Belirli bir blog yazısını siler.
* **GET /posts/user/{userId}**: Belirli bir kullanıcının yazılarını getirir.
* **GET /posts/category/{categoryId}**: Belirli bir kategorinin yazılarını getirir.
* **GET /posts/posts**: Tüm yazıları yayın tarihine göre getirir.

#### 3. CategoryController

* **GET /categories**: Tüm kategorileri getirir.
* **GET /categories/{categoryId}**: Belirli bir kategoriyi getirir.
* **POST /categories**: Yeni bir kategori oluşturur.
* **PUT /categories/{categoryId}**: Belirli bir kategoriyi günceller.
* **DELETE /categories/{categoryId}**: Belirli bir kategoriyi siler.

### **Servisler**

#### 1. UserService

* **getAllUsers()**: Tüm kullanıcıları getirir.
* **getUserById(userId)**: Belirli bir kullanıcıyı getirir.
* **createUser(userDto)**: Yeni bir kullanıcı oluşturur.
* **updateUser(userId, userDto)**: Belirli bir kullanıcıyı günceller.
* **deleteUser(userId)**: Belirli bir kullanıcıyı siler.

#### 2. PostService

* **getAllPosts()**: Tüm blog yazılarını getirir.
* **getPostById(postId)**: Belirli bir blog yazısını getirir.
* **createPost(postDto)**: Yeni bir blog yazısı oluşturur.
* **updatePost(postId, postDto)**: Belirli bir blog yazısını günceller.
* **deletePost(postId)**: Belirli bir blog yazısını siler.
* **getPostsByUser(userId)**: Belirli bir kullanıcının yazılarını getirir.
* **getPostsByCategory(categoryId)**: Belirli bir kategorinin yazılarını getirir.
* **getAllPostsOrderByPublishedDate()**: Tüm yazıları yayın tarihine göre getirir.

#### 3. CategoryService

* **getAllCategories()**: Tüm kategorileri getirir.
* **getCategoryById(categoryId)**: Belirli bir kategoriyi getirir.
* **createCategory(categoryDto)**: Yeni bir kategori oluşturur.
* **updateCategory(categoryId, categoryDto)**: Belirli bir kategoriyi günceller.
* **deleteCategory(categoryId)**: Belirli bir kategoriyi siler.

## **DTO'lar**

### **1. UserDto**

* **id**
* **firstName**
* **lastName**
* **email**
* **password**
* **posts**

### **2. PostDto**

* **id**
* **title**
* **content**
* **publishedAt**
* **userId**
* **categoryId**

### **3. CategoryDto**

* **id**
* **name**
* **description**

## **Veritabanı Tabloları**

### **1. users**

* **id**
* **firstName**
* **lastName**
* **email**
* **password**

### **2. posts**

* **id**
* **title**
* **content**
* **publishedAt**
* **user\_id**
* **category\_id**

### **3. categories**

* **id**
* **name**
* **description**

## **MapStruct Mapper'ları**

* UserMapper
* PostMapper
* CategoryMapper

## **Global Exception Handling**

Proje genelindeki hatalar için GlobalExceptionHandler bulunmaktadır. Proje, uygun HTTP durum kodları ve hata mesajlarıyla hataları yönetir.