



FULL STACK UYGULAMA DOKÜMANTASYONU

Geliştirici: Beyazıt Bestami Sarıkaya

E-Posta: bestami980@outlook.com

Cep No: 0538 696 2189

Web Site: bbestamis.com/

LinkedIn: [linkedin.com/in/beyazit-bestami-sarikaya-b016b8173](https://www.linkedin.com/in/beyazit-bestami-sarikaya-b016b8173)

V 1.0

04.12.2024, Ankara



İÇİNDEKİLER

İÇİNDEKİLER.....	2
1. <i>GİRİŞ (INTRODUCTION)</i>	3
2. <i>BAŞLANGIÇ (GETTING STARTED)</i>	4
3. <i>PROJE YAPISI (PROJECT STRUCTURE)</i>	4



1. GİRİŞ (INTRODUCTION)

Back-End Uygulamasının Tanımı: Python Django tabanlı bir API projesinin geliştirilmesi, yapılandırılması, test edilmesi ve dağıtılması süreçlerini detaylı olarak açıklamaktadır. Proje, kullanıcılara yönelik CRUD (Create, Read, Update, Delete) işlemleri ve ilgili kaynaklara erişim sağlayan bir RESTful API hizmeti sunar. Bu proje, güçlü bir API anahtarı doğrulama mekanizması, kapsamlı test senaryoları ve modüler bir yapı sunarak ölçeklenebilir ve sürdürülebilir bir çözüm hedeflemiştir.

Proje Kapsamı

- Kullanıcı, Todo, Post, Album, Comment ve Photo veri modellerine dayalı CRUD operasyonları.
- API güvenliği için x-api-key başlığı ile doğrulama.
- Verilerin performansını artırmak için önbellekleme (caching) mekanizması.
- Swagger ile API dokümantasyonu.
- Otomatik birim testleriyle sürekli entegrasyon için yapılandırma.
- Uygulama ve bağımlılıkları Docker konteynerleri kullanılarak çalıştırılabilir hale getirilmiştir

Teknik Özellikler

- Dil:** Python
- Framework:** Django + Django REST Framework
- Veritabanı:** PostgreSQL
- Cache:** Redis
- Testler:** Django TestCase, APIClient
- API Dokümantasyonu:** Swagger (drf-yasg)
- Kapsayıcı:** Docker + Docker Compose

Front-End Uygulamasının Tanımı: Vue.js uygulaması, kullanıcıların çeşitli özelliklere sahip veri setlerini (kullanıcılar, görevler, gönderiler, albümler ve fotoğraflar) kolayca görüntüleyebileceği, düzenleyebileceği ve yönetebileceği bir platform sunmayı hedeflemektedir.

Proje Kapsamı

- Kullanıcıların listelenmesi ve kullanıcı detaylarının gösterilmesi.
- Kullanıcılara ait görevlerin (todos) yönetimi ve durumlarının değiştirilmesi.
- Kullanıcı gönderilerinin (posts) görüntülenmesi ve detaylı içeriklerin gösterimi.
- Albüm ve fotoğraf galerisi özellikleriyle, medya görüntülenmesini kolaylaştırılması.

Teknik Özellikler

- Vue.js:** Modern, bileşen tabanlı bir JavaScript framework'ü olarak uygulamanın temel yapısını oluşturmak için kullanılmıştır.
- Pinia:** Durum yönetimi (state management) için kullanılmıştır. Uygulama genelinde kullanıcı bilgisi, aktif sekme ve hata durumlarını yönetir.
- Axios:** API çağrılarını gerçekleştirmek için tercih edilen HTTP istemcisi.
- Tailwind CSS:** Hızlı ve modern bir CSS framework'ü ile kullanıcı arayüzü tasarlanmıştır.
- Vue Router:** Uygulama içinde sayfa geçişlerini sağlamak için kullanılmıştır.



•**Local Storage:** Verilerin kalıcılığını sağlamak için kullanılmıştır (örneğin, görevlerin tamamlanma durumları).

2. BAŞLANGIÇ (GETTING STARTED)

- **Ön Gereksinimler (Prerequisites):** tarayıcı çalışması için gerekli yazılımlar (Front-End: Node.js, npm veya yarn, Tailwind CSS, Git. & Back-End: Python, Django, Django REST Framework, PostgreSQL, Redis, Docker)

- **Kurulum (Installation):**

Python kütüphaneleri requirements.txt dosyasında listelenmiştir ve proje için gereklidir.

Öncelikle .env dosyası içerisinde bulunan değişkenlere atama işlemi yapılması gerekmektedir. Bunlar; “POSTGRES_USER”, “POSTGRES_PASSWORD”, “POSTGRES_DB”, ” X_API_KEY”

Dockerize edilmiş olan uygulama sırasıyla;

- “docker compose build”
- “docker compose up -d”
- “docker exec -it n2mobil-backend-web-1 python manage.py migrate”

komutlarıyla uygulama hazır hale getirilir.

- **Ortam Değişkenleri (Environment Variables):**

X_API_KEY: 4815b9c3-ef30-48cd-9f41-49058a178b2b

3. PROJE YAPISI (PROJECT STRUCTURE)

Aşağıda uygulamada yer alan tüm endpointler ve işlevleri verilmiştir. Endpointlerin tamamı, x-api-key başlığı ile API anahtarı doğrulaması gerektirir.

Base Endpoint: /users/

- **GET** /users/ : Tüm kullanıcıları getirir.
- **GET** /users/{id}/ : Belirtilen kullanıcıyı getirir.
- **DELETE** /users/{id}/ : Belirtilen kullanıcıyı siler.
- **GET** /users/{id}/todos/ : Kullanıcıya ait tüm todo öğelerini listeler.
- **GET** /users/{id}/posts/ : Kullanıcıya ait tüm post öğelerini listeler.
- **GET** /users/{id}/albums/ : Kullanıcıya ait tüm albümleri listeler.

Base Endpoint: /todos/

- **GET** /todos/ : Tüm todo öğelerini getirir.
- **GET** /todos/{id}/ : Belirtilen todo öğesini getirir.
- **POST** /todos/ : Yeni bir todo oluşturur.
- **DELETE** /todos/{id}/ : Belirtilen todo öğesini siler.

Base Endpoint: /posts/

- **GET** /posts/ : Tüm postları getirir.



- **GET** /posts/{id}/: Belirtilen postu getirir.
- **POST** /posts/: Yeni bir post oluşturur.
- **DELETE** /posts/{id}/: Belirtilen postu siler.
- **GET** /posts/{id}/comments/: Belirtilen posta ait yorumları listeler.

Base Endpoint: /albums/

- **GET** /albums/: Tüm albümleri getirir.
- **GET** /albums/{id}/: Belirtilen albümü getirir.
- **POST** /albums/: Yeni bir albüm oluşturur.
- **DELETE** /albums/{id}/: Yeni bir albüm oluşturur.
- **GET** /albums/{id}/photos/: Belirtilen albüme ait fotoğrafları listeler.

Base Endpoint: /comments/

- **GET** /comments/: Tüm yorumları getirir.
- **GET** /comments/{id}/: Belirtilen yorumu getirir.
- **POST** /comments/: Yeni bir yorum oluşturur.
- **DELETE** /comments/{id}/: Belirtilen yorumu siler.

Base Endpoint: /photos/

- **GET** /photos/: Tüm fotoğrafları getirir.
- **GET** /photos/{id}/: Belirtilen fotoğrafı getirir.
- **POST** /photos/: Yeni bir fotoğraf oluşturur.
- **DELETE** /photos/{id}/: Belirtilen fotoğrafı siler.

Tüm API dökümantasyonunu görmek için aşağıdaki URL kullanılabilir:

http://localhost:8000/swagger/

End-Point Testleri: Proje test altyapısı, her bir API endpoint'inin işlevselliğini ve doğruluğunu doğrulamak için kapsamlı bir şekilde tasarlanmıştır. unittest ve Django'nun TestCase sınıfları kullanarak yazılan testler, setUpTestData metodu ile önceden tanımlanan verilerle çalışır. Her bir endpoint için GET, POST, DELETE işlemleri ayrı ayrı test edilmiştir ve API anahtarı doğrulaması (x-api-key) her testte geçerli olacak şekilde ayarlanmıştır. Testler, kullanıcılar, görevler (todos), gönderiler (posts), albümler, yorumlar ve fotoğraflar gibi tüm veri modellerini kapsamaktadır. Tüm testler, komut satırından **python manage.py test** ile çalıştırılarak sistemin güvenilirliği ve entegrasyon doğruluğu kontrol edilebilir.

Bu projede **ModelViewSet**, kullanıcılar, görevler (todos), gönderiler (posts), albümler, yorumlar ve fotoğraflar gibi ana modeller için API endpoint'lerini hızlı ve etkili bir şekilde oluşturmak amacıyla kullanıldı. Her model için bir ViewSet sınıfı tanımlandı ve bu sınıflar, Django REST Framework'ün sağladığı CRUD işlemlerini (Create, Read, Update, Delete) otomatikleştirdi. Projede, her modelin özelleştirilmiş davranışını belirlemek için list, retrieve, create, ve destroy gibi metodlar gerektiğinde override edildi. Ayrıca @action dekoratörü kullanarak, kullanıcıya özgü görevler, gönderiler, albümler gibi ilişkili verilere erişimi sağlayan özel endpoint'ler eklendi. API key tabanlı bir izin yapısı (APIKeyPermission) ve verimlilik için cache mekanizması entegre edildi. ModelViewSet kullanımı, her model için tekrar eden kod yazımını önleyerek API geliştirme sürecini hızlandırdı ve sürdürülebilir bir yapı sağladı.

Projede, başlangıçta fotoğraflar için **JSONPlaceholder API** kullanılacaktı, ancak bu API'nin sık sık **503 Service Unavailable** hatası vermesi nedeniyle fotoğraf verilerinin alınmasında sorunlar yaşanmıştır. Bu durumu çözmek için, projeye daha güvenilir ve stabil bir alternatif olan **Pexels API** entegre edilmiştir. Bu değişiklik ile fotoğraf yönetimi sorunsuz hale getirilmiştir.

Projede, kullanıcı deneyimini geliştirmek ve geniş bir cihaz yelpazesinde erişilebilirliği sağlamak için **Tailwind CSS** kullanılarak duyarlı (responsive) bir tasarım oluşturulmuştur. Bu sayede uygulama, masaüstü, tablet ve mobil cihazlarda sorunsuz bir şekilde çalışmaktadır.



Front-End uygulamasının public linki: <http://n2mobil.bbestamis.com/>
Projenin bulunduğu depo linki: <https://github.com/BBestamiS/n2mobil>