API IMSAKIYAH 1444 H

● Sekilas Tentang Api Imsakiyah

REST API Imsakiyah Ramadhan adalah antarmuka pemrograman aplikasi (API) yang menyediakan data Tanggal, waktu Imsak, Subuh, Terbit, Duha, Dzuhur, Ashar, Maghrib, dan Isya' untuk setiap hari selama bulan suci Ramadhan. Data waktu Imsak sangat penting bagi umat Muslim yang sedang menjalankan ibadah puasa selama bulan Ramadhan. Rest API Imsakiyah Ramadhan umumnya menyediakan informasi dalam format JSON yang dapat diakses melalui HTTP request. Pengguna dapat mengirimkan permintaan HTTP ke API dan menerima data waktu Imsakiyah Ramadhan dalam format JSON sebagai respons.

Informasi yang biasanya tersedia dalam Rest API Imsakiyah Ramadhan meliputi Tanggal, waktu Imsak, Subuh, Terbit, Duha, waktu Dzuhur, waktu Ashar, waktu Maghrib, dan waktu Isya' untuk setiap hari selama bulan Ramadhan. Rest API Imsakiyah Ramadhan dapat digunakan oleh pengembang aplikasi untuk membuat aplikasi atau layanan yang membantu umat Muslim untuk memantau waktu Imsakiyah Ramadhan dengan mudah.

Sumber Data yang digunakan sebagai acuan yakni berasal dari lembaga falakiyah dan astronomi (L-FAS). untuk data pada API ini menggunakan satu kabupaten yakni Lombok Timur.

Link :

https://drive.google.com/drive/folders/1N-FslH9BL8HkFBm3zlDRuEr6X7j0oFXQ?usp=sharing

● Kode Program

Pada pembuatan Rest Api ini menggunakan bahasa pemrograman Python dengan Framework Django, adapun framewok django mengusung konsep MVT (Model View Template).

1. Model

Adalah sebuah kelas Python yang merepresentasikan sebuah tabel dalam database relasional. Dalam model, kita menentukan atribut-atribut dari sebuah tabel seperti nama kolom, tipe data dan lain sebagainya.

Pada API imsyak ini kita membuat tabel pada file models.py dengan nama Jadwal yang memiliki kolom/atribut tanggal, imsak, subuh, terbit, dhua, zuhur, asar, magrib dan isya. untuk tanggal kita menggunakan tipe data CharField (str) agar kita bisa menginput data dengan format 1 Ramadhan 1444 h dan untuk kolom lainya kita menggunakan tipe data TimeField yang dimana formatnya sudah ditentukan yaitu berupa jam:menit:detik

2. Admin

Digunakan untuk mengatur antarmuka administratif (admin) dari sebuah aplikasi Django. Dalam file admin.py, kita mendefinisikan kelas-kelas admin yang merepresentasikan model-model yang ingin kita tampilkan di admin interface. Setiap kelas admin ini mewarisi kelas admin.ModelAdmin, dan kita dapat menentukan beberapa opsi untuk mengatur perilaku admin interface.

Pada kode tersebut, pertama-tama kita mengimpor modul admin dari package django.contrib dan model Jadwal dari file models.py di dalam aplikasi yang sama. Setelah itu, kita mendefinisikan sebuah kelas jadwalModelAdmin yang mewarisi kelas admin.ModelAdmin. Di dalam kelas ini, kita mengatur beberapa opsi yang akan digunakan dalam tampilan admin dari model Jadwal.

• readonly_fields digunakan untuk menentukan field-field yang hanya dapat dibaca pada tampilan admin. Di sini, kita memilih field created dan updated agar tidak bisa diubah oleh admin.

• list_display digunakan untuk menentukan kolom-kolom yang akan ditampilkan pada tampilan daftar data. Di sini, kita menampilkan kolom tanggal, waktu imsak, subuh, terbit, duha, zuhur, asar, magrib, isya, serta kolom created dan updated.

• list_filter digunakan untuk menentukan filter untuk menampilkan data berdasarkan kolom tertentu. Di sini, kita menentukan filter berdasarkan kolom tanggal.

Terakhir, kita mendaftarkan model Jadwal ke dalam tampilan

admin dengan mengggunakan admin.site.register(), dan mendaftarkan kelas jadwalModelAdmin sebagai opsi untuk model tersebut. Dengan menggunakan admin.ModelAdmin, kita dapat mengatur tampilan dan perilaku admin interface yang digunakan untuk mengakses dan memanipulasi data pada model Jadwal.

3. Serializers

Adalah sebuah kelas yang digunakan untuk mengkonversi model atau instance dari model menjadi bentuk-bentuk yang dapat dikirimkan dan diterima melalui HTTP, seperti JSON atau XML. Serializers memudahkan pengiriman data antara aplikasi web dan klien seperti aplikasi mobile atau aplikasi desktop.

Pada kode tersebut, kita mengimpor modul serializers dari package rest_framework. Modul ini menyediakan kelas-kelas yang digunakan untuk membuat serializer pada Django REST Framework. Kemudian, kita mendefinisikan sebuah kelas ImsyakSerializer yang mewarisi kelas serializers.ModelSerializer. Di dalam kelas ini, kita mendefinisikan sebuah nested kelas Meta, yang digunakan untuk mengkonfigurasi serializer. Dalam nested kelas Meta, kita menentukan model yang akan diserialisasi dengan memberikan nilai Jadwal pada atribut model. Kita juga menentukan field-field dari model yang akan disertakan dalam serializer dengan memberikan nilai '__all__' pada atribut fields. Nilai '__all__' ini menandakan bahwa semua field/kolom pada model akan disertakan dalam serializer. Dengan membuat sebuah serializer seperti ImsyakSerializer.

4. Views

Sebuah kelas atau fungsi yang berfungsi sebagai penghubung antara model dan template. Views digunakan untuk memproses request yang diterima dari klien, melakukan query pada database menggunakan model, melakukan operasi pada data, dan merender template yang akan ditampilkan pada klien.

Pertama, terdapat fungsi jadwal_detail yang menerima request

dengan method GET, PUT, atau DELETE pada endpoint /jadwal/<id>, dimana id adalah parameter yang dikirimkan. Fungsi ini akan mencari Jadwal dengan id yang sesuai, jika ditemukan akan melakukan operasi sesuai dengan method request yang diberikan.

• Jika metode request adalah GET, maka data Jadwal yang sudah diambil pada tahap sebelumnya akan diserialisasi dan dikembalikan sebagai respons.

• Jika metode request adalah PUT, maka data Jadwal yang sudah diambil pada tahap sebelumnya akan diperbarui dengan data yang diberikan dalam permintaan. Data baru tersebut kemudian diserialisasi dan dikembalikan sebagai respons jika data sudah valid, atau dikembalikan pesan kesalahan dan status error 400 jika data tidak valid.

• Terakhir, jika metode request adalah DELETE, maka data Jadwal yang sudah diambil pada tahap sebelumnya akan dihapus dari database, dan fungsi akan mengembalikan respons dengan status 204 (NO CONTENT).

Jika Jadwal tidak ditemukan, akan dikirimkan response error 404.

Kedua, terdapat fungsi jadwal_list yang menerima request dengan method GET atau POST pada endpoint /jadwal. Fungsi ini akan mengambil semua data Jadwal yang ada di database dan mengembalikannya dalam bentuk serialized jika request method adalah GET. Sedangkan jika request method adalah POST, fungsi ini akan membuat data Jadwal baru dengan data yang diberikan dalam request. Jika data yang diberikan valid, fungsi akan menyimpan data Jadwal baru tersebut dan mengembalikannya dalam bentuk serialized. Jika data tidak valid, akan dikirimkan response error 400.

5. Url

Berfungsi untuk menangani permintaan HTTP dan menentukan bagaimana aplikasi merespons permintaan tersebut. File urls.py akan memetakan setiap URL yang diakses ke view yang tepat. Dalam file

urls.py, kita dapat menentukan URL mana yang akan dipetakan ke view yang tepat. Kemudian, kita dapat menentukan view mana yang akan menangani permintaan tersebut. View yang digunakan dapat berupa function-based view atau class-based view.

Pada kode dibawah ini, kita mengimport path dari modul django.urls untuk membuat daftar URL (URL patterns) yang terhubung dengan fungsi (views) pada aplikasi web Django.

• path('jadwal/<int:id>/',views.jadwal_detail): URL pattern ini akan menangani permintaan GET, PUT, dan DELETE untuk jadwal dengan ID tertentu. Parameter id digunakan untuk menentukan id jadwal yang dimaksud.

• path('jadwals/',views.jadwal_list): URL pattern ini akan menangani permintaan GET dan POST untuk seluruh jadwal.

Selanjutnya, terdapat penggunaan fungsi include() yang digunakan untuk mengimpor dan memasukkan URL patterns dari sebuah aplikasi Django bernama 'apiApp'. URL patterns dari aplikasi tersebut akan ditambahkan sebagai sub-URLs dari root URL yang didefinisikan pada baris ini. Dengan menggunakan baris kode tersebut, setiap permintaan yang masuk ke URL utama dari proyek Django akan diteruskan ke 'apiApp.urls' untuk dicocokkan dengan URL patterns yang telah didefinisikan di dalamnya.

6. Respon

untuk melihat semua data : http://127.0.0.1:8000/jadwals/

untuk membuat/input data baru : http://127.0.0.1:8000/jadwals/

untuk melihat detail/satu data : http://127.0.0.1:8000/jadwal/<id>

untuk melihat update data : http://127.0.0.1:8000/jadwal/<id>

untuk melihat delet data : http://127.0.0.1:8000/jadwal/<id>