Setting.py

Setelah menginstall django dan djangorestframework serta membuat project dan aplikasinya kita dapat mendaftarkan app pada INSTALLED\_APPS di file setting.py

Models.py

Pada definisi model tersebut, kita mengimport modul `models` dari Django. Kemudian, kita mendefinisikan sebuah class `jadwalModels` yang merupakan subclass dari `models.Model`. Class ini akan mewarisi semua fitur yang disediakan oleh `models.Model`.

Pada definisi class tersebut, kita mendefinisikan beberapa atribut. Atribut pertama adalah `Tanggal`, yang merupakan sebuah field dengan tipe `CharField`. Field ini akan menyimpan tanggal dalam bentuk string dengan panjang maksimum 50 karakter.

Kemudian, kita mendefinisikan beberapa field dengan tipe `TimeField`, yaitu `Imsyak`, `Subuh`, `Terbit`, `Duha`, `Zuhur`, `Asar`, `Magrib`, dan `Isya`. Field-field ini akan menyimpan waktu dalam bentuk `datetime.time`.

Pada akhir definisi class, kita mendefinisikan sebuah method `\_\_str\_\_` yang akan mengembalikan representasi string dari objek model. Dalam contoh di atas, method ini akan mengembalikan nilai dari field `Tanggal`.

Dalam code ini, `jadwalModels` adalah sebuah class model yang merepresentasikan sebuah tabel pada database.

Serializers.py

Pada definisi `jadwalSerializer`, kita mengimport model `jadwalModels` yang didefinisikan pada file `models.py` dan kemudian kita mendefinisikan `jadwalSerializer` sebagai subclass dari `serializers.ModelSerializer`.

Kemudian, kita mendefinisikan sebuah class `Meta` yang akan menyediakan informasi tentang model yang akan diserialisasi. Pada contoh ini, kita menggunakan model `jadwalModels` sebagai model yang akan diserialisasi.

Pada atribut `fields`, kita menyatakan field-field dari model `jadwalModels` yang akan disertakan dalam serialisasi. Pada contoh di atas, kita memilih semua field yang ada pada model `jadwalModels`.

Dengan menggunakan serializer ini, kita dapat melakukan konversi data dari model `jadwalModels` menjadi format JSON atau XML dan sebaliknya. Hal ini memudahkan kita dalam melakukan operasi CRUD pada API yang kita bangun menggunakan Django REST framework.

Views.py

Pada definisi `jadwalViewset`, kita mengimport model `jadwalModels` yang didefinisikan pada file `models.py` dan serializer `jadwalSerializer` yang didefinisikan pada file `serializers.py`.

Kemudian, kita mendefinisikan `jadwalViewset` sebagai subclass dari `viewsets.ModelViewSet`. `ModelViewSet` adalah class dari Django REST framework yang menyediakan implementasi standar untuk tampilan REST API. Dengan menggunakan `ModelViewSet`, kita dapat secara otomatis menangani semua operasi CRUD (Create, Retrieve, Update, Delete) pada model `jadwalModels`.

Pada atribut `queryset`, kita menentukan semua data yang akan diambil dari database. Dalam hal ini, kita menggunakan `jadwalModels.objects.all()` untuk mengambil semua data dari model `jadwalModels`.

Pada atribut `serializer\_class`, kita menentukan serializer yang akan digunakan untuk mengubah data dari format model menjadi format JSON atau XML. Dalam hal ini, kita menggunakan `jadwalSerializer`.

Dengan menggunakan `jadwalViewset`, kita dapat membuat API endpoint yang mendukung semua operasi CRUD pada model `jadwalModels` dengan sangat mudah dan efisien.

Urls.py

Pertama-tama, kita mengimport `admin` dan `path` dari Django, serta `include` untuk menghubungkan ke URL patterns pada aplikasi lain. Kemudian, kita mengimport `jadwalViewset` dari `views.py` dan `routers` dari Django REST framework.

Selanjutnya, kita membuat instance dari `DefaultRouter()` dengan nama `router`, yang akan menghasilkan URL patterns secara otomatis untuk operasi CRUD pada model `jadwalModels` yang telah diatur pada `jadwalViewset`.

Kemudian, kita mendaftarkan URL patterns yang dibuat oleh `router` menggunakan `include()`. Dalam hal ini, kita menambahkan prefix URL kosong, sehingga endpoint API kita akan dapat diakses pada http://localhost:8000/. Selain itu, kita juga mendaftarkan URL admin dengan menggunakan `path()` dan `admin.site.urls`.

Dengan demikian, aplikasi kita sekarang memiliki endpoint API yang mendukung semua operasi CRUD pada model `jadwalModels`, dengan menggunakan `jadwalViewset` sebagai tampilan utama, dan dapat diakses pada http://localhost:8000/jadwalLombokTimur.