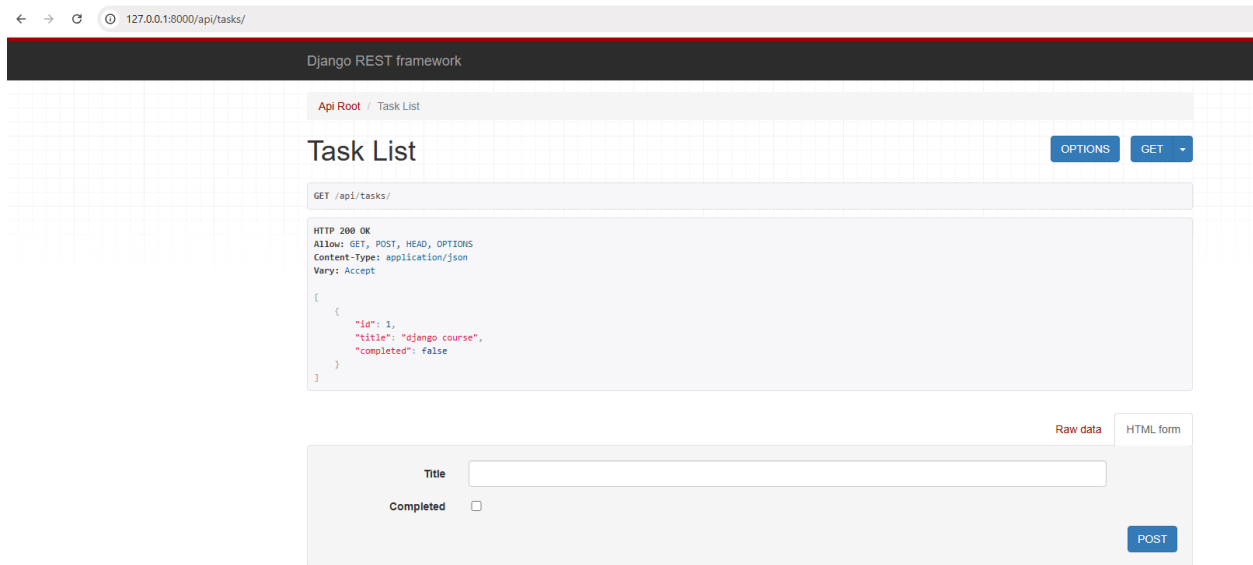


Django REST API Crud Example:-



Step 1: Set Up Django Project

Create virtual environment (optional)

```
python -m venv env
```

```
env\Scripts\activate
```

Start a new project

```
django-admin startproject todo_project
```

```
cd todo_project
```

Install Django and DRF

```
pip install django djangorestframework
```

Start an app

```
python manage.py startapp todo
```

as shown below example are :-

Command Prompt - python manage.py runserver

```
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Big Data>d:

D:\>env\scripts\activate

(env) D:\>django-admin startproject todo_project

(env) D:\>cd todo_project

(env) D:\todo_project>python manage.py startapp todo
```

Step 2: Register App & DRF in Settings.py

```
# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'rest_framework',
    'todo',
]

]

```

todo/models.py (todo app folder):-

```
from django.db import models
class Task(models.Model):
    title = models.CharField(max_length=255)
    completed = models.BooleanField(default=False)
    def __str__(self):
        return self.title
```

Then run migrations:

```
python manage.py makemigrations
```

```
python manage.py migrate
```

as shown example below :-

```
(env) D:\todo_project>python manage.py makemigrations
Migrations for 'todo':
  todo\migrations\0001_initial.py
    + Create model Task

(env) D:\todo_project>python manage.py migrate
Operations to perform:
```

Create Serializer In todo/serializers.py(todo app folder):

```
from rest_framework import serializers
from .models import Task
class TaskSerializer(serializers.ModelSerializer):
    class Meta:
        model = Task
        fields = '__all__'
```

todo/urls.py(todo app folder):-

```
from django.urls import path, include
from rest_framework.routers import DefaultRouter
from .views import TaskViewSet
router = DefaultRouter()
router.register(r'tasks', TaskViewSet)
urlpatterns = [
    path('', include(router.urls)),
]
```

todo/views.py (todo app folder):-

```
from rest_framework import viewsets
from .models import Task
from .serializers import TaskSerializer
class TaskViewSet(viewsets.ModelViewSet):
    queryset = Task.objects.all()
    serializer_class = TaskSerializer
```

In todo_project/urls.py (todo_project folder):

```
from django.contrib import admin
from django.urls import path, include
urlpatterns = [
    path('admin/', admin.site.urls),
    path('api/', include('todo.urls')),
]
```

Finally run:-

```
python manage.py runserver
```

for get and Post option:-

<http://127.0.0.1:8000/api/tasks/>

for Put & delete , GET option :-

The screenshot shows the Django REST framework interface for a Task Instance. The browser address bar displays `127.0.0.1:8000/api/tasks/1/`. The page title is "Task Instance" and the breadcrumb is "Api Root / Task List / Task Instance". There are three buttons: "DELETE" (red), "OPTIONS" (blue), and "GET" (blue with a dropdown arrow). Below the title, the URL `GET /api/tasks/1/` is shown. The response details are: `HTTP 200 OK`, `Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS`, `Content-Type: application/json`, and `Vary: Accept`. The JSON response is: `{ "id": 1, "title": "django course", "completed": false }`. At the bottom, there is a form with a "Title" field containing "django course" and a "Completed" checkbox. A "PUT" button is located at the bottom right of the form. There are also "Raw data" and "HTML form" tabs.

You can now:

- GET `/api/tasks/` – List all tasks
- POST `/api/tasks/` – Create a new task
- GET `/api/tasks/<id>/` – Retrieve a task
- PUT `/api/tasks/<id>/` – Update a task

- DELETE /api/tasks/<id>/ – Delete a task