Django and rest framework

Create a simple project with Django and rest framework. Add an app. Create a model named Blog. Blog should have title(CharField), description(Text), created_date(DateTime), modified_date(DateTime).

create APIs to handle

- Blog lists. Only list id, title and created_date field. Date should be formatted to user-friendly string

- A detail view of the blog. Show all fields.

- A blog create API

- An edit API

- A blog delete API

Create separate APIs for all purposes.
URLs and view names should be meaningful.

PROJECT NAME : BlogApi **APP NAME : Blogapp**

models.py

```
class Blog(models.Model):
    title = models.CharField(max_length=255)
    description = models.TextField()
    created_date = models.DateTimeField(auto_now_add=True)
    modified_date = models.DateTimeField(auto_now=True)
```

serializers.py

```
from .models import Blog
from rest_framework.serializers import ModelSerializer, DateTimeField


class BlogCreate(ModelSerializer):
    created_date = DateTimeField(format="%B %d, %Y, %I:%M %p", read_only=True)
    modified_date = DateTimeField(format="%B %d, %Y, %I:%M %p", read_only = True)

    class Meta:
        model = Blog
        fields = '__all__'


class BlogList(ModelSerializer):
    created_date = DateTimeField(format="%B %d, %Y, %I:%M %p")

    class Meta:
        model = Blog
        fields = ['id', 'title', 'created_date']
```

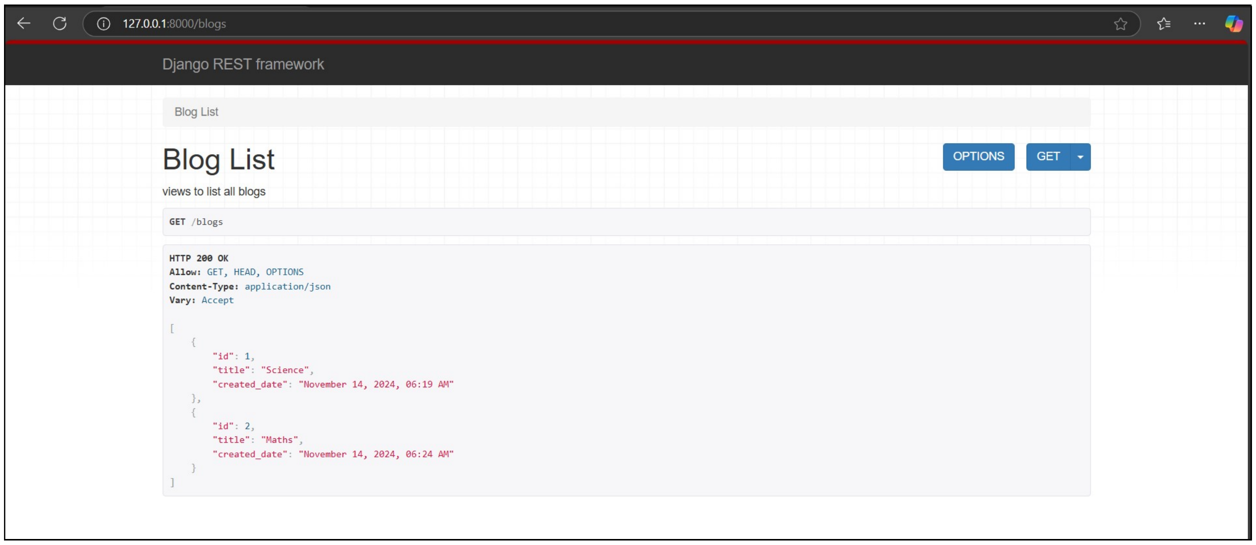
views.py

```python
from django.shortcuts import render

# Create your views here.

from rest_framework.generics import ListAPIView, RetrieveAPIView, CreateAPIView,
RetrieveUpdateAPIView, DestroyAPIView
from .models import Blog
from .serializers import BlogList, BlogCreate
from rest_framework.response import Response


class BlogCreateView(CreateAPIView):
    """views to create blogs"""
    queryset = Blog.objects.all()
    serializer_class = BlogCreate

    def create(self, request, *args, **kwargs):
        response=super().create(request, *args, **kwargs)
        return Response({"message": "Blog created successfully","data": response.data})


class BlogListView(ListAPIView):
    """views to list all blogs"""
    queryset = Blog.objects.all()
    serializer_class = BlogList


class BlogDetailView(RetrieveAPIView):
    """views to detail blogs"""
    queryset = Blog.objects.all()
    serializer_class = BlogCreate


class BlogUpdateView(RetrieveUpdateAPIView):
    """views to update blogs"""
    queryset = Blog.objects.all()
    serializer_class = BlogCreate

    def update(self, request, *args, **kwargs):
        response=super().update(request, *args, **kwargs)
        return Response({"message": "Blog updated successfully","data": response.data})


class BlogDeleteView(DestroyAPIView):
    """views to delete blogs"""
    queryset = Blog.objects.all()
    serializer_class = BlogCreate

    def destroy(self, request, *args, **kwargs):
        super().destroy(request, *args, **kwargs)
        return Response({"message": "Blog deleted successfully"})
```

urls.py

```python
from django.urls import path
from .views import  BlogCreateView, BlogListView, BlogDetailView, BlogUpdateView, BlogDeleteView

urlpatterns = [
    path('blogs/create', BlogCreateView.as_view(), name='create_blog'),
    path('blogs', BlogListView.as_view(), name='list_blog'),
    path('blogs/<int:pk>', BlogDetailView.as_view(), name='detail_blog'),
    path('blogs/<int:pk>/edit', BlogUpdateView.as_view(), name='edit_blog'),
    path('blogs/<int:pk>/delete', BlogDeleteView.as_view(), name='delete_blog'),
]
```

Blog list page : /blogs

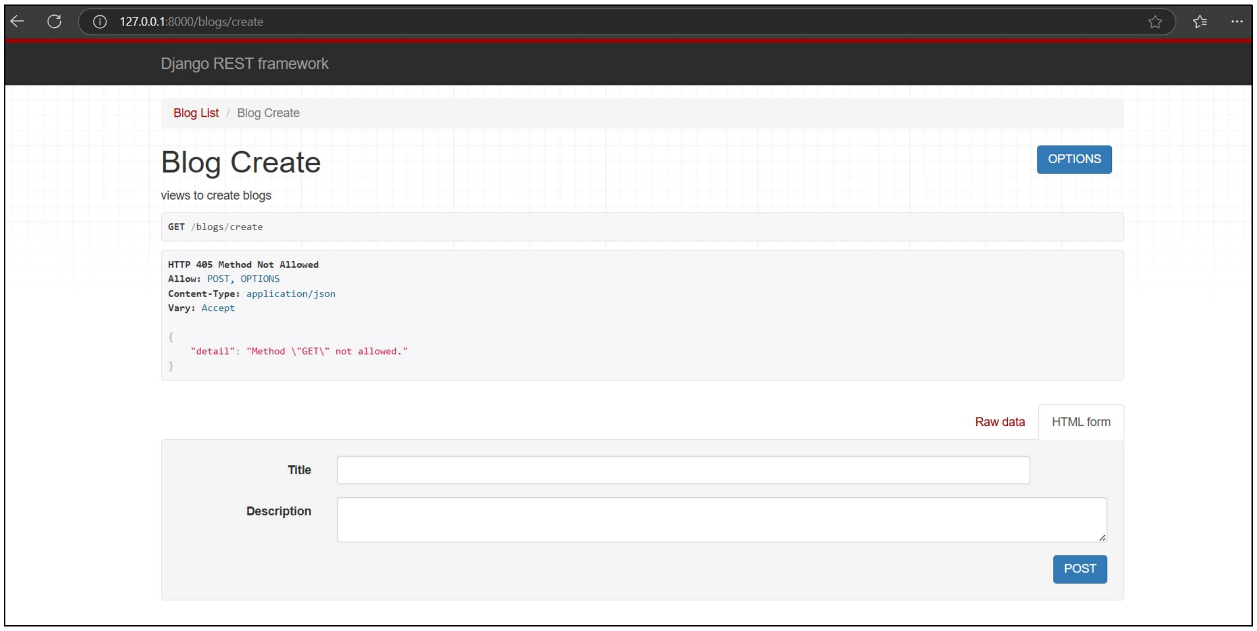


Blog List

views to list all blogs

```
GET /blogs

HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 1,
        "title": "Science",
        "created_date": "November 14, 2024, 06:19 AM"
    },
    {
        "id": 2,
        "title": "Maths",
        "created_date": "November 14, 2024, 06:24 AM"
    }
]
```

Blog create page : /blogs/create

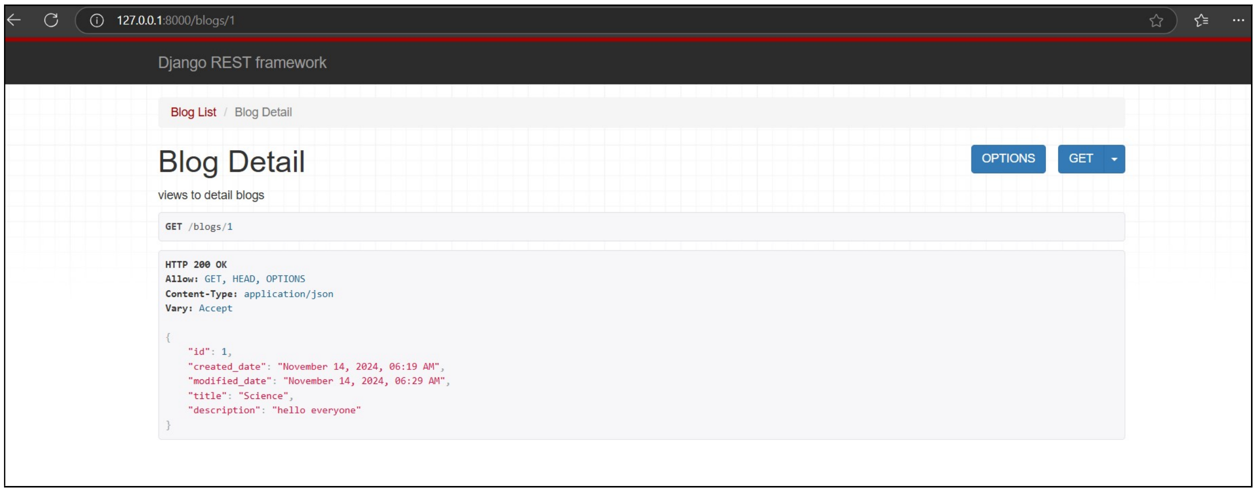


Blog Create

views to create blogs

```
GET /blogs/create

HTTP 405 Method Not Allowed
Allow: POST, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "detail": "Method \"GET\" not allowed."
}
```

Blog detail page : blogs/<int:pk> , <int:pk> = ID of the blog



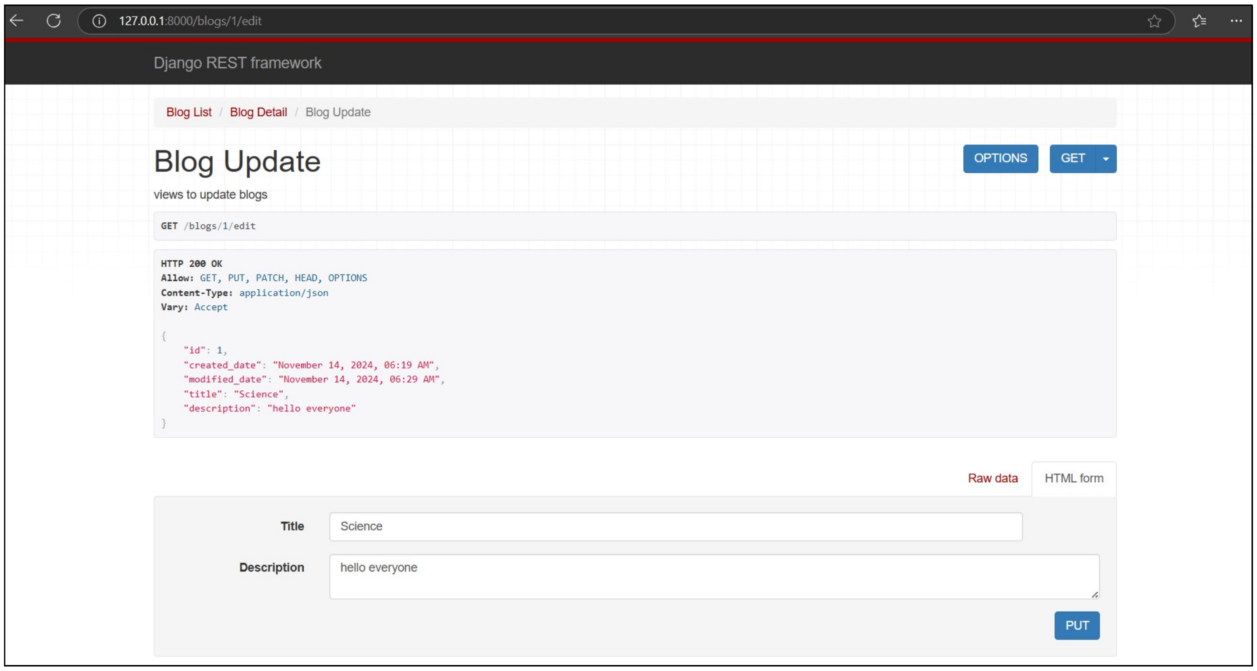
Blog Detail

views to detail blogs

```
GET /blogs/1
```

```
HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 1,
    "created_date": "November 14, 2024, 06:19 AM",
    "modified_date": "November 14, 2024, 06:29 AM",
    "title": "Science",
    "description": "hello everyone"
}
```

Blog edit page : blogs/<int:pk>/edit , <int:pk> = ID of the blog


127.0.0.1:8000/blogs/1/edit
Django REST framework
Blog List / Blog Detail / Blog Update
OPTIONS
GET
Raw data
HTML form
PUT

Blog Update

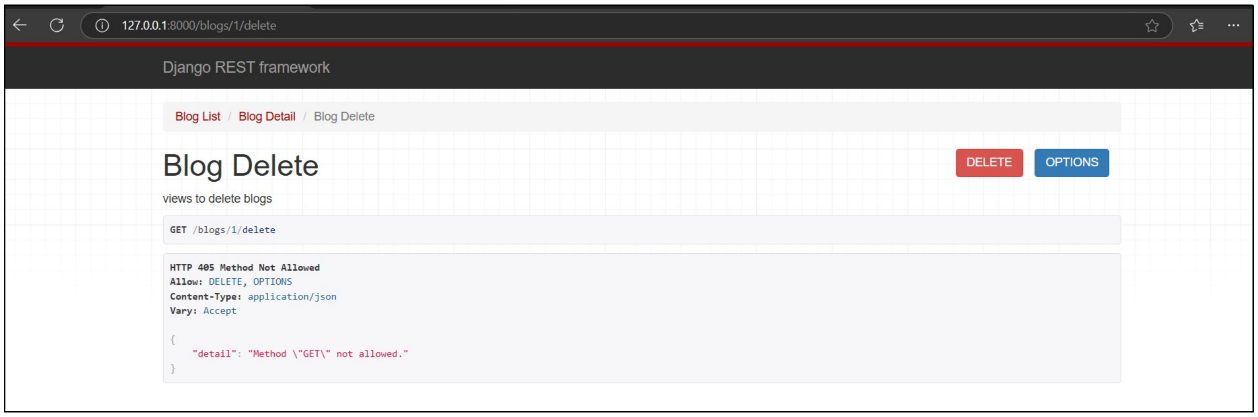
views to update blogs

```
GET /blogs/1/edit
```

```
HTTP 200 OK
Allow: GET, PUT, PATCH, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 1,
    "created_date": "November 14, 2024, 06:19 AM",
    "modified_date": "November 14, 2024, 06:29 AM",
    "title": "Science",
    "description": "hello everyone"
}
```

Title	Science
Description	hello everyone

Blog delete page : blogs/<int:pk>/delete , <int:pk> = ID of the blog


127.0.0.1:8000/blogs/1/delete
Django REST framework
Blog List / Blog Detail / Blog Delete
DELETE
OPTIONS

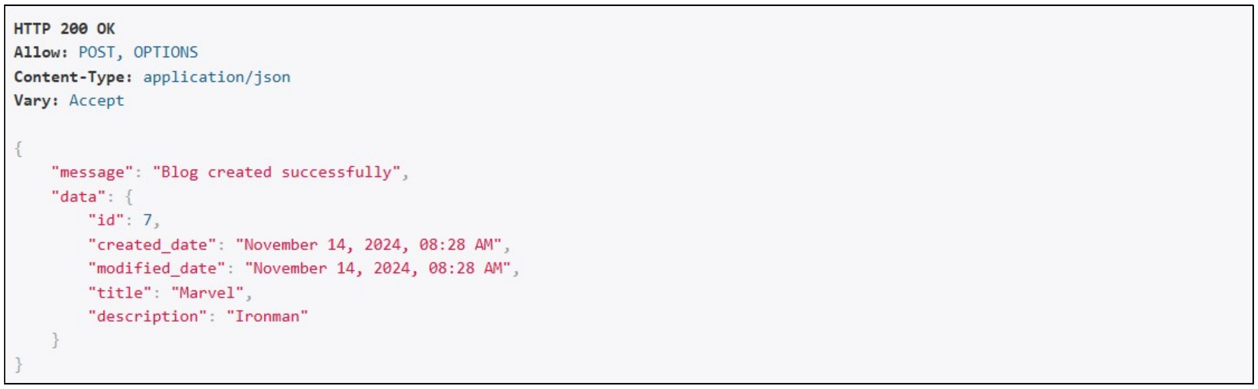
Blog Delete

views to delete blogs

```
GET /blogs/1/delete
```

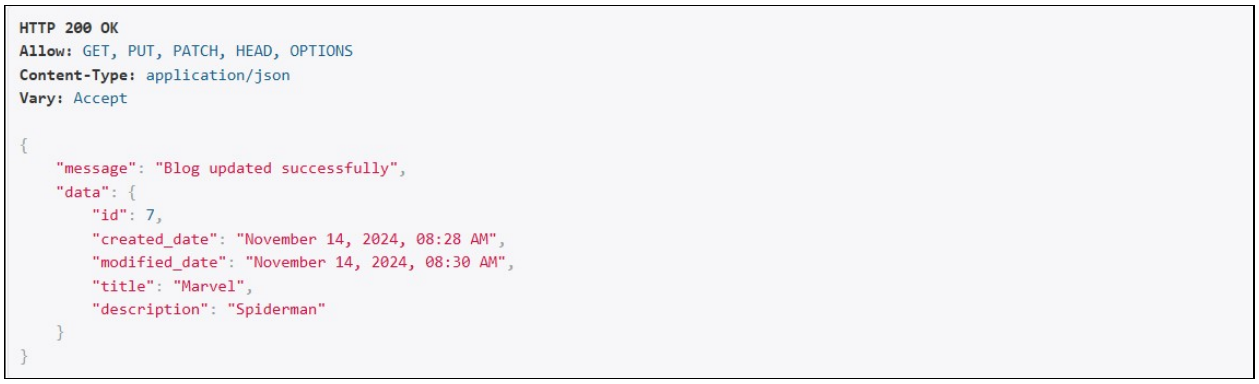
```
HTTP 405 Method Not Allowed
Allow: DELETE, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "detail": "Method \"GET\" not allowed."
}
```

Create response



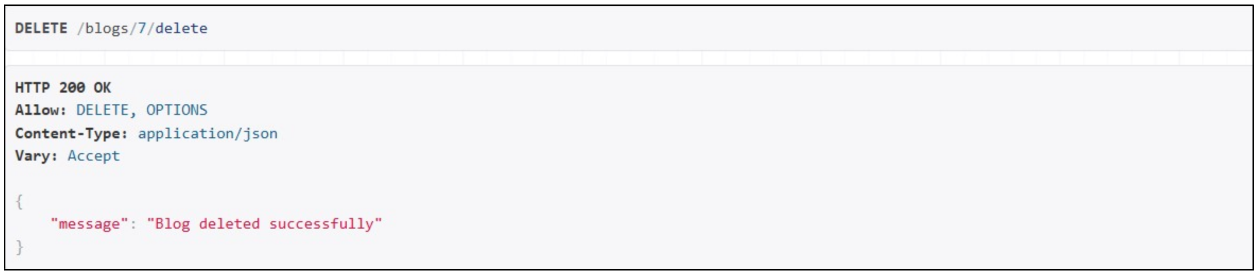
```
HTTP 200 OK
Allow: POST, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "message": "Blog created successfully",
    "data": {
        "id": 7,
        "created_date": "November 14, 2024, 08:28 AM",
        "modified_date": "November 14, 2024, 08:28 AM",
        "title": "Marvel",
        "description": "Ironman"
    }
}
```

Edit response



```
HTTP 200 OK
Allow: GET, PUT, PATCH, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "message": "Blog updated successfully",
    "data": {
        "id": 7,
        "created_date": "November 14, 2024, 08:28 AM",
        "modified_date": "November 14, 2024, 08:30 AM",
        "title": "Marvel",
        "description": "Spiderman"
    }
}
```

Delete response



```
DELETE /blogs/7/delete

HTTP 200 OK
Allow: DELETE, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "message": "Blog deleted successfully"
}
```