Django API Setup with MySQL and CORS

Step 5: Create a Model

```
# myapi/models.py
from django.db import models

class Products(models.Model):
    name = models.CharField(max_length=100)
    price = models.DecimalField(max_digits=10, decimal_places=2)
    created_at = models.DateTimeField(auto_now_add=True)
```

Step 6: Make Migrations & Migrate

```
python manage.py makemigrations
python manage.py migrate
```

Step 7: Create a Serializer

```
# myapi/serializers.py
from rest_framework import serializers
from .models import Products

class ProductsSerializer(serializers.ModelSerializer):
    class Meta:
        model = Products
        fields = '__all__'
```

Step 8: Create an API View

```
# myapi/views.py
from rest_framework.decorators import api_view
from rest_framework.response import Response
from .models import Products
from .serializers import ProductsSerializer

@api_view(['GET'])
def product_list(request):
    products = Products.objects.all()
    serializer = ProductsSerializer(products, many=True)
    return Response(serializer.data)
```

Step 9: Create URL Patterns

```
# myapi/urls.py
from django.urls import path
from .views import product_list

urlpatterns = [
    path('products/', product_list),
]
```

```
# myproject/urls.py
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('api/', include('myapi.urls')),
]
```

Step 10: Run the Server

```
python manage.py runserver
```

Visit: http://127.0.0.1:8000/api/products/ - You-ll get JSON data.