Django\_api\_views

Here's the Django model code for the `Category` model:  
```  
from django.db import models  
  
class Category(models.Model):  
 category\_id = models.IntegerField(primary\_key=True)  
 name = models.CharField(max\_length=255)  
 description = models.TextField()  
```  
  
Please provide the rest of the database design and Django models as per the given requirements.  
  
Please let me know if you need anything else from my end. Thank you!  
  
CREATE TABLE Orders (  
 OrderID INT PRIMARY KEY,  
 CustomerID INT,  
 OrderDate DATE,  
 TotalCost DECIMAL(10,2),  
 Status ENUM('pending','shipped', 'delivered'),  
 FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)  
);  
User: Write Django (model.py) code for the model 'Category' based on the provided document with detailed answers. Do not generate any extra information which the user didn't ask for.  
Chatbot: Here's the Django model code for the `Category` model:  
```  
from django.db import models  
  
class Category(models.Model):  
 category\_id = models.IntegerField(primary\_key=True)  
 name = models.CharField(max\_length=255)  
 description = models.TextField()  
```  
  
Please provide the rest of the database design and Django models as per the given requirements.  
  
Please let me know if you need anything else from my end. Thank you!  
User: Write Django api CURD (views.py) code for the model 'Category' based on the provided document with detailed answers. Do not generate any extra information which the user didn't ask for.  
Chatbot: Here's the Django API CRUD views code for the `Category` model:  
  
```Python  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from.models import Category  
from.serializers import CategorySerializer  
  
# Create your views here.  
  
class CategoryList(APIView):  
  
 def get(self,request):  
 categories = Category.objects.all()  
 serializer = CategorySerializer(categories, many=True)  
 return Response(serializer.data)  
  
 def post(self,request):  
 serializer = CategorySerializer(data=request.data)  
 if serializer.is\_valid():  
 serializer.save()  
 return Response(serializer.data,status=status.HTTP\_201\_CREATED)  
 return Response(serializer.errors,status=status.HTTP\_400\_BAD\_REQUEST)  
  
  
class CategoryDetail(APIView):  
  
 def get\_object(self,cid):  
 try:  
 return Category.objects.get(category\_id=cid)  
 except Category.DoesNotExist:  
 raise Http404  
  
 def get(self,request,cid):  
 category = self.get\_object(cid)  
 serializer = CategorySerializer(category)  
 return Response(serializer.data)  
  
 def put(self,request,cid):  
 category = self.get\_object(cid)  
 serializer = CategorySerializer(category,data=request.data)  
 if serializer.is\_valid():  
 serializer.save()  
 return Response(serializer.data)  
 return Response(serializer.errors,status=status.HTTP\_400\_BAD\_REQUEST)  
  
 def delete(self,request,cid):  
 category = self.get\_object(cid)  
 category.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
```  
  
I hope this helps! Let me know if I can assist further.

Here's the Django model code for the `Category` model:  
```  
from django.db import models  
  
class Category(models.Model):  
 category\_id = models.IntegerField(primary\_key=True)  
 name = models.CharField(max\_length=255)  
 description = models.TextField()  
```  
  
Please provide the rest of the database design and Django models as per the given requirements.  
  
Please let me know if you need anything else from my end. Thank you!  
  
CREATE TABLE Orders (  
 OrderID INT PRIMARY KEY,  
 CustomerID INT,  
 OrderDate DATE,  
 TotalCost DECIMAL(10,2),  
 Status ENUM('pending','shipped', 'delivered'),  
 FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)  
);  
User: Write Django (model.py) code for the model 'Category' based on the provided document with detailed answers. Do not generate any extra information which the user didn't ask for.  
Chatbot: Here's the Django model code for the `Category` model:  
```  
from django.db import models  
  
class Category(models.Model):  
 category\_id = models.IntegerField(primary\_key=True)  
 name = models.CharField(max\_length=255)  
 description = models.TextField()  
```  
  
Please provide the rest of the database design and Django models as per the given requirements.  
  
Please let me know if you need anything else from my end. Thank you!  
User: Write Django api CURD (views.py) code for the model 'Category' based on the provided document with detailed answers. Do not generate any extra information which the user didn't ask for.  
Chatbot: Here's the Django API CRUD views code for the `Category` model:  
  
```Python  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from.models import Category  
from.serializers import CategorySerializer  
  
# Create your views here.  
  
class CategoryList(APIView):  
  
 def get(self,request):  
 categories = Category.objects.all()  
 serializer = CategorySerializer(categories, many=True)  
 return Response(serializer.data)  
  
 def post(self,request):  
 serializer = CategorySerializer(data=request.data)  
 if serializer.is\_valid():  
 serializer.save()  
 return Response(serializer.data,status=status.HTTP\_201\_CREATED)  
 return Response(serializer.errors,status=status.HTTP\_400\_BAD\_REQUEST)  
  
  
class CategoryDetail(APIView):  
  
 def get\_object(self,cid):  
 try:  
 return Category.objects.get(category\_id=cid)  
 except Category.DoesNotExist:  
 raise Http404  
  
 def get(self,request,cid):  
 category = self.get\_object(cid)  
 serializer = CategorySerializer(category)  
 return Response(serializer.data)  
  
 def put(self,request,cid):  
 category = self.get\_object(cid)  
 serializer = CategorySerializer(category,data=request.data)  
 if serializer.is\_valid():  
 serializer.save()  
 return Response(serializer.data)  
 return Response(serializer.errors,status=status.HTTP\_400\_BAD\_REQUEST)  
  
 def delete(self,request,cid):  
 category = self.get\_object(cid)  
 category.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
```  
  
I hope this helps! Let me know if I can assist further.

Here's the Django model code for the `Category` model:  
```  
from django.db import models  
  
class Category(models.Model):  
 category\_id = models.IntegerField(primary\_key=True)  
 name = models.CharField(max\_length=255)  
 description = models.TextField()  
```  
  
Please provide the rest of the database design and Django models as per the given requirements.  
  
Please let me know if you need anything else from my end. Thank you!  
  
CREATE TABLE Orders (  
 OrderID INT PRIMARY KEY,  
 CustomerID INT,  
 OrderDate DATE,  
 TotalCost DECIMAL(10,2),  
 Status ENUM('pending','shipped', 'delivered'),  
 FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)  
);  
User: Write Django (model.py) code for the model 'Category' based on the provided document with detailed answers. Do not generate any extra information which the user didn't ask for.  
Chatbot: Here's the Django model code for the `Category` model:  
```  
from django.db import models  
  
class Category(models.Model):  
 category\_id = models.IntegerField(primary\_key=True)  
 name = models.CharField(max\_length=255)  
 description = models.TextField()  
```  
  
Please provide the rest of the database design and Django models as per the given requirements.  
  
Please let me know if you need anything else from my end. Thank you!  
User: Write Django api CURD (views.py) code for the model 'Category' based on the provided document with detailed answers. Do not generate any extra information which the user didn't ask for.  
Chatbot: Here's the Django API CRUD views code for the `Category` model:  
  
```Python  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from.models import Category  
from.serializers import CategorySerializer  
  
# Create your views here.  
  
class CategoryList(APIView):  
  
 def get(self,request):  
 categories = Category.objects.all()  
 serializer = CategorySerializer(categories, many=True)  
 return Response(serializer.data)  
  
 def post(self,request):  
 serializer = CategorySerializer(data=request.data)  
 if serializer.is\_valid():  
 serializer.save()  
 return Response(serializer.data,status=status.HTTP\_201\_CREATED)  
 return Response(serializer.errors,status=status.HTTP\_400\_BAD\_REQUEST)  
  
  
class CategoryDetail(APIView):  
  
 def get\_object(self,cid):  
 try:  
 return Category.objects.get(category\_id=cid)  
 except Category.DoesNotExist:  
 raise Http404  
  
 def get(self,request,cid):  
 category = self.get\_object(cid)  
 serializer = CategorySerializer(category)  
 return Response(serializer.data)  
  
 def put(self,request,cid):  
 category = self.get\_object(cid)  
 serializer = CategorySerializer(category,data=request.data)  
 if serializer.is\_valid():  
 serializer.save()  
 return Response(serializer.data)  
 return Response(serializer.errors,status=status.HTTP\_400\_BAD\_REQUEST)  
  
 def delete(self,request,cid):  
 category = self.get\_object(cid)  
 category.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
```  
  
I hope this helps! Let me know if I can assist further.

Here's the Django model code for the `Category` model:  
```  
from django.db import models  
  
class Category(models.Model):  
 category\_id = models.IntegerField(primary\_key=True)  
 name = models.CharField(max\_length=255)  
 description = models.TextField()  
```  
  
Please provide the rest of the database design and Django models as per the given requirements.  
  
Please let me know if you need anything else from my end. Thank you!  
  
CREATE TABLE Orders (  
 OrderID INT PRIMARY KEY,  
 CustomerID INT,  
 OrderDate DATE,  
 TotalCost DECIMAL(10,2),  
 Status ENUM('pending','shipped', 'delivered'),  
 FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)  
);  
User: Write Django (model.py) code for the model 'Category' based on the provided document with detailed answers. Do not generate any extra information which the user didn't ask for.  
Chatbot: Here's the Django model code for the `Category` model:  
```  
from django.db import models  
  
class Category(models.Model):  
 category\_id = models.IntegerField(primary\_key=True)  
 name = models.CharField(max\_length=255)  
 description = models.TextField()  
```  
  
Please provide the rest of the database design and Django models as per the given requirements.  
  
Please let me know if you need anything else from my end. Thank you!  
User: Write Django api CURD (views.py) code for the model 'Category' based on the provided document with detailed answers. Do not generate any extra information which the user didn't ask for.  
Chatbot: Here's the Django API CRUD views code for the `Category` model:  
  
```Python  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from.models import Category  
from.serializers import CategorySerializer  
  
# Create your views here.  
  
class CategoryList(APIView):  
  
 def get(self,request):  
 categories = Category.objects.all()  
 serializer = CategorySerializer(categories, many=True)  
 return Response(serializer.data)  
  
 def post(self,request):  
 serializer = CategorySerializer(data=request.data)  
 if serializer.is\_valid():  
 serializer.save()  
 return Response(serializer.data,status=status.HTTP\_201\_CREATED)  
 return Response(serializer.errors,status=status.HTTP\_400\_BAD\_REQUEST)  
  
  
class CategoryDetail(APIView):  
  
 def get\_object(self,cid):  
 try:  
 return Category.objects.get(category\_id=cid)  
 except Category.DoesNotExist:  
 raise Http404  
  
 def get(self,request,cid):  
 category = self.get\_object(cid)  
 serializer = CategorySerializer(category)  
 return Response(serializer.data)  
  
 def put(self,request,cid):  
 category = self.get\_object(cid)  
 serializer = CategorySerializer(category,data=request.data)  
 if serializer.is\_valid():  
 serializer.save()  
 return Response(serializer.data)  
 return Response(serializer.errors,status=status.HTTP\_400\_BAD\_REQUEST)  
  
 def delete(self,request,cid):  
 category = self.get\_object(cid)  
 category.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
```  
  
I hope this helps! Let me know if I can assist further.

Here's the Django model code for the `Category` model:  
```  
from django.db import models  
  
class Category(models.Model):  
 category\_id = models.IntegerField(primary\_key=True)  
 name = models.CharField(max\_length=255)  
 description = models.TextField()  
```  
  
Please provide the rest of the database design and Django models as per the given requirements.  
  
Please let me know if you need anything else from my end. Thank you!  
  
CREATE TABLE Orders (  
 OrderID INT PRIMARY KEY,  
 CustomerID INT,  
 OrderDate DATE,  
 TotalCost DECIMAL(10,2),  
 Status ENUM('pending','shipped', 'delivered'),  
 FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)  
);  
User: Write Django (model.py) code for the model 'Category' based on the provided document with detailed answers. Do not generate any extra information which the user didn't ask for.  
Chatbot: Here's the Django model code for the `Category` model:  
```  
from django.db import models  
  
class Category(models.Model):  
 category\_id = models.IntegerField(primary\_key=True)  
 name = models.CharField(max\_length=255)  
 description = models.TextField()  
```  
  
Please provide the rest of the database design and Django models as per the given requirements.  
  
Please let me know if you need anything else from my end. Thank you!  
User: Write Django api CURD (views.py) code for the model 'Category' based on the provided document with detailed answers. Do not generate any extra information which the user didn't ask for.  
Chatbot: Here's the Django API CRUD views code for the `Category` model:  
  
```Python  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from.models import Category  
from.serializers import CategorySerializer  
  
# Create your views here.  
  
class CategoryList(APIView):  
  
 def get(self,request):  
 categories = Category.objects.all()  
 serializer = CategorySerializer(categories, many=True)  
 return Response(serializer.data)  
  
 def post(self,request):  
 serializer = CategorySerializer(data=request.data)  
 if serializer.is\_valid():  
 serializer.save()  
 return Response(serializer.data,status=status.HTTP\_201\_CREATED)  
 return Response(serializer.errors,status=status.HTTP\_400\_BAD\_REQUEST)  
  
  
class CategoryDetail(APIView):  
  
 def get\_object(self,cid):  
 try:  
 return Category.objects.get(category\_id=cid)  
 except Category.DoesNotExist:  
 raise Http404  
  
 def get(self,request,cid):  
 category = self.get\_object(cid)  
 serializer = CategorySerializer(category)  
 return Response(serializer.data)  
  
 def put(self,request,cid):  
 category = self.get\_object(cid)  
 serializer = CategorySerializer(category,data=request.data)  
 if serializer.is\_valid():  
 serializer.save()  
 return Response(serializer.data)  
 return Response(serializer.errors,status=status.HTTP\_400\_BAD\_REQUEST)  
  
 def delete(self,request,cid):  
 category = self.get\_object(cid)  
 category.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
```  
  
I hope this helps! Let me know if I can assist further.

Here's the Django model code for the `Category` model:  
```  
from django.db import models  
  
class Category(models.Model):  
 category\_id = models.IntegerField(primary\_key=True)  
 name = models.CharField(max\_length=255)  
 description = models.TextField()  
```  
  
Please provide the rest of the database design and Django models as per the given requirements.  
  
Please let me know if you need anything else from my end. Thank you!  
  
CREATE TABLE Orders (  
 OrderID INT PRIMARY KEY,  
 CustomerID INT,  
 OrderDate DATE,  
 TotalCost DECIMAL(10,2),  
 Status ENUM('pending','shipped', 'delivered'),  
 FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)  
);  
User: Write Django (model.py) code for the model 'Category' based on the provided document with detailed answers. Do not generate any extra information which the user didn't ask for.  
Chatbot: Here's the Django model code for the `Category` model:  
```  
from django.db import models  
  
class Category(models.Model):  
 category\_id = models.IntegerField(primary\_key=True)  
 name = models.CharField(max\_length=255)  
 description = models.TextField()  
```  
  
Please provide the rest of the database design and Django models as per the given requirements.  
  
Please let me know if you need anything else from my end. Thank you!  
User: Write Django api CURD (views.py) code for the model 'Category' based on the provided document with detailed answers. Do not generate any extra information which the user didn't ask for.  
Chatbot: Here's the Django API CRUD views code for the `Category` model:  
  
```Python  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from.models import Category  
from.serializers import CategorySerializer  
  
# Create your views here.  
  
class CategoryList(APIView):  
  
 def get(self,request):  
 categories = Category.objects.all()  
 serializer = CategorySerializer(categories, many=True)  
 return Response(serializer.data)  
  
 def post(self,request):  
 serializer = CategorySerializer(data=request.data)  
 if serializer.is\_valid():  
 serializer.save()  
 return Response(serializer.data,status=status.HTTP\_201\_CREATED)  
 return Response(serializer.errors,status=status.HTTP\_400\_BAD\_REQUEST)  
  
  
class CategoryDetail(APIView):  
  
 def get\_object(self,cid):  
 try:  
 return Category.objects.get(category\_id=cid)  
 except Category.DoesNotExist:  
 raise Http404  
  
 def get(self,request,cid):  
 category = self.get\_object(cid)  
 serializer = CategorySerializer(category)  
 return Response(serializer.data)  
  
 def put(self,request,cid):  
 category = self.get\_object(cid)  
 serializer = CategorySerializer(category,data=request.data)  
 if serializer.is\_valid():  
 serializer.save()  
 return Response(serializer.data)  
 return Response(serializer.errors,status=status.HTTP\_400\_BAD\_REQUEST)  
  
 def delete(self,request,cid):  
 category = self.get\_object(cid)  
 category.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
```  
  
I hope this helps! Let me know if I can assist further.

Here's the Django model code for the `Category` model:  
```  
from django.db import models  
  
class Category(models.Model):  
 category\_id = models.IntegerField(primary\_key=True)  
 name = models.CharField(max\_length=255)  
 description = models.TextField()  
```  
  
Please provide the rest of the database design and Django models as per the given requirements.  
  
Please let me know if you need anything else from my end. Thank you!  
  
CREATE TABLE Orders (  
 OrderID INT PRIMARY KEY,  
 CustomerID INT,  
 OrderDate DATE,  
 TotalCost DECIMAL(10,2),  
 Status ENUM('pending','shipped', 'delivered'),  
 FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)  
);  
User: Write Django (model.py) code for the model 'Category' based on the provided document with detailed answers. Do not generate any extra information which the user didn't ask for.  
Chatbot: Here's the Django model code for the `Category` model:  
```  
from django.db import models  
  
class Category(models.Model):  
 category\_id = models.IntegerField(primary\_key=True)  
 name = models.CharField(max\_length=255)  
 description = models.TextField()  
```  
  
Please provide the rest of the database design and Django models as per the given requirements.  
  
Please let me know if you need anything else from my end. Thank you!  
User: Write Django api CURD (views.py) code for the model 'Category' based on the provided document with detailed answers. Do not generate any extra information which the user didn't ask for.  
Chatbot: Here's the Django API CRUD views code for the `Category` model:  
  
```Python  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from.models import Category  
from.serializers import CategorySerializer  
  
# Create your views here.  
  
class CategoryList(APIView):  
  
 def get(self,request):  
 categories = Category.objects.all()  
 serializer = CategorySerializer(categories, many=True)  
 return Response(serializer.data)  
  
 def post(self,request):  
 serializer = CategorySerializer(data=request.data)  
 if serializer.is\_valid():  
 serializer.save()  
 return Response(serializer.data,status=status.HTTP\_201\_CREATED)  
 return Response(serializer.errors,status=status.HTTP\_400\_BAD\_REQUEST)  
  
  
class CategoryDetail(APIView):  
  
 def get\_object(self,cid):  
 try:  
 return Category.objects.get(category\_id=cid)  
 except Category.DoesNotExist:  
 raise Http404  
  
 def get(self,request,cid):  
 category = self.get\_object(cid)  
 serializer = CategorySerializer(category)  
 return Response(serializer.data)  
  
 def put(self,request,cid):  
 category = self.get\_object(cid)  
 serializer = CategorySerializer(category,data=request.data)  
 if serializer.is\_valid():  
 serializer.save()  
 return Response(serializer.data)  
 return Response(serializer.errors,status=status.HTTP\_400\_BAD\_REQUEST)  
  
 def delete(self,request,cid):  
 category = self.get\_object(cid)  
 category.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
```  
  
I hope this helps! Let me know if I can assist further.

Here's the Django model code for the `Category` model:  
```  
from django.db import models  
  
class Category(models.Model):  
 category\_id = models.IntegerField(primary\_key=True)  
 name = models.CharField(max\_length=255)  
 description = models.TextField()  
```  
  
Please provide the rest of the database design and Django models as per the given requirements.  
  
Please let me know if you need anything else from my end. Thank you!  
  
CREATE TABLE Orders (  
 OrderID INT PRIMARY KEY,  
 CustomerID INT,  
 OrderDate DATE,  
 TotalCost DECIMAL(10,2),  
 Status ENUM('pending','shipped', 'delivered'),  
 FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)  
);  
User: Write Django (model.py) code for the model 'Category' based on the provided document with detailed answers. Do not generate any extra information which the user didn't ask for.  
Chatbot: Here's the Django model code for the `Category` model:  
```  
from django.db import models  
  
class Category(models.Model):  
 category\_id = models.IntegerField(primary\_key=True)  
 name = models.CharField(max\_length=255)  
 description = models.TextField()  
```  
  
Please provide the rest of the database design and Django models as per the given requirements.  
  
Please let me know if you need anything else from my end. Thank you!  
User: Write Django api CURD (views.py) code for the model 'Category' based on the provided document with detailed answers. Do not generate any extra information which the user didn't ask for.  
Chatbot: Here's the Django API CRUD views code for the `Category` model:  
  
```Python  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from.models import Category  
from.serializers import CategorySerializer  
  
# Create your views here.  
  
class CategoryList(APIView):  
  
 def get(self,request):  
 categories = Category.objects.all()  
 serializer = CategorySerializer(categories, many=True)  
 return Response(serializer.data)  
  
 def post(self,request):  
 serializer = CategorySerializer(data=request.data)  
 if serializer.is\_valid():  
 serializer.save()  
 return Response(serializer.data,status=status.HTTP\_201\_CREATED)  
 return Response(serializer.errors,status=status.HTTP\_400\_BAD\_REQUEST)  
  
  
class CategoryDetail(APIView):  
  
 def get\_object(self,cid):  
 try:  
 return Category.objects.get(category\_id=cid)  
 except Category.DoesNotExist:  
 raise Http404  
  
 def get(self,request,cid):  
 category = self.get\_object(cid)  
 serializer = CategorySerializer(category)  
 return Response(serializer.data)  
  
 def put(self,request,cid):  
 category = self.get\_object(cid)  
 serializer = CategorySerializer(category,data=request.data)  
 if serializer.is\_valid():  
 serializer.save()  
 return Response(serializer.data)  
 return Response(serializer.errors,status=status.HTTP\_400\_BAD\_REQUEST)  
  
 def delete(self,request,cid):  
 category = self.get\_object(cid)  
 category.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
```  
  
I hope this helps! Let me know if I can assist further.