Django\_api\_views

Here's your Django API Views Code for User Registration:  
  
```  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from rest\_framework import status  
from.serializers import RegisterSerializer  
from.models import Customer  
  
class RegisterUser(APIView):  
 serializer\_class = RegisterSerializer  
  
 def post(self, request):  
 serializer = self.serializer\_class(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 RetrieveUpdateDeleteUser(APIView):  
 queryset = Customer.objects.all()  
  
 def get\_object(self, pk):  
 try:  
 return Customer.objects.get(pk=pk)  
 except Customer.DoesNotExist:  
 return Response(status=status.HTTP\_404\_NOT\_FOUND)  
  
 def get(self, request, pk):  
 customer = self.get\_object(pk)  
 serializer = RegisterSerializer(customer)  
 return Response(serializer.data)  
  
 def put(self, request, pk):  
 customer = self.get\_object(pk)  
 serializer = RegisterSerializer(customer, 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, pk):  
 customer = self.get\_object(pk)  
 customer.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
  
```  
  
I hope that helps! Let me know if there's anything else I can assist you with.

Here's your Django API Views Code for User Registration:  
  
```  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from rest\_framework import status  
from.serializers import RegisterSerializer  
from.models import Customer  
  
class RegisterUser(APIView):  
 serializer\_class = RegisterSerializer  
  
 def post(self, request):  
 serializer = self.serializer\_class(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 RetrieveUpdateDeleteUser(APIView):  
 queryset = Customer.objects.all()  
  
 def get\_object(self, pk):  
 try:  
 return Customer.objects.get(pk=pk)  
 except Customer.DoesNotExist:  
 return Response(status=status.HTTP\_404\_NOT\_FOUND)  
  
 def get(self, request, pk):  
 customer = self.get\_object(pk)  
 serializer = RegisterSerializer(customer)  
 return Response(serializer.data)  
  
 def put(self, request, pk):  
 customer = self.get\_object(pk)  
 serializer = RegisterSerializer(customer, 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, pk):  
 customer = self.get\_object(pk)  
 customer.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
  
```  
  
I hope that helps! Let me know if there's anything else I can assist you with.

Here's your Django API Views Code for User Registration:  
  
```  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from rest\_framework import status  
from.serializers import RegisterSerializer  
from.models import Customer  
  
class RegisterUser(APIView):  
 serializer\_class = RegisterSerializer  
  
 def post(self, request):  
 serializer = self.serializer\_class(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 RetrieveUpdateDeleteUser(APIView):  
 queryset = Customer.objects.all()  
  
 def get\_object(self, pk):  
 try:  
 return Customer.objects.get(pk=pk)  
 except Customer.DoesNotExist:  
 return Response(status=status.HTTP\_404\_NOT\_FOUND)  
  
 def get(self, request, pk):  
 customer = self.get\_object(pk)  
 serializer = RegisterSerializer(customer)  
 return Response(serializer.data)  
  
 def put(self, request, pk):  
 customer = self.get\_object(pk)  
 serializer = RegisterSerializer(customer, 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, pk):  
 customer = self.get\_object(pk)  
 customer.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
  
```  
  
I hope that helps! Let me know if there's anything else I can assist you with.

Here's your Django API Views Code for User Registration:  
  
```  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from rest\_framework import status  
from.serializers import RegisterSerializer  
from.models import Customer  
  
class RegisterUser(APIView):  
 serializer\_class = RegisterSerializer  
  
 def post(self, request):  
 serializer = self.serializer\_class(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 RetrieveUpdateDeleteUser(APIView):  
 queryset = Customer.objects.all()  
  
 def get\_object(self, pk):  
 try:  
 return Customer.objects.get(pk=pk)  
 except Customer.DoesNotExist:  
 return Response(status=status.HTTP\_404\_NOT\_FOUND)  
  
 def get(self, request, pk):  
 customer = self.get\_object(pk)  
 serializer = RegisterSerializer(customer)  
 return Response(serializer.data)  
  
 def put(self, request, pk):  
 customer = self.get\_object(pk)  
 serializer = RegisterSerializer(customer, 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, pk):  
 customer = self.get\_object(pk)  
 customer.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
  
```  
  
I hope that helps! Let me know if there's anything else I can assist you with.

Here's your Django API Views Code for User Registration:  
  
```  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from rest\_framework import status  
from.serializers import RegisterSerializer  
from.models import Customer  
  
class RegisterUser(APIView):  
 serializer\_class = RegisterSerializer  
  
 def post(self, request):  
 serializer = self.serializer\_class(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 RetrieveUpdateDeleteUser(APIView):  
 queryset = Customer.objects.all()  
  
 def get\_object(self, pk):  
 try:  
 return Customer.objects.get(pk=pk)  
 except Customer.DoesNotExist:  
 return Response(status=status.HTTP\_404\_NOT\_FOUND)  
  
 def get(self, request, pk):  
 customer = self.get\_object(pk)  
 serializer = RegisterSerializer(customer)  
 return Response(serializer.data)  
  
 def put(self, request, pk):  
 customer = self.get\_object(pk)  
 serializer = RegisterSerializer(customer, 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, pk):  
 customer = self.get\_object(pk)  
 customer.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
  
```  
  
I hope that helps! Let me know if there's anything else I can assist you with.