Django\_api\_views

Here's the Django API CRUD views code for the CouponCode model:  
  
```  
from rest\_framework import status  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from.serializers import CouponCodeSerializer  
from.models import CouponCode  
  
class CouponCodeList(APIView):  
 def get(self, request):  
 coupons = CouponCode.objects.all()  
 serializer = CouponCodeSerializer(coupons, many=True)  
 return Response(serializer.data)  
  
 def post(self, request):  
 serializer = CouponCodeSerializer(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 CouponCodeDetail(APIView):  
 def get\_object(self, pk):  
 try:  
 return CouponCode.objects.get(pk=pk)  
 except CouponCode.DoesNotExist:  
 return Response(status=status.HTTP\_404\_NOT\_FOUND)  
  
 def get(self, request, pk):  
 coupon = self.get\_object(pk)  
 serializer = CouponCodeSerializer(coupon)  
 return Response(serializer.data)  
  
 def put(self, request, pk):  
 coupon = self.get\_object(pk)  
 serializer = CouponCodeSerializer(coupon, 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):  
 coupon = self.get\_object(pk)  
 coupon.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
```  
  
This code provides the following endpoints:  
- GET /api/couponcodes: Retrieves all coupon codes.  
- POST /api/couponcodes: Creates a new coupon code.  
- GET /api/couponcodes/{pk}: Retrieves a specific coupon code by its ID.  
- PUT /api/couponcodes/{pk}: Updates an existing coupon code.  
- DELETE /api/couponcodes/{pk}: Deletes a specific coupon code.  
  
The serializers used here are assumed to be defined elsewhere in your application; ensure they match the structure of your CouponCode model. If you have any questions or need help defining these serializers, feel free to ask!

Here's the Django API CRUD views code for the CouponCode model:  
  
```  
from rest\_framework import status  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from.serializers import CouponCodeSerializer  
from.models import CouponCode  
  
class CouponCodeList(APIView):  
 def get(self, request):  
 coupons = CouponCode.objects.all()  
 serializer = CouponCodeSerializer(coupons, many=True)  
 return Response(serializer.data)  
  
 def post(self, request):  
 serializer = CouponCodeSerializer(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 CouponCodeDetail(APIView):  
 def get\_object(self, pk):  
 try:  
 return CouponCode.objects.get(pk=pk)  
 except CouponCode.DoesNotExist:  
 return Response(status=status.HTTP\_404\_NOT\_FOUND)  
  
 def get(self, request, pk):  
 coupon = self.get\_object(pk)  
 serializer = CouponCodeSerializer(coupon)  
 return Response(serializer.data)  
  
 def put(self, request, pk):  
 coupon = self.get\_object(pk)  
 serializer = CouponCodeSerializer(coupon, 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):  
 coupon = self.get\_object(pk)  
 coupon.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
```  
  
This code provides the following endpoints:  
- GET /api/couponcodes: Retrieves all coupon codes.  
- POST /api/couponcodes: Creates a new coupon code.  
- GET /api/couponcodes/{pk}: Retrieves a specific coupon code by its ID.  
- PUT /api/couponcodes/{pk}: Updates an existing coupon code.  
- DELETE /api/couponcodes/{pk}: Deletes a specific coupon code.  
  
The serializers used here are assumed to be defined elsewhere in your application; ensure they match the structure of your CouponCode model. If you have any questions or need help defining these serializers, feel free to ask!

Here's the Django API CRUD views code for the CouponCode model:  
  
```  
from rest\_framework import status  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from.serializers import CouponCodeSerializer  
from.models import CouponCode  
  
class CouponCodeList(APIView):  
 def get(self, request):  
 coupons = CouponCode.objects.all()  
 serializer = CouponCodeSerializer(coupons, many=True)  
 return Response(serializer.data)  
  
 def post(self, request):  
 serializer = CouponCodeSerializer(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 CouponCodeDetail(APIView):  
 def get\_object(self, pk):  
 try:  
 return CouponCode.objects.get(pk=pk)  
 except CouponCode.DoesNotExist:  
 return Response(status=status.HTTP\_404\_NOT\_FOUND)  
  
 def get(self, request, pk):  
 coupon = self.get\_object(pk)  
 serializer = CouponCodeSerializer(coupon)  
 return Response(serializer.data)  
  
 def put(self, request, pk):  
 coupon = self.get\_object(pk)  
 serializer = CouponCodeSerializer(coupon, 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):  
 coupon = self.get\_object(pk)  
 coupon.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
```  
  
This code provides the following endpoints:  
- GET /api/couponcodes: Retrieves all coupon codes.  
- POST /api/couponcodes: Creates a new coupon code.  
- GET /api/couponcodes/{pk}: Retrieves a specific coupon code by its ID.  
- PUT /api/couponcodes/{pk}: Updates an existing coupon code.  
- DELETE /api/couponcodes/{pk}: Deletes a specific coupon code.  
  
The serializers used here are assumed to be defined elsewhere in your application; ensure they match the structure of your CouponCode model. If you have any questions or need help defining these serializers, feel free to ask!

Here's the Django API CRUD views code for the CouponCode model:  
  
```  
from rest\_framework import status  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from.serializers import CouponCodeSerializer  
from.models import CouponCode  
  
class CouponCodeList(APIView):  
 def get(self, request):  
 coupons = CouponCode.objects.all()  
 serializer = CouponCodeSerializer(coupons, many=True)  
 return Response(serializer.data)  
  
 def post(self, request):  
 serializer = CouponCodeSerializer(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 CouponCodeDetail(APIView):  
 def get\_object(self, pk):  
 try:  
 return CouponCode.objects.get(pk=pk)  
 except CouponCode.DoesNotExist:  
 return Response(status=status.HTTP\_404\_NOT\_FOUND)  
  
 def get(self, request, pk):  
 coupon = self.get\_object(pk)  
 serializer = CouponCodeSerializer(coupon)  
 return Response(serializer.data)  
  
 def put(self, request, pk):  
 coupon = self.get\_object(pk)  
 serializer = CouponCodeSerializer(coupon, 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):  
 coupon = self.get\_object(pk)  
 coupon.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
```  
  
This code provides the following endpoints:  
- GET /api/couponcodes: Retrieves all coupon codes.  
- POST /api/couponcodes: Creates a new coupon code.  
- GET /api/couponcodes/{pk}: Retrieves a specific coupon code by its ID.  
- PUT /api/couponcodes/{pk}: Updates an existing coupon code.  
- DELETE /api/couponcodes/{pk}: Deletes a specific coupon code.  
  
The serializers used here are assumed to be defined elsewhere in your application; ensure they match the structure of your CouponCode model. If you have any questions or need help defining these serializers, feel free to ask!

Here's the Django API CRUD views code for the CouponCode model:  
  
```  
from rest\_framework import status  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from.serializers import CouponCodeSerializer  
from.models import CouponCode  
  
class CouponCodeList(APIView):  
 def get(self, request):  
 coupons = CouponCode.objects.all()  
 serializer = CouponCodeSerializer(coupons, many=True)  
 return Response(serializer.data)  
  
 def post(self, request):  
 serializer = CouponCodeSerializer(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 CouponCodeDetail(APIView):  
 def get\_object(self, pk):  
 try:  
 return CouponCode.objects.get(pk=pk)  
 except CouponCode.DoesNotExist:  
 return Response(status=status.HTTP\_404\_NOT\_FOUND)  
  
 def get(self, request, pk):  
 coupon = self.get\_object(pk)  
 serializer = CouponCodeSerializer(coupon)  
 return Response(serializer.data)  
  
 def put(self, request, pk):  
 coupon = self.get\_object(pk)  
 serializer = CouponCodeSerializer(coupon, 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):  
 coupon = self.get\_object(pk)  
 coupon.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
```  
  
This code provides the following endpoints:  
- GET /api/couponcodes: Retrieves all coupon codes.  
- POST /api/couponcodes: Creates a new coupon code.  
- GET /api/couponcodes/{pk}: Retrieves a specific coupon code by its ID.  
- PUT /api/couponcodes/{pk}: Updates an existing coupon code.  
- DELETE /api/couponcodes/{pk}: Deletes a specific coupon code.  
  
The serializers used here are assumed to be defined elsewhere in your application; ensure they match the structure of your CouponCode model. If you have any questions or need help defining these serializers, feel free to ask!

Here's the Django API CRUD views code for the CouponCode model:  
  
```  
from rest\_framework import status  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from.serializers import CouponCodeSerializer  
from.models import CouponCode  
  
class CouponCodeList(APIView):  
 def get(self, request):  
 coupons = CouponCode.objects.all()  
 serializer = CouponCodeSerializer(coupons, many=True)  
 return Response(serializer.data)  
  
 def post(self, request):  
 serializer = CouponCodeSerializer(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 CouponCodeDetail(APIView):  
 def get\_object(self, pk):  
 try:  
 return CouponCode.objects.get(pk=pk)  
 except CouponCode.DoesNotExist:  
 return Response(status=status.HTTP\_404\_NOT\_FOUND)  
  
 def get(self, request, pk):  
 coupon = self.get\_object(pk)  
 serializer = CouponCodeSerializer(coupon)  
 return Response(serializer.data)  
  
 def put(self, request, pk):  
 coupon = self.get\_object(pk)  
 serializer = CouponCodeSerializer(coupon, 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):  
 coupon = self.get\_object(pk)  
 coupon.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
```  
  
This code provides the following endpoints:  
- GET /api/couponcodes: Retrieves all coupon codes.  
- POST /api/couponcodes: Creates a new coupon code.  
- GET /api/couponcodes/{pk}: Retrieves a specific coupon code by its ID.  
- PUT /api/couponcodes/{pk}: Updates an existing coupon code.  
- DELETE /api/couponcodes/{pk}: Deletes a specific coupon code.  
  
The serializers used here are assumed to be defined elsewhere in your application; ensure they match the structure of your CouponCode model. If you have any questions or need help defining these serializers, feel free to ask!

Here's the Django API CRUD views code for the CouponCode model:  
  
```  
from rest\_framework import status  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from.serializers import CouponCodeSerializer  
from.models import CouponCode  
  
class CouponCodeList(APIView):  
 def get(self, request):  
 coupons = CouponCode.objects.all()  
 serializer = CouponCodeSerializer(coupons, many=True)  
 return Response(serializer.data)  
  
 def post(self, request):  
 serializer = CouponCodeSerializer(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 CouponCodeDetail(APIView):  
 def get\_object(self, pk):  
 try:  
 return CouponCode.objects.get(pk=pk)  
 except CouponCode.DoesNotExist:  
 return Response(status=status.HTTP\_404\_NOT\_FOUND)  
  
 def get(self, request, pk):  
 coupon = self.get\_object(pk)  
 serializer = CouponCodeSerializer(coupon)  
 return Response(serializer.data)  
  
 def put(self, request, pk):  
 coupon = self.get\_object(pk)  
 serializer = CouponCodeSerializer(coupon, 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):  
 coupon = self.get\_object(pk)  
 coupon.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
```  
  
This code provides the following endpoints:  
- GET /api/couponcodes: Retrieves all coupon codes.  
- POST /api/couponcodes: Creates a new coupon code.  
- GET /api/couponcodes/{pk}: Retrieves a specific coupon code by its ID.  
- PUT /api/couponcodes/{pk}: Updates an existing coupon code.  
- DELETE /api/couponcodes/{pk}: Deletes a specific coupon code.  
  
The serializers used here are assumed to be defined elsewhere in your application; ensure they match the structure of your CouponCode model. If you have any questions or need help defining these serializers, feel free to ask!

Here's the Django API CRUD views code for the CouponCode model:  
  
```  
from rest\_framework import status  
from rest\_framework.response import Response  
from rest\_framework.views import APIView  
from.serializers import CouponCodeSerializer  
from.models import CouponCode  
  
class CouponCodeList(APIView):  
 def get(self, request):  
 coupons = CouponCode.objects.all()  
 serializer = CouponCodeSerializer(coupons, many=True)  
 return Response(serializer.data)  
  
 def post(self, request):  
 serializer = CouponCodeSerializer(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 CouponCodeDetail(APIView):  
 def get\_object(self, pk):  
 try:  
 return CouponCode.objects.get(pk=pk)  
 except CouponCode.DoesNotExist:  
 return Response(status=status.HTTP\_404\_NOT\_FOUND)  
  
 def get(self, request, pk):  
 coupon = self.get\_object(pk)  
 serializer = CouponCodeSerializer(coupon)  
 return Response(serializer.data)  
  
 def put(self, request, pk):  
 coupon = self.get\_object(pk)  
 serializer = CouponCodeSerializer(coupon, 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):  
 coupon = self.get\_object(pk)  
 coupon.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)  
```  
  
This code provides the following endpoints:  
- GET /api/couponcodes: Retrieves all coupon codes.  
- POST /api/couponcodes: Creates a new coupon code.  
- GET /api/couponcodes/{pk}: Retrieves a specific coupon code by its ID.  
- PUT /api/couponcodes/{pk}: Updates an existing coupon code.  
- DELETE /api/couponcodes/{pk}: Deletes a specific coupon code.  
  
The serializers used here are assumed to be defined elsewhere in your application; ensure they match the structure of your CouponCode model. If you have any questions or need help defining these serializers, feel free to ask!