**ViewSet CreateAPIView**

**For creating a record we use CreateAPIView**

**views.py**

**from rest\_framework.generics import ListAPIView,CreateAPIView**

**……**

**class EmployeeListApiView(ListAPIView):**

**………..**

**class EmployeeCreateApiView(CreateAPIView):**

**queryset=Employee.objects.all()**

**serializer\_class=EmployeeSerializer**

**urls.py**

**from django.contrib import admin**

**from django.urls import path**

**from myApp.views import \***

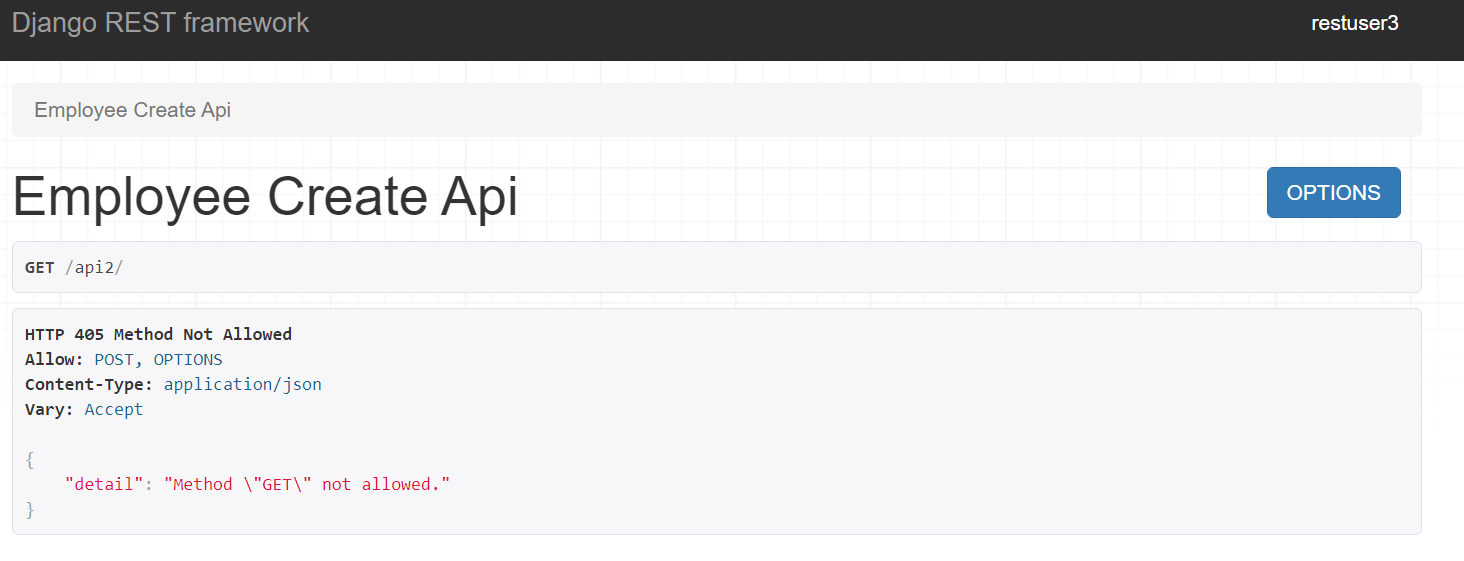
**urlpatterns = [**

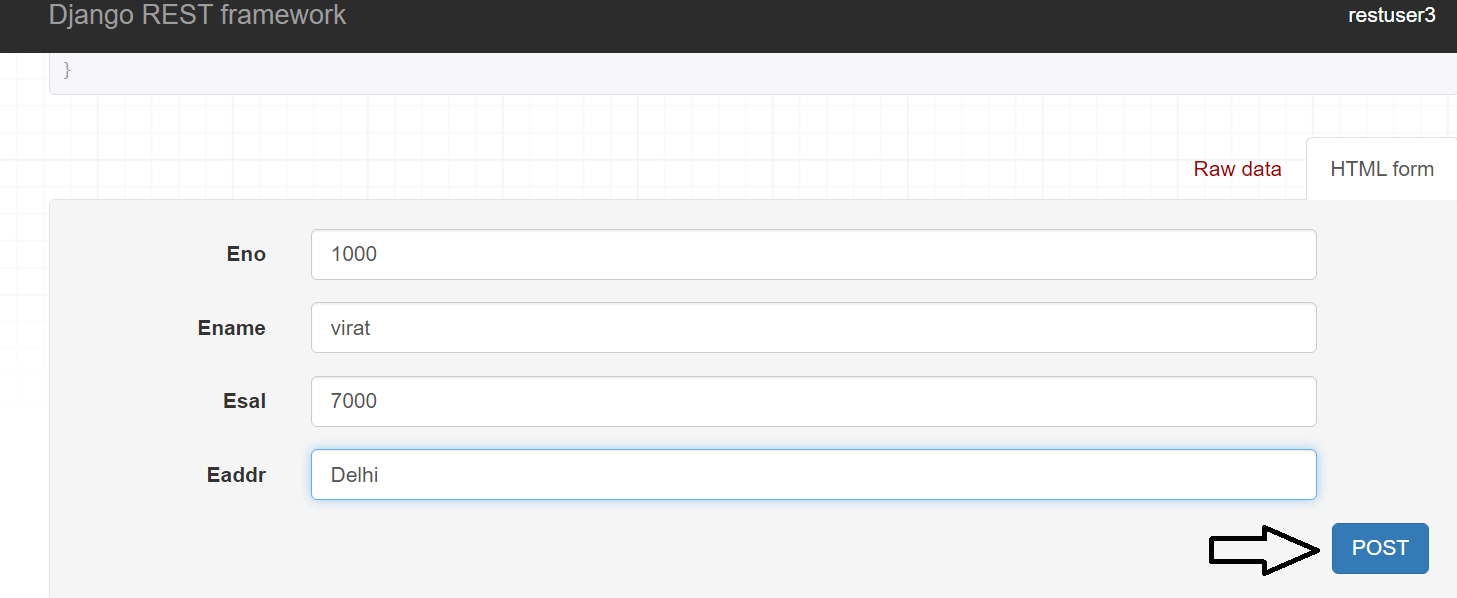
**path('admin/', admin.site.urls),**

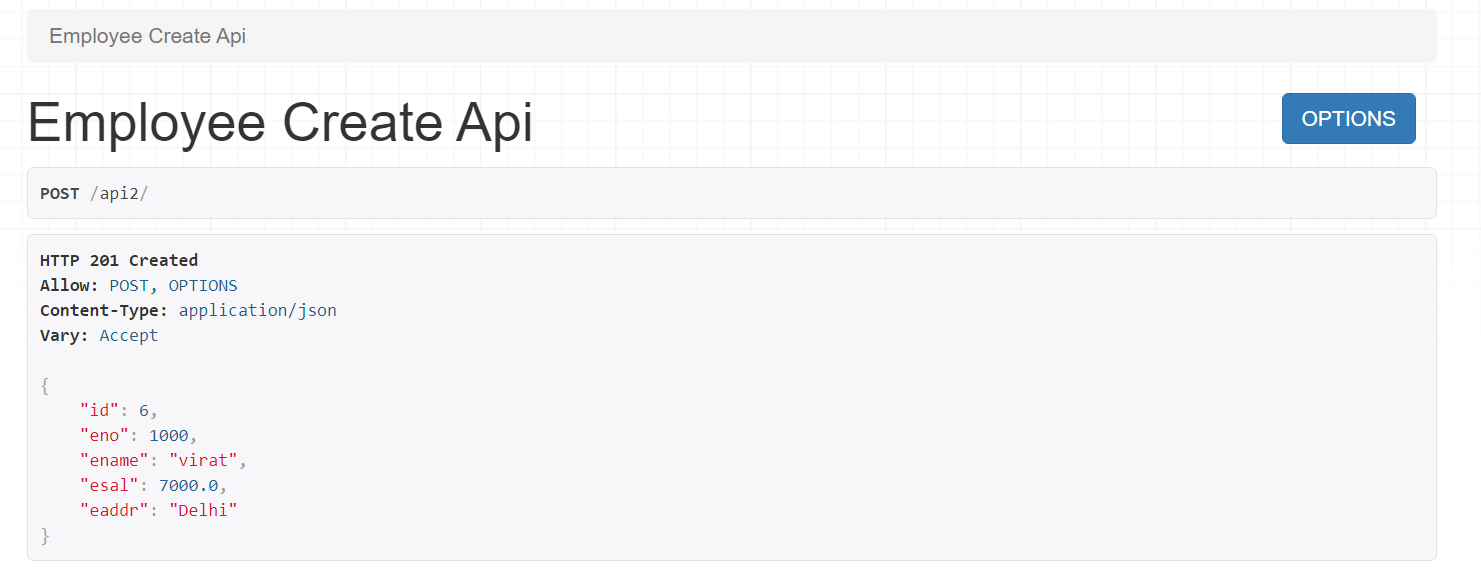
**path('api1/',EmployeeListApiView.as\_view()),**

**path('api2/',EmployeeCreateApiView.as\_view())**

**]**

****

****

****

**Retrieve operation**

**To get a particular record we use RetrieveAPIView.**

**views.py**

**from rest\_framework.generics import ListAPIView,CreateAPIView,RetrieveAPIView**

**class EmployeeListApiView(ListAPIView):**

**……**

**class EmployeeCreateApiView(CreateAPIView):**

**….**

**class EmployeeRetrieveApiView(RetrieveAPIView):**

**queryset=Employee.objects.all()**

**serializer\_class=EmployeeSerializer**

**urls.py**

**from django.contrib import admin**

**from django.conf.urls import url**

**from myApp.views import \***

**urlpatterns = [**

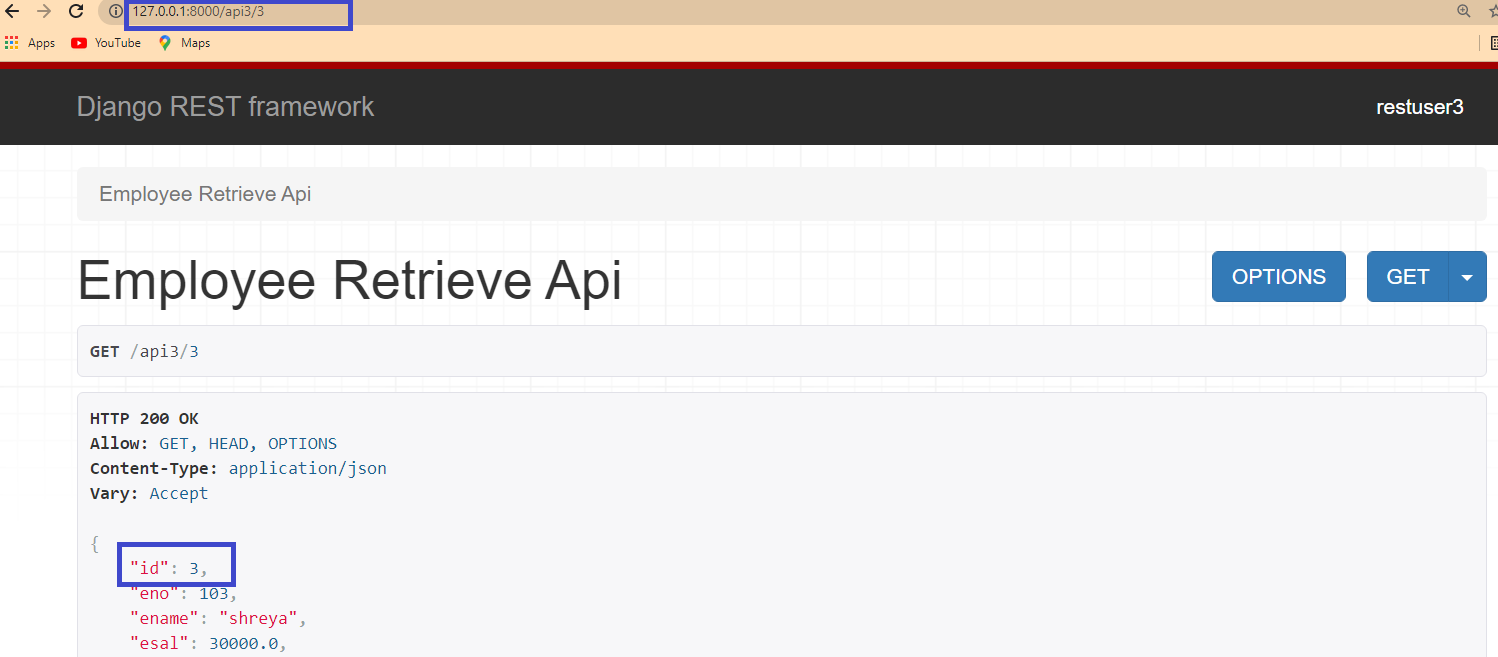
**url('admin/', admin.site.urls),**

**url('api1/',EmployeeListApiView.as\_view()),**

**url('api2/',EmployeeCreateApiView.as\_view()),**

**url('api3/(?P<pk>\d+)',EmployeeRetrieveApiView.as\_view())**

**]**

****

**To replace,pk by id,give the line lookup\_field='id' in views**

**class EmployeeRetrieveApiView(RetrieveAPIView):**

**queryset=Employee.objects.all()**

**serializer\_class=EmployeeSerializer**

**lookup\_field='id'**

**Update operation🡺UpdateAPIView**

**views.py**

**from rest\_framework.generics import UpdateAPIView**

**……**

**class EmployeeUpdateApiView(UpdateAPIView):**

**queryset=Employee.objects.all()**

**serializer\_class=EmployeeSerializer**

**lookup\_field='id'**

**urls.py**

**from django.contrib import admin**

**from django.conf.urls import url**

**from myApp.views import \***

**urlpatterns = [**

**url('admin/', admin.site.urls),**

**url('api1/',EmployeeListApiView.as\_view()),**

**url('api2/',EmployeeCreateApiView.as\_view()),**

**url('api3/(?P<id>\d+)',EmployeeRetrieveApiView.as\_view()),**

**url('api4/(?P<id>\d+)',EmployeeUpdateApiView.as\_view())**

**]**

**DestroyAPIView**

**views.py**

**from rest\_framework.generics import DestroyAPIView**

**class EmployeeDeleteApiView(DestroyAPIView):**

**queryset=Employee.objects.all()**

**serializer\_class=EmployeeSerializer**

**lookup\_field='id'**

**urls.py**

**from django.contrib import admin**

**from django.conf.urls import url**

**from myApp.views import \***

**urlpatterns = [**

**…….,**

**url('api5/(?P<id>\d+)',EmployeeDeleteApiView.as\_view())**

**]**

**Implement both List And Create Simultaneously**

**Use ListCreateAPIView**

**views.py**

**from rest\_framework.generics import ListCreateAPIView**

**class EmployeeListCreateApiView(ListCreateAPIView):**

**queryset=Employee.objects.all()**

**serializer\_class=EmployeeSerializer**

**urls.py**

**from django.contrib import admin**

**from django.conf.urls import url**

**from myApp.views import \***

**urlpatterns = [**

**url('admin/', admin.site.urls),**

**url('api/',EmployeeListCreateApiView.as\_view()),**

**]**

**Implement both Retrieve and update operations**

**views.py**

**from django.shortcuts import render**

**from myApp.models import Employee**

**from myApp.serializers import EmployeeSerializer**

**from rest\_framework.generics import RetrieveUpdateAPIView**

**class EmployeeRetrieveUpdateApiView(RetrieveUpdateAPIView):**

**queryset=Employee.objects.all()**

**serializer\_class=EmployeeSerializer**

**lookup\_field='id'**

**urls.py**

**from django.contrib import admin**

**from django.conf.urls import url**

**from myApp.views import EmployeeRetrieveUpdateApiView**

**urlpatterns = [**

**url('admin/', admin.site.urls),**

**url('api/(?P<id>\d+)',EmployeeRetrieveUpdateApiView.as\_view())**

**]**

**Implement Both Retrieve and Delete operations**

**views.py**

**from django.shortcuts import render**

**from myApp.models import Employee**

**from myApp.serializers import EmployeeSerializer**

**from rest\_framework.generics import RetrieveDestroyAPIView**

**class EmployeeRetrieveDeleteApiView(RetrieveDestroyAPIView):**

**queryset=Employee.objects.all()**

**serializer\_class=EmployeeSerializer**

**lookup\_field='id'**

**from django.contrib import admin**

**from django.conf.urls import url**

**from myApp.views import EmployeeRetrieveDeleteApiView**

**urlpatterns = [**

**url('admin/', admin.site.urls),**

**url('api/(?P<id>\d+)',EmployeeRetrieveDeleteApiView.as\_view())**

**]**

**Perfrom Retrieve update delete operations simultaneously**

**views.py**

**from django.shortcuts import render**

**from myApp.models import Employee**

**from myApp.serializers import EmployeeSerializer**

**from rest\_framework.generics import RetrieveUpdateDestroyAPIView**

**class EmployeeRetrieveUpdateDeleteApiView(RetrieveUpdateDestroyAPIView):**

**queryset=Employee.objects.all()**

**serializer\_class=EmployeeSerializer**

**lookup\_field='id'**

**urls.py**

**from django.contrib import admin**

**from django.conf.urls import url**

**from myApp.views import EmployeeRetrieveUpdateDeleteApiView**

**urlpatterns = [**

**url('admin/', admin.site.urls),**

**url('api/(?P<id>\d+)',EmployeeRetrieveUpdateDeleteApiView.as\_view())**

**]**

**Implementing all CRUD operations using 2 endpoints**

**1.EmployeeListCreateAPIView**

**2.EmployeeRetrieveUpdateDestroyAPIView**

**views.py**

**from django.shortcuts import render**

**from myApp.models import Employee**

**from myApp.serializers import EmployeeSerializer**

**from rest\_framework.generics import ListCreateAPIView,RetrieveUpdateDestroyAPIView**

**class EmployeeListCreateApiView(ListCreateAPIView):**

**queryset=Employee.objects.all()**

**serializer\_class=EmployeeSerializer**

**class EmployeeRetrieveUpdateDestroyApiView(RetrieveUpdateDestroyAPIView):**

**queryset=Employee.objects.all()**

**serializer\_class=EmployeeSerializer**

**lookup\_field='id'**

**urls.py**

**from django.contrib import admin**

**from django.conf.urls import url**

**from myApp.views import \***

**urlpatterns = [**

**url('admin/', admin.site.urls),**

**url('api/$',EmployeeListCreateApiView.as\_view()),**

**url('api/(?P<id>\d+)/$',EmployeeRetrieveUpdateDestroyApiView.as\_view())**

**]**

**DRF VIEWS**

**1.APIView**

**2.ListAPIView**

**3.CreateAPIView**

**4.RetrieveAPIView**

**5.UpdateAPIView**

**6.DestroyAPIView**

**7.ListCreateAPIView**

**8.RetrieveUpdateAPIView**

**9.RetrieveUpdateDestroyAPIView**

**10.RetrieveDestroyAPIView**

**MIXINS**

**1.Normal classes meant for reausability**

**2.Direct child classes of object**

**3.Contain only the methods**

**rest\_framework.mixins🡪**

**ListModelMixin🡪 list(resquest,\*args,\*\*kwargs)**

**CreateModelMixin🡪create(resquest,\*args,\*\*kwargs)**

**UpdateModelMixin🡪 update(resquest,\*args,\*\*kwargs)**

**🡪partial\_update (resquest,\*args,\*\*kwargs)**

**RetrieveModelMixin🡪 retrieve(resquest,\*args,\*\*kwargs)**

**DestroyModelMixin🡪 destroy(resquest,\*args,\*\*kwargs)**

**Ex1**

**views.py**

**from rest\_framework import mixins**

**from rest\_framework.generics import ListAPIView**

**from django.shortcuts import render**

**from myApp.models import Employee**

**from myApp.serializers import EmployeeSerializer**

**class EmployeeListModelMixin(ListAPIView):**

**queryset=Employee.objects.all()**

**serializer\_class=EmployeeSerializer**

**urls.py**

**from django.contrib import admin**

**from django.conf.urls import url**

**from myApp.views import EmployeeListModelMixin**

**urlpatterns = [**

**url('admin/', admin.site.urls),**

**url('api/$',EmployeeListModelMixin.as\_view())**

**]**

**views.py**

**from rest\_framework.mixins import \***

**from rest\_framework.generics import \***

**from django.shortcuts import render**

**from myApp.models import Employee**

**from myApp.serializers import EmployeeSerializer**

**class EmployeeListCreateModelMixin(CreateModelMixin,ListAPIView):**

**queryset=Employee.objects.all()**

**serializer\_class=EmployeeSerializer**

**def post(self,request,\*args,\*\*kwargs):**

**return self.create(request,\*args,\*\*kwargs)**

**urls.py**

**from django.contrib import admin**

**from django.conf.urls import url**

**from myApp.views import \***

**urlpatterns = [**

**url('admin/', admin.site.urls),**

**url('api/$',EmployeeListCreateModelMixin.as\_view())**

**]**