**REST API (Django Rest Framework-DRF)**

**Session-11**

**Generic API View and mixins:**

* Generic API views are set of commonly used patterns.
* The purpose of Generic API view is to quickly build API views that map closely to our database models without repeating the code.
* Generic API view is more loaded version of APIView
* **mixin:** a mixin is a class which contains a combination of methods from other classes.
* Mixins provides bits of common behavior; they cannot be used standalone that means we cannot use mixins without Generic API view.
* Mixins must pair with Gneric API view to make a functional view.

Types of mixin:

* CreateModelMixin: Create a model instance
* ListModelMixin: List a queryset
* RetrieveModelMixin : Retrieve a model instance
* UpdateModelMixin: Update a model instance
* DestroyModelMixin: Delete a model instance

**Settings.py:**

INSTALLED\_APPS = [  
 **'django.contrib.admin'**,  
 **'django.contrib.auth'**,  
 **'django.contrib.contenttypes'**,  
 **'django.contrib.sessions'**,  
 **'django.contrib.messages'**,  
 **'django.contrib.staticfiles'**,  
 **'myapp.apps.MyappConfig'**,  
 **'rest\_framework'**,  
  
 **'myapp4'**,  
]

**Models.py:**

**from** django.db **import** models  
  
*# Create your models here.***class** Customer(models.Model):  
 name=models.CharField(max\_length=20)  
 address=models.CharField(max\_length=20)  
 mail = models.CharField(max\_length=20)  
 age = models.IntegerField()

**admin.py:**

**from** django.contrib **import** admin  
**from** myapp4.models **import** Customer  
*# Register your models here.*@admin.register(Customer)  
**class** CustomerAdmin(admin.ModelAdmin):  
 list\_display = [**'id'**,**'name'**,**'address'**,**'mail'**,**'age'**]

**Now go to terminal and then type the following commands**

* Python manage.py makemigrations
* Python manage.py migrate
* Python manage.py createsuperuser
* Now run server: python manage.py runserver
* Now go to browser: <http://127.0.0.1:8000/admin>
* Insert few records

Create a new python file with the name serializers.py.

**serializer.py:**

**from** rest\_framework **import** serializers  
**from** myapp4.models **import** Customer  
  
**class** CustomerSerializer(serializers.ModelSerializer):  
 **class** Meta:  
 model=Customer  
 fields=[**'id'**,**'name'**,**'address'**,**'mail'**,**'age'**]

**views.py:**

**from** django.shortcuts **import** render  
**from** myapp4.serializers **import** CustomerSerializer  
**from** myapp4.models **import** Customer  
**from** rest\_framework.generics **import** GenericAPIView  
**from** rest\_framework.mixins **import** ListModelMixin,CreateModelMixin,RetrieveModelMixin,UpdateModelMixin,DestroyModelMixin  
  
*# Create your views here.***class** CustomerList(GenericAPIView,ListModelMixin):  
 queryset = Customer.objects.all()  
 serializer\_class =CustomerSerializer  
  
 **def** get(self,request,\*args,\*\*kwargs):  
 **return** self.list(request,\*args,\*\*kwargs)  
  
**class** CustomerCreate(GenericAPIView,CreateModelMixin):  
 queryset = Customer.objects.all()  
 serializer\_class =CustomerSerializer  
  
 **def** post(self,request,\*args,\*\*kwargs):  
 **return** self.create(request,\*args,\*\*kwargs)  
  
  
**class** CustomerRetrieve(GenericAPIView,RetrieveModelMixin):  
 queryset = Customer.objects.all()  
 serializer\_class =CustomerSerializer  
  
 **def** get(self,request,\*args,\*\*kwargs):  
 **return** self.retrieve(request,\*args,\*\*kwargs)  
  
**class** CustomerUpdate(GenericAPIView,UpdateModelMixin):  
 queryset = Customer.objects.all()  
 serializer\_class =CustomerSerializer  
  
 **def** put(self,request,\*args,\*\*kwargs):  
 **return** self.update(request,\*args,\*\*kwargs)  
  
  
**class** CustomerDelete(GenericAPIView,DestroyModelMixin):  
 queryset = Customer.objects.all()  
 serializer\_class =CustomerSerializer  
  
 **def** delete(self,request,\*args,\*\*kwargs):  
 **return** self.destroy(request,\*args,\*\*kwargs)  
  
  
*#Retrieve,Update,Delete***class** Retrive\_Update\_Delete(GenericAPIView,RetrieveModelMixin,UpdateModelMixin,DestroyModelMixin):  
 queryset = Customer.objects.all()  
 serializer\_class = CustomerSerializer  
  
 **def** get(self, request, \*args, \*\*kwargs):  
 **return** self.retrieve(request, \*args, \*\*kwargs)  
  
 **def** put(self,request,\*args,\*\*kwargs):  
 **return** self.update(request,\*args,\*\*kwargs)  
  
 **def** delete(self, request, \*args, \*\*kwargs):  
 **return** self.destroy(request, \*args, \*\*kwargs)  
  
*#List ,Create***class** List\_Create(GenericAPIView,ListModelMixin, CreateModelMixin):  
 queryset = Customer.objects.all()  
 serializer\_class = CustomerSerializer  
  
 **def** get(self, request, \*args, \*\*kwargs):  
 **return** self.list(request, \*args, \*\*kwargs)  
  
 **def** post(self, request, \*args, \*\*kwargs):  
 **return** self.create(request, \*args, \*\*kwargs)

**urls.py :**

**from** django.contrib **import** admin  
**from** django.urls **import** path  
**from** myapp4 **import** views  
  
urlpatterns = [  
 path(**'admin/'**, admin.site.urls),  
 *#path('customerlist/',views.CustomerList.as\_view()),  
 #path('customercreate/',views.CustomerCreate.as\_view()),  
 #path('customerretrieve/<int:pk>', views.CustomerRetrieve.as\_view()),  
 #path('customerupdate/<int:pk>', views.CustomerUpdate.as\_view()),  
 #path('customerdelete/<int:pk>', views.CustomerDelete.as\_view()),* path(**'RUD/<int:pk>'**,views.Retrive\_Update\_Delete.as\_view()),  
 path(**'LC/'**,views.List\_Create.as\_view()),  
  
]