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

**Session-13**

**Working with ViewSet:**

* Django REST framework allows you to combine the logic for a set of related views in a single class, called a ViewSet.
* In other frameworks you may also find conceptually similar implementations named something like 'Resources' or 'Controllers'.
* A ViewSet class is simply a type of class-based View, that does not provide any method handlers such as .get() or .post(), and instead provides actions such as .list() and .create().
* The method handlers for a ViewSet are only bound to the corresponding actions at the point of finalizing the view, using the .as\_view() method.
* Typically, rather than explicitly registering the views in a viewset in the urlconf, you'll register the viewset with a router class, that automatically determines the urlconf for you.
* Routers are used with ViewSets in django rest framework to auto config the urls.
* Routers provides a simple, quick and consistent way of wiring ViewSet logic to a set of URLs.
* Router automatically maps the incoming request to proper viewset action based on the request method type(i.e GET, POST, etc).

**ViewSet attributes:**

**basename** :

* The base to use for the URL names that are created.
* If unset the basename will be automatically generated based on the queryset attribute of the viewset.
* Note that if the viewset does not include a queryset attribute then you must set basename when registering the viewset.

**action** :

* The name of the current action(list, create)

**detail:**

* Return Boolean, indicates that the current action is configured for a list or detail view.

**name** :

* The display name for the view set.

**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'**,  
  
 **'myapp6'**,  
]

**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** myapp6.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** myapp6.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** rest\_framework.response **import** Response  
**from** myapp6.models **import** Customer  
**from** myapp6.serializers **import** CustomerSerializer  
**from** rest\_framework **import** status  
**from** rest\_framework **import** viewsets  
  
*# Create your views here.***class** CustomerViewset(viewsets.ViewSet):  
 **def** list(self,request):  
 print(**"Basename:"**,self.basename)  
 print(**"Action:"**,self.action)  
 print(**"Detail:"**,self.detail)  
 print(**"Name:"**,self.name)  
 cust=Customer.objects.all()  
 serializer=CustomerSerializer(cust,many=**True**)  
 **return** Response(serializer.data)  
  
 **def** retrieve(selfself,request,pk=**None**):  
 id=pk  
 **if** id **is not None**:  
 cust=Customer.objects.get(id=id)  
 serializer=CustomerSerializer(cust)  
 **return** Response(serializer.data)  
  
 **def** create(self,request):  
 serializer=CustomerSerializer(data=request.data)  
 **if** serializer.is\_valid():  
 serializer.save()  
 **return** Response({**"message"**:**"Data inserted"**},status=status.HTTP\_201\_CREATED)  
 **return** Response(serializer.errors,status=status.HTTP\_400\_BAD\_REQUEST)  
  
 **def** update(self,request,pk):  
 id=pk  
 cust=Customer.objects.get(pk=id)  
 serializer = CustomerSerializer(cust,data=request.data)  
 **if** serializer.is\_valid():  
 serializer.save()  
 **return** Response({**"message"**:**"Data updated"**})  
 **return** Response(serializer.errors, status=status.HTTP\_400\_BAD\_REQUEST)  
  
 **def** partial\_update(self,request,pk):  
 id=pk  
 cust=Customer.objects.get(pk=id)  
 serializer = CustomerSerializer(cust,data=request.data,partial=**True**)  
 **if** serializer.is\_valid():  
 serializer.save()  
 **return** Response({**"message"**:**"Data is partially updated"**})  
 **return** Response(serializer.errors, status=status.HTTP\_400\_BAD\_REQUEST)  
  
 **def** destroy(selfself,request,pk):  
 id=pk  
 cust=Customer.objects.get(pk=id)  
 cust.delete()  
 **return** Response({**"message"**:**"Record deleted"**})

**urls.py :**

**from** django.contrib **import** admin  
**from** django.urls **import** path,include  
**from** myapp6 **import** views  
**from** rest\_framework.routers **import** DefaultRouter  
  
*#create a router object*router=DefaultRouter()  
  
*#register your viewset with router*router.register(**'CustomerViewset'**,views.CustomerViewset,basename=**'customer'**)  
  
urlpatterns = [  
 path(**'admin/'**, admin.site.urls),  
 path(**''**,include(router.urls)),  
  
]