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

**Session-10**

**Ex with CRUD api\_view using function based view:**

**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'**,  
  
 **'myapp2'**,  
]

**Models.py:**

**from** django.db **import** models  
  
*# Create your models here.***class** Trainer(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** myapp2.models **import** Trainer  
*# Register your models here.*@admin.register(Trainer)  
**class** TrainerAdmin(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** myapp2.models **import** Trainer  
  
**class** TrainerSerializer(serializers.ModelSerializer):  
 **class** Meta:  
 model=Trainer  
 fields=[**'id'**,**'name'**,**'address'**,**'mail'**,**'age'**]

**views.py:**

**from** django.shortcuts **import** render  
**from** rest\_framework.decorators **import** api\_view  
**from** rest\_framework.response **import** Response  
**from** myapp2.models **import** Trainer  
**from** myapp2.serializers **import** TrainerSerializer  
  
*# Create your views here.*@api\_view([**'GET'**,**'POST'**,**'PUT'**,**'DELETE'**])  
**def** trainer\_api(request):  
 **if** request.method == **'GET'**:  
 id = request.data.get(**'id'**)  
 **if** id **is not None**:  
 tr=Trainer.objects.get(id=id)  
 serializer=TrainerSerializer(tr)  
 **return** Response(serializer.data)  
 tr=Trainer.objects.all()  
 serializer=TrainerSerializer(tr,many=**True**)  
 **return** Response(serializer.data)  
  
 **if** request.method==**'POST'**:  
 serializer=TrainerSerializer(data=request.data)  
 **if** serializer.is\_valid():  
 serializer.save()  
 **return** Response({**"message"**:**"Data inserted"**})  
 **return** Response(serializer.errors)  
  
 **if** request.method==**'PUT'**:  
 id=request.data.get(**'id'**)  
 tr=Trainer.objects.get(pk=id)  
 serializer=TrainerSerializer(tr,data=request.data,partial=**True**)  
 **if** serializer.is\_valid():  
 serializer.save()  
 **return** Response({**"message"**:**"Data updated"**})  
 **return** Response(serializer.errors)  
  
 **if** request.method==**'DELETE'**:  
 id=request.data.get(**'id'**)  
 tr=Trainer.objects.get(pk=id)  
 tr.delete()  
 **return** Response({**"message"**:**"Record deleted"**})

**urls.py :**

**from** django.contrib **import** admin  
**from** django.urls **import** path  
**from** myapp2 **import** views  
  
urlpatterns = [  
 path(**'admin/'**, admin.site.urls),  
 path(**'trainer/'**,views.trainer\_api),  
]

Create a new python file with the name test.py inside the application.

**test.py:**

**import** requests  
**import** json  
  
URL=**" http://127.0.0.1:8000/trainer/"  
  
def** get\_record(id=**None**):  
 data={}  
 **if** id **is not None**:  
 data={**'id'**:id}  
 jsondata=json.dumps(data)  
  
 headers={**'content-Type'**:**'application/json'**}  
 r=requests.get(url=URL,headers=headers,data=jsondata)  
 data=r.json()  
 print(data)  
  
*#get\_record(1)  
#get\_record()***def** post\_record():  
 data = {  
 **'name'**: **'sairam'**,  
 **'address'**: **'hyd'**,  
 **'mail'**: **'sairam@gmail.com'**,  
 **'age'**: 26  
 }  
 jsondata=json.dumps(data)  
 headers = {**'content-Type'**: **'application/json'**}  
 r=requests.post(url=URL,headers=headers,data=jsondata)  
 data=r.json()  
 print(data)  
  
*#post\_record()***def** update\_record():  
 data = {  
 **'id'**:1,  
 **'name'**: **'mohan'**,  
 **'address'**: **'srnagar'**,  
 **'mail'**:**'mohan@gmail.com'**,  
 **'age'**: 30  
 }  
 jsondata=json.dumps(data)  
 headers = {**'content-Type'**: **'application/json'**}  
 r=requests.put(url=URL,headers=headers,data=jsondata)  
 data=r.json()  
 print(data)  
  
  
*#update\_record()***def** delete\_data():  
 data={**'id'**:4}  
  
 jsondata=json.dumps(data)  
 headers = {**'content-Type'**: **'application/json'**}  
 r=requests.delete(url=URL,headers=headers,data=jsondata)  
 data=r.json()  
 print(data)  
  
delete\_data()

**Ex with CRUD APIView** **using class based view with browser API testing:**

* REST framework provides an APIView class, which subclasses Django's View class.
* APIView classes are different from regular View classes in the following ways: Requests passed to the handler methods will be REST framework's Request instances, not Django's HttpRequest instances.

**Models.py :**

**from** django.db **import** models  
  
*# Create your models here.***class** Manager(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** myapp3.models **import** Manager  
*# Register your models here.*@admin.register(Manager)  
**class** ManagerAdmin(admin.ModelAdmin):  
 list\_display = [**'id'**,**'name'**,**'address'**,**'mail'**,**'age'**]

**serializer.py :**

**from** rest\_framework **import** serializers  
**from** myapp3.models **import** Manager  
  
**class** ManagerSerializer(serializers.ModelSerializer):  
 **class** Meta:  
 model=Manager  
 fields=[**'id'**,**'name'**,**'address'**,**'mail'**,**'age'**]

**views.py :**

**from** django.shortcuts **import** render  
**from** rest\_framework.views **import** APIView  
**from** myapp3.serializer **import** ManagerSerializer  
**from** rest\_framework.response **import** Response  
**from** rest\_framework **import** status  
**from** myapp3.models **import** Manager  
*# Create your views here.***class** ManagerAPI(APIView):  
 **def** get(self,request,pk=**None**,format=**None**):  
 id=pk  
 **if** id **is not None**:  
 m=Manager.objects.get(id=id)  
 serializer=ManagerSerializer(m)  
 **return** Response(serializer.data)  
 m=Manager.objects.all()  
 serializer=ManagerSerializer(m,many=**True**)  
 **return** Response(serializer.data)  
  
 **def** post(self,request,format=**None**):  
 serializer=ManagerSerializer(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** put(self,request,pk,format=**None**):  
 id=pk  
 m=Manager.objects.get(pk=id)  
 serializer=ManagerSerializer(m,data=request.data)  
 **if** serializer.is\_valid():  
 serializer.save()  
 **return** Response({**'message'**:**"Data is updated"**})  
 **return** Response(serializer.errors,status=status.HTTP\_400\_BAD\_REQUEST)  
  
 **def** patch(self,request,pk,format=**None**):  
 id=pk  
 m=Manager.objects.get(pk=id)  
 serializer=ManagerSerializer(m,data=request.data,partial=**True**)  
 **if** serializer.is\_valid():  
 serializer.save()  
 **return** Response({**'message'**:**"partiallly updated"**})  
 **return** Response(serializer.errors,status=status.HTTP\_400\_BAD\_REQUEST)  
  
 **def** delete(self,request,pk,format=**None**):  
 id=pk  
 m=Manager.objects.get(pk=id)  
 m.delete()  
 **return** Response({**'message'**:**"Record is deleted"**})

**urls.py :**

**from** django.contrib **import** admin  
**from** django.urls **import** path  
**from** myapp3 **import** views  
  
urlpatterns = [  
 path(**'admin/'**, admin.site.urls),  
 path(**'manager/'**,views.ManagerAPI.as\_view()),  
 path(**'manager/<int:pk>'**,views.ManagerAPI.as\_view()),  
]

Note: to test this, go to browser and then type

http://127.0.0.1:8000/manager/

