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

**Session-8**

**DRF Model Serializer Class:**

* Model serializer class will provide automatic serialize class with fields that are related to model fields.
* Model serializer class is just model based serializer class.
* It will create automatically set of fields based on the model.
* It will generate validators automatically.
* It will provide default implementation of create and update methods.

**Syntax:**

from rest\_framework import serializers

class StudentSerialzer (serializers.ModelSerializer):

class Meta:

model=Student

fields=['id','name','address','mail','age']

#fields='\_\_all\_\_'

**Create a new project and application**

**Settings.py:**

INSTALLED\_APPS = [  
 **'django.contrib.admin'**,  
 **'django.contrib.auth'**,  
 **'django.contrib.contenttypes'**,  
 **'django.contrib.sessions'**,  
 **'django.contrib.messages'**,  
 **'django.contrib.staticfiles'**,  
 **'myapp1'**,  
 **'rest\_framework'**,  
]

**models.py :**

**from** django.db **import** models  
  
*# Create your models here.***class** Student(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** myapp1.models **import** Student  
*# Register your models here.*@admin.register(Student)  
**class** StudentAdmin(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

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

**serializers.py:**

**from** rest\_framework **import** serializers  
**from** myapp1.models **import** Student  
  
**class** StudentSerializer(serializers.ModelSerializer):  
 *# validators* **def** starts\_with\_s(value):  
 **if** value[0].lower() != **'s'**:  
 **raise** serializers.ValidationError(**'Name should starts with letter s'**)  
 *#address=serializers.CharField(read\_only=True)* name=serializers.CharField(validators=[starts\_with\_s])  
 **class** Meta:  
 model=Student  
 *#fields=['name','address','mail','age']* fields=**'\_\_all\_\_'** *#read\_only\_fields=['address','age']*

**views.py:**

**from** django.shortcuts **import** render  
**import** io  
**from** rest\_framework.parsers **import** JSONParser  
**from** myapp1.serializers **import** StudentSerializer  
**from** myapp1.models **import** Student  
**from** rest\_framework.renderers **import** JSONRenderer  
**from** django.http **import** HttpResponse  
**from** django.views.decorators.csrf **import** csrf\_exempt  
**from** django.utils.decorators **import** method\_decorator  
**from** django.views **import** View  
  
*# Create your views here.*@method\_decorator(csrf\_exempt,name=**'dispatch'**)  
**class** studentdata(View):  
 **def** get(self,request,\*args,\*\*kwargs):  
 **if** request.method == **'GET'**:  
 jsondata = request.body  
 stream = io.BytesIO(jsondata)  
 py\_data = JSONParser().parse(stream)  
 id = py\_data.get(**'id'**, **None**)  
 **if** id **is not None**:  
 emp = Student.objects.get(id=id)  
 serializer = StudentSerializer(emp)  
 jsondata = JSONRenderer().render(serializer.data)  
 **return** HttpResponse(jsondata, content\_type=**'application/json'**)  
 emp = Student.objects.all()  
 serializer = StudentSerializer(emp, many=**True**)  
 jsondata = JSONRenderer().render(serializer.data)  
 **return** HttpResponse(jsondata, content\_type=**'application/json'**)  
  
 **def** post(self, request, \*args, \*\*kwargs):  
 **if** request.method == **'POST'**:  
 jsondata = request.body  
 stream = io.BytesIO(jsondata)  
 py\_data = JSONParser().parse(stream)  
 serializer = StudentSerializer(data=py\_data)  
 **if** serializer.is\_valid():  
 serializer.save()  
 result = {**'message'**: **'Data inserted into database'**}  
 jsondata = JSONRenderer().render(result)  
 **return** HttpResponse(jsondata, content\_type=**'application/json'**)  
 jsondata = JSONRenderer().render(serializer.errors)  
 **return** HttpResponse(jsondata, content\_type=**'application/json'**)  
  
 **def** put(self, request, \*args, \*\*kwargs):  
 **if** request.method == **'PUT'**:  
 jsondata = request.body  
 stream = io.BytesIO(jsondata)  
 py\_data = JSONParser().parse(stream)  
 id = py\_data.get(**'id'**)  
 emp = Student.objects.get(id=id)  
 *# serializer=EmployeeSerialzer(emp,data=py\_data,partial=True)* serializer = StudentSerializer(emp, data=py\_data)  
 **if** serializer.is\_valid():  
 serializer.save()  
 result = {**'message'**: **'Data updated into database'**}  
 jsondata = JSONRenderer().render(result)  
 **return** HttpResponse(jsondata, content\_type=**'application/json'**)  
 jsondata = JSONRenderer().render(serializer.errors)  
 **return** HttpResponse(jsondata, content\_type=**'application/json'**)  
  
 **def** delete(self, request, \*args, \*\*kwargs):  
 **if** request.method == **'DELETE'**:  
 jsondata = request.body  
 stream = io.BytesIO(jsondata)  
 py\_data = JSONParser().parse(stream)  
 id = py\_data.get(**'id'**)  
 emp = Student.objects.get(id=id)  
 emp.delete()  
 result = {**'message'**: **'Data deleted from database'**}  
 jsondata = JSONRenderer().render(result)  
 **return** HttpResponse(jsondata, content\_type=**'application/json'**)

**urls.py:**

**from** django.contrib **import** admin  
**from** django.urls **import** path  
**from** myapp1 **import** views  
  
urlpatterns = [  
 path(**'admin/'**, admin.site.urls),  
 path(**'student/'**,views.studentdata.as\_view()),  
  
]

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

**api.py:**

**import** requests  
**import** json  
  
URL=**"http://127.0.0.1:8000/student/"  
  
def** get\_record(id=**None**):  
 data={}  
 **if** id **is not None**:  
 data={**'id'**:id}  
 jsondata=json.dumps(data)  
 r=requests.get(url=URL,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)  
 r=requests.post(url=URL,data=jsondata)  
 data=r.json()  
 print(data)  
  
post\_record()  
  
**def** update\_record():  
 data = {  
 **'id'**:1,  
 **'name'**: **'durga'**,  
 **'address'**: **'hyd'**,  
 **'mail'**:**'durga@gmail.com'**,  
 **'age'**: 45  
 }  
 jsondata=json.dumps(data)  
 r=requests.put(url=URL,data=jsondata)  
 data=r.json()  
 print(data)  
  
  
*#update\_record()***def** delete\_data():  
 data={**'id'**:4}  
  
 jsondata=json.dumps(data)  
 r=requests.delete(url=URL,data=jsondata)  
 data=r.json()  
 print(data)  
  
*#delete\_data()*

* Now start server : python manage.py runserver
* Python api.py( in new terminal window)