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

**Session-7**

**CRUD API using Class Based View:**

* **dumps ( ):** It is used to convert python object JSON string.
* **loads ( ):** It is used to pars JSON string.

**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'**,  
 **'myapp.apps.MyappConfig'**,  
 **'rest\_framework'**,  
]

**models.py :**

**from** django.db **import** models  
  
*# Create your models here.***class** Employee(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** myapp.models **import** Employee  
*# Register your models here.*@admin.register(Employee)  
**class** EmployeeAdmin(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** myapp.models **import** Employee  
  
**class** EmployeeSerialzer(serializers.Serializer):  
 name=serializers.CharField(max\_length=20)  
 address=serializers.CharField(max\_length=20)  
 mail=serializers.CharField(max\_length=20)  
 age=serializers.IntegerField()  
  
 **def** create(self, validated\_data):  
 **return** Employee.objects.create(\*\*validated\_data)  
  
 **def** update(self, instance, validated\_data):  
 instance.name=validated\_data.get(**'name'**,instance.name)  
 instance.address = validated\_data.get(**'address'**, instance.address)  
 instance.mail = validated\_data.get(**'mail'**, instance.mail)  
 instance.age = validated\_data.get(**'age'**, instance.age)  
 instance.save()  
 **return** instance

**views.py:**

**from** django.shortcuts **import** render  
**import** io  
**from** rest\_framework.parsers **import** JSONParser  
**from** myapp.serializers **import** EmployeeSerialzer  
**from** myapp.models **import** Employee  
**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** empdata(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 = Employee.objects.get(id=id)  
 serializer = EmployeeSerialzer(emp)  
 jsondata = JSONRenderer().render(serializer.data)  
 **return** HttpResponse(jsondata, content\_type=**'application/json'**)  
 emp = Employee.objects.all()  
 serializer = EmployeeSerialzer(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 = EmployeeSerialzer(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 = Employee.objects.get(id=id)  
 *# serializer=EmployeeSerialzer(emp,data=py\_data,partial=True)* serializer = EmployeeSerialzer(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 = Employee.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** myapp **import** views  
  
urlpatterns = [  
 path(**'admin/'**, admin.site.urls),  
 path(**'emp/'**,views.empdata.as\_view()),  
  
]

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/emp/"  
  
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'**: **'kiran'**,  
 **'address'**: **'vizag'**,  
 **'mail'**: **'kiran@gmail.com'**,  
 **'age'**: 28  
 }  
 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'**: **'ramesh'**,  
 **'address'**: **'vizag'**,  
 **'mail'**:**'ramesh@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'**:2}  
  
 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 test.py( in new terminal window)

**DRF Validations:**

* Validation is the process of checking whether user entered data is correct or not.
* We can perform validation in 3 different ways

1. **Field level validation**
2. **Object level validation**
3. **Validators**

**Field level validation:** This can be used to validate specific field

Syntax: def validate\_fieldname (self, value):

Here value is the field value which requires the validation.

**Object level validation:** This can be used to perform validation on all fields or multiple fields.

Syntax: def validate (self, data):

Here data is python dictionary of field values.

**Validators:** This can be used to create a validation function with logic that can be reuse in the application.

**Ex:**

**Note:** I recommend executing the above example as it is just by changing

the serializers.py file code only.

**Serializers.py:**

**from** rest\_framework **import** serializers  
**from** myapp.models **import** Employee  
  
*#validators***def** starts\_with\_s(value):  
 **if** value[0].lower()!=**'s'**:  
 **raise** serializers.ValidationError(**'Name should starts with letter s'**)  
  
**class** EmployeeSerialzer(serializers.Serializer):  
 name=serializers.CharField(max\_length=20,validators=[starts\_with\_s])  
 address=serializers.CharField(max\_length=20)  
 mail=serializers.CharField(max\_length=20)  
 age=serializers.IntegerField()  
  
 **def** create(self, validated\_data):  
 **return** Employee.objects.create(\*\*validated\_data)  
  
 **def** update(self, instance, validated\_data):  
 instance.name=validated\_data.get(**'name'**,instance.name)  
 instance.address = validated\_data.get(**'address'**, instance.address)  
 instance.mail = validated\_data.get(**'mail'**, instance.mail)  
 instance.age = validated\_data.get(**'age'**, instance.age)  
 instance.save()  
 **return** instance  
  
 *#field level validation* **def** validate\_age(self,value):  
 **if** value>100:  
 **raise** serializers.ValidationError(**"Age should not exceed 100"**)  
 **return** value  
  
 *#object level validation* **def** validate(self, data):  
 name=data.get(**'name'**)  
 addr=data.get(**'address'**)  
 **if** name.lower()==**"durga" and** addr.lower()!=**'hyd'**:  
 **raise** serializers.ValidationError(**"address should be hyd"**)  
 **return** data