CRUD using web api without REST framework

Settings.py🡪Add the application

models.py

from django.db import models

class Employee(models.Model):

eno=models.IntegerField()

ename=models.CharField(max\_length=40)

esal=models.FloatField()

eaddr=models.CharField(max\_length=100)

def \_\_str\_\_(self):

return self.ename

admin.py

from django.contrib import admin

from myApp.models import Employee

class EmployeeAdmin(admin.ModelAdmin):

l=['id','eno','ename','esal','eaddr']

admin.site.register(Employee,EmployeeAdmin)

perform migration,create super user,run server and add records in database.

Retrieve operation:Get the information based on id

views.py

from django.shortcuts import render

from django.models import Employee

from myApp.models import Employee

from django.http import HttpResponse

import json

from django.views.generic import View

class EmployeeDetails(View):

def get(self,request,\*args,\*\*kwargs):

emp=Employee.objects.get(id=1)

emp\_data={'eno':emp.eno,'ename':emp.ename,'esal':emp.esal,'eaddr':emp.eaddr}

json\_data=json.dumps(emp\_data)

return HttpResponse(json\_data,content\_type="application/json")

urls.py

from django.contrib import admin

from django.urls import path

from myApp.views import EmployeeDetails

urlpatterns = [

path('admin/', admin.site.urls),

path('api/',EmployeeDetails.as\_view())

]

At project level create test.py

test.py

import requests

BASE\_URL='http://127.0.0.1:8000/'

ENDPOINT='api/'

def get\_resource():

resp=requests.get(BASE\_URL+ENDPOINT)

print(resp.status\_code)

print(resp.json())

get\_resource()

**Status Codes**

Represents status of response

1XX🡺100-199[Informational] Ex:Don’t close the browser

2XX🡺200-299[Successful]

3XX🡺300-399[Redirectional]

4XX🡺400-499[Page not found]

5XX🡺500-599[Server error]

**Getting id from user and display the details**

**views.py**

**from django.shortcuts import render**

**from myApp.models import Employee**

**from django.http import HttpResponse**

**from django.views.generic import View**

**import json**

**class EmployeeDetails(View):**

**def get(self,request,id,\*args,\*\*kwargs):**

**emp=Employee.objects.get(id=id)**

**emp\_data={'eno':emp.eno,'ename':emp.ename,'esal':emp.esal,'eaddr':emp.eaddr}**

**json\_data=json.dumps(emp\_data)**

**return HttpResponse(json\_data,content\_type="application/json")**

**urls.py**

**from django.contrib import admin**

**from django.conf.urls import url**

**from myApp.views import EmployeeDetails**

**urlpatterns = [**

**url('admin/', admin.site.urls),**

**url('api/(?P<id>\d+)/$',EmployeeDetails.as\_view())**

**]**

**test.py**

**import requests**

**BASE\_URL='http://127.0.0.1:8000/'**

**ENDPOINT='api/'**

**def get\_resource(id):**

**resp=requests.get(BASE\_URL+ENDPOINT+id+'/')**

**#http://127.0.0.1:8000/api/id/**

**print(resp.status\_code)**

**print(resp.json())**

**id=input("Enter id:")**

**get\_resource(id)**

**models.py**

**from django.db import models**

**class Employee(models.Model):**

**eno=models.IntegerField()**

**ename=models.CharField(max\_length=40)**

**esal=models.FloatField()**

**eaddr=models.CharField(max\_length=100)**

**def \_\_str\_\_(self):**

**return self.ename**

**admin.py**

**from django.contrib import admin**

**from myApp.models import Employee**

**class EmployeeAdmin(admin.ModelAdmin):**

**l=['id','eno','ename','esal','eaddr']**

**admin.site.register(Employee,EmployeeAdmin)**

**cmd prompt1🡪run django server**

**cmd prompt2🡪run python application**

**SERIALIZATION**

**Converting python dictionary to json obeject.Converting an employee record into a dictionary is easy.If it is query set,If we use Employee.objects.all()**

**Gives many records.Convert them into dictionary is very difficult.To overcome this problem,we use a module called serializers.**

**Serializers🡪django inbuilt method that contains a method called serialize**

**Using serialize we can convert query set into json very easily.**

**json\_data=serializers.serialize(‘json’,qs)**

**views.py**

**from django.shortcuts import render**

**from myApp.models import Employee**

**from django.http import HttpResponse**

**from django.views.generic import View**

**from django.core.serializers import serialize**

**import json**

**class EmployeeDetails(View):**

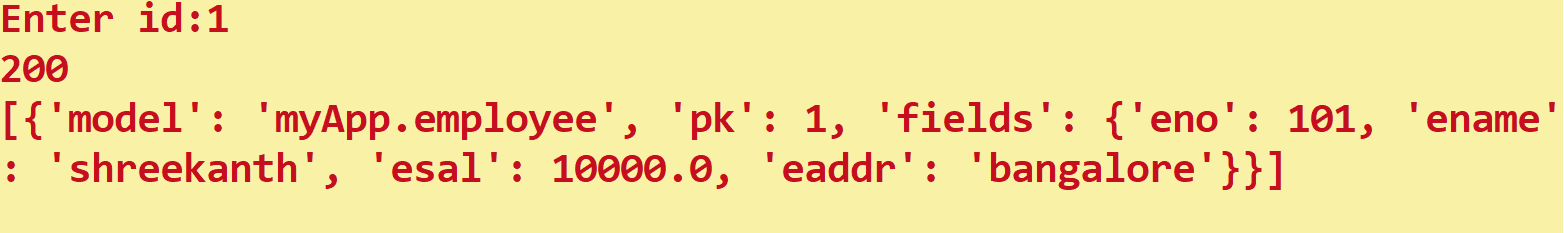
**def get(self,request,id,\*args,\*\*kwargs):**

**emp=Employee.objects.get(id=id)**

**json\_data=serialize('json',[emp]) #only one record**

**return HttpResponse(json\_data,content\_type="application/json")**

**py test.py**

****

**Extracting only the specified fields**

**from django.shortcuts import render**

**from myApp.models import Employee**

**from django.http import HttpResponse**

**from django.views.generic import View**

**from django.core.serializers import serialize**

**import json**

**class EmployeeDetails(View):**

**def get(self,request,id,\*args,\*\*kwargs):**

**emp=Employee.objects.get(id=id)**

**json\_data=serialize('json',[emp],fields=('eno','ename')) #only one record**

**return HttpResponse(json\_data,content\_type="application/json")**

**CRUD operation:1🡪Retrieve based on provided id**

**CRUD operation:2🡪Retrieve all records**

**views.py**

**from django.shortcuts import render**

**from myApp.models import Employee**

**from django.http import HttpResponse**

**from django.views.generic import View**

**from django.core.serializers import serialize**

**import json**

**class EmployeeDetails(View):**

**def get(self,request,\*args,\*\*kwargs):**

**emp=Employee.objects.all()**

**json\_data=serialize('json',emp) #only one record**

**return HttpResponse(json\_data,content\_type="application/json")**

**urls.py**

**from django.contrib import admin**

**from django.conf.urls import url**

**from myApp.views import EmployeeDetails**

**urlpatterns = [**

**url('admin/', admin.site.urls),**

**url('api/$',EmployeeDetails.as\_view())**

**]**

**test.py**

**import requests**

**BASE\_URL='http://127.0.0.1:8000/'**

**ENDPOINT='api/'**

**def get\_resource():**

**resp=requests.get(BASE\_URL+ENDPOINT)**

**#http://127.0.0.1:8000/api/id/**

**print(resp.status\_code)**

**print(resp.json())**

**get\_resource()**

**Removing Extra information**

**[{'model': 'myApp.employee', 'pk': 1, 'fields': {'eno': 101, 'ename': 'shreekanth', 'esal': 10000.0,……}]**

**Step 1:Convert json data into dict by using loads()**

**Step 2:For each object in dict,extract the required fields**

**json\_data=serialize(‘json’,qs)**

**pdict=json.loads(json\_data) # Step 1**

**result=[]#to hold python data**

**for ob in pdict:**

**emp\_data=ob[‘fields’] #step 2**

**result.append(emp\_data)**

**json\_data=json.dumps(result)**

**views.py**

**from django.shortcuts import render**

**from myApp.models import Employee**

**from django.http import HttpResponse**

**from django.views.generic import View**

**from django.core.serializers import serialize**

**import json**

**class EmployeeDetails(View):**

**def get(self,request,\*args,\*\*kwargs):**

**emp=Employee.objects.all()**

**json\_data=serialize('json',emp)**

**p\_dict=json.loads(json\_data)**

**result=[]**

**for ob in p\_dict:**

**emp\_data=ob['fields']**

**result.append(emp\_data)**

**json\_data=json.dumps(result)**

**return HttpResponse(json\_data,content\_type="application/json")**