**test.py**

**import requests**

**import json**

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

**ENDPOINT='api/'**

**def get\_resource1():**

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

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

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

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

**def get\_resource2(id):**

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

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

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

**def create\_resource():**

**new\_emp={'eno':150,'ename':'Arun','esal':70000,'eaddr':'Mumbai'}**

**resp=requests.post(BASE\_URL+ENDPOINT,data=json.dumps(new\_emp))**

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

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

**print("Enter 1-->to display all records 2-->to display a particular record 3-->to create a new resource")**

**ch=int(input("Enter your choice"))**

**if ch==1:**

**get\_resource1()**

**if ch==2:**

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

**get\_resource2(id)**

**if ch==3:**

**create\_resource()**

**views.py**

**from django.shortcuts import render**

**from myApp.models import Employee**

**from django.http import HttpResponse**

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

**from myApp.mixins import SerializeMixin,HttpResponseMixin**

**import json**

**class EmployeeDetails(SerializeMixin,HttpResponseMixin,View):**

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

**try:**

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

**except:**

**json\_data=json.dumps({'msg':'The requested resource is unavailable'})**

**return self.render\_to\_http\_response(json\_data,status=404)**

**else:**

**json\_data=self.fun([emp])**

**return self.render\_to\_http\_response(json\_data)**

**class EmployeeList(SerializeMixin,HttpResponseMixin,View):**

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

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

**json\_data=self.fun(qs)**

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

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

**json\_data=json.dumps({'msg':'This is from post method'})**

**return self.render\_to\_http\_response(json\_data)**

**Error: Forbidden (CSRF cookie not set.): /api/**

**Solution:We have to disable csrf token,because we are running our application through test.py file.We can do this by using the following ways**

**1.At method level**

**2.At class level**

**3.At project level**

**1.At method level:**

**from django.views.decorators.csrf import csrf\_exempt**

**@csrf\_ exempt**

**def view1(request):**

**….**

**….**

**2.At class level:**

**from django.views.decorators.csrf import csrf\_exempt**

**from django.utils.decorators import method\_decorator**

**@method\_decorator(csrf\_exempt,name=’dispatch’)**

**class EmployeeList(View):**

**-------**

**--------**

**3.disable the following line**

**MIDDLEWARE = [**

**------**

**# 'django.middleware.csrf.CsrfViewMiddleware',**

**--------**

**]**

**from django.shortcuts import render**

**from myApp.models import Employee**

**from django.http import HttpResponse**

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

**from myApp.mixins import SerializeMixin,HttpResponseMixin**

**import json**

**from django.views.decorators.csrf import csrf\_exempt**

**from django.utils.decorators import method\_decorator**

**class EmployeeDetails(SerializeMixin,HttpResponseMixin,View):**

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

**try:**

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

**except:**

**json\_data=json.dumps({'msg':'The requested resource is unavailable'})**

**return self.render\_to\_http\_response(json\_data,status=404)**

**else:**

**json\_data=self.fun([emp])**

**return self.render\_to\_http\_response(json\_data)**

**@method\_decorator(csrf\_exempt,name="dispatch")**

**class EmployeeList(SerializeMixin,HttpResponseMixin,View):**

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

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

**json\_data=self.fun(qs)**

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

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

**json\_data=json.dumps({'msg':'This is from post method'})**

**return self.render\_to\_http\_response(json\_data)**

**cmd prompt1:Run django sever  
cmd prompt2:py test.py**

**Enter 1-->to display all records 2-->to display a particular record 3-->to create a new resource**

**Enter your choice3**

**{'msg': 'This is from post method'}**

**200**

**We are sending json data to django application,django application should grab the data and save this in data base table.To check if user has sent json data or not we use the following code.**

**myApp/utils.py**

**import json**

**def is\_json(data):**

**try:**

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

**valid=True**

**except:**

**valid=False**

**return valid**

**views.py**

**from django.shortcuts import render**

**from myApp.models import Employee**

**from django.http import HttpResponse**

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

**from myApp.mixins import SerializeMixin,HttpResponseMixin**

**import json**

**from django.views.decorators.csrf import csrf\_exempt**

**from django.utils.decorators import method\_decorator**

**from myApp.utils import is\_json**

**from myApp.forms import EmployeeForm**

**class EmployeeDetails(SerializeMixin,HttpResponseMixin,View):**

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

**try:**

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

**except:**

**json\_data=json.dumps({'msg':'The requested resource is unavailable'})**

**return self.render\_to\_http\_response(json\_data,status=404)**

**else:**

**json\_data=self.fun([emp])**

**return self.render\_to\_http\_response(json\_data)**

**@method\_decorator(csrf\_exempt,name="dispatch")**

**class EmployeeList(SerializeMixin,HttpResponseMixin,View):**

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

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

**json\_data=self.fun(qs)**

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

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

**data=request.body #To fetch partner application provided data**

**valid=is\_json(data)**

**if not valid:**

**json\_data=json.dumps({'msg':'Invalid data form'})**

**return self.render\_to\_http\_response(json\_data,status=404)**

**empdata=json.loads(data) #emp data is a dict**

**form=EmployeeForm(empdata) #create a form with user data**

**if form.is\_valid():**

**form.save(commit=True)**

**json\_data=json.dumps({'msg':'Resource has been created successfully'})**

**return self.render\_to\_http\_response(json\_data)**

**if form.errors:**

**json\_data=json.dumps({'msg':'Form contains errors'})**

**return self.render\_to\_http\_response(json\_data)**

**myApp/forms.py**

**from django import forms**

**from myApp.models import Employee**

**class EmployeeForm(forms.ModelForm):**

**class Meta:**

**model=Employee**

**fields="\_\_all\_\_"**

**test.py**

**import requests**

**import json**

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

**ENDPOINT='api/'**

**def get\_resource1():**

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

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

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

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

**def get\_resource2(id):**

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

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

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

**def create\_resource():**

**new\_emp={'eno':150,'ename':'Arun','esal':70000,'eaddr':'Mumbai'}**

**resp=requests.post(BASE\_URL+ENDPOINT,data=json.dumps(new\_emp))**

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

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

**print("Enter 1-->to display all records 2-->to display a particular record 3-->to create a new resource")**

**ch=int(input("Enter your choice"))**

**if ch==1:**

**get\_resource1()**

**if ch==2:**

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

**get\_resource2(id)**

**if ch==3:**

**create\_resource()**

**Form validation**

**from django import forms**

**from myApp.models import Employee**

**class EmployeeForm(forms.ModelForm):**

**def clean\_esal(self):**

**sal=self.cleaned\_data['esal']**

**if sal<15000:**

**raise forms.ValidationError("sal cannot be <15000")**

**return sal**

**class Meta:**

**model=Employee**

**fields="\_\_all\_\_"**

**is\_valid internally calls clean\_esal if form is invalid,then the statement form.errors gets executed**

**UPDATING THE RECORD**

**To perform update operation,we use put method and we have to update a particular record based on id  
Note:**

**original\_data={'eno':100,'ename':'shrinath','esal':10000}**

**provided\_data={'esal':20000}**

**original\_data.update(provided\_data)**

**print(original\_data)**

**o/p**

**{'eno': 100, 'ename': 'shrinath', 'esal': 20000}**

**1.form=EmployeeForm(original\_data)**

**here,with original\_data a new record gets created**

**2. form=EmployeeForm(original\_data,instance=emp)**

**Original\_data gets updated to existing emp object i.e a new record does not get created rather,existing record gets updated**

**test.py**

**import requests**

**import json**

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

**ENDPOINT='api/'**

**def get\_resource1():**

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

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

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

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

**def get\_resource2(id):**

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

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

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

**def create\_resource():**

**new\_emp={'eno':151,'ename':'Anu','esal':16000,'eaddr':'Delhi'}**

**resp=requests.post(BASE\_URL+ENDPOINT,data=json.dumps(new\_emp))**

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

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

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

**new\_emp={'esal':25000,'eaddr':'Bangalore'}**

**resp=requests.put(BASE\_URL+ENDPOINT+id+'/',data=json.dumps(new\_emp))**

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

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

**print("Enter 1-->to display all records 2-->to display a particular record 3-->to create a new resource 4-->to update a resource")**

**ch=int(input("Enter your choice"))**

**if ch==1:**

**get\_resource1()**

**if ch==2:**

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

**get\_resource2(id)**

**if ch==3:**

**create\_resource()**

**if ch==4:**

**id=input("enter id")**

**update\_resource(id)**

**views.py**

**from django.shortcuts import render**

**from myApp.models import Employee**

**from django.http import HttpResponse**

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

**from myApp.mixins import SerializeMixin,HttpResponseMixin**

**import json**

**from django.views.decorators.csrf import csrf\_exempt**

**from django.utils.decorators import method\_decorator**

**from myApp.utils import is\_json**

**from myApp.forms import EmployeeForm**

**@method\_decorator(csrf\_exempt,name="dispatch")**

**class EmployeeDetails(SerializeMixin,HttpResponseMixin,View):**

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

**try:**

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

**except:**

**json\_data=json.dumps({'msg':'The requested resource is unavailable'})**

**return self.render\_to\_http\_response(json\_data,status=404)**

**else:**

**json\_data=self.fun([emp])**

**return self.render\_to\_http\_response(json\_data)**

**def get\_object\_by\_id(self,id):**

**try:**

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

**except Employee.DoesNotExist:**

**emp=None**

**return emp**

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

**emp=self.get\_object\_by\_id(id=id)**

**if emp is None:**

**json\_data=json.dumps({'msg':'No such Record'})**

**return self.render\_to\_http\_response(json\_data,status=404)**

**data=request.body**

**valid=is\_json(data)**

**if not valid:**

**json\_data=json.dumps({'msg':'invalid data form'})**

**return self.render\_to\_http\_response(json\_data,status=404)**

**provided\_data=json.loads(data) #givev by partner application**

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

**origanal\_data.update(provided\_data)**

**form=EmployeeForm(origanal\_data,instance=emp)**

**if form.is\_valid():**

**form.save(commit=True)**

**json\_data=json.dumps({'msg':'Resource has been updated successfully'})**

**return self.render\_to\_http\_response(json\_data)**

**if form.errors:**

**json\_data=json.dumps({'msg':'Form contains errors'})**

**return self.render\_to\_http\_response(json\_data)**

**@method\_decorator(csrf\_exempt,name="dispatch")**

**class EmployeeList(SerializeMixin,HttpResponseMixin,View):**

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

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

**json\_data=self.fun(qs)**

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

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

**data=request.body #To fetch partner application provided data**

**valid=is\_json(data)**

**if not valid:**

**json\_data=json.dumps({'msg':'Invalid data form'})**

**return self.render\_to\_http\_response(json\_data,status=404)**

**empdata=json.loads(data) #emp data is a dict**

**form=EmployeeForm(empdata) #create a form with user data**

**if form.is\_valid():**

**form.save(commit=True)**

**json\_data=json.dumps({'msg':'Resource has been created successfully'})**

**return self.render\_to\_http\_response(json\_data)**

**if form.errors:**

**json\_data=json.dumps({'msg':'Form contains errors'})**

**return self.render\_to\_http\_response(json\_data)**

**Delete Operation**

**Deleting a particular record based on id**

**test.py**

**import requests**

**import json**

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

**ENDPOINT='api/'**

**def get\_resource1():**

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

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

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

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

**def get\_resource2(id):**

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

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

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

**def create\_resource():**

**new\_emp={'eno':151,'ename':'Anu','esal':16000,'eaddr':'Delhi'}**

**resp=requests.post(BASE\_URL+ENDPOINT,data=json.dumps(new\_emp))**

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

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

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

**new\_emp={'esal':25000,'eaddr':'Bangalore'}**

**resp=requests.put(BASE\_URL+ENDPOINT+id+'/',data=json.dumps(new\_emp))**

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

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

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

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

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

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

**print("Enter 1-->to display all records\n 2-->to display a particular record\n 3-->to create a new resource \n 4-->to update a resource \n 5-->to delete a resource")**

**ch=int(input("Enter your choice"))**

**if ch==1:**

**get\_resource1()**

**if ch==2:**

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

**get\_resource2(id)**

**if ch==3:**

**create\_resource()**

**if ch==4:**

**id=input("enter id")**

**update\_resource(id)**

**if ch==5:**

**id=input("enter id")**

**delete\_resource(id)**

**views.py**

**from django.shortcuts import render**

**from myApp.models import Employee**

**from django.http import HttpResponse**

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

**from myApp.mixins import SerializeMixin,HttpResponseMixin**

**import json**

**from django.views.decorators.csrf import csrf\_exempt**

**from django.utils.decorators import method\_decorator**

**from myApp.utils import is\_json**

**from myApp.forms import EmployeeForm**

**@method\_decorator(csrf\_exempt,name="dispatch")**

**class EmployeeDetails(SerializeMixin,HttpResponseMixin,View):**

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

**try:**

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

**except:**

**json\_data=json.dumps({'msg':'The requested resource is unavailable'})**

**return self.render\_to\_http\_response(json\_data,status=404)**

**else:**

**json\_data=self.fun([emp])**

**return self.render\_to\_http\_response(json\_data)**

**def get\_object\_by\_id(self,id):**

**try:**

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

**except Employee.DoesNotExist:**

**emp=None**

**return emp**

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

**emp=self.get\_object\_by\_id(id=id)**

**if emp is None:**

**json\_data=json.dumps({'msg':'No such Record'})**

**return self.render\_to\_http\_response(json\_data,status=404)**

**data=request.body**

**valid=is\_json(data)**

**if not valid:**

**json\_data=json.dumps({'msg':'invalid data form'})**

**return self.render\_to\_http\_response(json\_data,status=404)**

**provided\_data=json.loads(data) #givev by partner application**

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

**origanal\_data.update(provided\_data)**

**form=EmployeeForm(origanal\_data,instance=emp)**

**if form.is\_valid():**

**form.save(commit=True)**

**json\_data=json.dumps({'msg':'Resource has been updated successfully'})**

**return self.render\_to\_http\_response(json\_data)**

**if form.errors:**

**json\_data=json.dumps({'msg':'Form contains errors'})**

**return self.render\_to\_http\_response(json\_data)**

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

**emp=self.get\_object\_by\_id(id=id)**

**if emp is None:**

**json\_data=json.dumps({'msg':'No such Record'})**

**return self.render\_to\_http\_response(json\_data,status=404)**

**t=emp.delete()**

**print(t)**

**@method\_decorator(csrf\_exempt,name="dispatch")**

**class EmployeeList(SerializeMixin,HttpResponseMixin,View):**

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

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

**json\_data=self.fun(qs)**

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

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

**data=request.body #To fetch partner application provided data**

**valid=is\_json(data)**

**if not valid:**

**json\_data=json.dumps({'msg':'Invalid data form'})**

**return self.render\_to\_http\_response(json\_data,status=404)**

**empdata=json.loads(data) #emp data is a dict**

**form=EmployeeForm(empdata) #create a form with user data**

**if form.is\_valid():**

**form.save(commit=True)**

**json\_data=json.dumps({'msg':'Resource has been created successfully'})**

**return self.render\_to\_http\_response(json\_data)**

**if form.errors:**

**json\_data=json.dumps({'msg':'Form contains errors'})**

**return self.render\_to\_http\_response(json\_data)**