[**http://127.0.0.1:8000/api/1==> extra**](http://127.0.0.1:8000/api/1==%3e%20extra) **information**

**To avoid extra information at id level,wrt server,we need the previous functionality one more time,Hence we use mixin to generalize the functionality.[Customization purpose]**

**test.py**

**import requests**

**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.status\_code)**

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

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

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

**if ch==1:**

**get\_resource1()**

**if ch==2:**

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

**get\_resource2(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**

**import json**

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

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

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

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

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

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

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

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

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

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

**myApp/mixins.py**

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

**import json**

**class SerializeMixin(object):**

**def fun(self,qs):**

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

**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 json\_data**

**urls.py**

**from django.contrib import admin**

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

**from myApp.views import \***

**urlpatterns = [**

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

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

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

**]**

**Exception Handling**

**It is not recommended to display the error information directly to the partner application,such errors must be handled**

**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**

**import json**

**class EmployeeDetails(SerializeMixin,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'})**

**else:**

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

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

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

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

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

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

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

**Assuming that our views.py does not handle the exception,then we can also handle the exception as shown below:**

**import requests**

**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+'/')**

**if resp.status\_code not in range(200,300):#successfull**

**print("Something went wrong")**

**else:**

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

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

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

**if ch==1:**

**get\_resource1()**

**if ch==2:**

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

**get\_resource2(id)**

**Adding status code to the response**

**test.py**

**import requests**

**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)**

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

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

**if ch==1:**

**get\_resource1()**

**if ch==2:**

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

**get\_resource2(id)**

**mixins.py**

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

**import json**

**from django.http import HttpResponse**

**class HttpResponseMixin(object):**

**def render\_to\_http\_response(self,json\_data,status=200):#default status=200**

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

**class SerializeMixin(object):**

**def fun(self,qs):**

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

**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 json\_data**

**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,View):**

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

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

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

**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 \***

**urlpatterns = [**

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

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

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

**]**

**It is possible to dump db data to a console**

**py manage.py dumpdata myApp.Employee🡪provides employee data to the console json form**

**py manage.py dumpdata myApp.Employee --indent 4**

**we can print the data in xml format**

**py manage.py dumpdata myApp.Employee --format xml --indent 4**

**we can save the data in a file say emp.xml**

**py manage.py dumpdata myApp.Employee --format xml>emp.xml --indent 4**