IT Lab 10 : Rest API

Name: Rajvardhan Reddy

Roll Number: 19

Section: B

Batch: B1

Registration Number: 180905093

settings.py: (Common to all programs)

```
from pathlib import Path

import os

# Build paths inside the project like this: BASE_DIR / 'subdir'.

BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/3.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET_KEY = 'django-insecure-6hv)wr9a8u3p4#rru)f)f3(^pf-
$5g&i97#4m+ku$dry8$%y64'

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED_HOSTS = ['127.0.0.1']

# Application definition

INSTALLED_APPS = [

'prob4.apps.Prob4Config',

'prob3.apps.Prob3Config',

'prob2.apps.Prob2Config',

'prob1.apps.Prob1Config',
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles','rest_framework',
]
MIDDLEWARE = [
'django.middleware.security.SecurityMiddleware',
'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
ROOT_URLCONF = 'week10v2.urls'
TEMPLATES = [
{
'BACKEND': 'django.template.backends.django.DjangoTemplates',
'DIRS': [os.path.join(BASE_DIR,'templates')],
'APP_DIRS': True,
'OPTIONS': {
'context_processors': [
'django.template.context_processors.debug',
'django.template.context_processors.request',
'django.contrib.auth.context_processors.auth',
'django.contrib.messages.context_processors.messages',
],
},
},
]
WSGI_APPLICATION = 'week10v2.wsgi.application'
# Database
# https://docs.djangoproject.com/en/3.2/ref/settings/#databases
DATABASES = {
'default': {
'ENGINE': 'django.db.backends.postgresql',
'NAME': 'lab10',
'USER': 'user','PASSWORD': 'password',
'HOST': 'localhost',
}
}
# Password validation
# https://docs.djangoproject.com/en/3.2/ref/settings/#auth-password-
validators
AUTH_PASSWORD_VALIDATORS = [
{
'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
},
{
'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
},
{
'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
},
{
'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
},
]
# Internationalization
# https://docs.djangoproject.com/en/3.2/topics/i18n/
LANGUAGE_CODE = 'en-us'
TIME_ZONE = 'UTC'
USE_I18N = True
USE_L10N = True
USE_TZ = True
# Static files (CSS, JavaScript, Images)#
https://docs.djangoproject.com/en/3.2/howto/static-files/
STATIC_URL = '/static/'
# Default primary key field type
# https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

P1)

manage.py:

```
import os
import sys
def main():
"""Run administrative tasks."""
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'week10v2.settings')
try:
from django.core.management import execute_from_command_line
except ImportError as exc:
raise ImportError(
"Couldn't import Django. Are you sure it's installed and "
"available on your PYTHONPATH environment variable? Did you "
"forget to activate a virtual environment?"
) from excexecute_from_command_line(sys.argv)
if __name__ == '__main__':
main()
```

<u>models.py:</u>

```
from re import T
from django.db import models
# Create your models here.
class User(models.Model):
username =
models.CharField(unique=True,null=False,blank=False,max_length=200)
email = models.EmailField(null = True,blank=True)
phno = models.PositiveBigIntegerField(null=True,blank=True)
password = models.CharField(null=False,blank=False,max_length=200)
def __str__(self):
```

```
return self.username
class Blog(models.Model):
title = models.CharField(max_length=200)
desc = models.TextField()
date = models.DateField()
user = models.ForeignKey(User,on_delete=models.CASCADE,default = None)
def __str__(self):
return self.title
class Comment(models.Model):
user = models.ForeignKey(User,on_delete=models.CASCADE,null=True)
Blog = models.ForeignKey(Blog,on_delete=models.CASCADE)
comment = models.TextField()
date = models.DateField()
```

<u>serializers.py:</u>

```
from django.db.models import fields
from rest_framework import serializers
from .models import Comment,Blog,User
class UserSerializer(serializers.ModelSerializer):
class Meta:
fields = (
'id',
'username',
'email',
'phno',
'password',
)
```

```
model = User

class CommentSerializer(serializers.ModelSerializer):

class Meta:

fields = (

'id',

'user',

'Blog',

'comment',

'date'

)

model = Comment

class BlogSerializer(serializers.ModelSerializer):

class Meta:

fields = (

'id',

'title',

'desc',

'date',

'user',

)

model = Blog
```

<u>views.py:</u>

```
from django.shortcuts import render

from django.http import request

from .models import *

from .serializers import *
from rest_framework import generics
import getpass
class ListBlogs(generics.ListCreateAPIView):
queryset = Blog.objects.all()
serializer_class = BlogSerializer
class DetailBlog(generics.RetrieveUpdateDestroyAPIView):
queryset = Blog.objects.all()
serializer_class = BlogSerializer
class ListComment(generics.ListCreateAPIView):
queryset = Comment.objects.all()
serializer_class = CommentSerializer
class DetailComment(generics.RetrieveUpdateDestroyAPIView):
queryset = Comment.objects.all()
serializer_class = CommentSerializer
class ListUser(generics.ListCreateAPIView):
queryset = User.objects.all()
serializer_class = UserSerializer
class DetailUser(generics.RetrieveUpdateDestroyAPIView):
queryset = User.objects.all()
serializer_class = UserSerializer
```

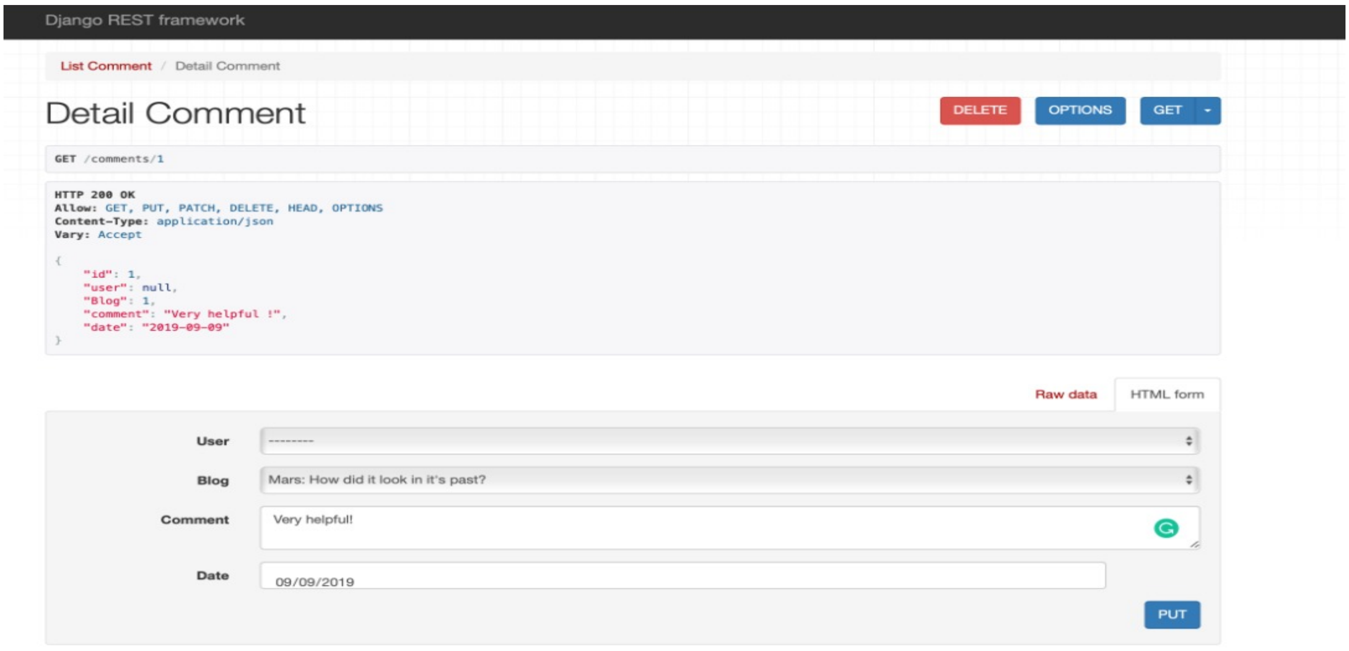
urls.py:

```
from django.urls import path
from .views import *
urlpatterns = [
path("blogs",ListBlogs.as_view(),name = "ListBlog"),
path("blogs/<int:pk>",DetailBlog.as_view(),name = "Blog"),
path("comments",ListComment.as_view(),name="comments"),

path("users",ListUser.as_view(),name = "users"),

path("users/<int:pk>",DetailUser.as_view(),name = "User"),

path("comments/<int:pk>",DetailComment.as_view(),name="comment"),

]
```

urls.py:

```
from django.contrib import admin

from django.urls import path,include

urlpatterns = [

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

path('',include("prob1.urls")),

#path('',include("prob2.urls")),

#path('',include("prob3.urls")),

#path('',include("prob4.urls")),

]
```

Output:

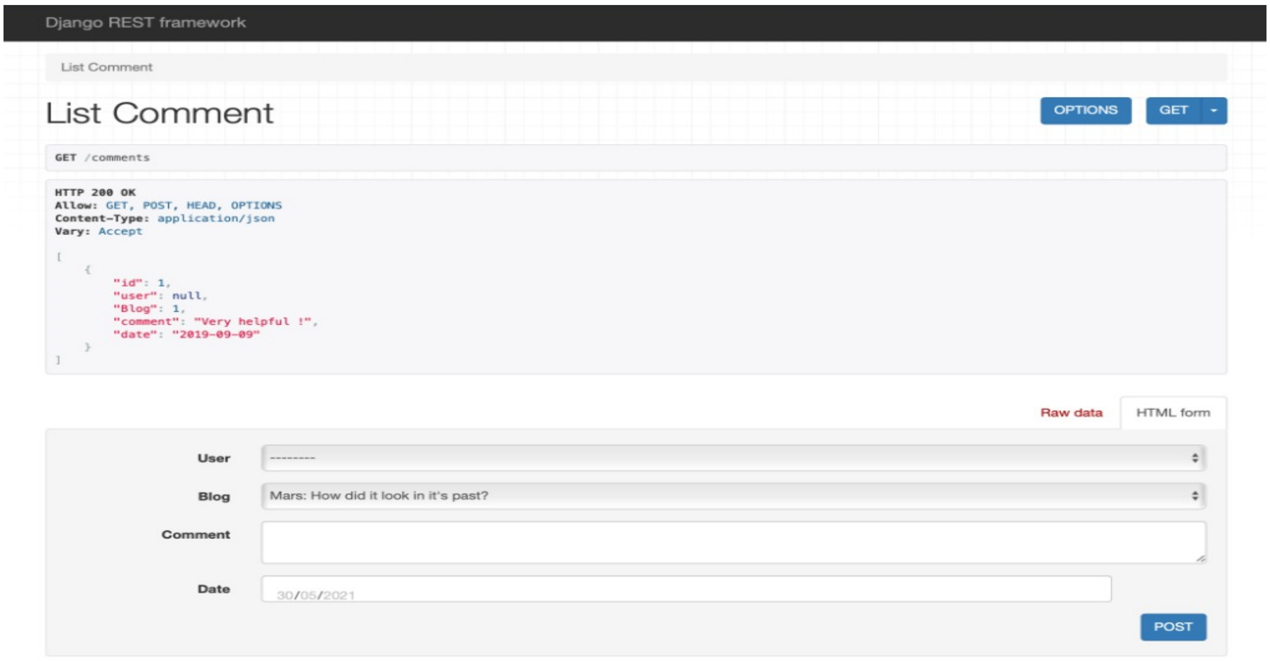


Detail Comment

```
GET /comments/1
```

```
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 1,
    "user": null,
    "Blog": 1,
    "comment": "Very helpful !",
    "date": "2019-09-09"
}
```

User	---------
Blog	Mars: How did it look in it's past?
Comment	Very helpful!
Date	09/09/2019

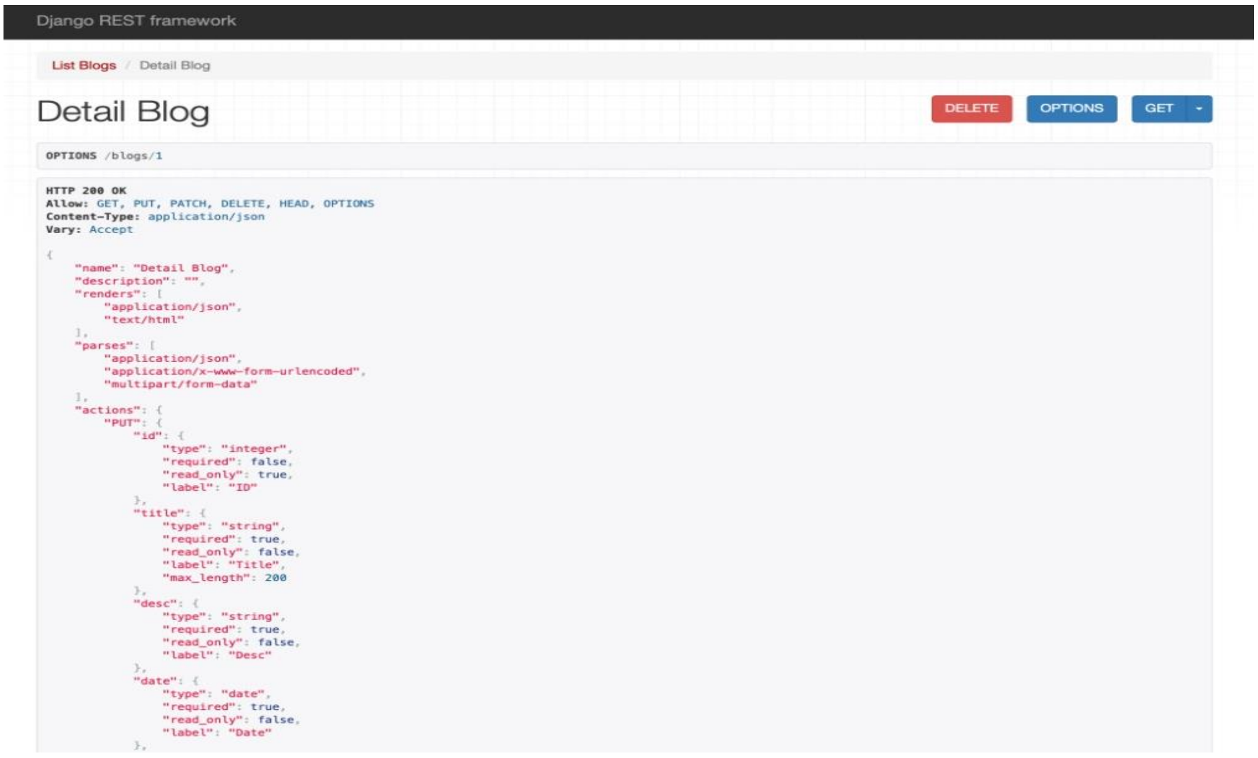


List Comment

```
GET /comments
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 1,
        "user": null,
        "Blog": 1,
        "comment": "Very helpful !",
        "date": "2019-09-09"
    }
]
```

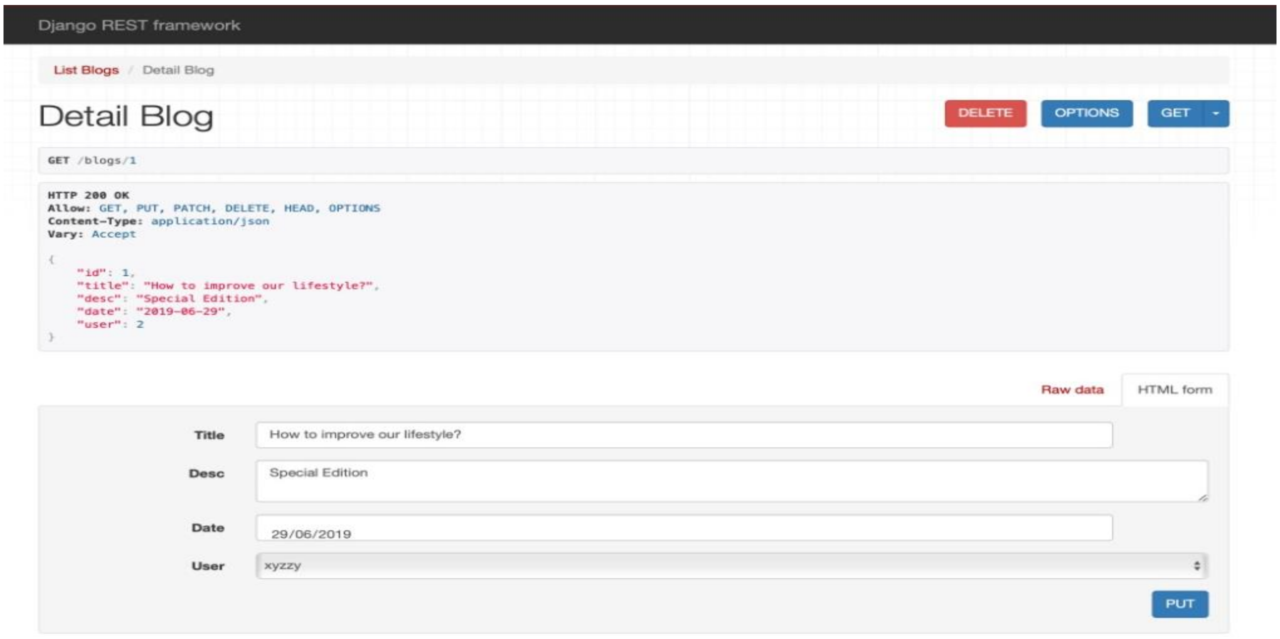
User	--------
Blog	Mars: How did it look in it's past?
Comment	
Date	30/05/2021



Detail Blog

```
OPTIONS /blogs/1
```

```
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "name": "Detail Blog",
    "description": "",
    "renders": [
        "application/json",
        "text/html"
    ],
    "parses": [
        "application/json",
        "application/x-www-form-urlencoded",
        "multipart/form-data"
    ],
    "actions": {
        "PUT": {
            "id": {
                "type": "integer",
                "required": false,
                "read_only": true,
                "label": "ID"
            },
            "title": {
                "type": "string",
                "required": true,
                "read_only": false,
                "label": "Title",
                "max_length": 200
            },
            "desc": {
                "type": "string",
                "required": true,
                "read_only": false,
                "label": "Desc"
            },
            "date": {
                "type": "date",
                "required": true,
                "read_only": false,
                "label": "Date"
            },
```

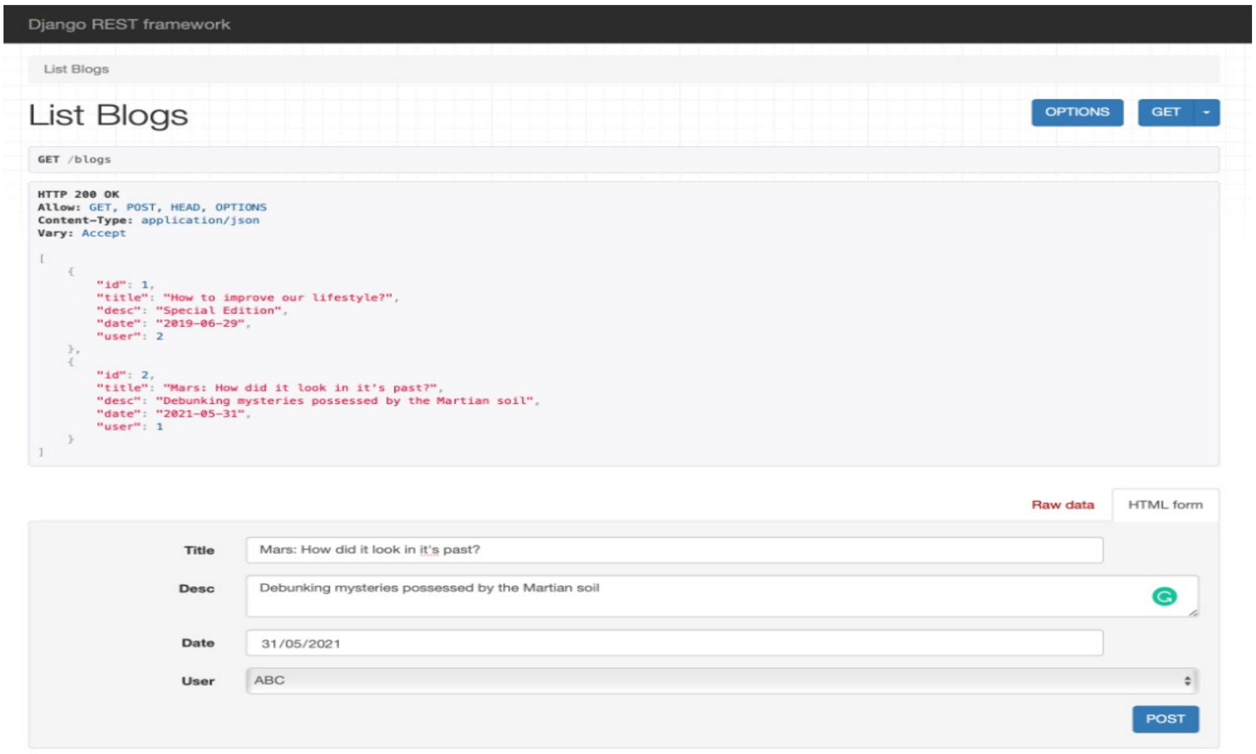


Detail Blog

```
GET /blogs/1
```

```
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 1,
    "title": "How to improve our lifestyle?",
    "desc": "Special Edition",
    "date": "2019-06-29",
    "user": 2
}
```

Title	How to improve our lifestyle?
Desc	Special Edition
Date	29/06/2019
User	xyzzy



List Blogs

```
GET /blogs
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 1,
        "title": "How to improve our lifestyle?",
        "desc": "Special Edition",
        "date": "2019-06-29",
        "user": 2
    },
    {
        "id": 2,
        "title": "Mars: How did it look in it's past?",
        "desc": "Debunking mysteries possessed by the Martian soil",
        "date": "2021-05-31",
        "user": 1
    }
]
```

Title	Mars: How did it look in it's past?
Desc	Debunking mysteries possessed by the Martian soil
Date	31/05/2021
User	ABC

P2)

models.py:

```
from django.db import models

# Create your models here.

class UserData(models.Model):

name = models.CharField(max_length=100)

contact = models.PositiveBigIntegerField()

def __str__(self):

return self.name

class UserLocation(models.Model):

user = models.ForeignKey(UserData,on_delete=models.CASCADE)

latitude = models.DecimalField(max_digits=7,decimal_places=4)

longitude = models.DecimalField(max_digits=7,decimal_places=4)

def __str__(self):

return '{}_{}_{}'.format(self.latitude,self.longitude,self.user.name)

class VehicleInfo(models.Model):

driverName = models.CharField(max_length = 100)

vehicleName = models.CharField(max_length = 100)

vehicleRegNo = models.CharField(max_length=10)

contact = models.PositiveBigIntegerField()

def __str__(self):

return self.driverName+"_"+self.vehicleName+"_"+self.vehicleRegNo

class TravelStatus(models.Model):

userLocation =
models.ForeignKey(UserLocation,on_delete=models.CASCADE)vehicle =
models.ForeignKey(VehicleInfo,on_delete=models.CASCADE)

eta = models.TimeField()

fare = models.PositiveIntegerField()
```

serializers.py:

```
from django.db.models import fields

from rest_framework import serializers

from .models import *

class UserDataserializer(serializers.ModelSerializer):

class Meta:

fields = '__all__'

model = UserData

class UserLocationserializer(serializers.ModelSerializer):

class Meta:

fields = '__all__'

model = UserLocation

class VehicleInfoserializer(serializers.ModelSerializer):

class Meta:

fields = '__all__'

model = VehicleInfo

class TravelStatusserializer(serializers.ModelSerializer):

class Meta:

fields = '__all__'

model = TravelStatus
```

views.py:

```
from django.shortcuts import render

from .serializers import *

from .models import *

from rest_framework import generics

# Create your views here.class ListUserData(generics.ListCreateAPIView):
```

```
queryset = UserData.objects.all()
serializer_class = UserDataserializer
class DetailUserData(generics.RetrieveUpdateDestroyAPIView):
queryset = UserData.objects.all()
serializer_class = UserDataserializer
class ListUserLocation(generics.ListCreateAPIView):
queryset = UserLocation.objects.all()
serializer_class = UserLocationserializer
class DetailUserLocation(generics.RetrieveUpdateDestroyAPIView):
queryset = UserLocationserializer
serializer_class = UserLocationserializer
class ListVehicleInfo(generics.ListCreateAPIView):
queryset = VehicleInfo.objects.all()
serializer_class = VehicleInfoserializer
class DetailVehicleInfo(generics.RetrieveUpdateDestroyAPIView):
queryset = VehicleInfo.objects.all()
serializer_class = VehicleInfoserializer
class ListTravelStatus(generics.ListCreateAPIView):
queryset = TravelStatus.objects.all()
serializer_class = TravelStatusserializer
class DetailTravelStatus(generics.RetrieveUpdateDestroyAPIView):
queryset = TravelStatus.objects.all()
serializer_class = TravelStatusserializer
```

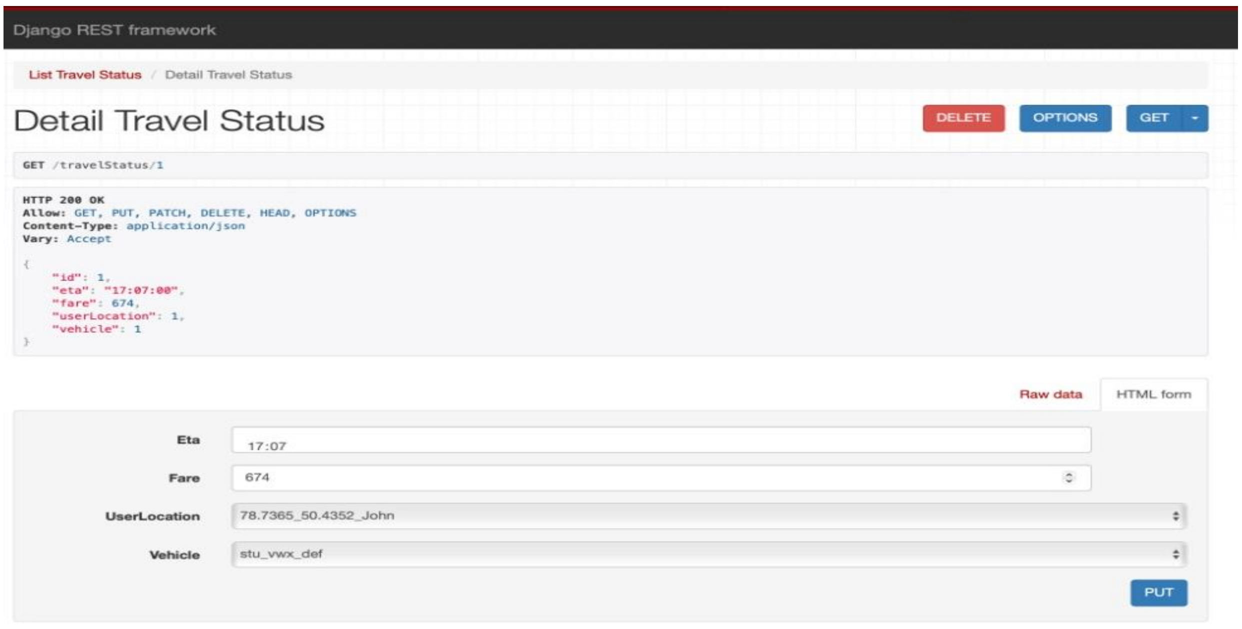
urls.py:

```
from django.urls import path
from django.urls.resolvers import URLPattern
from .views import *

urlpatterns = [

path("userData",ListUserData.as_view(),name =
"userData"),path("userLocation",ListUserLocation.as_view(),name =
"usersLocation"),

path("vehicleInfo",ListVehicleInfo.as_view(),name = "vehiclesInfo"),

path("travelStatus",ListTravelStatus.as_view(),name = "travelStatuses"),

path("userData/<int:pk>",DetailUserData.as_view(),name = "userDatum"),

path("userLocation/<int:pk>",DetailUserLocation.as_view(),name =
"userLocation"),

path("vehicleInfo/<int:pk>",DetailVehicleInfo.as_view(),name = "vehicleInfo"),

path("travelStatus/<int:pk>",DetailTravelStatus.as_view(),name =
"travelStatus"),

]
```

<u>urls.py:</u>

```
from django.contrib import admin

from django.urls import path,include

urlpatterns = [

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

#path('',include("prob1.urls")),

path('',include("prob2.urls")),

#path('',include("prob3.urls")),

#path('',include("prob4.urls")),

]
```

<u>Output:</u>

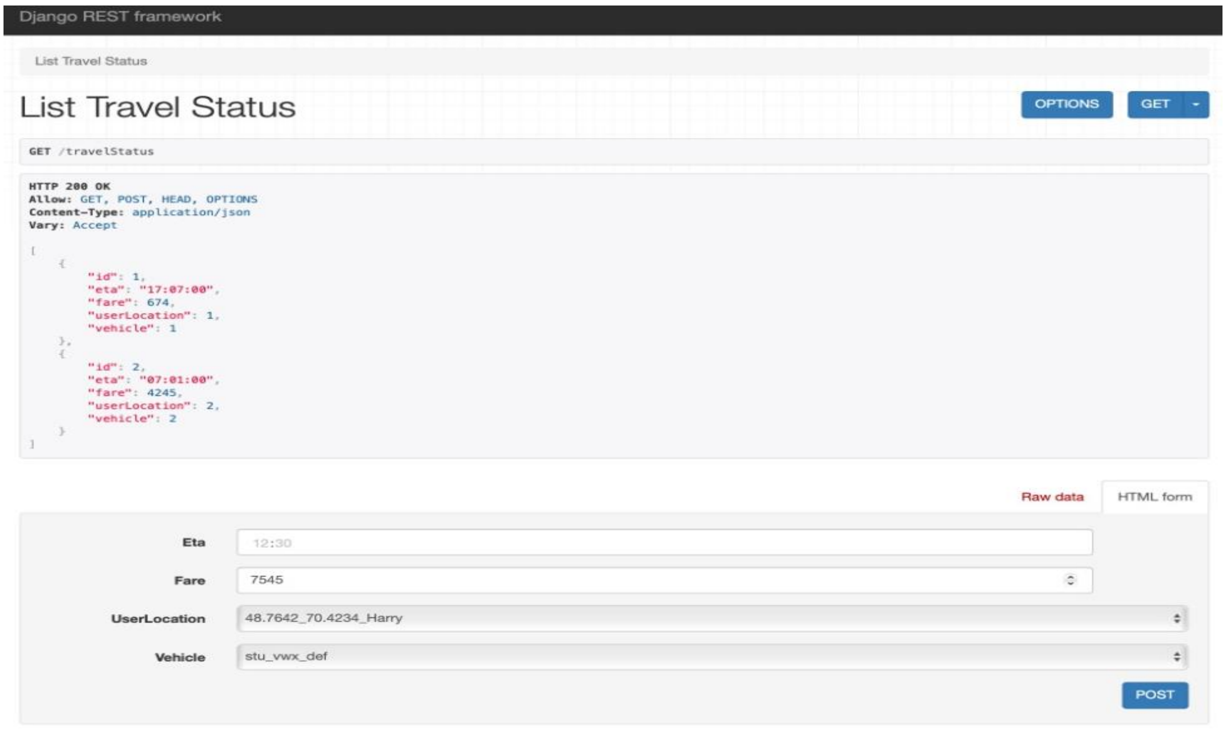


Detail Travel Status

```
GET /travelStatus/1
```

```
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 1,
    "eta": "17:07:00",
    "fare": 674,
    "userLocation": 1,
    "vehicle": 1
}
```

Eta	17:07
Fare	674
UserLocation	78.7365_50.4352_John
Vehicle	stu_vwx_def

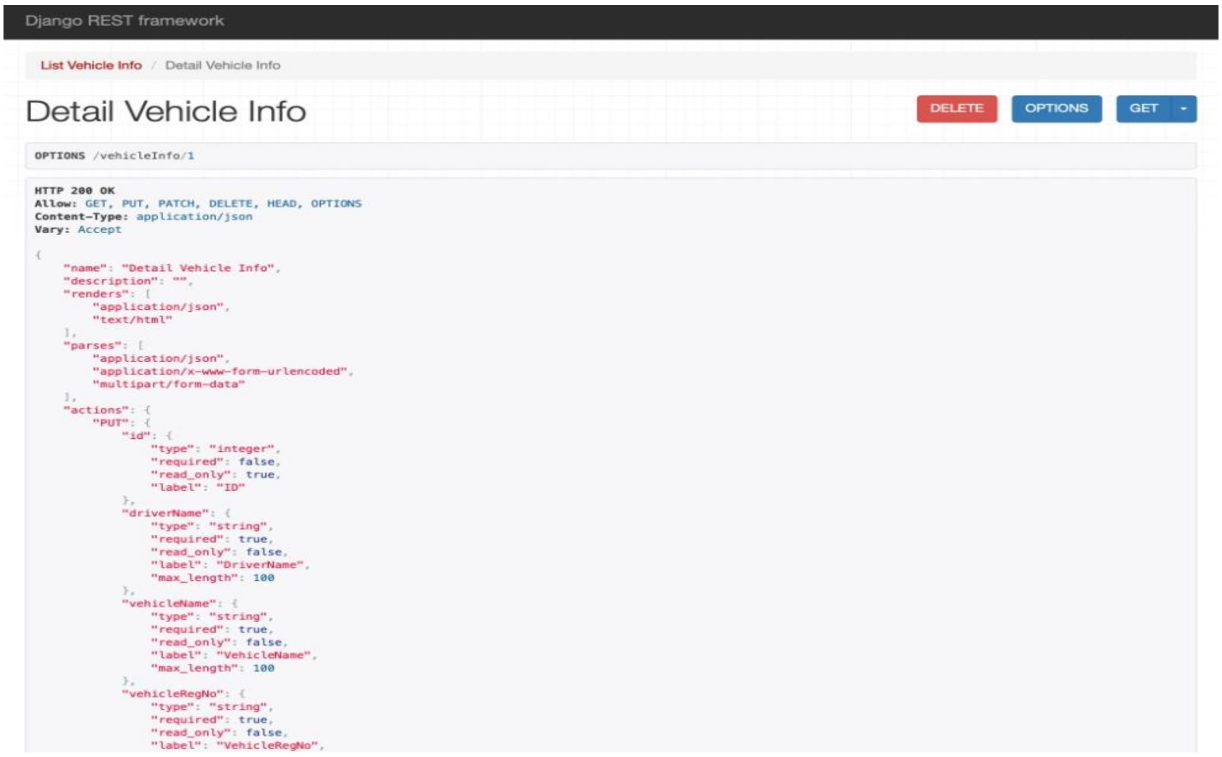


List Travel Status

```
GET /travelStatus
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
[
    {
        "id": 1,
        "eta": "17:07:00",
        "fare": 674,
        "userLocation": 1,
        "vehicle": 1
    },
    {
        "id": 2,
        "eta": "07:01:00",
        "fare": 4245,
        "userLocation": 2,
        "vehicle": 2
    }
]
```

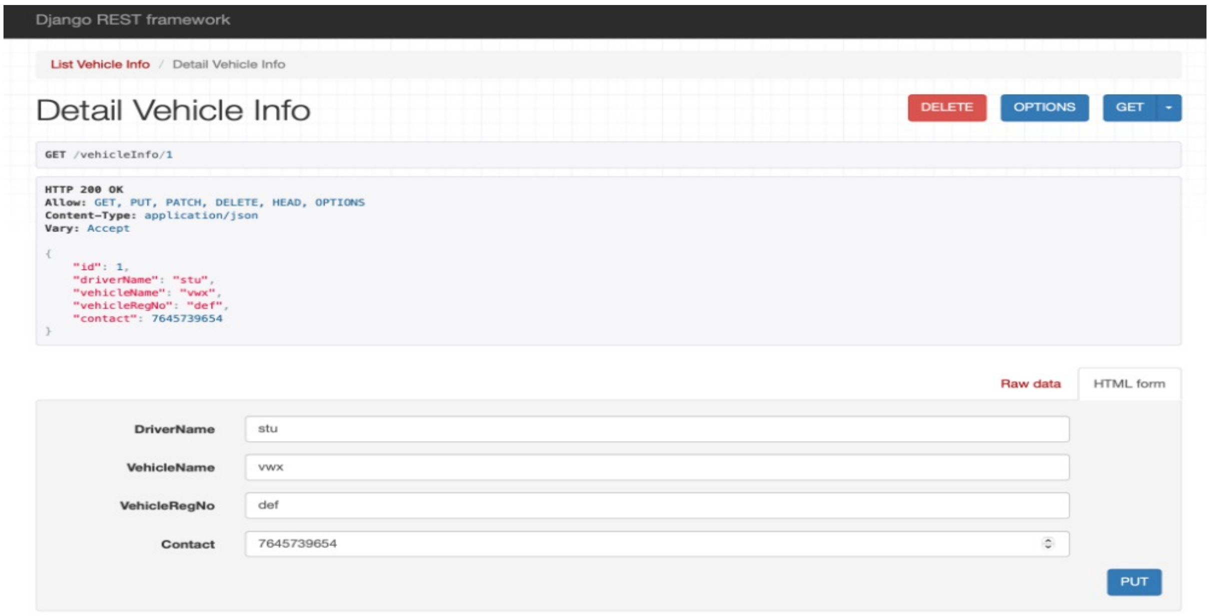
Eta	12:30
Fare	7545
UserLocation	48.7642_70.4234_Harry
Vehicle	stu_vwx_def



Detail Vehicle Info

```
OPTIONS /vehicleInfo/1
```

```
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "name": "Detail Vehicle Info",
    "description": "",
    "renders": [
        "application/json",
        "text/html"
    ],
    "parses": [
        "application/json",
        "application/x-www-form-urlencoded",
        "multipart/form-data"
    ],
    "actions": {
        "PUT": {
            "id": {
                "type": "integer",
                "required": false,
                "read_only": true,
                "label": "ID"
            },
            "driverName": {
                "type": "string",
                "required": true,
                "read_only": false,
                "label": "DriverName",
                "max_length": 100
            },
            "vehicleName": {
                "type": "string",
                "required": true,
                "read_only": false,
                "label": "VehicleName",
                "max_length": 100
            },
            "vehicleRegNo": {
                "type": "string",
                "required": true,
                "read_only": false,
                "label": "VehicleRegNo",
```

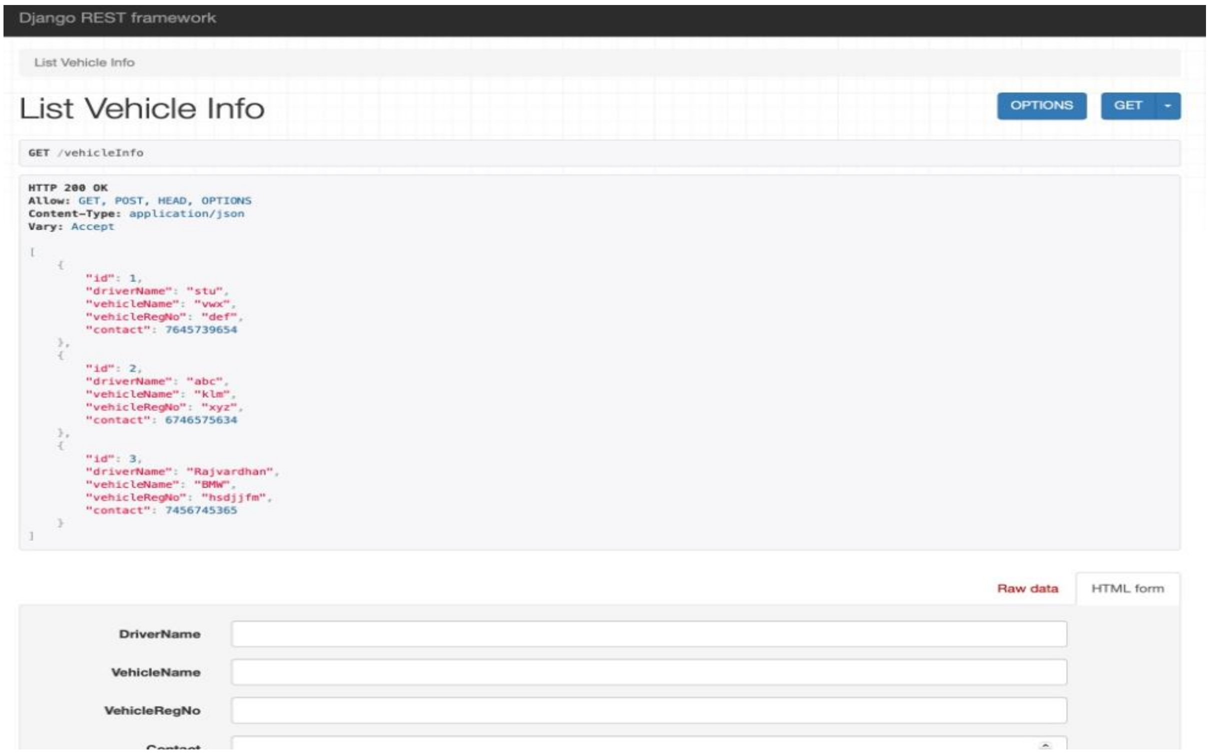


Detail Vehicle Info

```
GET /vehicleInfo/1
```

```
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 1,
    "driverName": "stu",
    "vehicleName": "vwx",
    "vehicleRegNo": "def",
    "contact": 7645739654
}
```

DriverName	stu
VehicleName	vwx
VehicleRegNo	def
Contact	7645739654

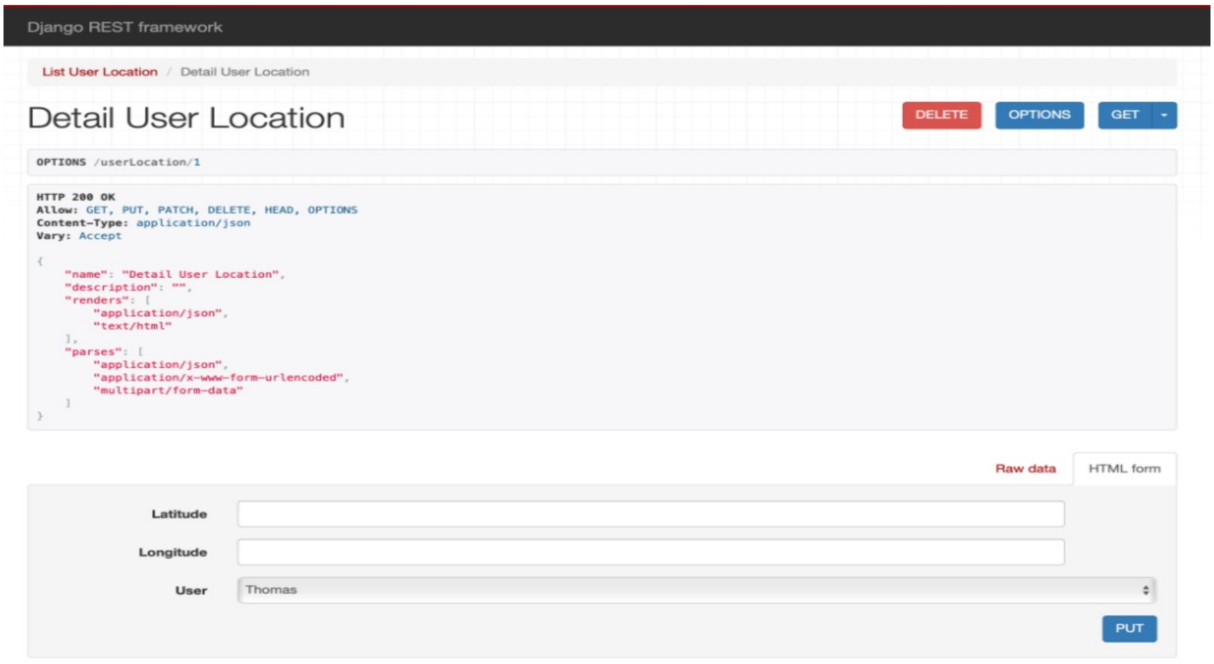


List Vehicle Info

```
GET /vehicleInfo
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 1,
        "driverName": "stu",
        "vehicleName": "vwx",
        "vehicleRegNo": "def",
        "contact": 7645739654
    },
    {
        "id": 2,
        "driverName": "abc",
        "vehicleName": "klm",
        "vehicleRegNo": "xyz",
        "contact": 6746575634
    },
    {
        "id": 3,
        "driverName": "Rajvardhan",
        "vehicleName": "BMW",
        "vehicleRegNo": "hsdjjfm",
        "contact": 7456745365
    }
]
```

DriverName	
VehicleName	
VehicleRegNo	
Contact	

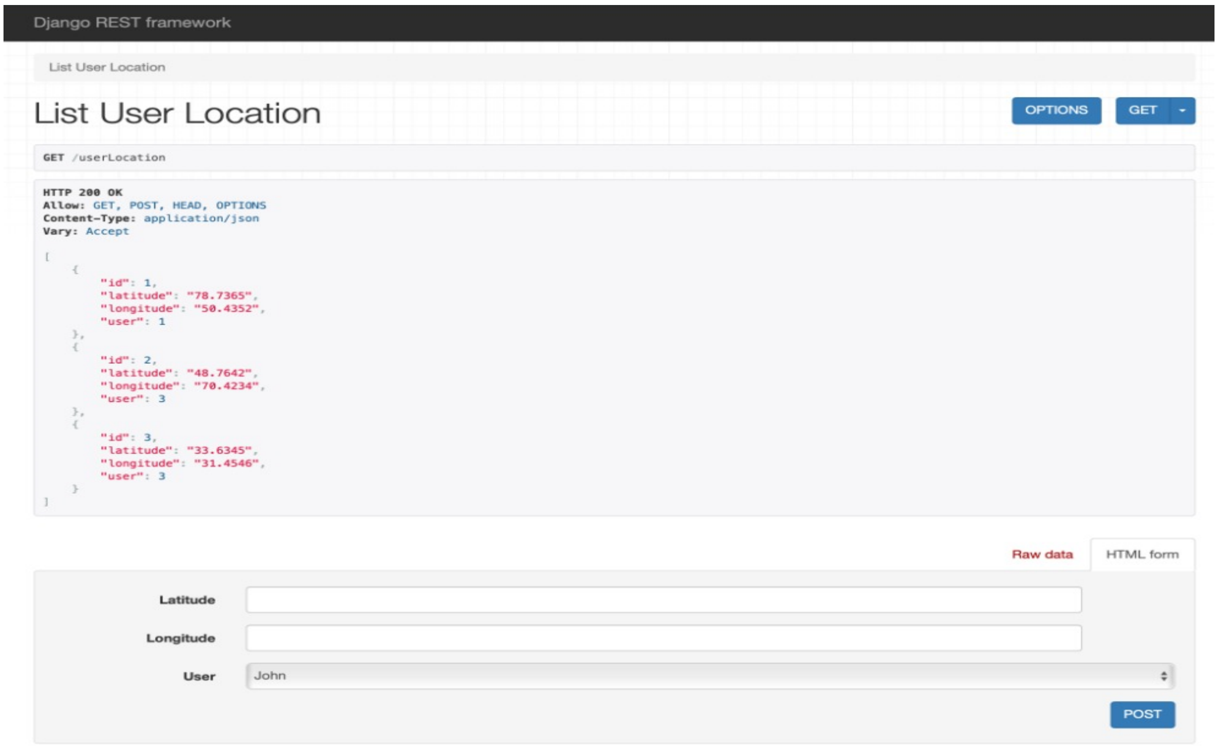


Detail User Location

```
OPTIONS /userLocation/1
```

```
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "name": "Detail User Location",
    "description": "",
    "renders": [
        "application/json",
        "text/html"
    ],
    "parses": [
        "application/json",
        "application/x-www-form-urlencoded",
        "multipart/form-data"
    ]
}
```

Latitude	
Longitude	
User	Thomas



List User Location

```
GET /userLocation

HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 1,
        "latitude": "78.7365",
        "longitude": "50.4352",
        "user": 1
    },
    {
        "id": 2,
        "latitude": "48.7642",
        "longitude": "70.4234",
        "user": 3
    },
    {
        "id": 3,
        "latitude": "33.6345",
        "longitude": "31.4546",
        "user": 3
    }
]
```

Latitude	
Longitude	
User	John

P3)

models.py:

```
from django.db import models

# Create your models here.

class Customer(models.Model):

name = models.CharField(max_length=100)

contact = models.PositiveBigIntegerField()

class Staff(models.Model):

name = models.CharField(max_length=100)

designation = models.CharField(max_length=200)

contact = models.PositiveBigIntegerField()

class Restaurant(models.Model):

name = models.CharField(max_length=200)

cuisine = models.CharField(max_length=100)
```

```
location = models.CharField(max_length=100)
contact = models.PositiveBigIntegerField()
```

serializers.py:

```
from django.db.models import fields
from rest_framework import serializers
from .models import *
class CustomerSerializer(serializers.ModelSerializer):
class Meta:
fields = '__all__'
model = Customer
class StaffSerializer(serializers.ModelSerializer):
class Meta:fields = '__all__'
model = Staff
class RestaurantSerializer(serializers.ModelSerializer):
class Meta:
fields = '__all__'
model = Restaurant
```

views.py:

```
from django.shortcuts import render
from rest_framework import generics,filters
from .serializers import *
from .models import *
# Create your views here
class ListCustomer(generics.ListCreateAPIView):
queryset = Customer.objects.all()
serializer_class = CustomerSerializer
```

```
class DetailCustomer(generics.RetrieveUpdateDestroyAPIView):

queryset = Customer.objects.all()

serializer_class = CustomerSerializer

class ListStaff(generics.ListCreateAPIView):

queryset = Staff.objects.all()

serializer_class = StaffSerializer

class DetailStaff(generics.RetrieveUpdateDestroyAPIView):

queryset = Staff.objects.all()

serializer_class = StaffSerializer

class ListRestaurant(generics.ListCreateAPIView):

queryset = Restaurant.objects.all()

serializer_class = RestaurantSerializer

filter_backends = [filters.SearchFilter]

search_fields = ['name','cuisine','location']

class DetailRestaurant(generics.RetrieveUpdateDestroyAPIView):

queryset = Restaurant.objects.all()

serializer_class = RestaurantSerializer
```

urls.py:

```
from django.urls import path

from .views import *

urlpatterns = [

path("customers",ListCustomer.as_view(),name = "customers"),

path("staff",ListStaff.as_view(),name = "staffs"),

path("restaurants",ListRestaurant.as_view(),name = "restaurants"),

path("customers/<int:pk>",DetailCustomer.as_view(),name = "customer"),

path("staff/<int:pk>",DetailStaff.as_view(),name = "staff"),
```

path("restaurants/<int:pk>",DetailRestaurant.as_view(),name = "restaurant"),

]

urls.py:

from django.contrib import admin

from django.urls import path,include

urlpatterns = [

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

#path('',include("prob1.urls")),

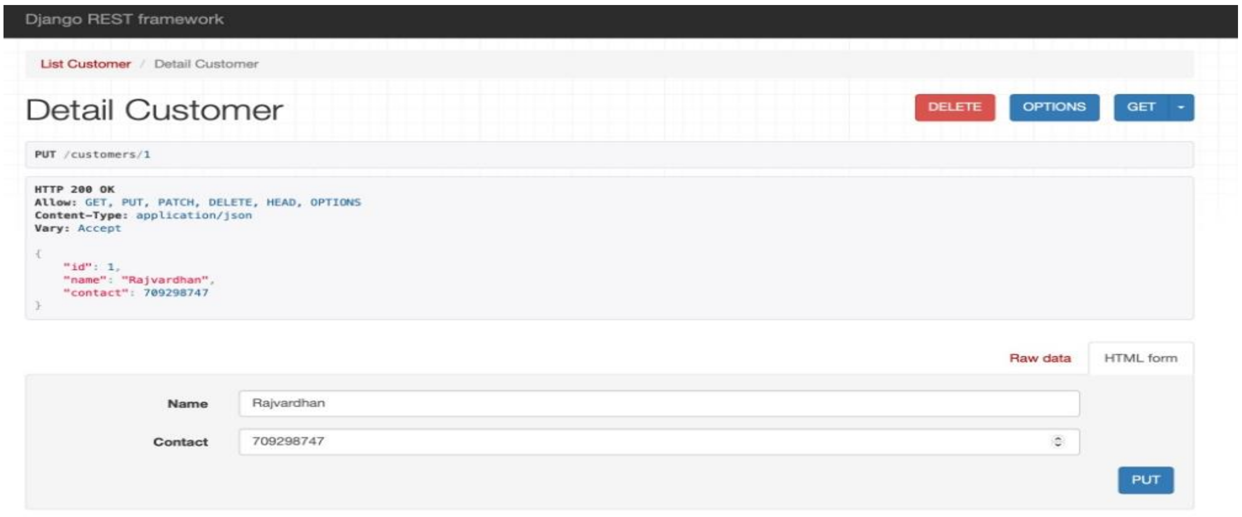
#path('',include("prob2.urls")),

path('',include("prob3.urls")),

#path('',include("prob4.urls")),

]

Output:

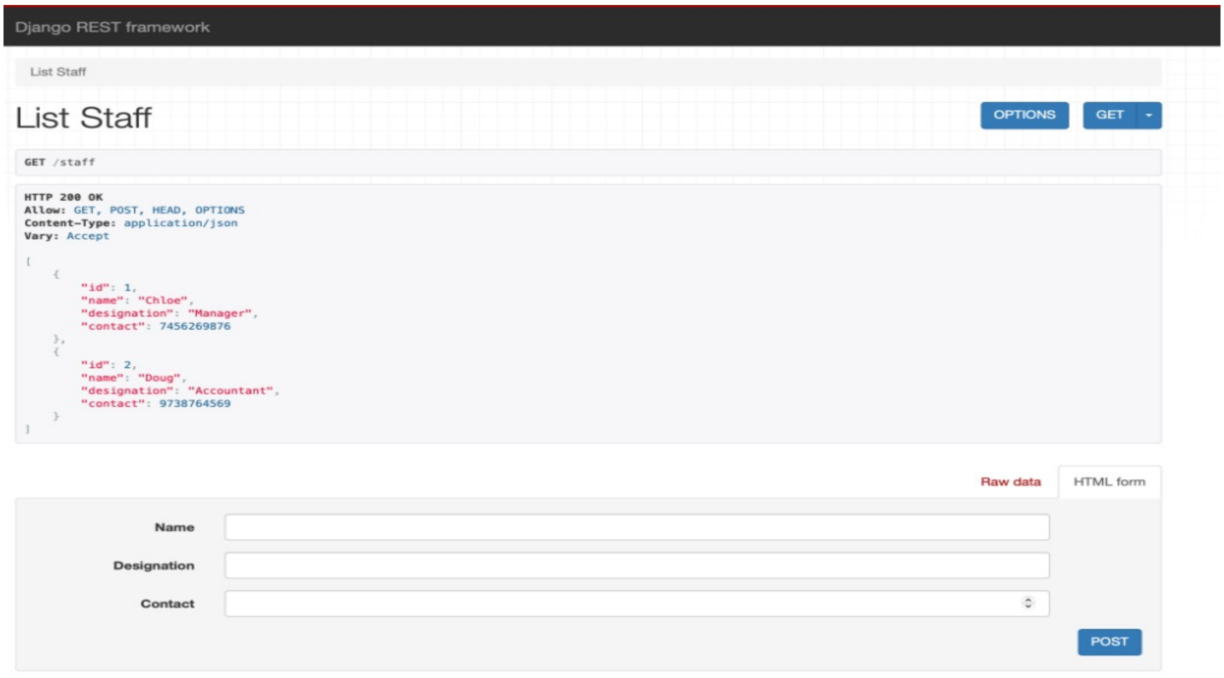


```
PUT /customers/1
```

```
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 1,
    "name": "Rajvardhan",
    "contact": 709298747
}
```

Name: Rajvardhan

Contact: 709298747

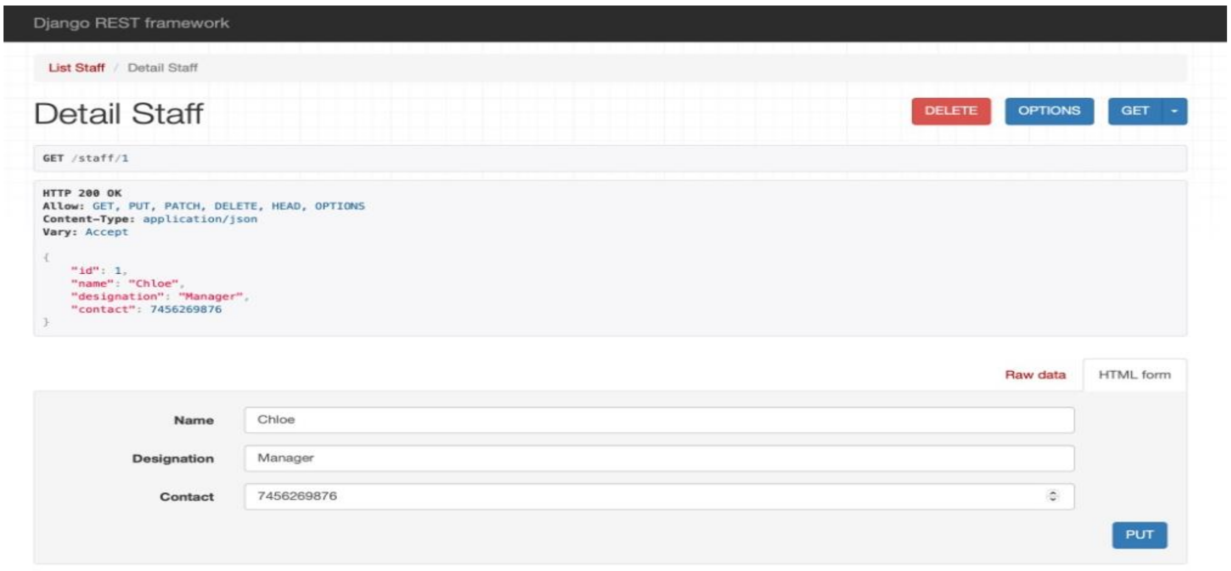


List Staff

```
GET /staff
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 1,
        "name": "Chloe",
        "designation": "Manager",
        "contact": 7456269876
    },
    {
        "id": 2,
        "name": "Doug",
        "designation": "Accountant",
        "contact": 9738764569
    }
]
```

Name	
Designation	
Contact	

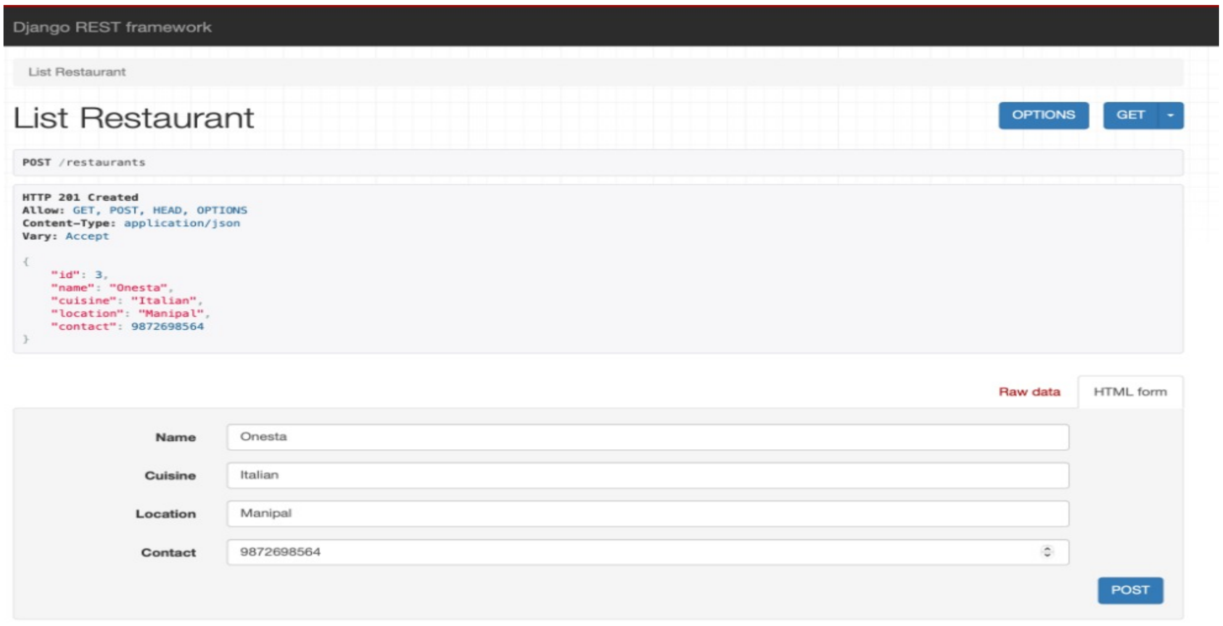


Detail Staff

```
GET /staff/1
```

```
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 1,
    "name": "Chloe",
    "designation": "Manager",
    "contact": 7456269876
}
```

Name	Chloe
Designation	Manager
Contact	7456269876

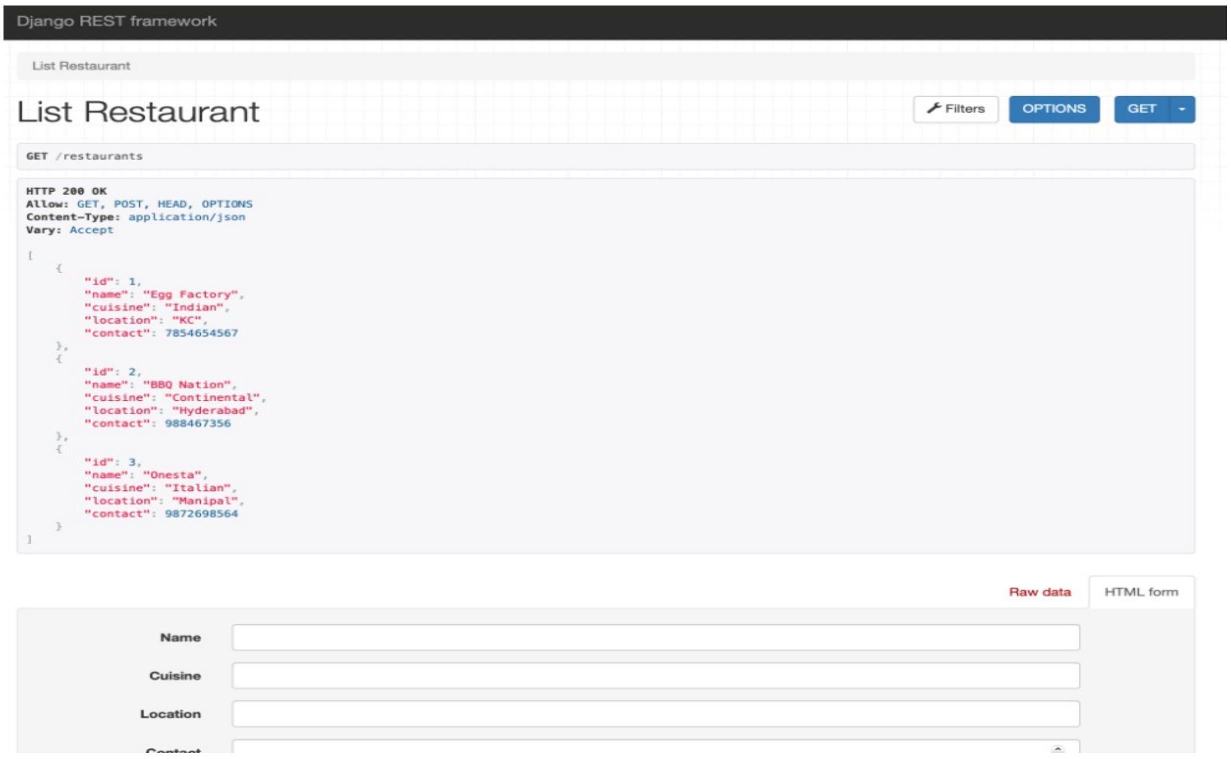


List Restaurant

```
POST /restaurants
```

```
HTTP 201 Created
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 3,
    "name": "Onesta",
    "cuisine": "Italian",
    "location": "Manipal",
    "contact": 9872698564
}
```

Name	Onesta
Cuisine	Italian
Location	Manipal
Contact	9872698564

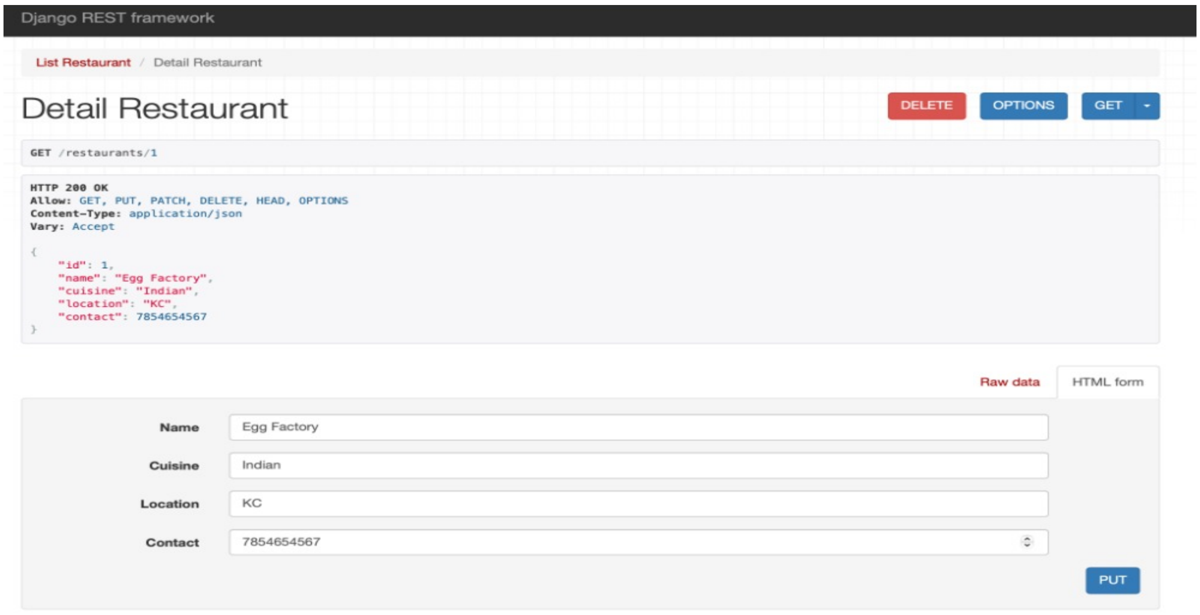


List Restaurant

```
GET /restaurants
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 1,
        "name": "Egg Factory",
        "cuisine": "Indian",
        "location": "KC",
        "contact": 7854654567
    },
    {
        "id": 2,
        "name": "BBQ Nation",
        "cuisine": "Continental",
        "location": "Hyderabad",
        "contact": 988467356
    },
    {
        "id": 3,
        "name": "Onesta",
        "cuisine": "Italian",
        "location": "Manipal",
        "contact": 9872698564
    }
]
```

Name	
Cuisine	
Location	
Contact	



Detail Restaurant

```
GET /restaurants/1
```

```
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 1,
    "name": "Egg Factory",
    "cuisine": "Indian",
    "location": "KC",
    "contact": 7854654567
}
```

Name	Egg Factory
Cuisine	Indian
Location	KC
Contact	7854654567

P4)

models.py:

```
from django.db import models

# Create your models here.

class Category(models.Model):

name = models.CharField(max_length=100)

def __str__(self):

return self.name

class Service(models.Model):

name = models.CharField(max_length=200)

provider = models.CharField(max_length=200)

location = models.CharField(max_length=200)

category = models.ForeignKey(Category,on_delete=models.CASCADE)

cost = models.IntegerField()

def __str__(self):

return self.name
```

```
class Customer(models.Model):
name = models.CharField(max_length=100)
contact = models.PositiveBigIntegerField()
def __str__(self):
return self.name
class ServiceRequested(models.Model):
customer = models.ForeignKey(Customer,on_delete=models.CASCADE)
service = models.ForeignKey(Service,on_delete=models.CASCADE)
```

serializers.py:

```
from django.db.models import fields
from rest_framework import serializers
from .models import *
class CategorySerializer(serializers.ModelSerializer):
class Meta:
fields = '__all__'
model = Category
class ServiceSerializer(serializers.ModelSerializer):
class Meta:
fields = '__all__'
model = Service
class CustomerSerializer(serializers.ModelSerializer):
class Meta:
fields = '__all__'
model = Customer
class ServiceRequestedSerializer(serializers.ModelSerializer):
class Meta:
```

```
fields = '__all__'
model = ServiceRequested
```

views.py:

```
from django.shortcuts import render
from rest_framework import generics,filters
from .models import *
from .serializers import *
# Create your views here.
class ListCategory(generics.ListCreateAPIView):
queryset = Category.objects.all()
serializer_class = CategorySerializer
class DetailCategory(generics.RetrieveUpdateDestroyAPIView):
queryset = Category.objects.all()
serializer_class = CategorySerializerclass
ListService(generics.ListCreateAPIView):
queryset = Service.objects.all()
serializer_class = ServiceSerializer
filter_backends = [filters.SearchFilter]
search_fields = ['name','location']
class DetailService(generics.RetrieveUpdateDestroyAPIView):
queryset = Service.objects.all()
serializer_class = ServiceSerializer
class ListCustomer(generics.ListCreateAPIView):
queryset = Customer.objects.all()
serializer_class = CustomerSerializer
class DetailCustomer(generics.RetrieveUpdateDestroyAPIView):
queryset = Customer.objects.all()
```

```
serializer_class = CustomerSerializer

class ListServiceRequested(generics.ListCreateAPIView):

queryset = ServiceRequested.objects.all()

serializer_class = ServiceRequestedSerializer

class DetailServiceRequested(generics.RetrieveUpdateDestroyAPIView):

queryset = ServiceRequested.objects.all()

serializer_class = ServiceRequestedSerializer
```

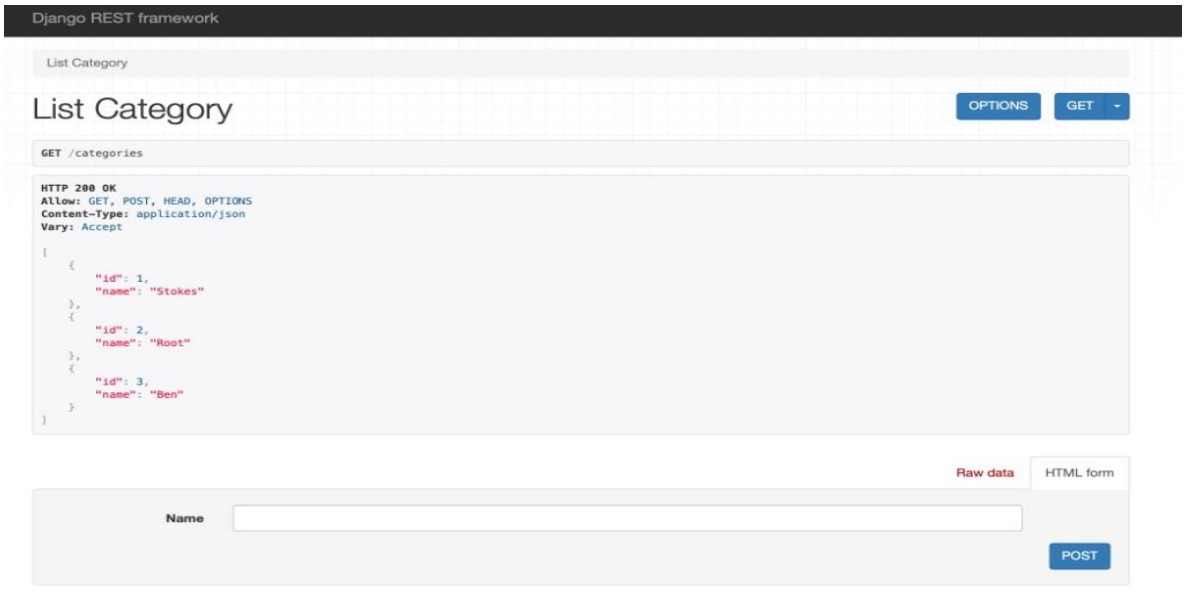
<u>urls.py:</u>

```
from django.urls import path

from .views import *

urlpatterns = [

path('categories',ListCategory.as_view(),name="categories"),

path('services',ListService.as_view(),name="services"),

path('customers',ListCustomer.as_view(),name="customers"),

path('requests',ListServiceRequested.as_view(),name = "requests"),

path('categories/<int:pk>',DetailCategory.as_view(),name="category"),

path('services/<int:pk>',DetailService.as_view(),name="service"),path('custom
ers/<int:pk>',DetailCustomer.as_view(),name="customer"),

path('requests/<int:pk>',DetailServiceRequested.as_view(),name = "request"),

]
```

<u>urls.py:</u>

```
from django.contrib import admin

from django.urls import path,include

urlpatterns = [

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

#path('',include("prob1.urls")),

#path('',include("prob2.urls")),
#path('',include("prob3.urls")),
path('',include("prob4.urls")),
]
```

<u>Output:</u>

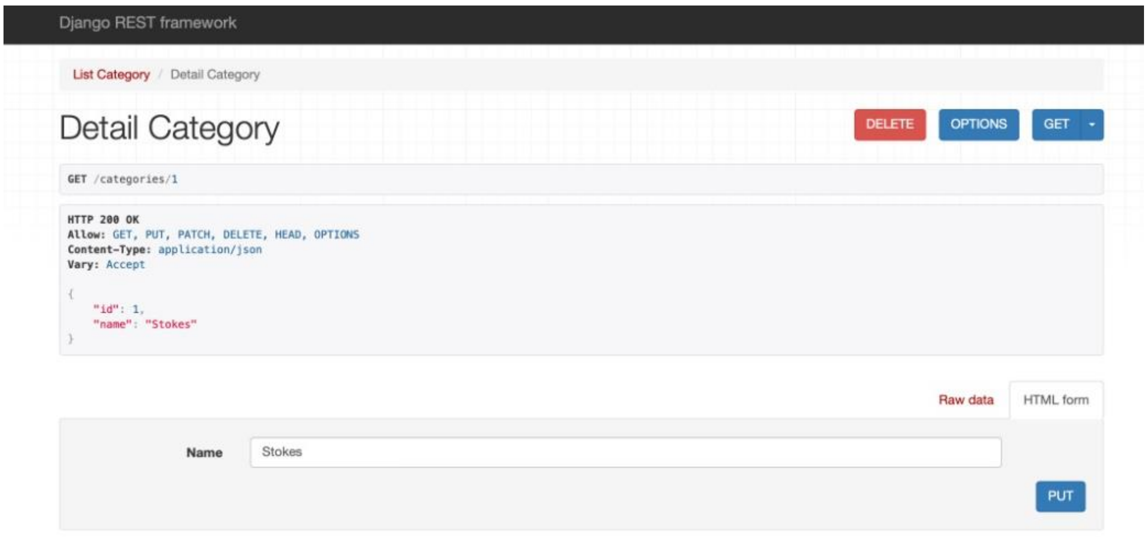


List Category

```
GET /categories
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 1,
        "name": "Stokes"
    },
    {
        "id": 2,
        "name": "Root"
    },
    {
        "id": 3,
        "name": "Ben"
    }
]
```

Name	

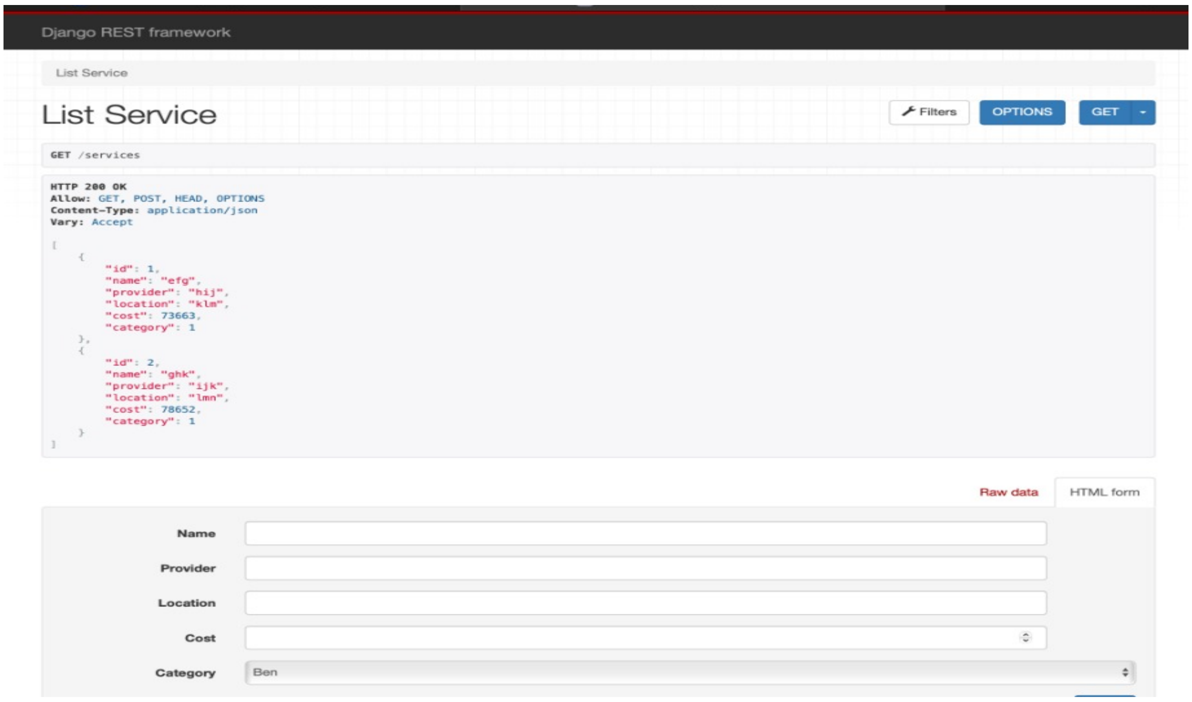


Detail Category

```
GET /categories/1
```

```
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 1,
    "name": "Stokes"
}
```

Name	Stokes

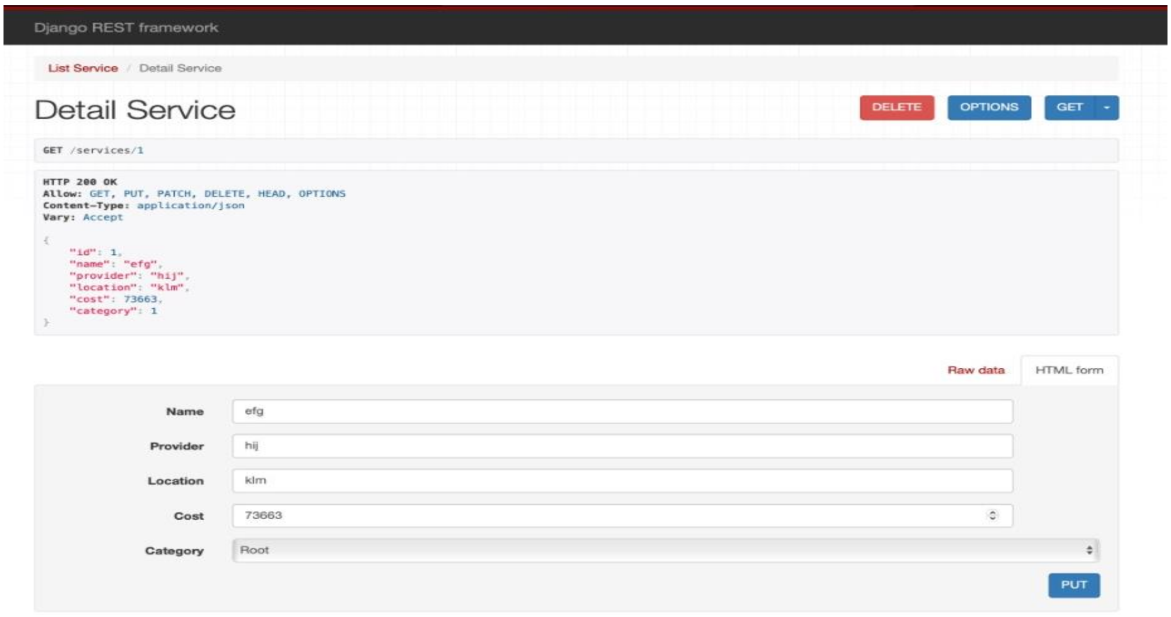


List Service

```
GET /services
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 1,
        "name": "efg",
        "provider": "hij",
        "location": "klm",
        "cost": 73663,
        "category": 1
    },
    {
        "id": 2,
        "name": "ghk",
        "provider": "ijk",
        "location": "lmn",
        "cost": 78652,
        "category": 1
    }
]
```

Name	
Provider	
Location	
Cost	
Category	Ben

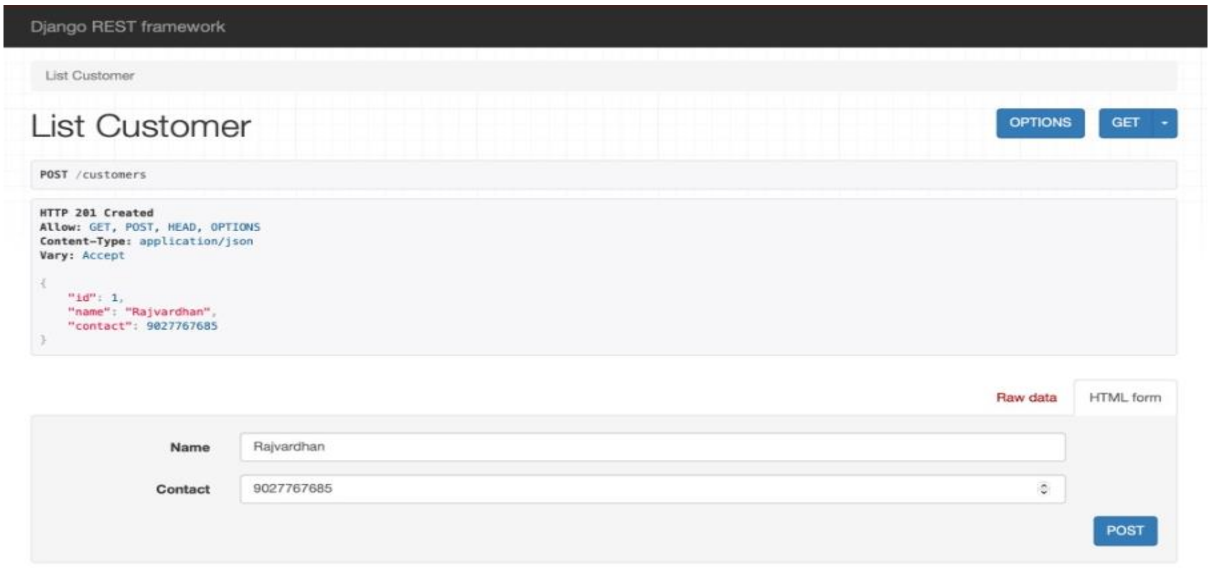


Detail Service

```
GET /services/1
```

```
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 1,
    "name": "efg",
    "provider": "hij",
    "location": "klm",
    "cost": 73663,
    "category": 1
}
```

Name	efg
Provider	hij
Location	klm
Cost	73663
Category	Root

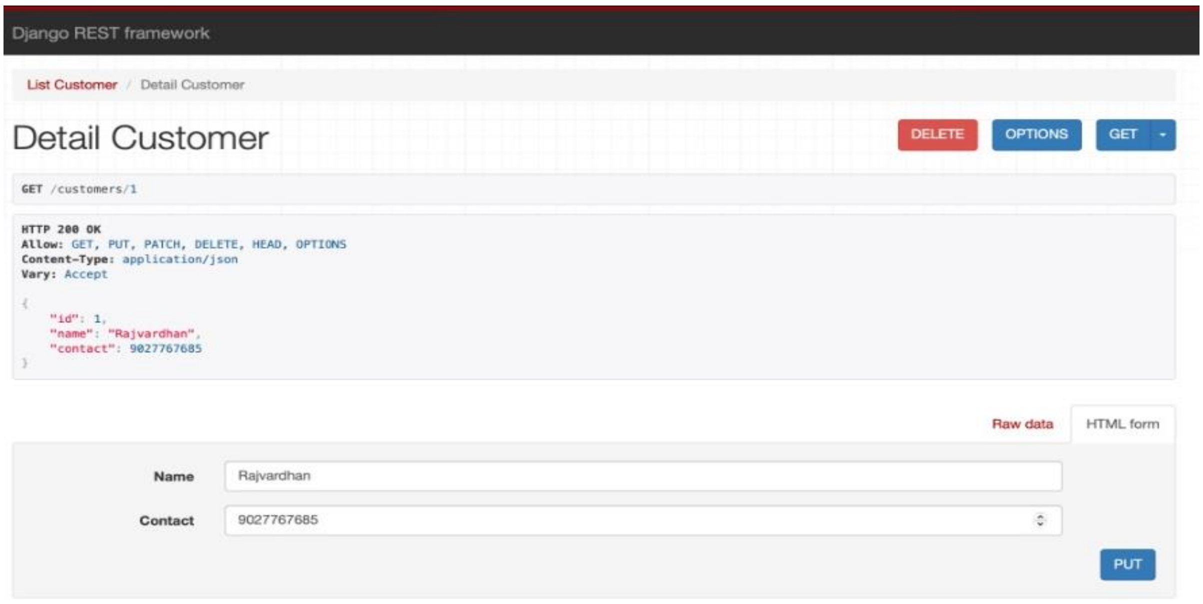


List Customer

```
POST /customers
```

```
HTTP 201 Created
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 1,
    "name": "Rajvardhan",
    "contact": 9027767685
}
```

Name	Rajvardhan
Contact	9027767685

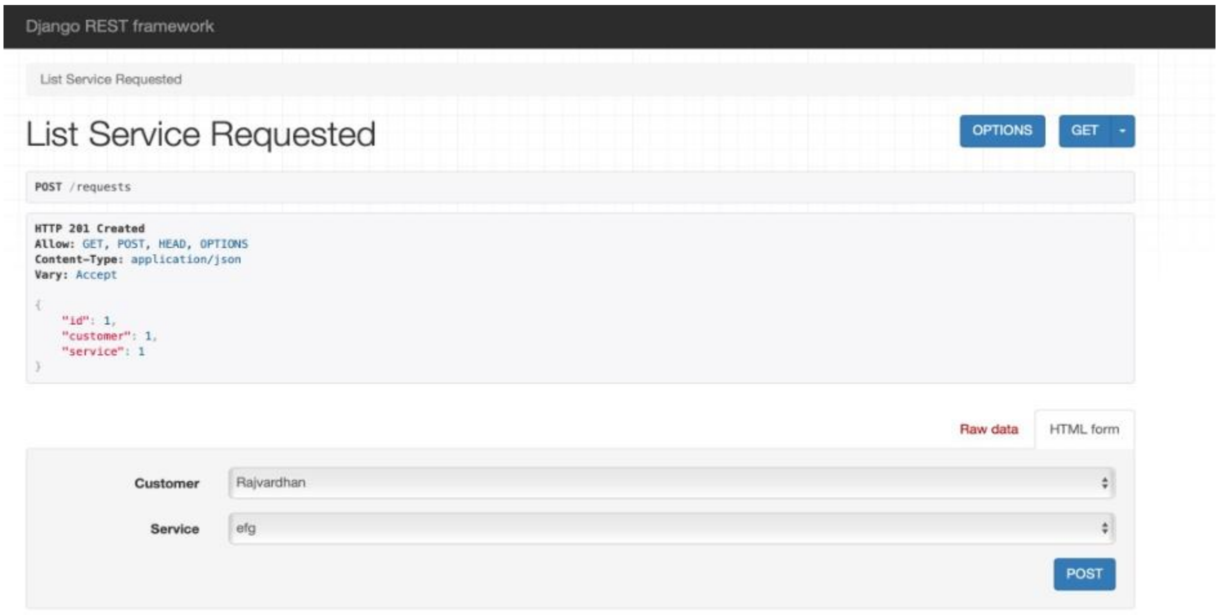


Detail Customer

```
GET /customers/1
```

```
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 1,
    "name": "Rajvardhan",
    "contact": 9027767685
}
```

Name	Rajvardhan
Contact	9027767685

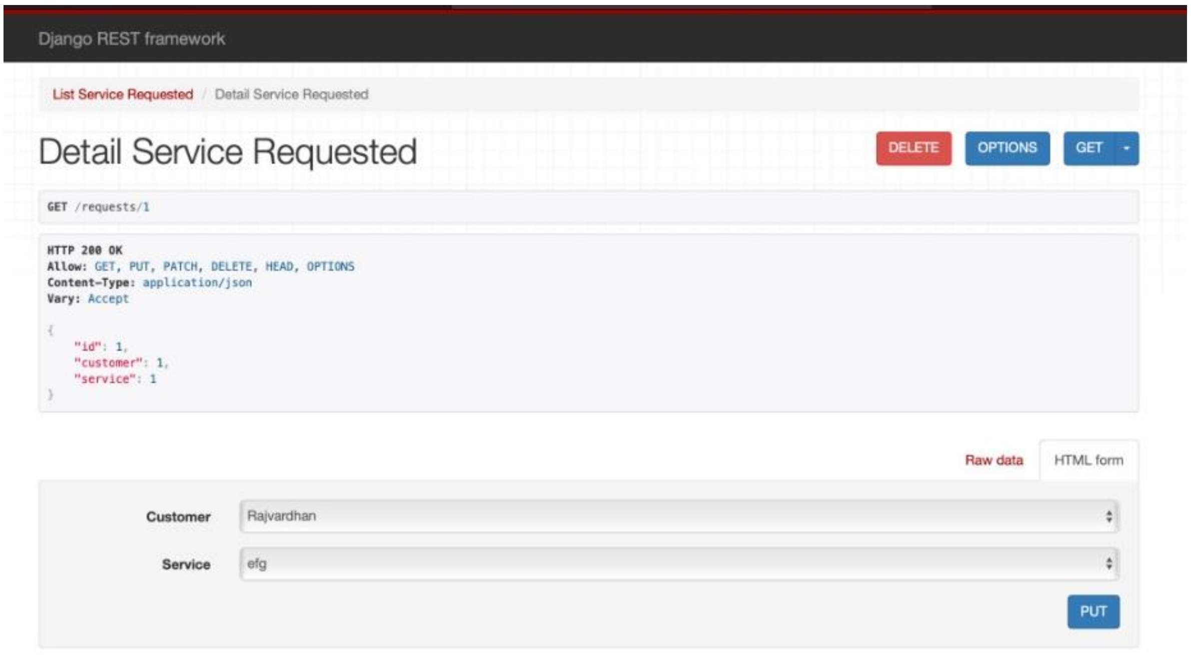


List Service Requested

```
POST /requests
```

```
HTTP 201 Created
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 1,
    "customer": 1,
    "service": 1
}
```

Customer	Rajvardhan
Service	efg



Detail Service Requested

```
GET /requests/1
```

```
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 1,
    "customer": 1,
    "service": 1
}
```

Customer	Rajvardhan
Service	efg