

DJANGO REST FRAMEWORK CRUD

A. DJANGO(settings.py)

1. Install the rest framework and corsheaders using the following commands.

```
Pip install djangorestframework
Python -m pip install django-cors-headers
Pip install markdown
```

2. Add 'rest_framework' and 'corsheaders' to your INSTALLED_APPS setting

```
INSTALLED_APPS = [
    ....
    'rest_framework'
    'corsheaders'
]
```

3. Add a middleware class to listen in on responses

```
MIDDLEWARE = [
    ....
    'corsheaders.middleware.CorsMiddleware',
    'django.middleware.common.CommonMiddleware',
    ....
]
```

4. Configure the behaviour of the middleware in your settings

```
CORS_ORIGIN_ALLOW_ALL = True
```

B. DJANGO(models.py)

```
from django.db import models

class ModelName(models.Model):
    userId=models.AutoField(primary_key=True)
    username=models.CharField(max_length=200)
    password=models.CharField(max_length=200)

    class Meta:
        db_table='ModelName'
```

C. DJANGO(serializers.py)

```
from rest_framework import serializers
from appName.models import ModelName

class ModelNameSerializer(serializers.ModelSerializer):
    class Meta:
        model = ModelName
        fields = "__all__"
```

D. DJANGO GET METHOD(views.py)

```
from django.shortcuts import render
from app1.models import UsersApi
from django.http import HttpResponse, JsonResponse #1
from django.views.decorators.csrf import csrf_exempt #2
from rest_framework.parsers import JSONParser #3
from appName.serializers import ModelNameSerializer

@csrf_exempt #a must to include
def users(request):
    if request.method == 'GET':
        ALL = UsersApi.objects.all()
        serializer = ModelNameSerializer(ALL, many=True)
        return JsonResponse(serializer.data, safe=False)
```

D. DJANGO POST METHOD(views.py)

```
from django.shortcuts import render
from app1.models import UsersApi
from django.http import HttpResponse, JsonResponse #1
from django.views.decorators.csrf import csrf_exempt #2
from rest_framework.parsers import JSONParser #3
from appName.serializers import ModelNameSerializer

@csrf_exempt #a must to include
def users(request):
    if request.method == 'POST':
        Alldata = JSONParser().parse(request)
        serializer = ModelNameSerializer(data=Alldata)
        if serializer.is_valid():
            serializer.save()
        return JsonResponse(serializer.data)
```

COMPILED BY :JUDAH T CHISARE

:JUNIOR FULL STACK DEVELOPER /JUNIOR DATA ENGINEER

CONTACT :0785430774/0777222531

:jaycea247@gmail.com/ Judah.Chisare@cassavasmartech.co.zw

E.DJANGO PUT METHOD([views.py](#))

```
from django.shortcuts import render
from app1.models import UsersApi
from django.http import HttpResponse, JsonResponse #1
from django.views.decorators.csrf import csrf_exempt #2
from rest_framework.parsers import JSONParser #3
from rest_framework.decorators import api_view #4
from rest_framework.response import Response #5
from appName.serializers import ModelNameSerializer

@api_view(['PUT'])
def updateUser(request,pk):
    user = clientss.objects.get(id=pk)
    serializer=clientssSerializer(instance=user,data=request.data)
    if serializer.is_valid():
        serializer.save()
        return JsonResponse(serializer.data,safe=False)
    return Response(serializer.data)
```

F.DJANGO DELETE METHOD([views.py](#))

```
from django.shortcuts import render
from app1.models import UsersApi
from django.http import HttpResponse, JsonResponse #1
from django.views.decorators.csrf import csrf_exempt #2
from rest_framework.parsers import JSONParser #3
from rest_framework.decorators import api_view #4
from rest_framework.response import Response #5
from appName.serializers import ModelNameSerializer

@api_view(['DELETE'])
def userDelete(request,pk):
    user = clientss.objects.get(id=pk)
    user.delete()

    return JsonResponse("user deleted",safe=False)
```

COMPILED BY :JUDAH T CHISARE

:JUNIOR FULL STACK DEVELOPER /JUNIOR DATA ENGINEER

CONTACT :0785430774/0777222531

:jaycea247@gmail.com/ Judah.Chisare@cassavasmartech.co.zw