**API Documentation**

**DJANGO Rest API – FullThrottle Labs**

Prepared by Amit Yadav

Table of Contents

[1 Requirements 3](#_Toc36700749)

[2 Dummy Data & Custom Commands 4](#_Toc36700750)

[3 Database Commands 5](#_Toc36700751)

[4 API Testing 5](#_Toc36700752)

[5 Appendix & References 10](#_Toc36700753)

[Exceptions 10](#_Toc36700754)

# Requirements

Django Python Web application for given models has been implemented. Uses of libraries is registered in Requirements.txt

Serialization, Authentication are implemented using Rest\_Framework (DRF). API is implemented separately as a separate application in this project so that it can be isolated & can be reused as per requirement.

Hosting is done on AWS Cloud using AWS ElasticBeanStalk & on Heroku. Github Repostiory can also be cloned to run in local.

Results from API testing run on local server can also be found in the documentation.

**Packages Used –**

autopep8==1.5

Django==2.1.5

djangorestframework==3.11.0

drf-nested-serializer==0.1.0

factory-boy==2.12.0

Faker==4.0.2

pycodestyle==2.5.0

python-dateutil==2.8.1

pytz==2019.3

six==1.14.0

sqlparse==0.3.1

text-unidecode==1.3

**TEST JSON File Provided – Readme.md**

** **

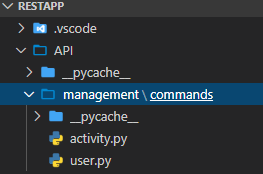
# Dummy Data & Custom Commands

Here in the code, factory\_boy is the package which is used. It involves two additional packages as Faker, Factory which have providers that can used to generate dummy data. It can also be used as standard providers & also local providers are available to. It is used here to generate Dummy Data.

**Path** to locate these files in application is –

API/management/commands/user.py

API/management/commands/activity.py



**Model** used – **UserFactory** & **ActivityPeriodFactory .**

**Library** Used –

import factory

**Python Custom Commands :**

Commands below can be used accordingly as per Heroku( preceed heroku run ) & Local server -

**Dummy Users Data creation :**

python manage.py runserver user --createuser 5

**Dummy Activities Data creation :**

python manage.py runserver activity –createactivity 5

Above Command creates 5 of each dummy data in respective models.

# Database Commands

Command to **migrate** -

python manage.py migrate

Command to **makemigrations** in db

python manage.py makemigrations API

Command to create **Superuser**

python manage.py createsuperuser

# API Testing

API testing is done using POSTMAN. Only **ADMIN** is given rights to access API Data. Rest\_Framework authentication is enabled for this application.

**ADMIN Credentials :**

Username : admin

Password : arbyadav

**Authentication URL –** <http://127.0.0.1:8000/api/auth/>

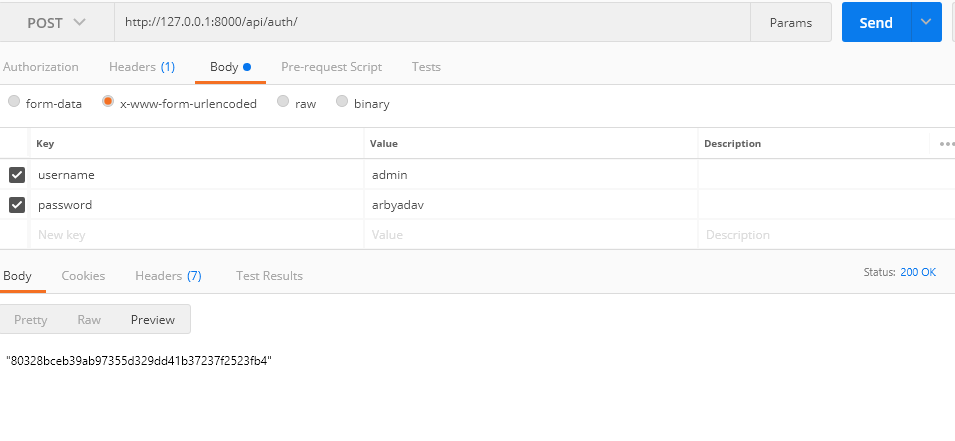
The above URL can be used to generate token as the implementation is based on Token Authentication. Token Key generated can be used to access API and retrieve lists of users. Any method can be worked with this API. It would fail authentication for wrong credentials & for other users with no super access.

**Method** : POST , OPTIONS

**Content-Type** : application/x-www-form-urlencoded

**Credentials** should be passed through **BODY**

**Status :** 200 OK



**ALL USERS LIST URL :** <http://127.0.0.1:8000/api/users_list/>

**ALL MEMBERS LIST URL :** <http://127.0.0.1:8000/api/members_list/>

**ALL ACTIVITIES LIST URL :** <http://127.0.0.1:8000/api/activities_list/> ( Post purposely deactivated in code)

The above URL’s can be used to Retrieve Users List from Database. It also allows POST Method to create new data. To access this you need token\_key obtained from Authentication URL. It should be passed in headers as – Authorization : Token {token\_key}

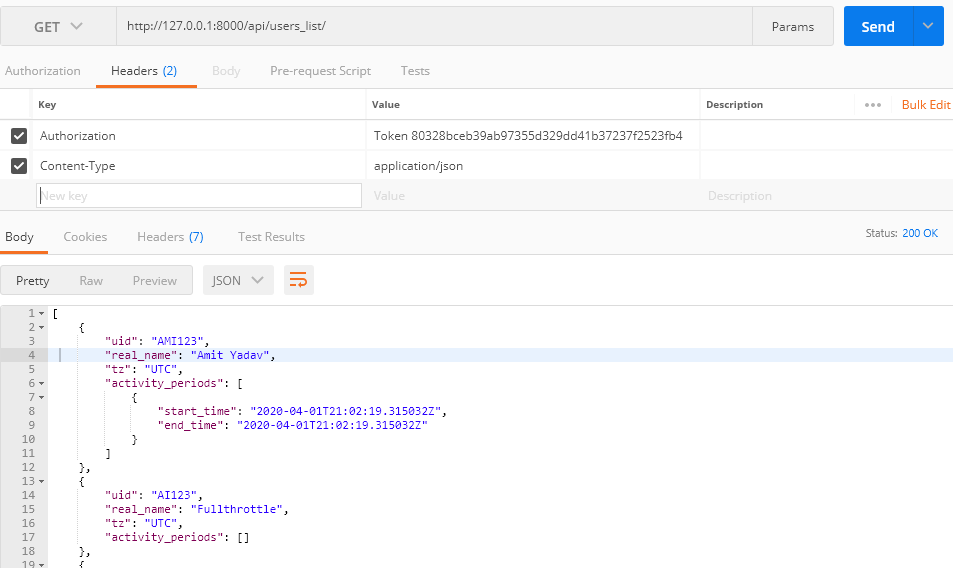
**Allows Method** : GET, POST, OPTIONS

**Content-Type** : application/json

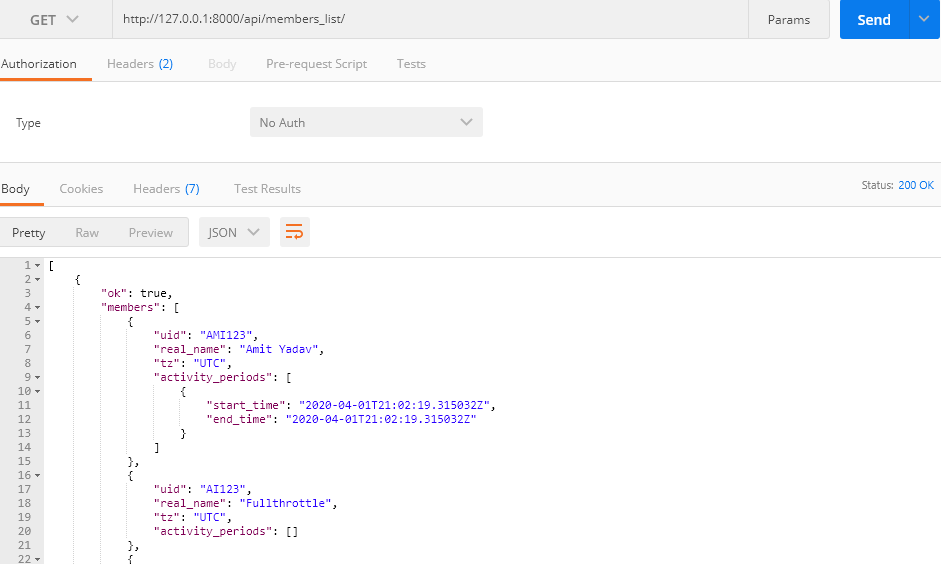
**Headers** –

**Authorization** : Token {token\_key}

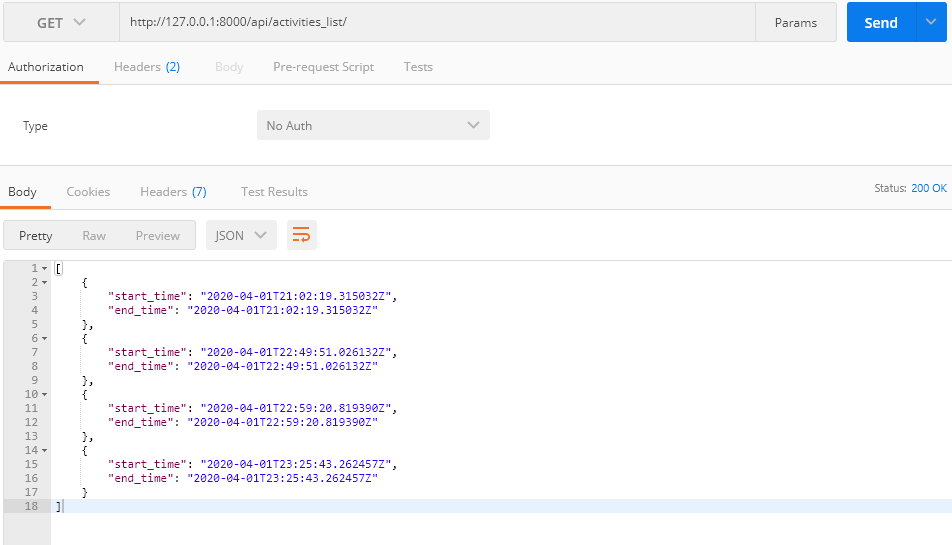
**Users List –**



**Members\_List –**



**Activities List –**



**USERS DETAIL URL :** [http://127.0.0.1:8000/api/users\_list/](http://127.0.0.1:8000/api/user/)1/

**ACTIVITIES DETAIL URL :** <http://127.0.0.1:8000/api/activities_list/1/> ( only GET Activated in code )

**MEMBERS DETAIL URL :** <http://127.0.0.1:8000/api/members_list/1/> ( only GET Activated in code )

The above URL’s can be used to retrieve User List from Database based on ID. It also allows GET Method to check user data as per ID passed. It also allows you to update resource using PUT Method and delete resource using DELETE METHOD.

To access this you need token\_key obtained from Authentication URL. It should be passed in headers as – Authorization : Token {token\_key}

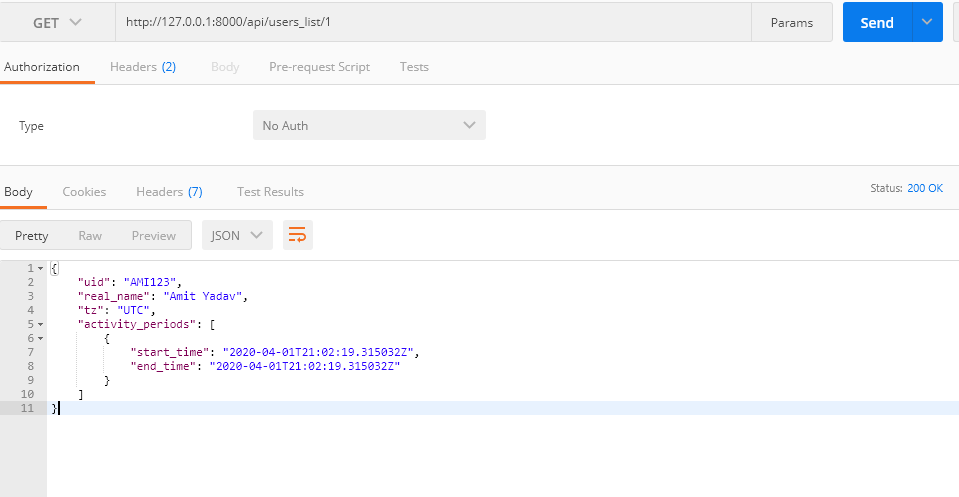
**Allows Method** : GET, PUT, DELETE, OPTIONS

**Content-Type** : application/json

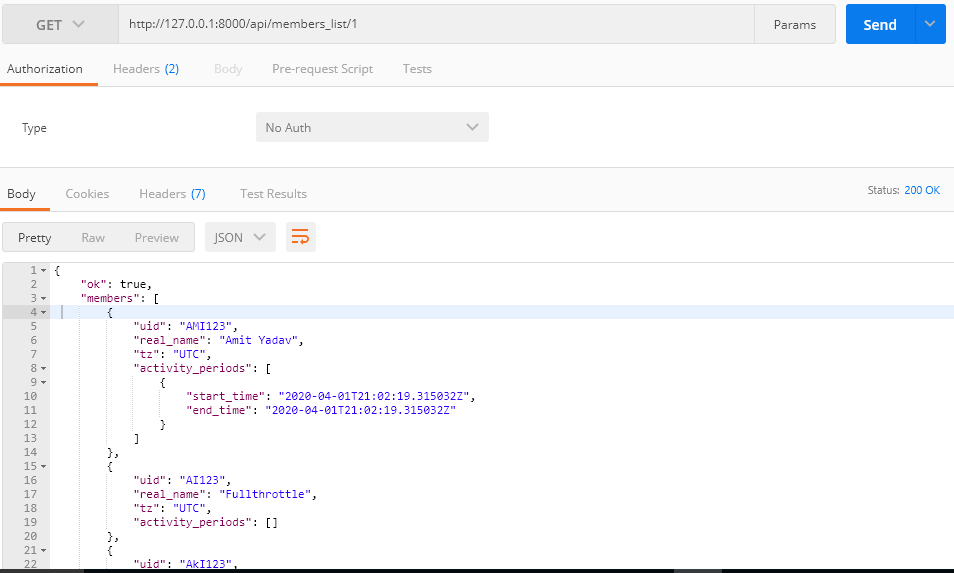
**Headers** –

**Authorization** : Token {token\_key}

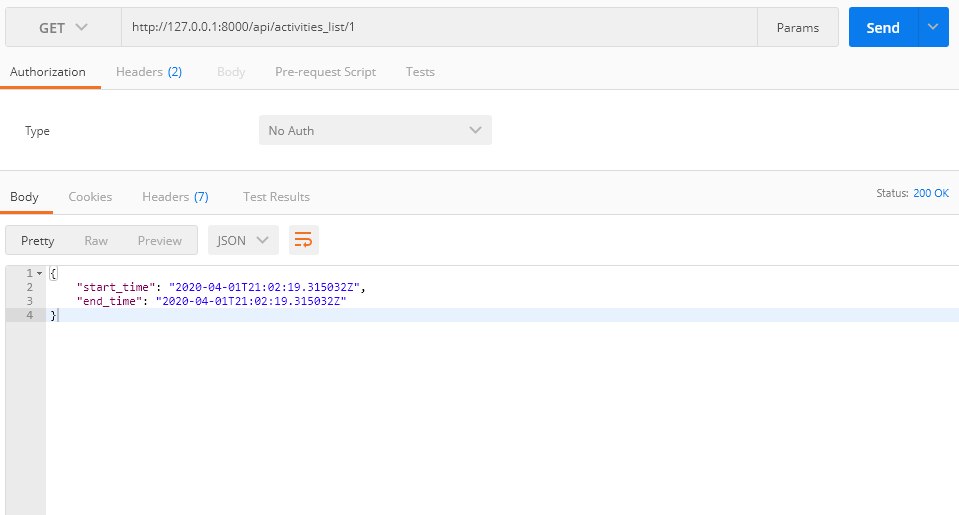
**Users Detail –**



**Members Detail -**



**Activity Detail -**



# Appendix & References

## Exceptions

No exceptions

**References –** Django Documentation, Django Rest Framework Documentation

**Thanks,**

**Amit Yadav**