# Documentation for the Mobile teams
# on
# “Accessing the database from webhub using APIs”

**What is an API ?**

API stands for Application Programming Interface. It is a set of functions and procedures that allow the creation of applications which access the features or data of an operating system, application, or other service.
In our case, API will allow the mobile applications to access and sync their apps with the Peacetrack database.

**Which are the different frameworks available to form the APIs ?**

There are n number of frameworks to implement the APIs in python-django.
A few of them are django-tastypie, django-rest-framework, django-piston.

**Which is the framework implemented in the Systers GSoC project ? Why ?**

I have implemented django-rest-framework. Following are some of the plus points to django-rest-framework.

- Easy to extend their Serializer base class to create custom serializers and resources
- Well-documented
- More flexible than TastyPie, less opinionated, and wonderfully architected.
- API Browser that comes automatically with DRF has proved itself to be invaluable.
- Ease-of-use

**Online Documentation of the Django REST Framework :**

Here is the documentation of Django REST Framework - http://www.django-rest-framework.org/, which contains everything from installation details, tutorials, quick start guide, community support to security and license. It’s indeed a very useful tool. Have a look !

**Authentication with the APIs :**

The APIs are Browsable APIs, so there has to be some authentication so that untrusted people can not access or modify our database. So, the APIs have been authenticated by adding several different authentication classes.
While interacting with the API through the web browser, we can login, and the browser session will provide the required authentication for the requests.

If we're interacting with the API programmatically we need to explicitly provide the authentication credentials on each request.

If we try to create a snippet without authenticating, we'll get an error:

for eg. http://192.168.33.10:8000/api/ptposts/?format=json might provide you with the following - {"detail": "Authentication credentials were not provided."}

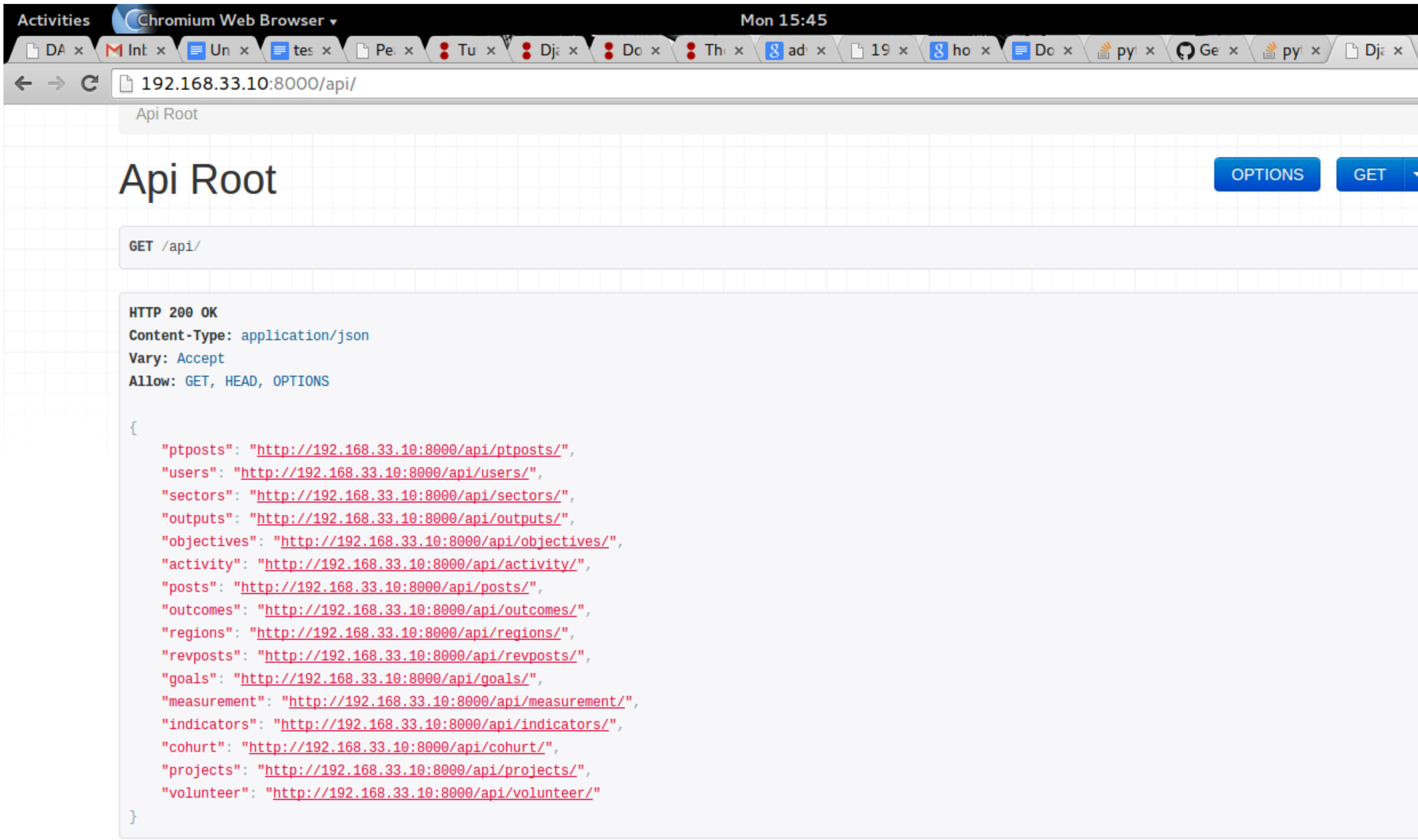
We can make a successful request by including the username and password of one of the users we created earlier.

```
curl -X POST http://127.0.0.1:8000/snippets/ -d "code=print 789" -u tom:password
```

```
{"id": 5, "owner": "tom", "title": "foo", "code": "print 789", "linenos": false, "language":
"python", "style": "friendly"}
```

**Format of the APIs and the way I have implemented them :**

Below is the screenshot of the APIs I have implemented in Django REST framework



Api Root

```
GET /api/
```

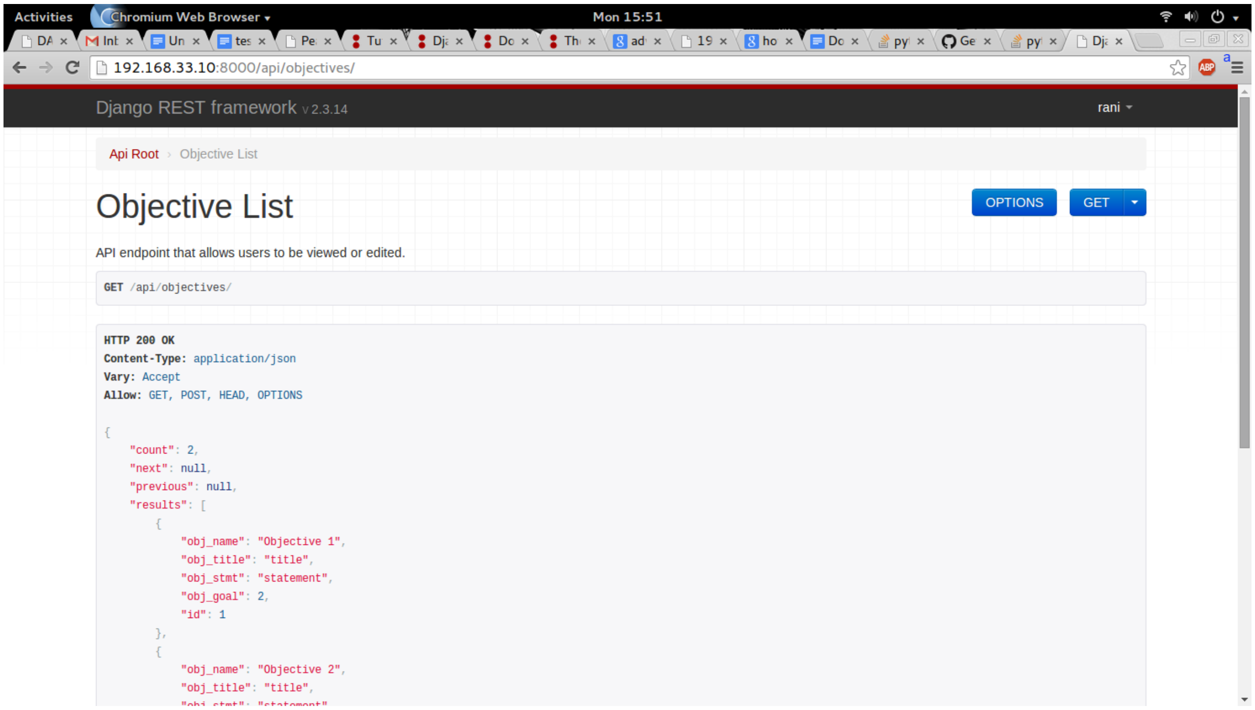
```
HTTP 200 OK
Content-Type: application/json
Vary: Accept
Allow: GET, HEAD, OPTIONS

{
    "ptposts": "http://192.168.33.10:8000/api/ptposts/",
    "users": "http://192.168.33.10:8000/api/users/",
    "sectors": "http://192.168.33.10:8000/api/sectors/",
    "outputs": "http://192.168.33.10:8000/api/outputs/",
    "objectives": "http://192.168.33.10:8000/api/objectives/",
    "activity": "http://192.168.33.10:8000/api/activity/",
    "posts": "http://192.168.33.10:8000/api/posts/",
    "outcomes": "http://192.168.33.10:8000/api/outcomes/",
    "regions": "http://192.168.33.10:8000/api/regions/",
    "revposts": "http://192.168.33.10:8000/api/revposts/",
    "goals": "http://192.168.33.10:8000/api/goals/",
    "measurement": "http://192.168.33.10:8000/api/measurement/",
    "indicators": "http://192.168.33.10:8000/api/indicators/",
    "cohurt": "http://192.168.33.10:8000/api/cohurt/",
    "projects": "http://192.168.33.10:8000/api/projects/",
    "volunteer": "http://192.168.33.10:8000/api/volunteer/"
}
```

How will you access the APIs ?

You will be able to access the API by the url itself. Moreover if you wish to get the data as json object, you will have to add “?format=json” at the end.

Here is an example :
Objectives (with dummy database)



Objective List

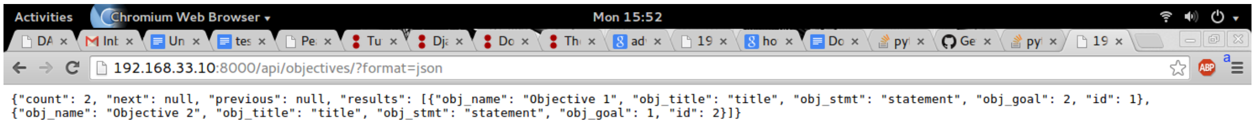
API endpoint that allows users to be viewed or edited.

```
GET /api/objectives/
```

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

{
    "count": 2,
    "next": null,
    "previous": null,
    "results": [
        {
            "obj_name": "Objective 1",
            "obj_title": "title",
            "obj_stmt": "statement",
            "obj_goal": 2,
            "id": 1
        },
        {
            "obj_name": "Objective 2",
            "obj_title": "title",
            "obj_stmt": "statement"
```

Objectives (as json)



```
{"count": 2, "next": null, "previous": null, "results": [{"obj_name": "Objective 1", "obj_title": "title", "obj_stmt": "statement", "obj_goal": 2, "id": 1},
{"obj_name": "Objective 2", "obj_title": "title", "obj_stmt": "statement", "obj_goal": 1, "id": 2}]}
```

### How to push back the data via the APIs ?

Django REST framework provides an HTML form in the browser to add the data back into the main database.
commands Yet to be added for the authentication panel

### Doubts any ?

Do let me know if you have any other doubts/difficulties on the above documentation.
at ranihaileydesai@gmail.com