**API Project**

**1-create repository in git**

**2-after publish repository open it with vscode**

**3-virtualenv venv**

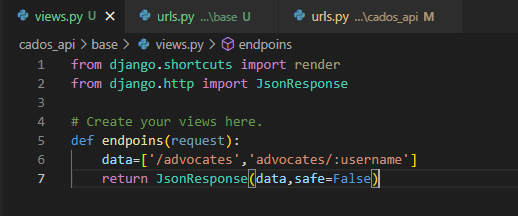
**source/venv/Scripts/activate**

**4-django-admin startproject cados\_api**

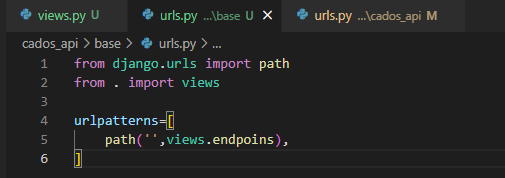
**5-cd cados\_api>python manage.py startapp base**

**6-add base app to installed apps in the settings**

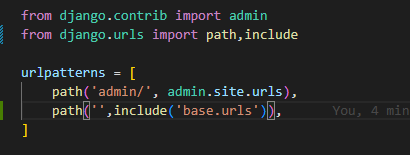
**7-base app views.py:**



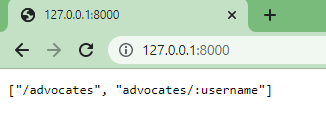
**8-base app urls.py:**



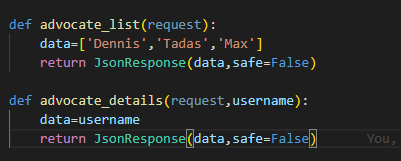
**9-cados\_api urls.py:**



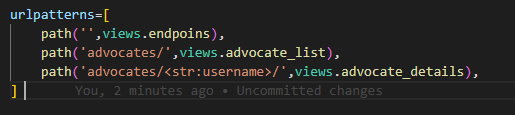
**10-Test:before test do migrate**



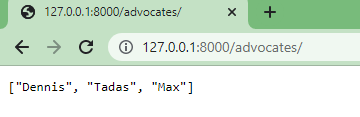
**11-base app views.py:**

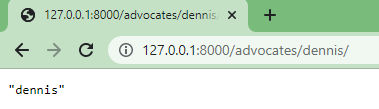


**12-base app urls.py:**



**13-Test:**

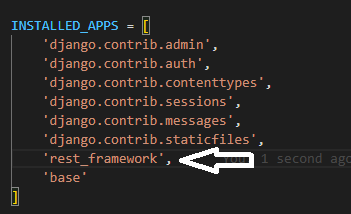




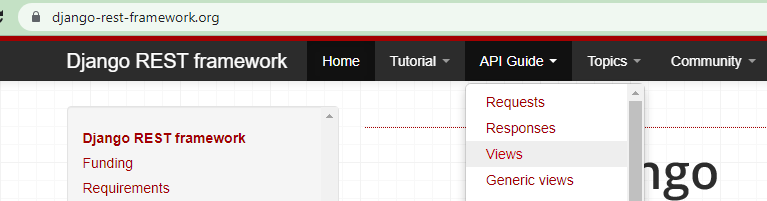
**14-django restframework:**

**pip install djangorestframework**

**then add it to installed apps in settings.py**



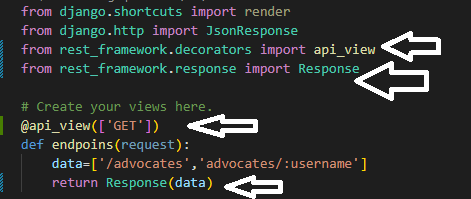
**Good site too learn:**



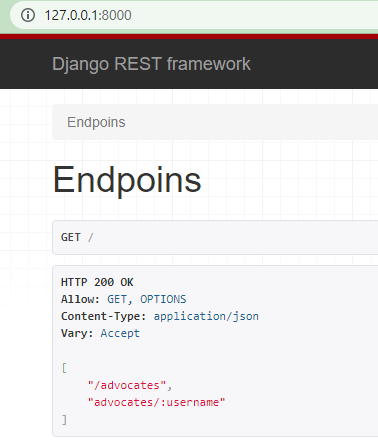
**Good example:**



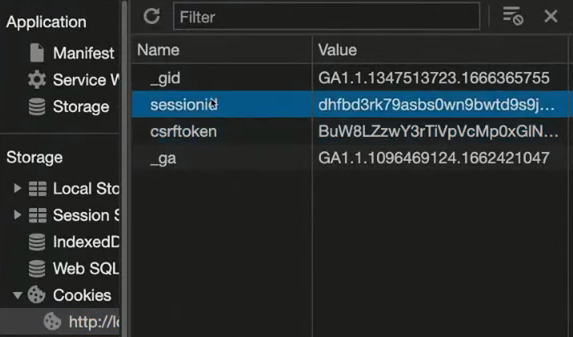
**base app view.py:**



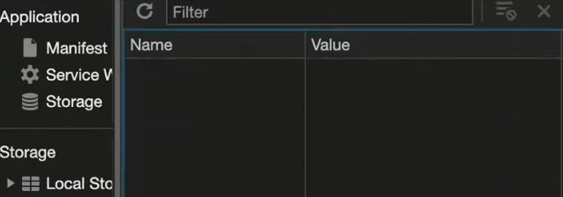
**Result:**



**But for teacher dosen’t work because he said he wroks with other django application and its session id cause this error and to solve it go to inspect:**

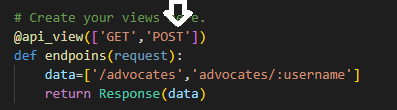


**And delete all of them:**

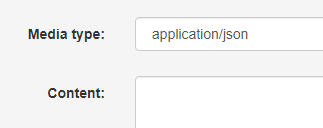


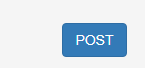
**And problem solved.**

**If we add POST request to decorator:**



**And refresh server we see a form to send data:**

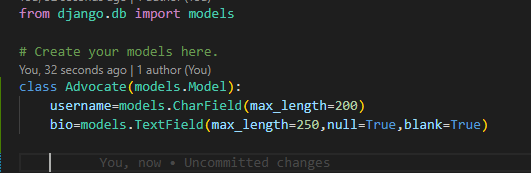




**Ao as you can see django is very powerfull and you don’t need something like postman to see rest apis.**

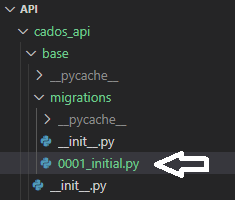
**15-design database in django**

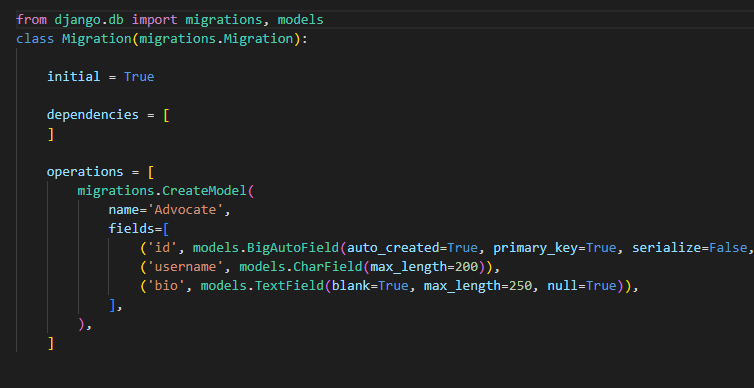
**open models.py for base app:**



**Then migration commands**

**After it open this file:**

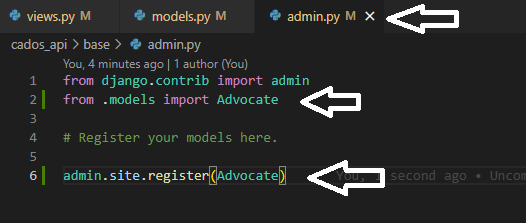


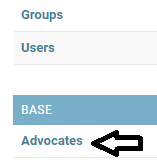


**You see it add id automatically and we don’t need to add id in models.**

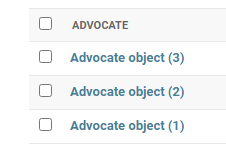
**create super user to can access admin panel.after it login to admin panel.we see users and groups table here but we don’t see Advocate table.**

**admin.py base app:**

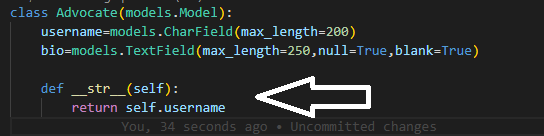




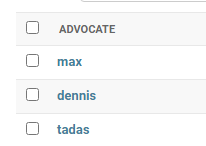
**we add 3 username:**



**To fix shows username instead of object:**

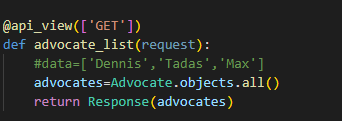


**Result:**

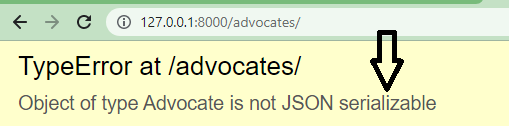


**16-another api to get all usernames**

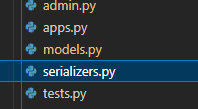


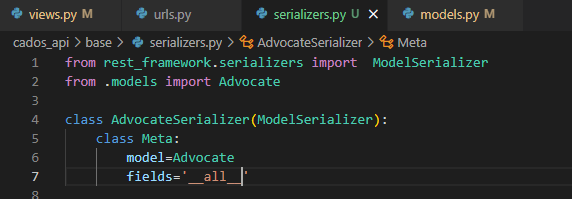


**If we runserver we have error:**



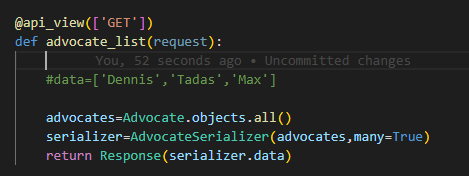
**To fix it create a new file in base app named serializers.py:**





**Views.py:**



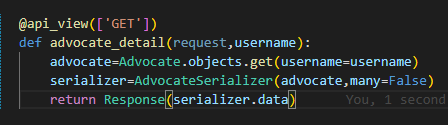


**Test:**

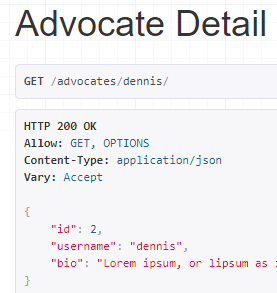




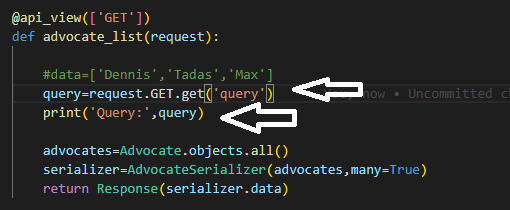
**16-another api to get 1 username**







**17-get the query from the front-end to get apecific username:**



**If we run the server and go to 127.0.0.1:8000/advocates we see query is None:**

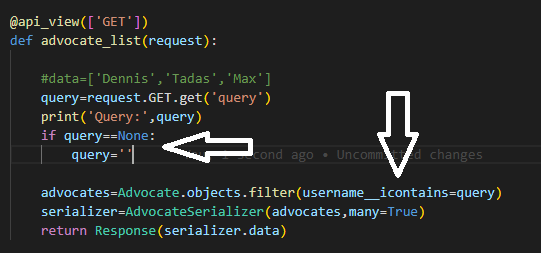


**But if we pass dennis:**





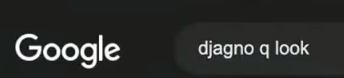
**But all usernames shows.**





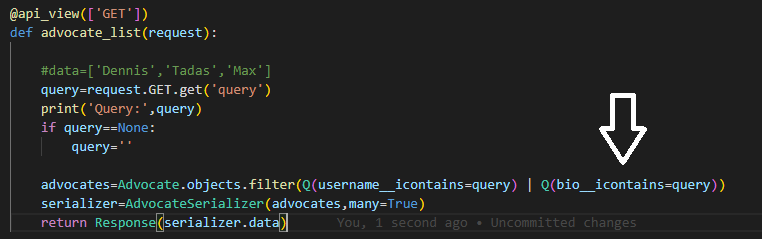


**18-two condition:**





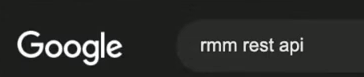


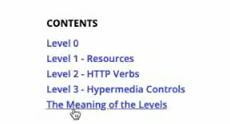






**You should know what is differenve between APIs and Rest APIs.**



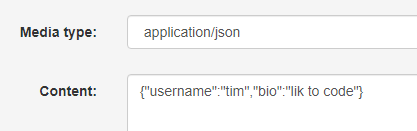


**19-handle GET and POST in one restfull API:**



**Test:**







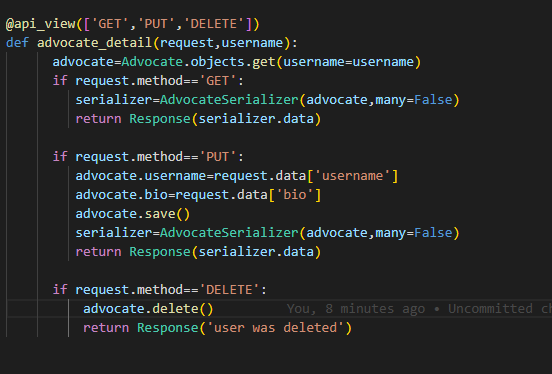


**If we refresh GET request:**





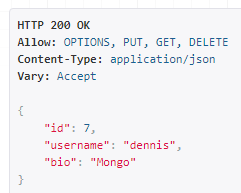
**20-GET PUT DELETE**



**Test:**

**Get 1 username:**





**Put:we want to change dennis informations**





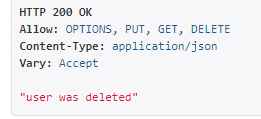


**And it works good.**

**We want to delete javidkhan username:**







**User deleted and if we get all usernames javidkhan was deleted successfully.**



**21-rewrite advocate\_detail view as class based view**

**First comment this view.and then go to this site:**



**We do need to import the APIView then pass the APIView to actuall class:**

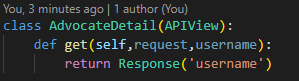




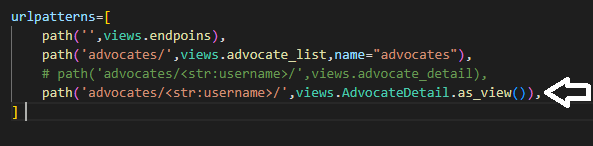
**Then all we need to do is create a Http method.and we don’t longer to check is this a GET request or is this a POST request and we don’t need to worry about any of that.**



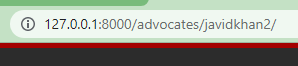
**views.py base app:**



**urls.py base app:**

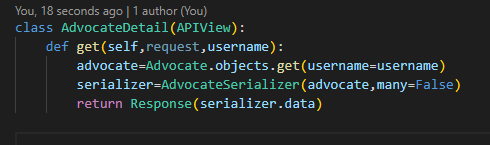


**Test:**



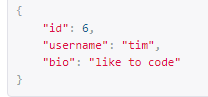


**So we complete the class based view:**

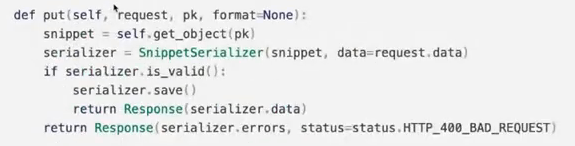


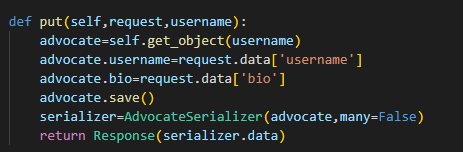
**Test:**



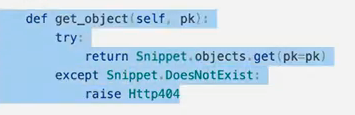


**Then we need post request:**

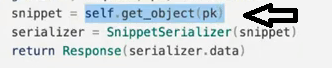




**And for delete we need to get single object:**



**But we modify this method and we use username instead of pk.and for call this method:**

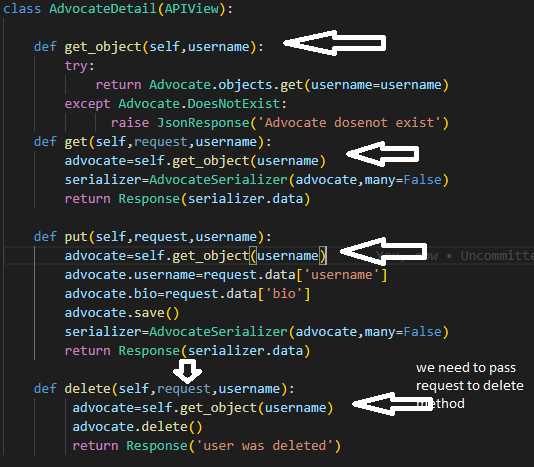


**So we change this line in get and put method:**



**To:**





**Test:**













**Then put:**



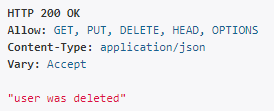
**And good work.**

**Delete Test:**









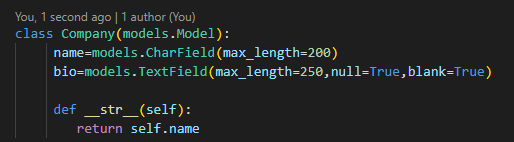




**And good work.**

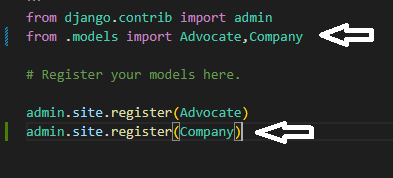
**22-Relationship**

**In base app views.py create another class:**

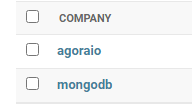


**Migrate it.**

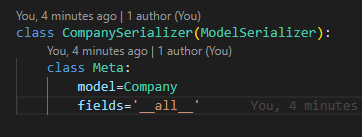
**Admin.py:**



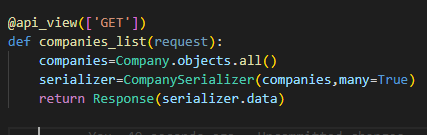
**Add 2 company in admin panel:**



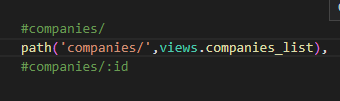
**serializer.py:**



**views.py:**



**urls.py:**



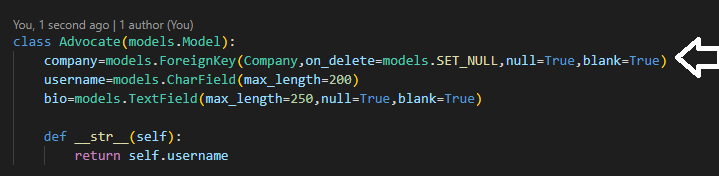
**Test:**





**And good work.**

**We want to make one-to-many relationship.because in real world it is possible a person works multiple companies.so we add company to Advocate model:**



**Migrate.then in admin panel add company to persons in the Advocate table.**

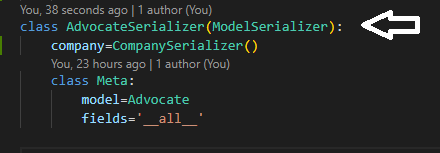
**Test:**





**We see company id .**

**Note:you should put company seriailizer top of the Advocate Serializer.and add this line:**

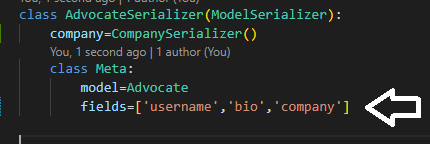


**Test:**





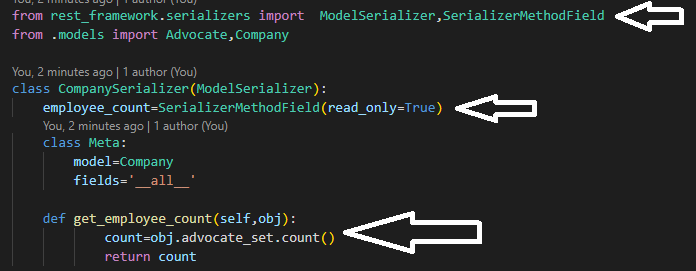
**If you change this line:**



**Test:**



**We can add custome filed to CompanySerializer:**



**Test:**

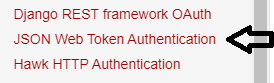


**And work good.**

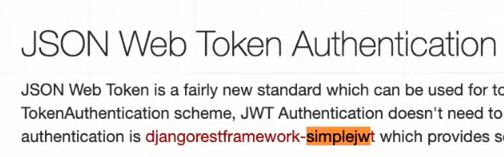
**23:Authentication**

**Json Web Token**





**Serach simple JWT:**





**There is a package I like to use and there is a link to official documentation.**

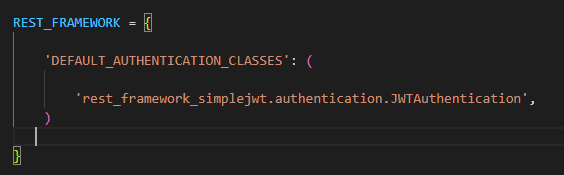




**Installation:**



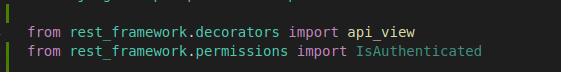
**Then we have to change authentication mechanisem in the settings.py file.so add this to settings.py:**





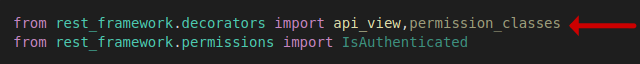
****

**then we import IsAuthenticated in the views.py base app:**

****

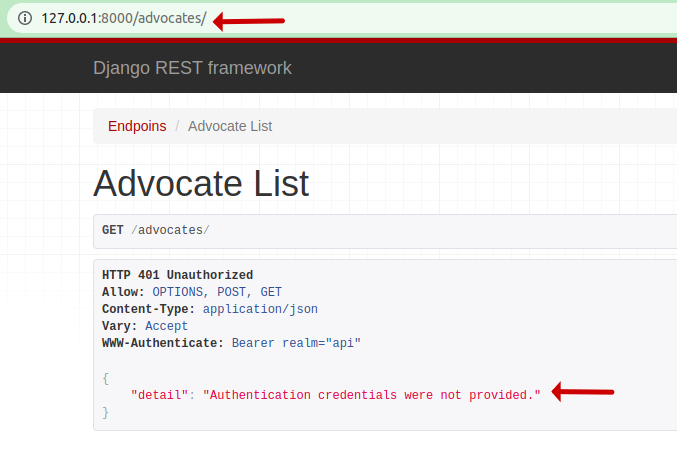
**then we have to import this:**

****

**so fo each request all we gonna to do is say hey wa wanna protect le’ts say all advocate list here:**

****

**the user has to be authenticated to see this.so if we go advocates:**

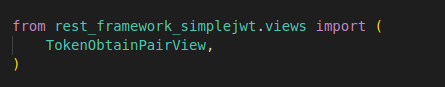
****

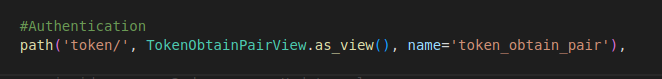
**so how to we get authenticated?first of all we need to get this token and we need to send it on every single request.**

****

**So in urls.py base app we import just TokenObtainPairView and its url path and remove api from the path:**

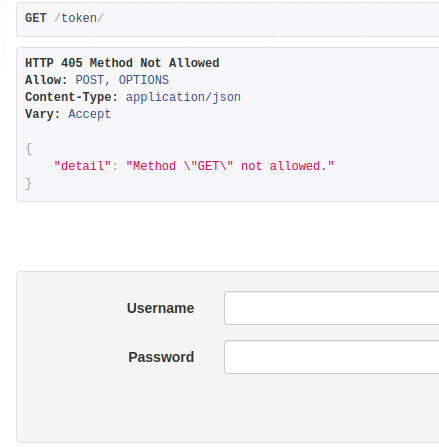
**note:it is a class based view.**

****

**So if we gonna to go endpoint we are gonna to see a form and have the ability to actually create the token itself.**

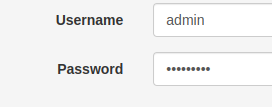
**So if we go token url:**

****

****

****

**so input admin user cerdential:**

****

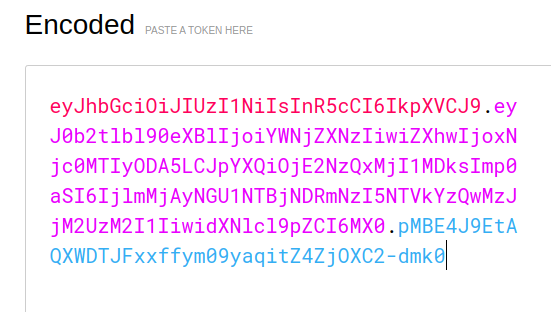
**we have access token and refresh token:**

****

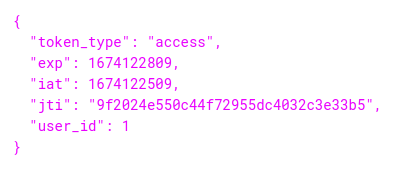
**go to JWT decoder:**

****

**copy access token in it:**

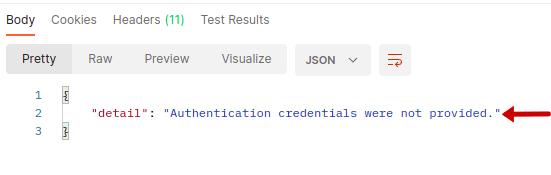
****

**we see all information:**

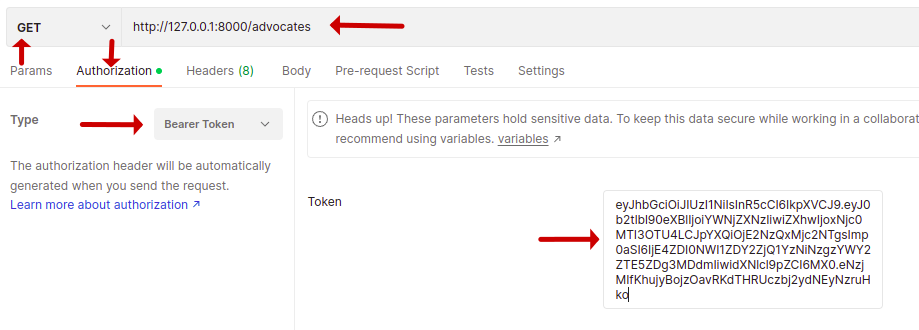
****

**open postman and:**

****

****

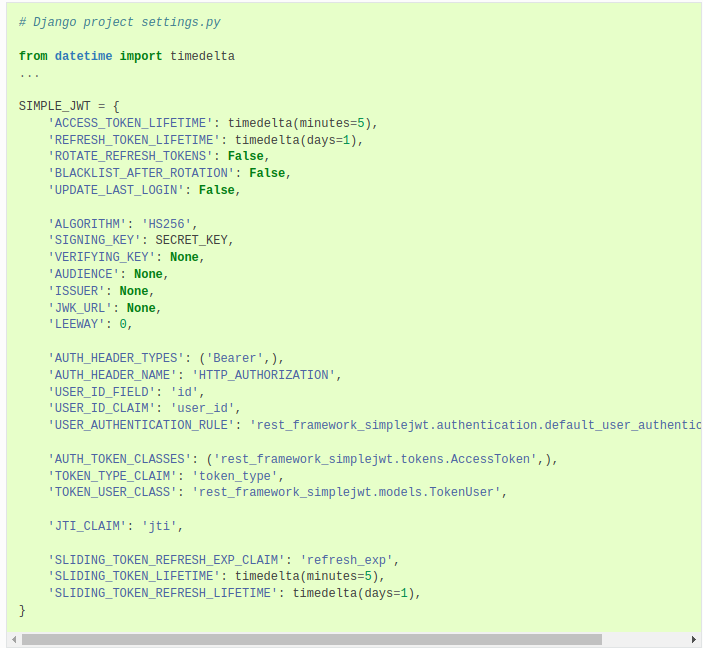
**we need t send token with request and we have access token so copy it:**



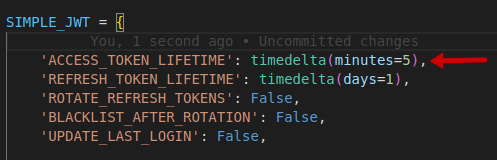


**so we passed a Bearer token and I can actually send this on every single request.one’s that token is sent in the headers I can access that data.**

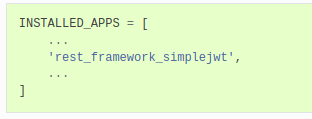
**This is the default configuration for the tokens:**

****

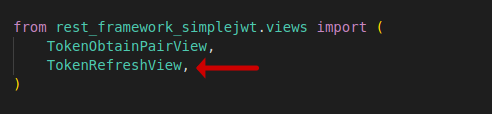
**copy these configurations to settings.py.**

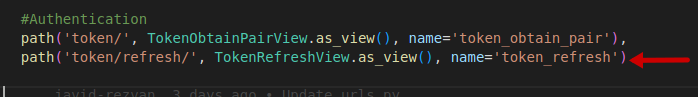
****

**As you see life time of the access token is 5 minute.and we have to add this to installed apps:**

****

**add refrsh token endpoint:**

****

****

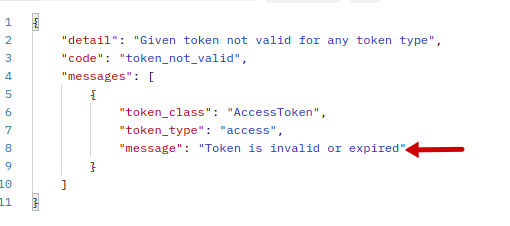
**we remove api of api/token/refresh and change it to token/refresh.**

**Test:**

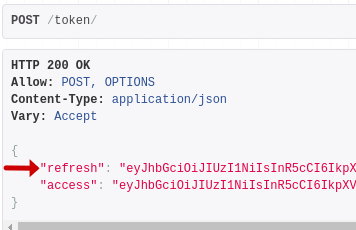
****

****

**the token has expired:**

****

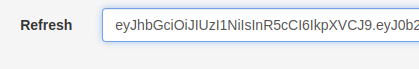
**so we need to refresh token.so copy the refrsh token:**

****

**go to this url:**

****

**paste refresh token here and then post it:**

****

**gave us a new access token:**

****

**and we can use it to accsess this endpoint in the postman**

****

**and we can access that endpoint:**

****

**everything is good.**

**Very important note:**

**Teacher changed access token life time to 1 second so it looks before we get access to the endpoint with that token it will expired but it worked!solution that teacher used was to unsitall djangorestframework-simplejwt and install another vesion:**

****

**and it worked.**

**cros-origin:**

**cross-origin resources sharing basically uh you can block other applications from requesting access to your API before you give them permission so I want to show you how to configure that**