**REST SOCKET DJANGO REST FRAMEWORK**

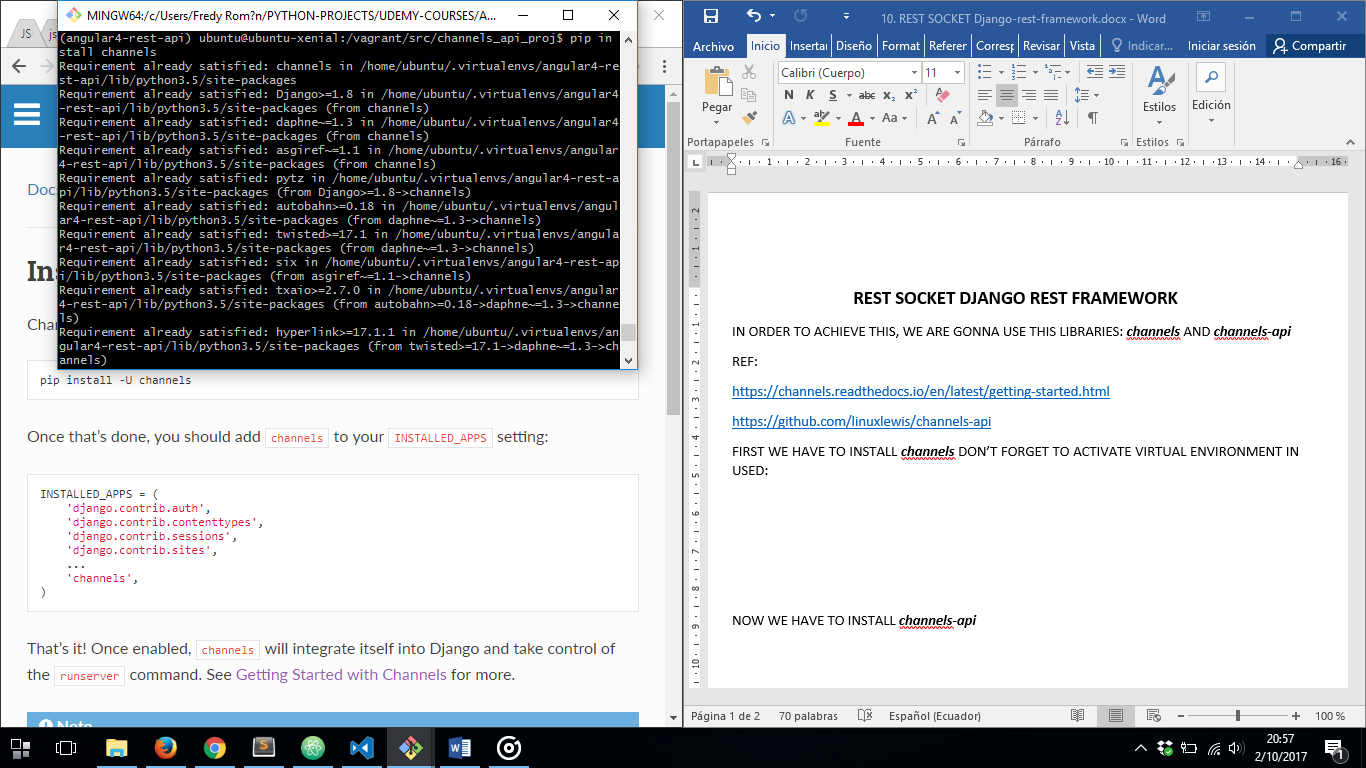
IN ORDER TO ACHIEVE THIS, WE ARE GONNA USE THIS LIBRARIES: ***channels*** AND ***channels-api***

REF:

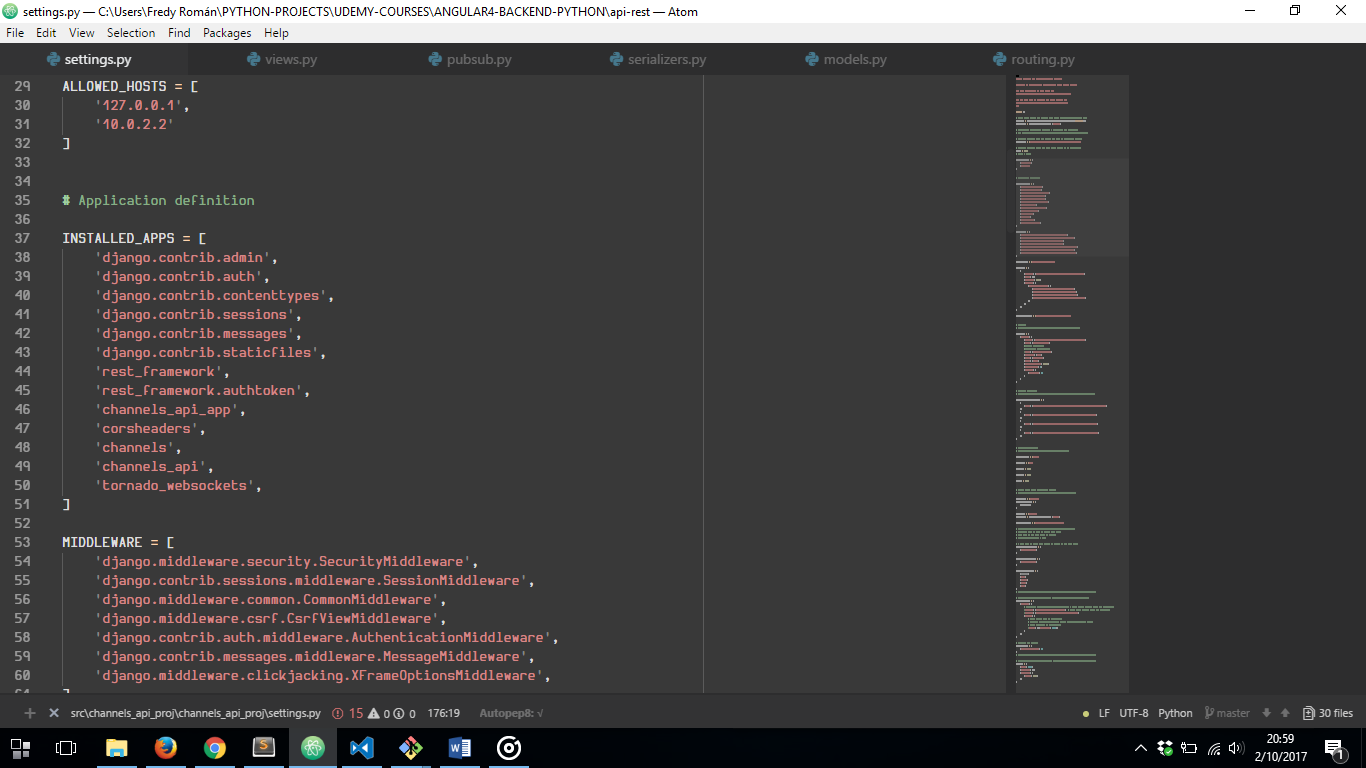
<https://channels.readthedocs.io/en/latest/getting-started.html>

<https://github.com/linuxlewis/channels-api>

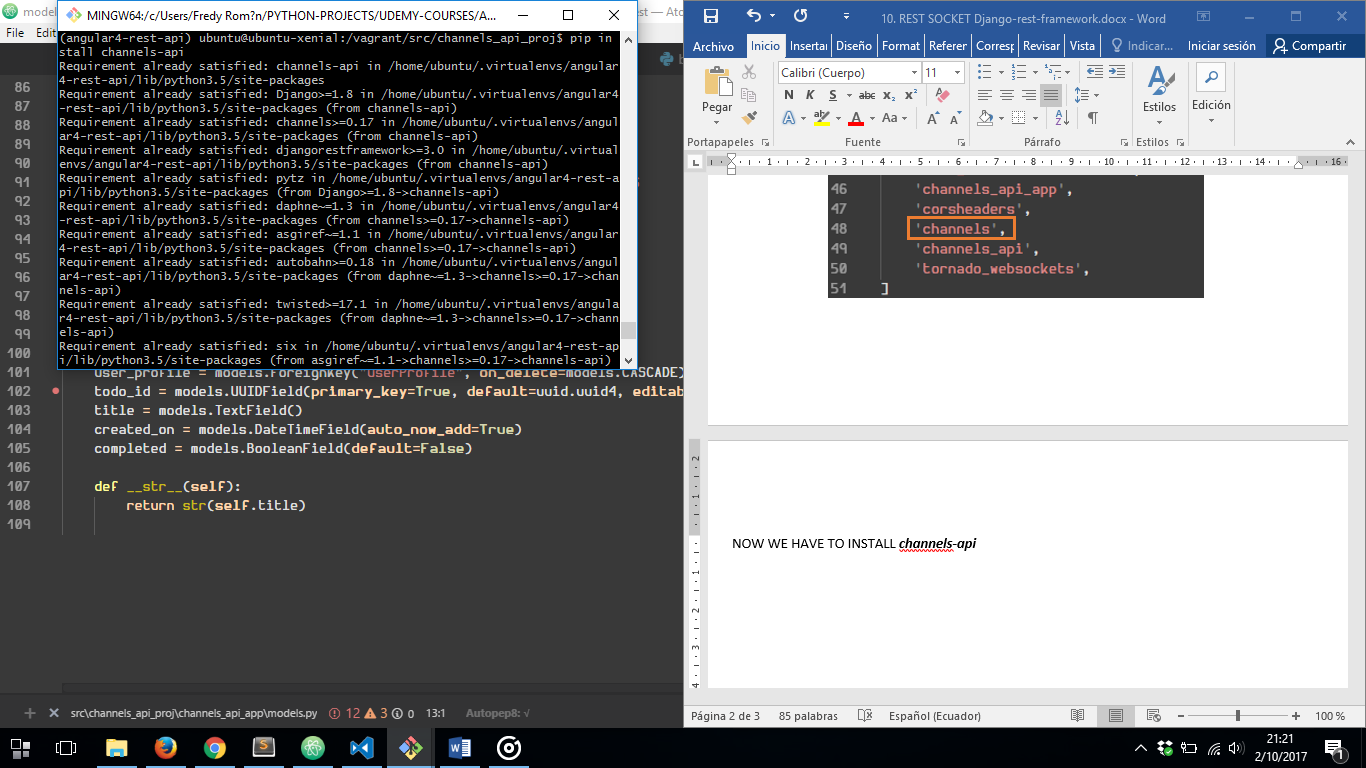
FIRST WE HAVE TO INSTALL ***channels*** DON’T FORGET TO ACTIVATE VIRTUAL ENVIRONMENT IN USED: ***pip install channels***



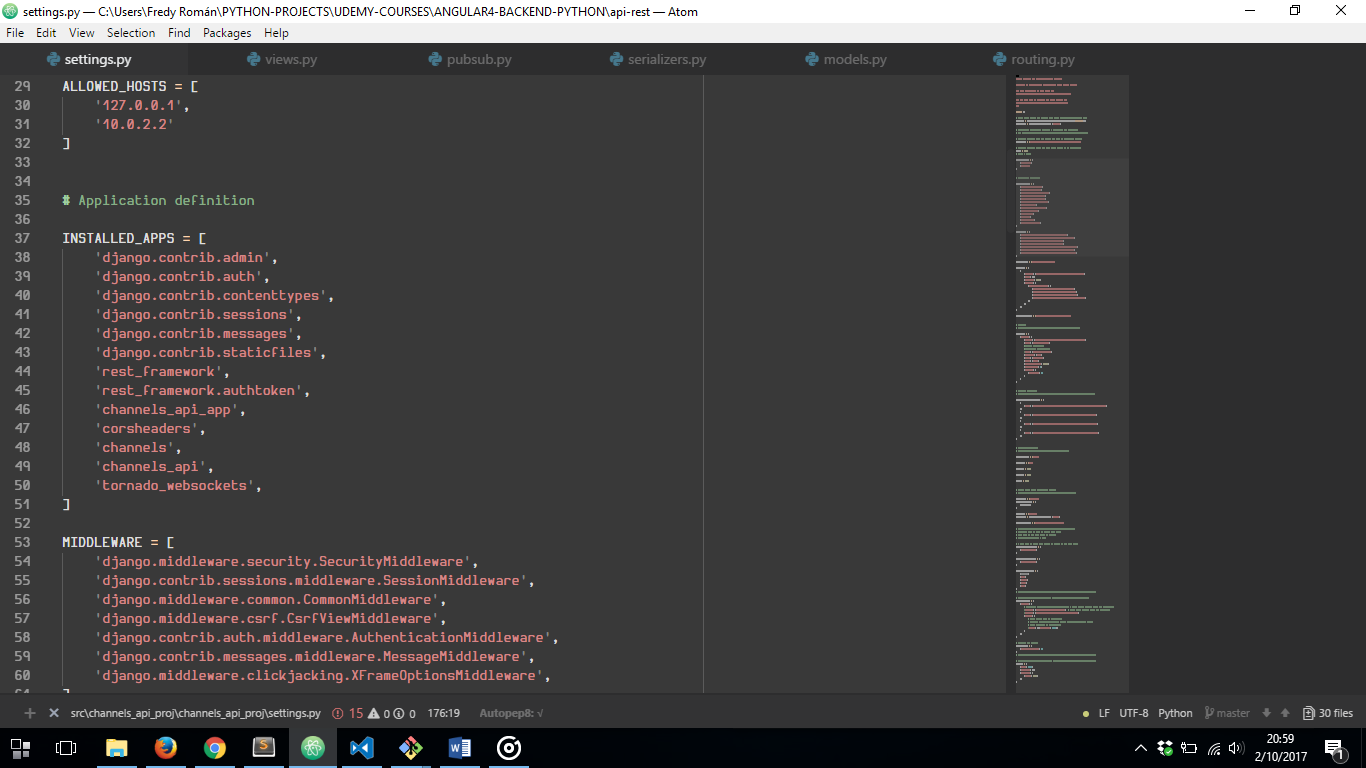
AFTER THAT WE HAVE TO CONFIGURE OUR ***settings.py*** FILE TO USE CHANNELS:



NOW WE HAVE TO INSTALL ***channels-api***

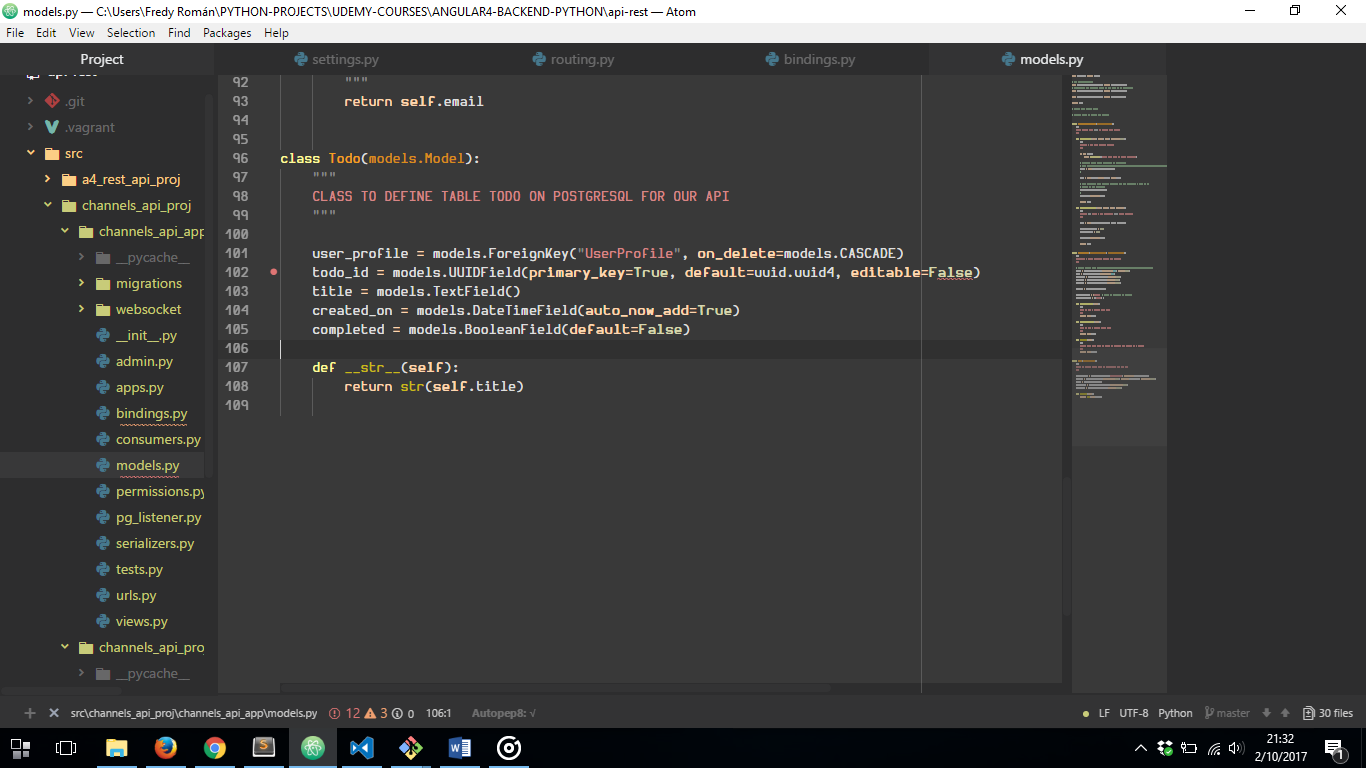


AFTER THAT WE HAVE TO CONFIGURE OUR ***settings.py*** FILE TO USE CHANNELS-API:

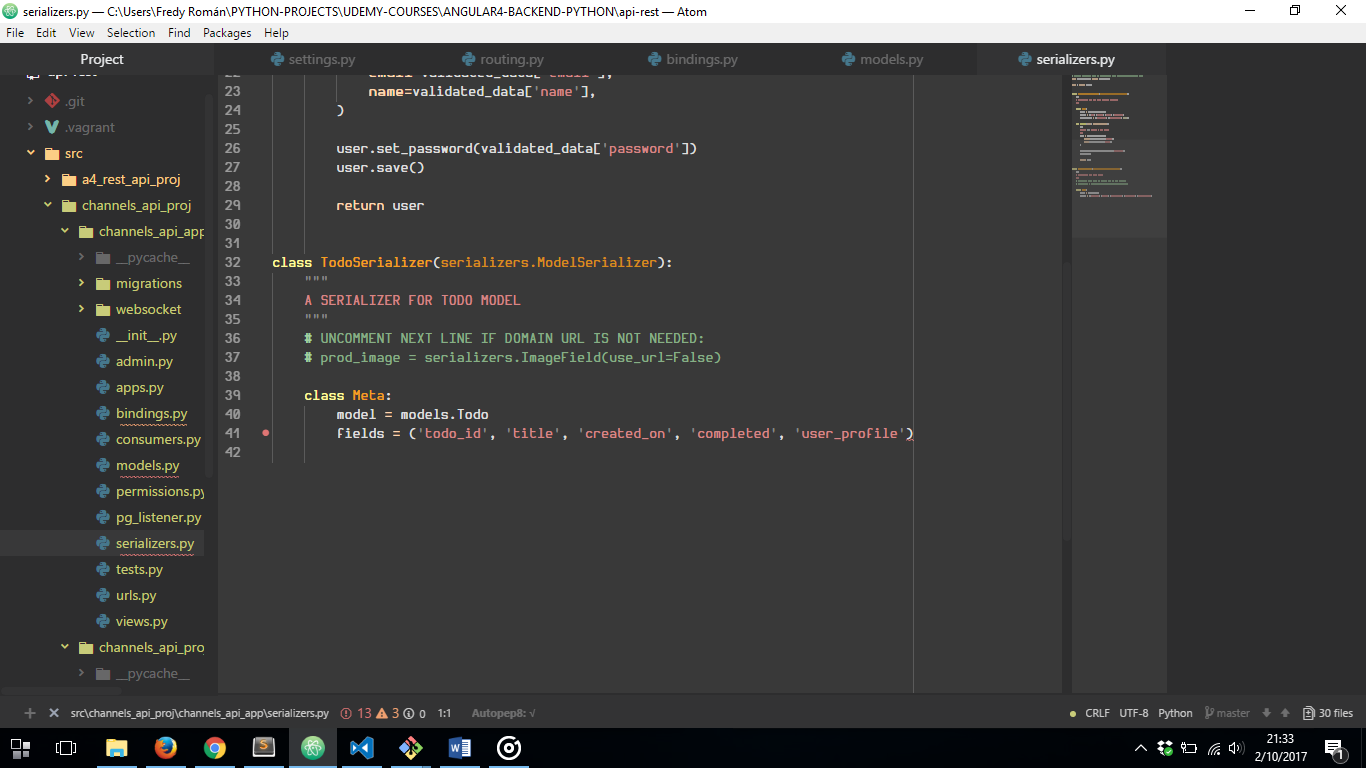


FINNALLY WE’RE GONNA CONFIGURE OUR DJANGO REST API SOCKET USING CHANNELS:

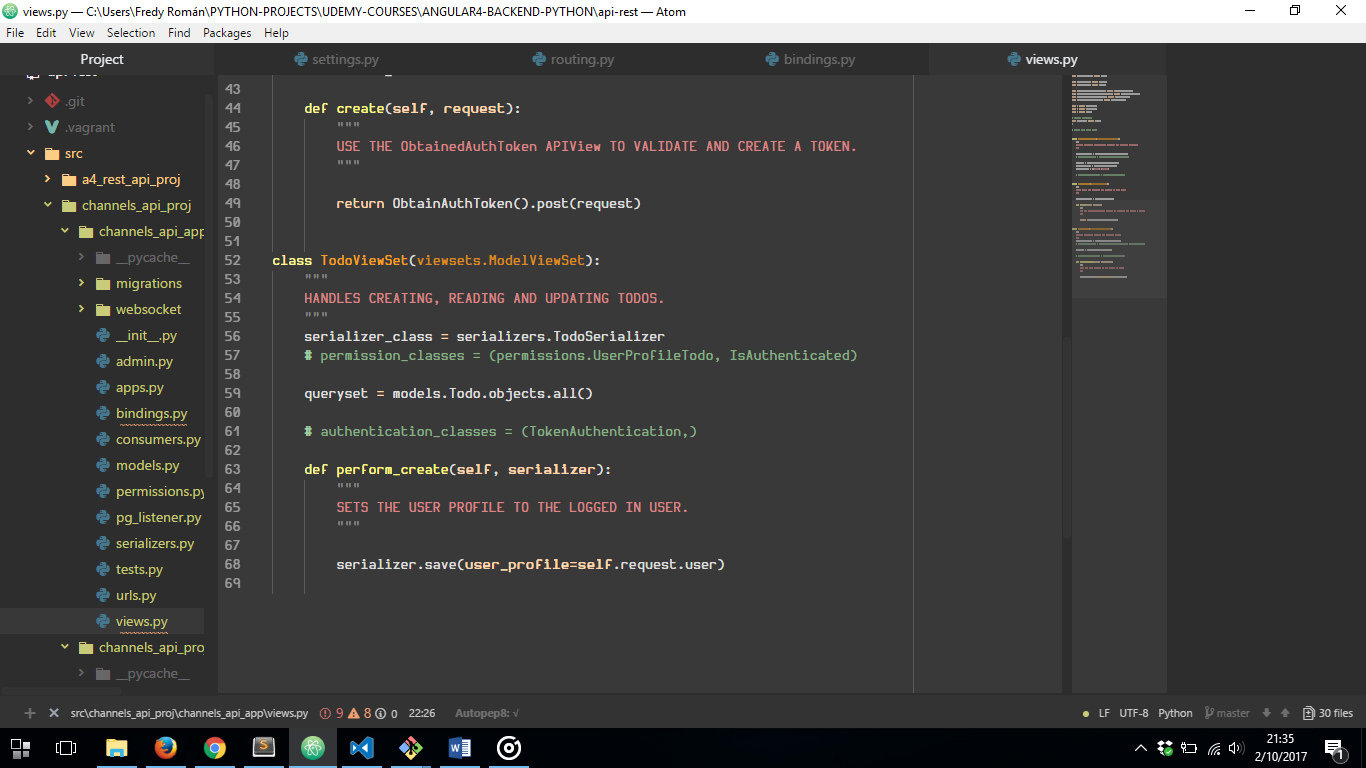
1. CREATING OUR MODEL



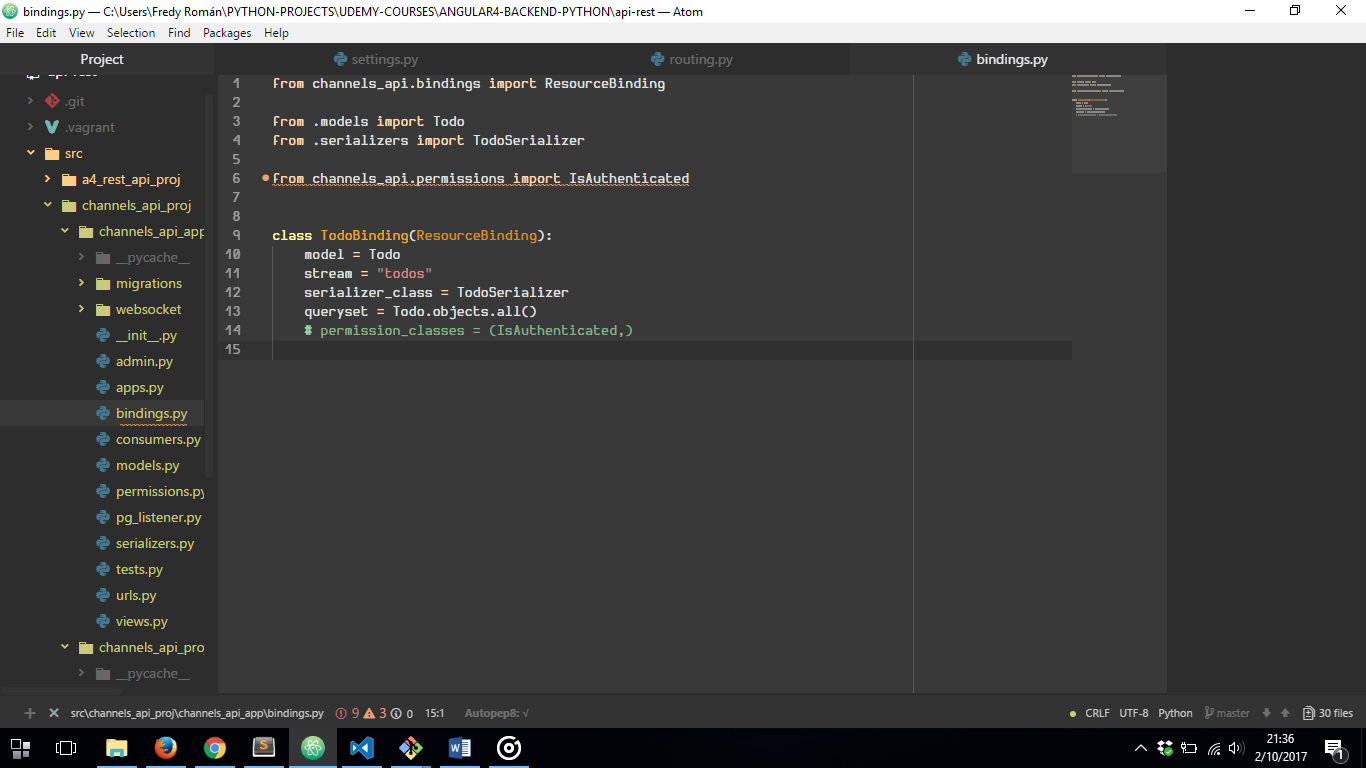
1. CREATING OUR SERIALIZER FILE



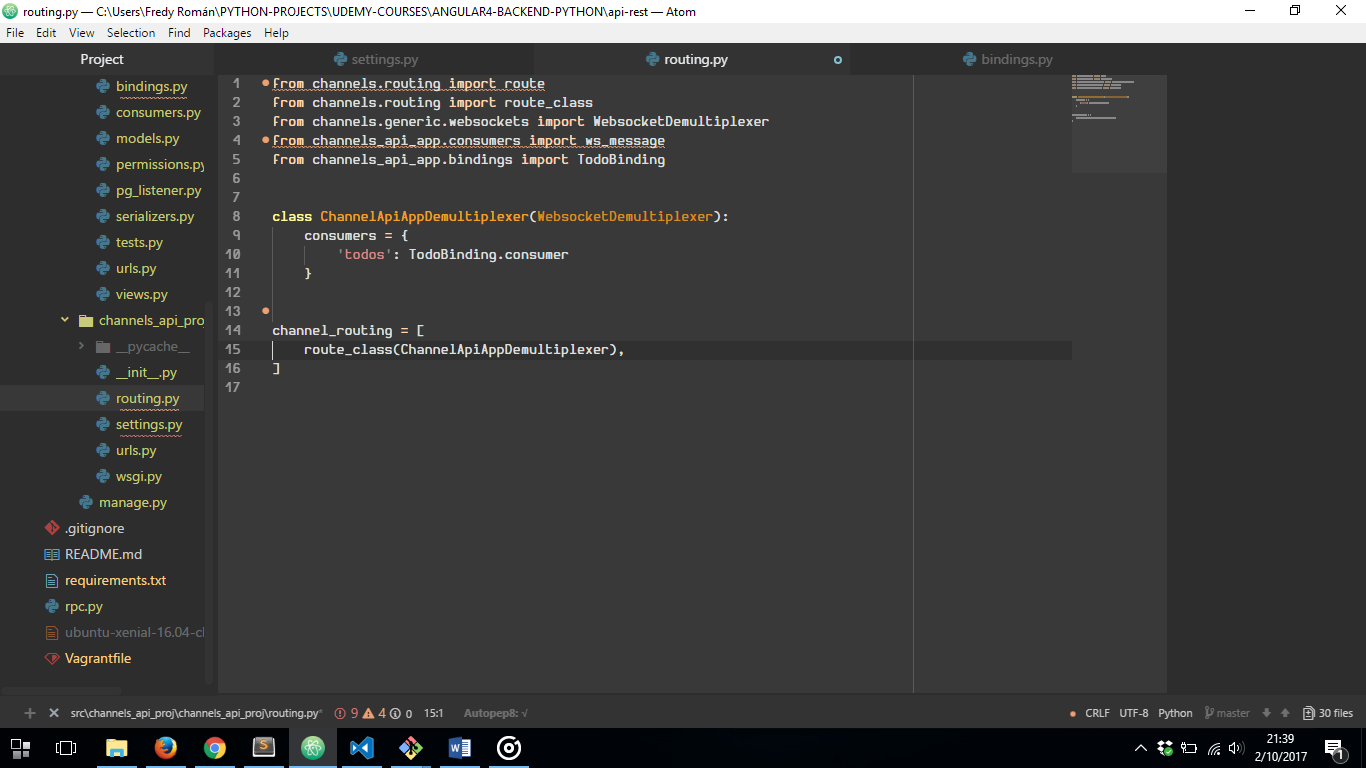
1. CREATING OUR VIEWSET



1. CREATING OUR BINDING TO PROPAGATE REST API CHANGES NOTIFICATIONS

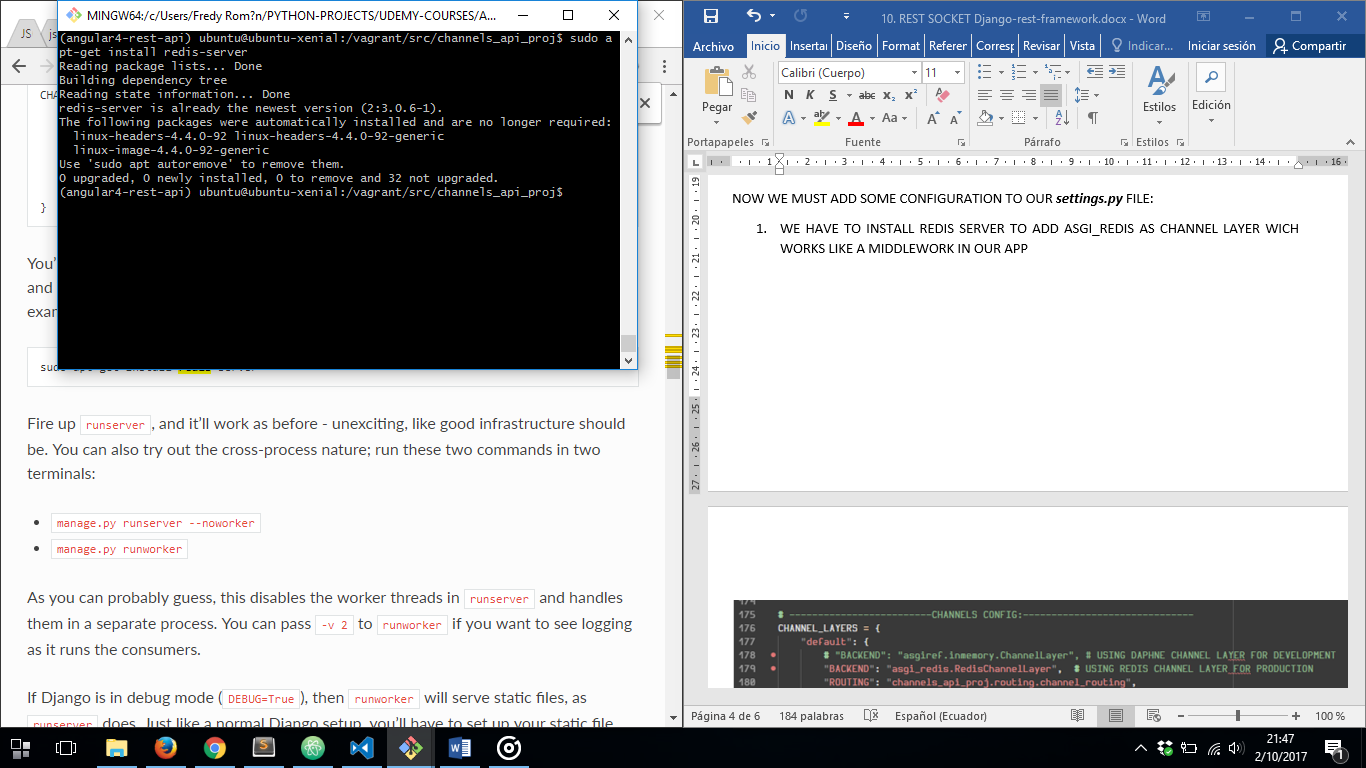


1. CREATING OUR ***ROUTING.PY*** FILE INSIDE OUR PROJECT FOLDER TO CONFIGURE OUR WEB SOCKET WHICH IS GONNA NOTIFY OUR API REST CHANGES:

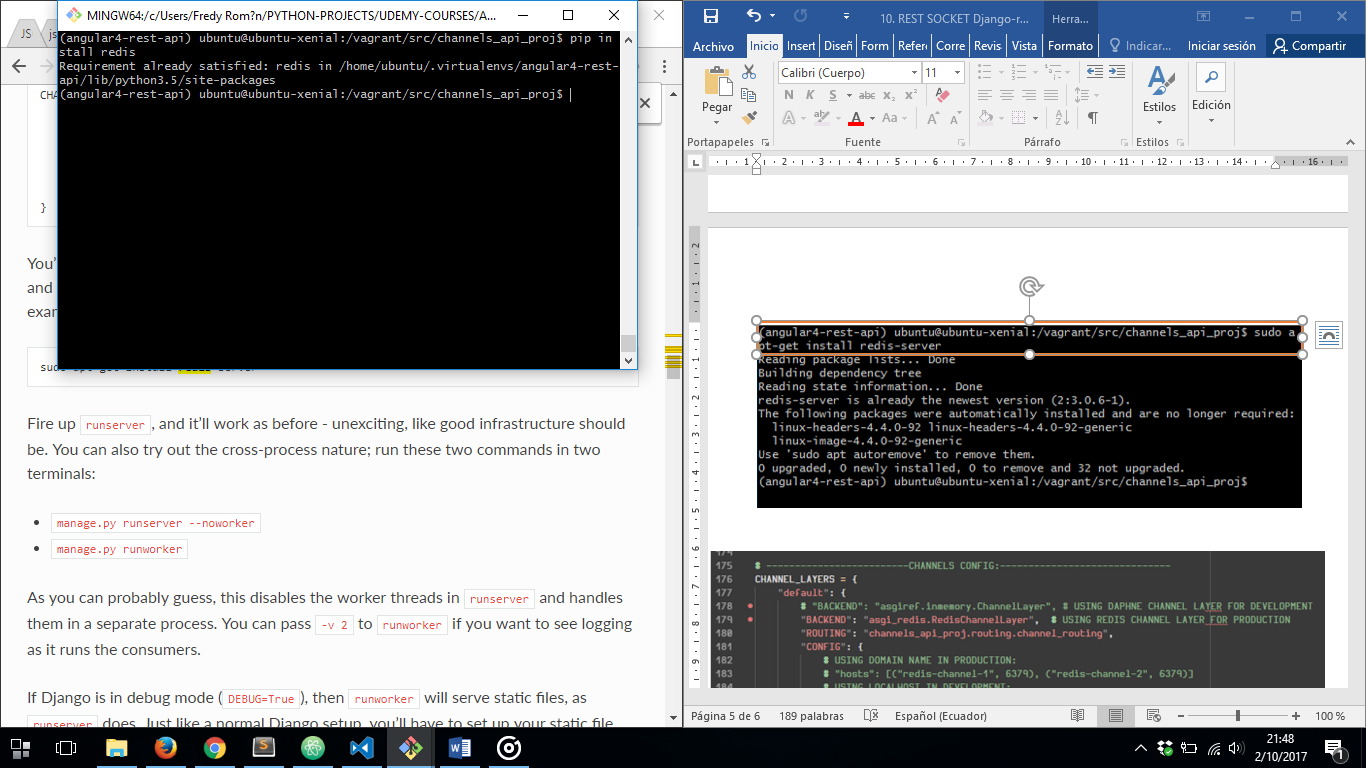


NOW WE MUST ADD SOME CONFIGURATION TO OUR ***settings.py*** FILE:

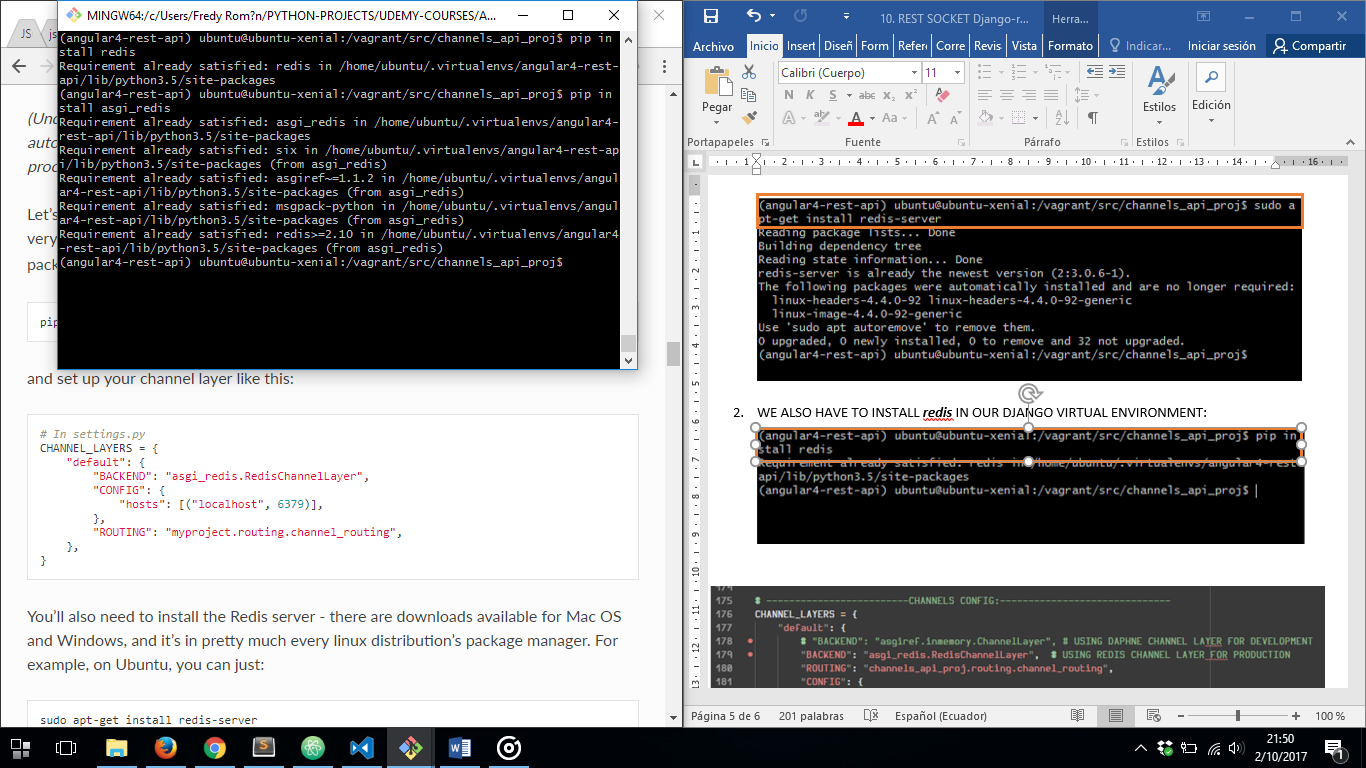
1. WE HAVE TO INSTALL REDIS SERVER IN OUR UBUNTU XENIAL SEVER TO ADD ASGI\_REDIS AS CHANNEL LAYER WICH WORKS LIKE A MIDDLEWORK IN OUR APP



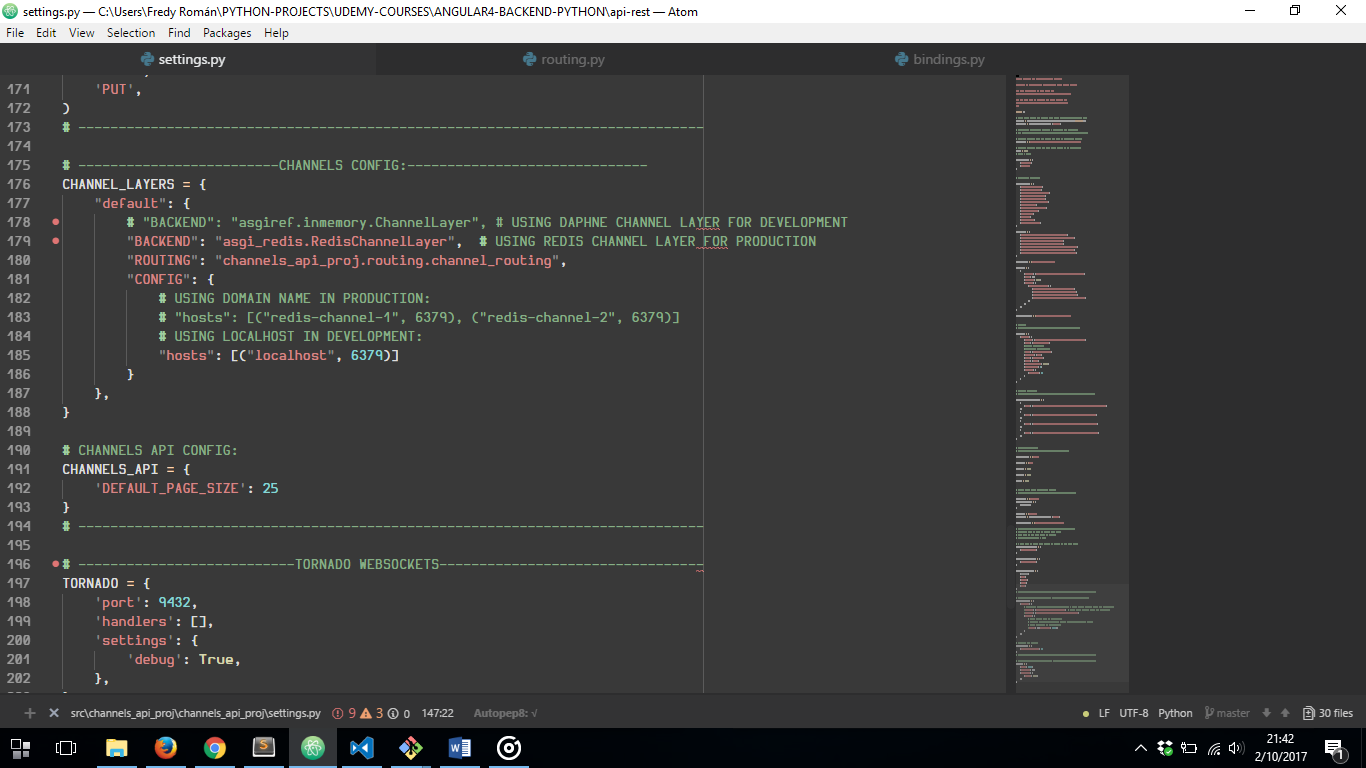
1. WE ALSO HAVE TO INSTALL ***redis*** IN OUR DJANGO VIRTUAL ENVIRONMENT:



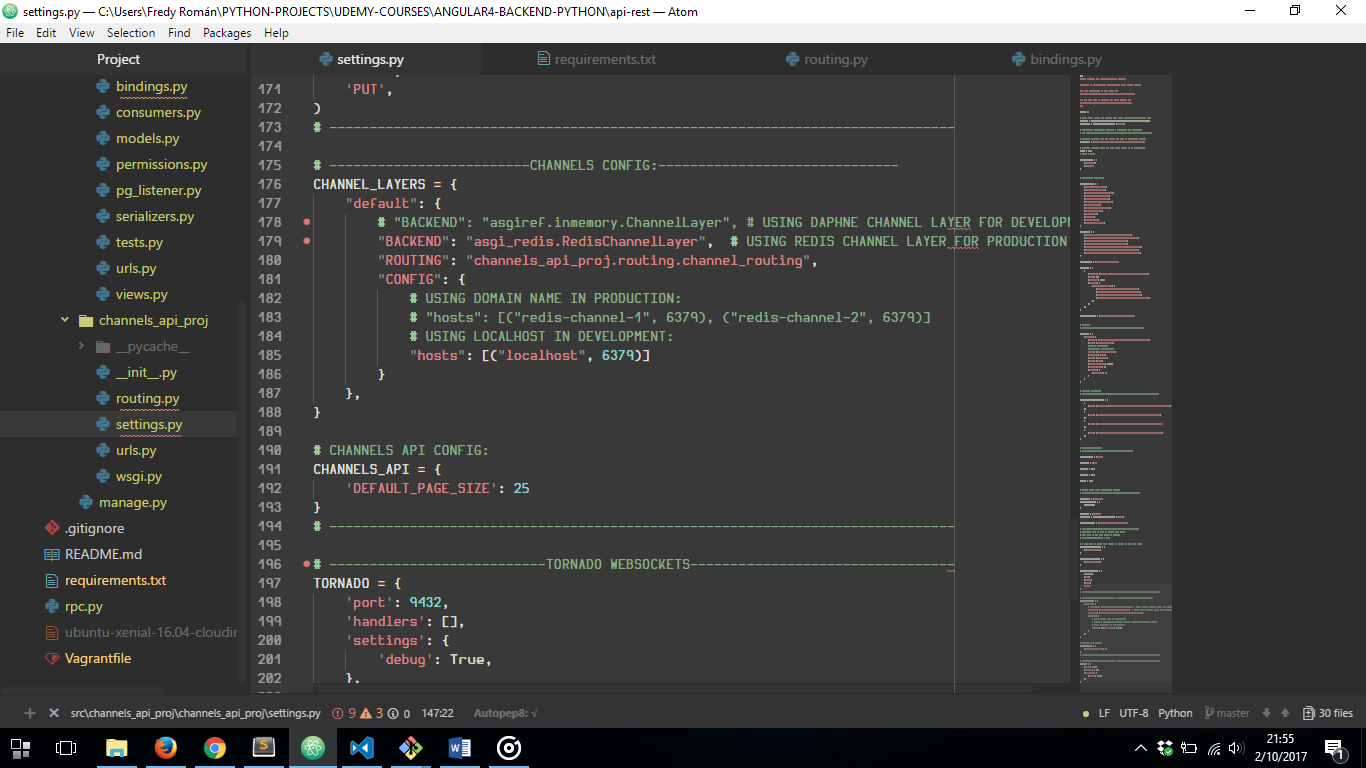
1. THEN WE HAVE TO INSTALL ***asgi\_redis*** TO USE IT LIKE A CHANNEL LAYER



1. FINNALY WE HAVE TO CONFIGURE OUR ***SETTINGS.PY*** FILE TO USE REDIS AS OUR CHANNEL RUNNING REDIS SERVER JUST INSTALLED, USING 6379 PORT BY DEFAULT



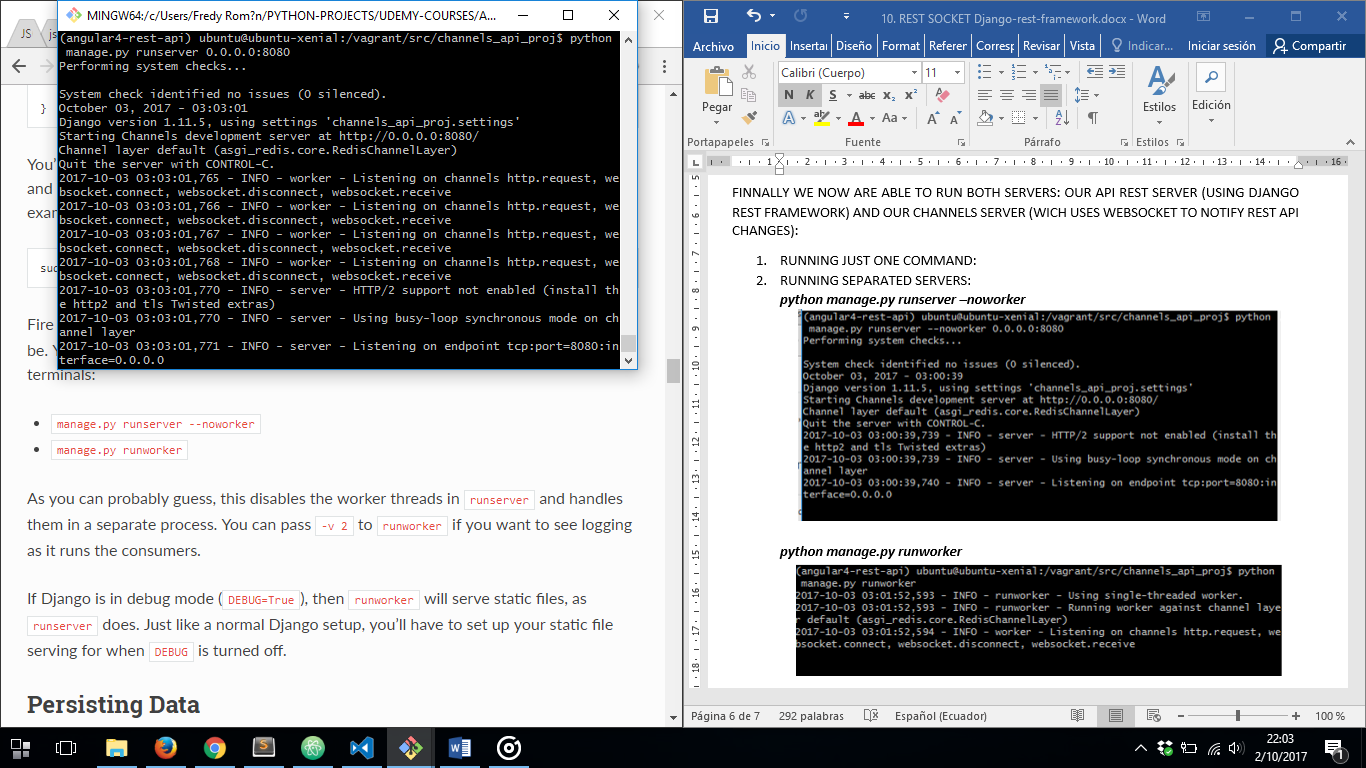
AND THEN OPTIONALLY CONFIGURE NEXT INSIDE ***settings.py***



FINNALLY WE NOW ARE ABLE TO RUN BOTH SERVERS: OUR API REST SERVER (USING DJANGO REST FRAMEWORK) AND OUR CHANNELS SERVER (WICH USES WEBSOCKET TO NOTIFY REST API CHANGES):

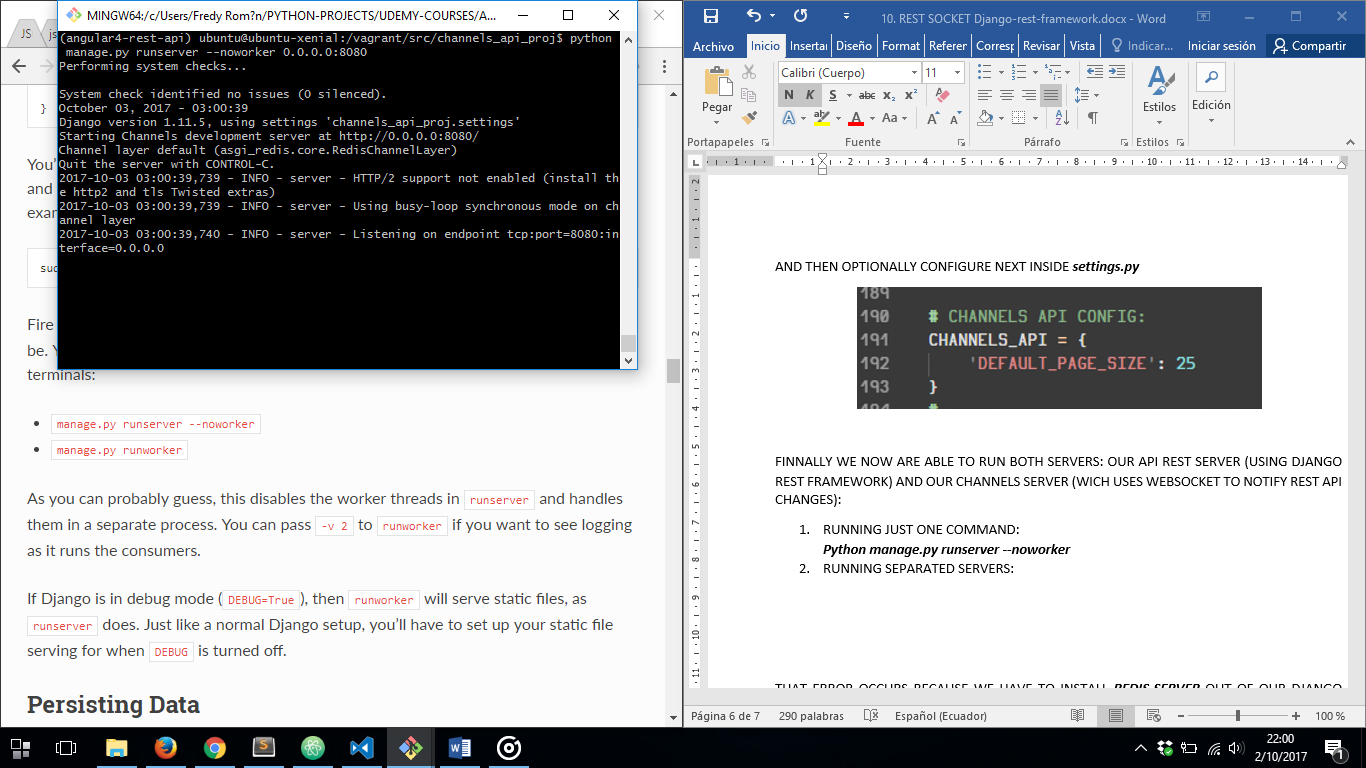
1. RUNNING JUST ONE COMMAND:

***python manage.py runserver 0.0.0.0:8080***

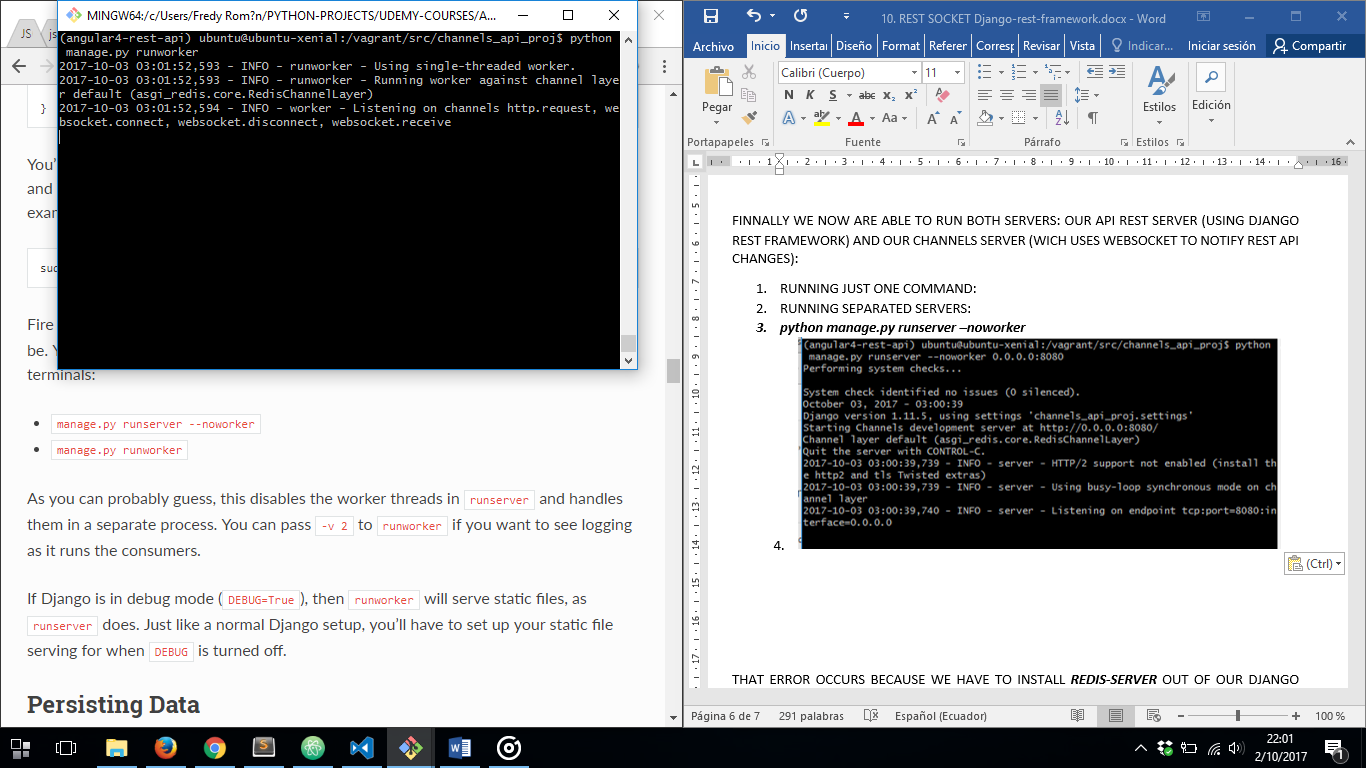


1. RUNNING SEPARATED SERVERS:

***python manage.py runserver –noworker 0.0.0.0:8080***



***python manage.py runworker***



NOW WE HAVE OUR REST API AND REST SOCKET BACKEND READY TO BE ACCESSED BY A JAVASCRIPT CLIENT.