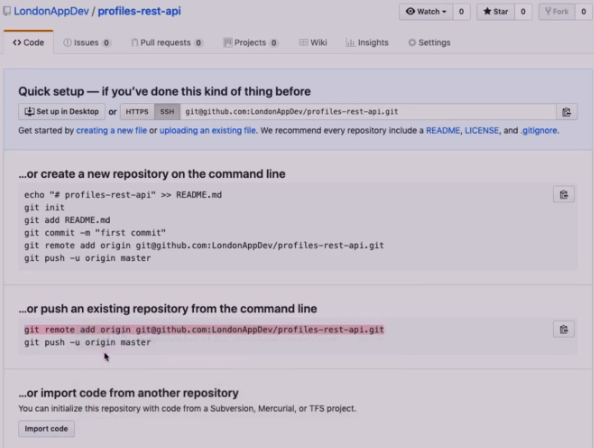
Setting up the Project to Local Machine and Github

1. Download and Install: Git, Oracle VM Virtual Box, Vagrant, Atom.
2. Create a workspace (a folder) and give it a name (e.g. REST-API).
3. Open Atom, Drag and drop the workspace on Atom.
4. Open Git Bash, go to the workspace destination with the cd command.
5. In Git Bash type ***git Init*** to initialize a git local repository.
6. Restart Atom and create a “READ\_ME.md” file (optional) to describe your project and a “.gitgnore” file (optional) to have a license.
7. In Git Bash type ***git add .*** to add the previously created files to the git project
8. Type ***git commit -am “****initial commit****”*** in order to commit the changes to the git project. The ***-am “****initial commit****”*** is just a message, describing the changes that we made.
9. Type ***ssh-keygen -t rsa -b 4096 -C “****your email.com****”*** to create a SHA256 key. Then, add an extra passphrase (optional) for further security. Now, if you can type ***ls ~/.sha*** to check if your private (id\_rsa) and public (id\_rsa.pub) key exist.
10. Type ***cat ~/.ssh /id\_rsa.pub*** and then copy your public key.
11. Go to Github.com, go to your profile (top right of the screen), go to your settings, go to SSH and GPG Keys, click add new, paste your public key on the “Key” field and type the name of the machine that you are authenticating this key to the “title” field and hit “add key”.
12. On Github.com create a Github repository. After creating it, you will see under the “…or push an existing repository from the command line” 2 commands.

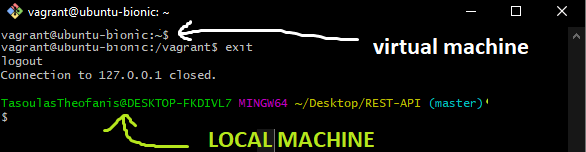


1. Go to Git Bash and type those 2 commands (***git remote add origin git@github.com:****username/project-name****.git*** and ***git push -u origin master***).

Congrats, you created your repository on your local machine and at github.com (even though it’s empty :P)

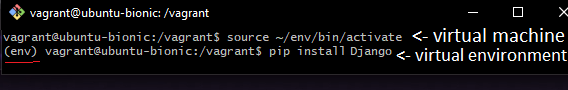
Setting up the Virtual Machine

1. Go to Git Bash and type ***vagrant init*** ***ubuntu/bionic64*** to create a new vagrant file.
2. Go to Atom and to the vagrant file and replace its content with the content of this one [here](https://gist.github.com/LondonAppDev/199eef145a21587ea866b69d40d28682).
3. Go to Git Bash and type ***vagrant up*** to download the base image (or in other words the virtual machine (vm) on our local machine) specified at the vagrant file (in our case an ubuntu bionic64).
4. Type ***vagrant ssh*** to connect to the virtual machine. The command line tells you whether you work on your local or your vm.

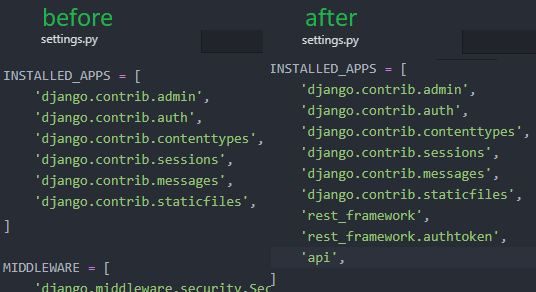


Keep in mind that Vagrant synchronizes files from your virtual to your local machine. Pretty cool huh?

1. In the vm, type ***cd /vagrant*** to switch to the vagrant directory. Type ***python -m venv ~/env*** or ***python3 -m venv ~/env*** to create a python virtual environment (install python on vm) on a destination. In this case, we want to create the python environment to the vagrant server, because we don’t want to synchronize this python environment to out local machine. For instance, if we want destroy and recreate the vagrants over from the scratch, you can do that from a fresh python environment. That’s why, we created the “env” file, to store the python environment there.
2. Type ***source ~/env/bin/activate*** to work in the “env” environment. Now we are on the “env” virtual python environment. You can switch to the vm by typing ***deactivate*** (don’t do that yet).



1. Type ***pip install Django*** and then ***pip install Djangorestframework***. Now we downloaded the Django and the djangoframework packages.
2. ***django-admin.py startproject api\_project .*** to create a new Django project (a folder with the essential files) called api\_project right here in the root location. If you want to add your project to a subfolder just remove the dat and type ***django-admin.py startproject api\_project***.
3. Type ***python manage.py startapp api*** to create an api app folder with the essential files. In this case, the folder will be named “api”.
4. In order to enable an app in a project is to open the “api” folder, open the “settings.py” file, which is the configuration file for the Django. Find the line with the “INSTALLED\_APPS = [” and add after those the “rest\_framework”, “rest\_framework.authtoken”, “api”. That’s how you add apps to your Django project.



1. Type ***python manage.py runserver 0.0.0.0:8000*** on the “env” virtual python environment, to start the Django development server in port 8000.
2. Open your browser and go to <http://127.0.0.1:8000/>. MAGIC!