

## Deploying a Django Application to PythonAnywhere:

The following steps assume you created a GitHub account and have a working Django application to deploy.

1. Edit settings.py to add PythonAnywhere as an allowed host and to set debug = false

```
ALLOWED_HOSTS = ['127.0.0.1', 'pythonanywhere.com']
```

2. Create a new GitHub.com repository for the deployed version of your application; clone this repo to your local computer; copy the project files to this new folder; then push your code to this GitHub.com repository. **This should be a separate repository on your personal GitHub account, not the repository you use to submit your work for grading.** When copying your application files to this new repo, you won't need to include the .idea file nor the venv folder. They are not needed for deploying to PythonAnywhere.

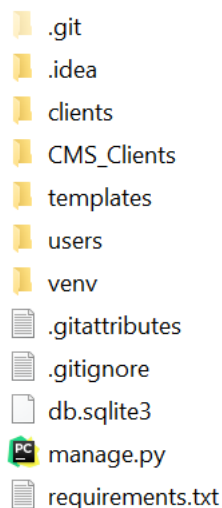
- a. Before creating the repo, you should create a simple text file called '.gitignore' to ensure your virtual environment, database, and PyCharm files are not pushed to the repo. Sample contents of the '.gitignore' are shown below – note that the virtual environment was called 'venv'. Change the name if your virtual environment has a different name:

```
venv
*.pyc
__pycache__
local_settings.py
.idea
db.sqlite3
```

- b. Verify you have a valid 'requirements.txt' file in the same folder with the .gitignore that includes all the required installations for your project. For this project, the 'requirements.txt' file should contain:

```
django=3.0
django-crispy-forms
```

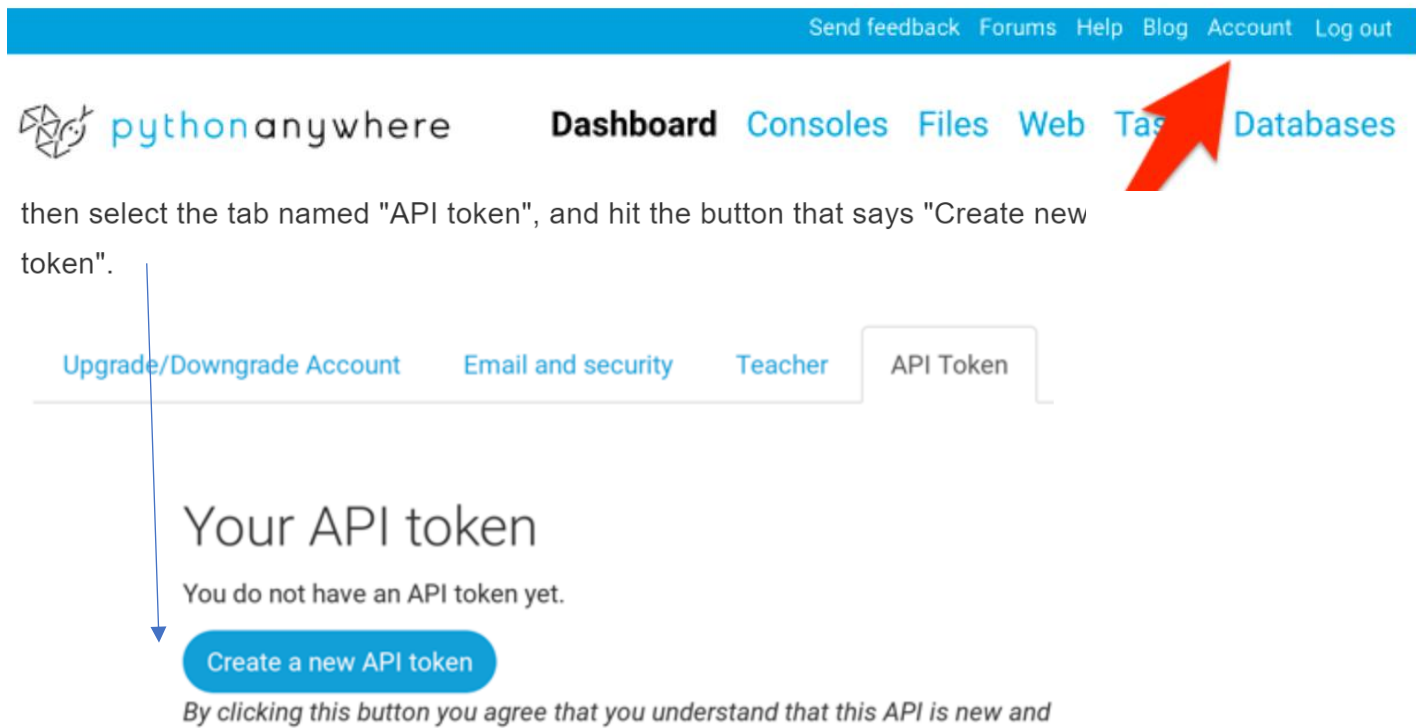
- c. **Do not move your application code inside another folder before pushing to GitHub.** The manage.py file along with the application folders should be at the 'top' level or the application will not deploy correctly. An example of a repository for an application called CMS\_Clients, with clients and users should look like the image below (if you included the .idea and venv when you copied your files to the new repo):



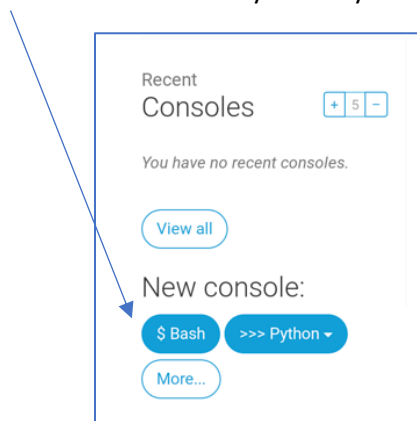
```
.git
.idea
clients
CMS_Clients
templates
users
venv
.gitattributes
.gitignore
db.sqlite3
manage.py
requirements.txt
```

3. Open the page on GitHub that allows you to Clone the repository and copy the Clone address.
4. Sign up for a free PythonAnywhere account if you have not already done so. [www.pythonanywhere.com](https://www.pythonanywhere.com)

- a. You will need to create an API token (this only needs to be done the first time you deploy). Once you log on to your new PythonAnywhere account, click Account:



5. Go to the main [PythonAnywhere Dashboard](#) by clicking on the logo, and choose the option to start a "Bash" console – that's the PythonAnywhere version of a command line.



Note PythonAnywhere is based on Linux, so if you're on Windows, the console will look a little different from the one on your computer.

6. Deploying a web app on PythonAnywhere involves pulling down your code from GitHub, then configuring PythonAnywhere to recognize it and start serving it as a web application. There are manual ways of doing it, but PythonAnywhere provides a helper tool that will do it all for you. Let's install it first:

In the Bash console PythonAnywhere command-line – enter the following command if you have not done this in the past (don't enter the \$):

```
$ pip3.8 install --user pythonanywhere
```

That command should print out some things like Collecting pythonanywhere, and eventually end with a line saying Successfully installed (...) pythonanywhere- (...).

- Next, run the helper to automatically configure our app from GitHub. Type the following into the console on PythonAnywhere (don't forget to use your GitHub username in place of <your-github-username>, so that the URL matches the clone URL from GitHub). PythonAnywhere command-line (don't type the \$):

```
$ pa_autoconfigure_django.py --python=3.8 yourGitHubRepoCloneAddress
```

If you get an error running the above command that indicates you should use the --nuke option, then reenter the command with that option to replace a previous application:

```
$ pa_autoconfigure_django.py --python=3.8 yourGitHubRepoCloneAddress --nuke
```

As you watch that running, you'll be able to see what it's doing:

Downloading your code from GitHub

Creating a virtualenv on PythonAnywhere, just like the one on your own computer

Updating your settings file with some deployment settings

Setting up a database on PythonAnywhere using the manage.py migrate command

Setting up your static files (if any)

And configuring PythonAnywhere to serve your web app via its API

On PythonAnywhere all those steps are automated, but they're the same steps you would have to go through with any other server provider.

- The main thing to notice right now is that your database on PythonAnywhere is completely separate from the database on your own computer. It will have different clients and admin accounts. As a result, just as we did on your own computer, we need to initialize the admin account with **createsuperuser**. PythonAnywhere has automatically activated your virtualenv for you, so all you need to do is run in the PythonAnywhere Bash Console command-line (as before, don't type the \$):

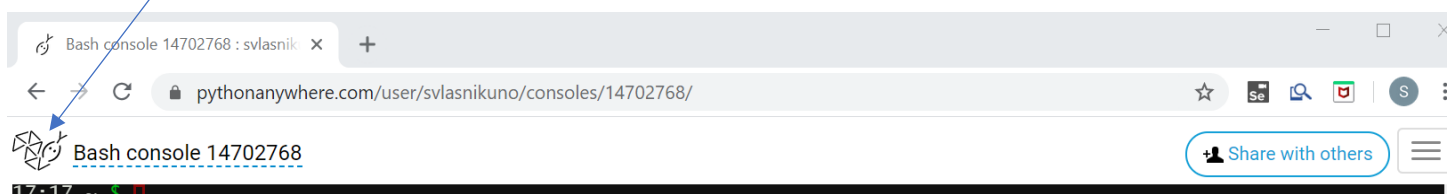
```
$ python manage.py createsuperuser
```

Type in the details for your admin user.

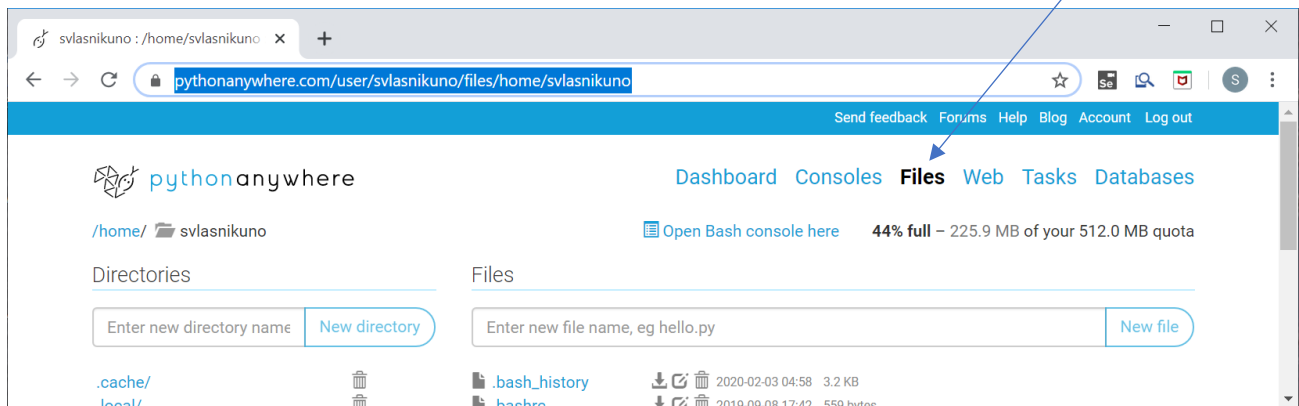
- Now, if you like, you can view your code on PythonAnywhere using the ls (lower case LS) command (short for list). You will see the folders created for your App:

```
$ ls
```

- Return to the PythonAnywhere Dashboard by clicking the 'snake' symbol in the upper left-hand corner of the Bash console:



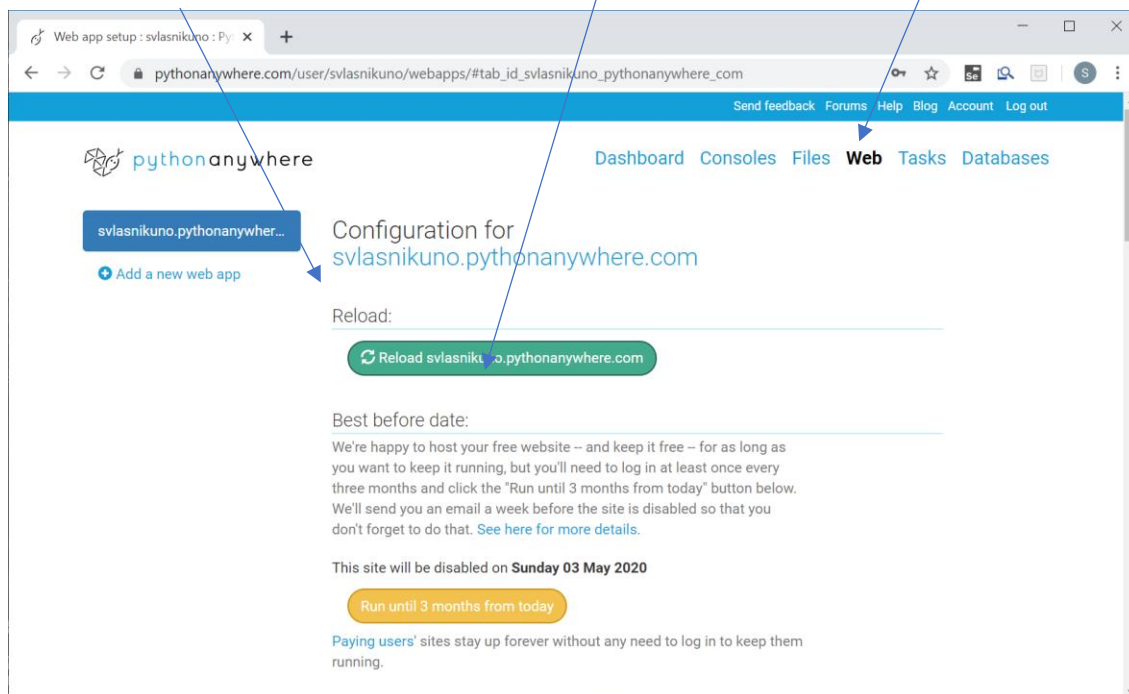
11. You can go to the "Files" page on PythonAnywhere and navigate using PythonAnywhere's built-in file browser. (From the Console page, you can get to other PythonAnywhere pages from the menu button in the upper right corner. Once you're on one of the pages, there are links to the other ones near the top.)



12. Your site should now be live on the public Internet! To view your site, Click 'Web' on the top of the PythonAnywhere Dashboard page.

Click Reload yourUsername.pythonanywhere.com

Click the link shown to go to your PythonAnywhere "Web" page. You can share this link with anyone you like.



Note This is a beginners' tutorial, and in deploying this site we've taken a few shortcuts which aren't ideal from a security point of view. If and when you decide to build on this project, or start a new project, you should review the [Django deployment checklist](#) for some tips on securing your site. [1]

Deploy to Heroku: See this tutorial for deployment steps of a simple application: <https://tutorial-extensions.djangogirls.org/en/heroku/>

## References

[1] D. G. Tutorial, "Django Girls Tutorial - Deploy," Django Girls, [Online]. Available: <https://tutorial.djangogirls.org/en/deploy/>. [Accessed Feb 2020].