

# API in Django

Thursday, February 4, 2016 1:19 PM

1. Go get a new Blob Storage and put a bunch of cool files there.
2. Install a Django website.
  - a. Figure out settings and how many bits and stuff...
3. Python virtual environment
4. Install only what you need to return some JSON (or CSV)
5. Add a new Python file called api.py as part of the app folder
6. Alter the URL file (within the 'project' sub-folder; not the app folder) per example to establish API calls
  - a. Example `url(r'api/music','app.api.music')`
    - i. Notice 'app.api.music': This means Go to app folder, api.py file, run `def music()`:
7. Add a tools folder
8. Add a builder.py file in the tools folder
  - a. Why no `__init__.py`?
    - i. It means 'live stuff lives here' and it is empty.
  - b. What does the red dot mean?
    - i. I added `__init__.py` and Built and it went away
9. Follow the music man example to create a simple API call: in the api.py file.

Ok I did all of the above and now we have a sort of model:

```
urls.py >> api.py [>> builder.py]
```

Now api.py is where a simple blob access call goes; so I looked it up to find this website:

<https://azure.microsoft.com/en-us/documentation/articles/storage-python-how-to-use-blob-storage/>

Notice we need this import in api.py:

```
from azure.storage.blob import BlobService
```

This will not work unless we pip install the azure machinery. So: Right click on the Python virtual environment and select Open Command Prompt Here. In that command prompt window type `Script \activate` so that the prompt begins with '(env)'. This means that your commands will now register into the Python virtual environment 'env'. Now do

```
% pip install azure
```