

ADEPT Profiler



Building this project

Docker image

1. From project root,

```
run docker build --tag adept_profiler:latest .
```

2. List images

run **docker images** to validate docker image exists with name *adept_profiler* with the latest tag

3. To test running the image,

```
docker run adept_profiler:latest --version
```

```
docker run adept_profiler:latest --help
```

```
docker run adept_profiler:latest --input_file_name </path/to/file> --  
output_location </path/to/output/directory>
```

4. To tag and push image to ECR/ Hub etc

```
export IMAGE_TAG=0.8.0  
docker build --tag adept_pg_dump_assessor:$IMAGE_TAG .  
aws ecr get-login-password --region us-east-1 | docker login --username  
AWS --password-stdin 745001225527.dkr.ecr.us-east-1.amazonaws.com  
docker tag adept_pg_dump_assessor:$IMAGE_TAG 745001225527.dkr.ecr.us-east-  
1.amazonaws.com/adept_pg_dump_assessor:latest  
docker tag adept_pg_dump_assessor:$IMAGE_TAG 745001225527.dkr.ecr.us-east-  
1.amazonaws.com/adept_pg_dump_assessor:$IMAGE_TAG  
docker push 745001225527.dkr.ecr.us-east-  
1.amazonaws.com/adept_pg_dump_assessor:$IMAGE_TAG  
docker push 745001225527.dkr.ecr.us-east-  
1.amazonaws.com/adept_pg_dump_assessor:latest  
docker images
```

To run this application with python

```
python3 run --input_file_name </path/to/file> --output_location  
</path/to/output/directory>
```

To run this application with docker

```
aws ecr get-login-password --region us-east-1 | docker login --username  
AWS --password-stdin 745001225527.dkr.ecr.us-east-1.amazonaws.com  
  
docker run 745001225527.dkr.ecr.us-east-  
1.amazonaws.com/adept_pg_dump_assessor:latest --help
```

Developing this project

Creating a virtual environment in the terminal

macOS/Linux

You may need to run `sudo apt-get install python3-venv` first on Debian-based OSs

```
python3 -m venv .py-venv
```

To activate the environment run `source .py-venv/bin/activate`

To deactivate the Virtual environment run `deactivate`

Windows

You can also use `py -3 -m venv .py-venv`

```
python -m venv .py-venv
```

To activate the environment run `.py-venv\Scripts\activate.bat`

To deactivate the Virtual environment run `deactivate`

To display all of the packages installed in the virtual environment

```
run python -m pip list
```

To save all of the packages installed in the virtual environment to a *requirements.txt*

```
run python -m pip freeze > requirements.txt
```