

Configure the Comparative Pathology Workbench (CPW) in a Development Environment

This document describes how to set up and run the Software for the Comparative Pathology Workbench (CPW) in a DEVELOPMENT environment ONLY!

See the Document “CPW_Configure-Production.docx” for instructions on how to setup the CPW in a simulated PRODUCTION environment.

Mike Wicks
19th May 2023

Step 1 - Prerequisites

Install Postgres and create a database with a database user.

Pull the software from the Git repository: <https://github.com/Comparative-Pathology/comparativepathologyworkbench>

Install “Pipenv” or other software that runs a virtual environment. Pipenv requires a “**Pipfile**” that specifies all the required Library dependencies. A “**requirements.txt**” file is also provided, should you use different virtual environment software.

Step 2 – Set Up a Virtual Environment

Move to the “**app**” folder - Create a virtual environment:

pipenv install

Start the a virtual environment:

pipenv shell

Step 2 – Configure Software with a Database

The system needs a file to hold environment variables.

In the “**app/config**” sub-folder create a “**.env**” file with the following environment variables.

An example “**.env**” file is shown here:

```
#
# CPW Environment Settings
#
SECRET_KEY=a_secret_key
ENCRYPT_KEY=an_encrypt_key
CPW_CIPHER_STRING=a_cpw_cipher_string
#
DEBUG=True
ALLOWED_HOSTS=localhost, 127.0.0.1
#
# PostGres Database Settings
#
DB_ENGINE=django.db.backends.postgresql_psycopg2
DB_NAME=a_schema_name
DB_USER=a_database_user
DB_PASSWORD=a_password
DB_HOST=localhost
DB_PORT=5432
DB_ATOMIC_REQUESTS=True
#
```

```
# Dummy Email File Backend
#
EMAIL_BACKEND=django.core.mail.backends.filebased.EmailBackend
EMAIL_FILE_PATH=a_folder_somewhere
#
SESSION_EXPIRE_AT_BROWSER_CLOSE=True
SESSION_COOKIE_AGE=86400
# Default: 1209600 (2 weeks, in seconds)
```

Step 3 - Set up an Empty Database

Run all Django Migrations, to set up an empty database.

python manage.py migrate

Create Superuser Account

python manage.py createsuperuser

Step 4 - Set up Base Data in the Database

The following commands populate the database with the basic data required for the system to function.

```
python manage.py loaddata fixtures/protocol_prod.json --app matrices.protocol
python manage.py loaddata fixtures/type_prod.json --app matrices.type
python manage.py loaddata fixtures/location_prod.json --app matrices.location
python manage.py loaddata fixtures/command_prod.json --app matrices.command
python manage.py loaddata fixtures/blog_prod.json --app matrices.blog
python manage.py loaddata fixtures/environment_prod.json --app matrices.environment
python manage.py loaddata fixtures/authority_prod.json --app matrices.authority
python manage.py loaddata fixtures/collectionauthority_prod.json --app
matrices.collectionauthority
python manage.py loaddata fixtures/server_prod.json --app matrices.server
```

Step 5 – Run the System

python manage.py runserver

Point a Browser at <http://localhost:8000>

You should now see the CPW Home Page

WordPress Blogging Engine

The Workbench requires access to a WordPress blogging instance to provide the full functionality of the system.

See the document “CPW_Configure-Production.docx”, Part B, for instructions on how to setup a suitable WordPress “stack”.