

The Comparative Pathology Workbench (Technical) README

This document describes how to set up and run the Software for the Comparative Pathology Workbench

Mike Wicks
22nd March 2019

Prerequisites

Install Python and Install Python dependencies in the supplied “Pipfile”.
(If Production, install “UWSGI” as well)

Install Postgres, and create a database.

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

In the “matrices” sub-folder create a “.env” file with the following environment variables:

Django defined variables:

```
SECRET_KEY=<secret_key>
ENCRYPT_KEY=<encrypt_key>
DEBUG=True (in development)
ALLOWED_HOSTS - must include “localhost” in development
SESSION_EXPIRE_AT_BROWSER_CLOSE=True
SESSION_COOKIE_AGE=1800 (Default is 1209600; 2 weeks, in seconds)
```

Database configuration variables:

```
DB_ENGINE=django.db.backends.postgresql_psycopg2 (if Postgres)
DB_NAME=<schema_name>
DB_USER=<db_user>
DB_PASSWORD=<db_password>
DB_HOST=db (typically)
DB_PORT=5432 (typically)
```

Email configuration variables:

```
EMAIL_HOST
EMAIL_PORT
EMAIL_HOST_USER
EMAIL_HOST_PASSWORD
EMAIL_USE_TLS
DEFAULT_FROM_EMAIL
```

(Email is used to authenticate and validate any users registering on the system, including changing and resetting passwords – these parameters are all dependent on the Email Server you decide to use.)

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

Create Superuser Account
 (“python manage.py createsuperuser ...”)

NB. You may need to configure the database directly via the Postgres console.

Execution

The Workbench software can run either as

1. DEVELOPMENT
 - a. Python (Django) application directly
 - i. “python manage.py runserver”
2. PRODUCTION
 - a. A uWSGI application within an Nginx web server
 - i. “uwsgi --ini matrices.ini &”

NGINX Web Server

In a Production environment, the Workbench application is designed to be served by a an NGINX (or other) web server, via an application server such as uWSGI.

Configuration of the NGINX web server to direct traffic to the Workbench is outside the scope of this document.

WordPress Blogging Engine

The Workbench requires access to a WordPress blogging instance to provide the required commentary facilities.

Installation and configuration of a WordPress instance for the Workbench is outside the scope of this document.

Workbench Configuration

Configuration of the workbench is achieved by using a superuser account, when the application is running.

Logging in as a Superuser provides 2 more main menu options on the main menu:

1. Maintenance
2. Authorisation

Adding Wordpress Server Functionality

The Workbench creates, updates and deletes Blog Posts on an associated Wordpress instance.

In order to converse with a Wordpress instance, the Workbench needs configuring for 1, Credentials on that instance, 2, Commands that use the REST API of that instance.

1. Wordpress Credentials

Log in as a Superuser, and access the “Maintenance” main menu option.

For each User of the Workbench, there must be matching Wordpress credentials in order to ensure, Blog Commentary is possible for each Matrix and Cell. – if matching credentials are not found, no Blog Commentary is created.

Add matching credentials as a “New Blog Credential” – you will need to supply the “User Name”, “Wordpress ID” and “Application. Password”, obtained from your Wordpress instance.

(Your Wordpress Instance must have the “Application Password” Plugin installed and activated – each Wordpress user must have a consequent application password generated, and this must be stored in the Workbench)

2. Wordpress Commands

Log in as a Superuser, and access the “Maintenance” main menu option

Blog Posts are created, updated and deleted from an associated Wordpress instance that you have access to; the Workbench must be updated with the relevant REST API commands to enable this functionality to work. Add commands using the “Add New Blog Command” button.

There must be the entries added for ALL the following commands:

Name	Protocol	URL	Application	Preamble	Postamble
PostAPost	http	<URL of your Wordpress Instance>	wp-json/wp/v2	posts	<None>
GetPostComments	http	<URL of your Wordpress Instance>	wp-json/wp/v2	comments?post=	<None>
PostAComment	http	<URL of your Wordpress Instance>	wp-json/wp/v2	comments	<None>
GetPost	http	<URL of your Wordpress Instance>	wp-json/wp/v2	posts	<None>
LinkPost	http	<URL of your Wordpress Instance>	archives	<None>	<None>
DeletePost	http	<URL of your Wordpress Instance>	wp-json/wp/v2	posts	<None>

Adding Image Server Functionality

In order to converse with an Image Server instance, the Workbench needs configuring for 1, Protocols to access Image Servers, 2, Type of Image Servers, 3, Commands that use the REST API for a given Type of Image Server

1. *Image Server Type*

Log in as a Superuser, and access the “Maintenance” main menu option

Scroll down to “Server Types”, and press “Add New Type”

At present the Workbench can only cater for 1 type of Image Server – “OMERO_5.4.7”. Add this type here.

2. *Image Server Protocol*

Log in as a Superuser, and access the “Maintenance” main menu option

Scroll down to “Protocols”, and press “Add New Protocol”

At present the Workbench can only cater for 2 types of Protocol– “HTTP” and “HTTPS”. Add these types here.

3. Image Server (API) Commands

Log in as a Superuser, and access the “Maintenance” main menu option

Scroll down to “API Commands”, and press “Add New Command”

In order for the Workbench to successfully access OMERO 5.4.7 Image servers, the following commands need to be added, thus:

Name	Type	Protocol	Application	Preamble	Postamble
Images	OMERO_5.4.7	https	api/v0/m	images	
ImageDatasets	OMERO_5.4.7	https	api/v0/m	images	datasets
Api	OMERO_5.4.7	https	api		
Token	OMERO_5.4.7	https	api/v0/token		
Login	OMERO_5.4.7	https	api/v0/login		
GroupDatasets	OMERO_5.4.7	https	api/v0/m	datasets/?group=	&childCount=true
GroupImages	OMERO_5.4.7	https	api/v0/m	images/?group=	&childCount=true
Projects	OMERO_5.4.7	https	api/v0/m	projects	&childCount=true
GroupProjects	OMERO_5.4.7	https	api/v0/m	projects/?group=	&childCount=true
Region	OMERO_5.4.7	https	webgateway	render_image_region	0/0/?region=
Thumbnail	OMERO_5.4.7	https	webgateway	render_thumbnail	
Viewer	OMERO_5.4.7	https	iviewer	?images=	
BirdsEye	OMERO_5.4.7	https	webgateway	render_birds_eye_view	175
Dataset	OMERO_5.4.7	https	api/v0/m	datasets	&childCount=true
DatasetImages	OMERO_5.4.7	https	api/v0/m	datasets	images/?childCount=true
DatasetProjects	OMERO_5.4.7	https	api/v0/m	datasets	projects
ImageROIs	OMERO_5.4.7	https	api/v0/m	images	rois/?childCount=true
ProjectDatasets	OMERO_5.4.7	https	api/v0/m	projects	datasets/?childCount=true
PublicViewer	OMERO_5.4.7	https	webclient	img_detail	

Adding Image Servers

In order for the Workbench to access Image Servers, it must be supplied with the connection details of an image server.

Log in as an ordinary User, and access the “View Imaging Host List” main menu option.

Ordinary Users can add, update and delete Imaging Servers; Only Superusers can maintain the “infrastructure” required to access Imaging Servers (See “Adding Image Server Functionality”).

Scroll down the list of Servers, and press “Add New Server”.

You must supply a Name for the Server, a URL of the server, and credentials on the server (User id (Uid) , and Password (Pwd)). At present the “Type” can only be “OMERO_5.4.7”.

OMERO 5.4.7 servers can either be:

- A. Fully Public
no credentials required, simply a URL without a Uid and Pwd.
- B. Private
Specific Credentials are required with the URL to ALL data; Uid and Pwd mandatory
- C. Semi-Public
credentials typically required for a “public user”, to access all data; Uid and Pwd mandatory

Authorization

Log in as a Superuser, and access the “Authorisation” main menu option.

A Superuser can edit and delete Workbench users here.

Ordinary users can via the login drop-down menu, as well as reset and change their passwords here. This is achieved via email, using the email address supplied during Registration.

Mike Wicks
22nd March 2019