

Populating or updating the ToMCAT database

This procedure will show how to setup, initialize, and configure the virtual environment needed to run code and processes that will parse files saved during the experiments and save the data to the "**postgres tomcat database**". Also, a "**sqlite**" copy of this database can be generated to be fed to the "**datasette interface**" which exposes the dataset to the public in a webpage. The procedure and examples below assumes that you already have a "**postgres**" cluster installed, configured and running on "**gauss**", and that you have a database named "**tomcat**" running in that cluster. Note: "<username>" in the examples below is your username that you are logged in as on "**gauss**".

Installing Miniconda and activating a virtual environment

If not already installed, "**miniconda**" will need to be installed on "**gauss**":

```
mkdir -p ~/miniconda3
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh -O ~/miniconda3/miniconda.sh
bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3
rm ~/miniconda3/miniconda.sh
```

Create a virtual environment using "**conda**":

```
conda create --name tomcat-database python=3.9
```

Activate the "**tomcat-database**" environment:

```
conda activate tomcat-database
```

Prerequisites

Before building the "**datasette**" interface or populating the dataset, some dependencies need to be installed. Run the following command to do so:

```
pip install -r requirements.txt
```

Datasette interface

The following `make` commands can be used to perform different operations on the "**datasette**" interface.

1. **create_datasette**: creates a Docker image for Datasette with plugins.
2. **launch_datasette**: launches the datasette Docker image.

Populating the ToMCAT database

The tomcat dataset is in a Postgres cluster that lives in `/space/paulosoares/postgres`. A series of `make` commands can be used to perform different operations on this dataset. The commands run by default in `development` mode, which uses an sqlite database created under `/space/USER/tomcat/.dev`. To modify the Postgres database, the commands need to be executed in `production` mode, which can be done by adding the following before each command:

```
working_env=production db_pass=<user_postgres_pass>
```

Your user must have permission to modify the Postgres database in production mode.

Commands

1. **clear_db**: [WARNING] drops the tables in the ToMCAT database. It must only be called if one really wants a fresh copy.
2. **create_tables**: creates new tables and indices. If a table already exists, it won't be changed.
3. **update_raw**: Adds new raw data to the relevant tables. This can be called to update the database with new experiment data. It will skip experiments already processed.
4. **sync_raw**: Filters and synchronizes fNIRS and EEG signals with a main clock with frequency 200Hz that starts 1 minute before the rest_state task and ends 1 minute after end of the last minecraft trial. Synchronized signals are saved to the `fnirs_sync` and `eeg_sync` tables.
5. **to_sqlite**: Copies the Postgres database to an SQLite database for publication in the datasette interface. It runs in production mode automatically to make sure to read from

the Postgres database. The environment variable `TBS` can be used in conjunction with this command to specify a subset of tables one wants to copy, otherwise, all tables will be copied. Be patient and run this in a tmux session as this process can take several days depending on the size of the tables.

Miscellaneous

The following `make` commands can be used to perform different operations.

1. **update_inspect_file**: Updates the `inspect-data.json` with the newest sqlite ToMCAT database.
2. **generate_diagram**: Generates the database diagram from the newest sqlite ToMCAT database.
3. **screenshots_to_server**: Copies screenshot images to the web server for access through a public URL.

Revision #7

Created 27 August 2024 22:16:48 by Rick Champlin

Updated 6 November 2025 14:15:13 by Rick Champlin