**MeerCat Installation Guide**

MeerCat consists of 3 main components. This installation guide will go over how to setup each component.

1. MySQL database for storing repository data such as commits, diffs, pull requests, issues, events, etc.
2. Django app for serving the MeerCat web application.
3. Python scripts that use git commands and the GitHub API to retrieve repository data and store it in the database.

**I. Installing MySQL:**

1. Install MySQL. Mac users: use homebrew: brew install mysql
2. For other operating systems or if brew is not installed, look at:

<https://dev.mysql.com/doc/mysql-installation-excerpt/5.7/en/installing.html>

1. Start the MySQL server: brew services start mysql
2. To connect to the MySQL server as root to create:

* Initially login using : mysql -u root;
* After initial login set the password using:

ALTER USER 'root'@'localhost' IDENTIFIED BY 'new\_password';

* Then login using the below: mysql -u root -p;

***Note****: Please set a password for root after initial login.*

1. Create a new database:

CREATE DATABASE [database\_name];

1. Create a new user that has read/write privileges to the database you just created.

Creating a new user:

CREATE USER '[username]'@'localhost' IDENTIFIED WITH mysql\_native\_password BY '[password]';

Adding privileges:

GRANT ALL PRIVILEGES ON [database\_name].\* TO '[username]'@'localhost' WITH GRANT OPTION;

1. Test the new user account by connecting to the MySQL as new user:

* That is, terminate the current terminal from mysql and reconnect to it using the below command.

mysql -u [username] -p;

1. To ensure the database is setup properly, next type SHOW DATABASES; and make sure the new database is listed.

***Note****: The names must be in quotes. The .\* allows all privileges to the user on all of the database’s tables. The user must at least have read/write privileges to run. Keep track of username and password as you will need them later.*

**II. Running Meercat:**

1. Download or clone meercat from: <https://github.com/HPCL/ideas-uo>
2. In meercat/website/settings.py modify the database information to match the database you created (around line 100 in settings.py)

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.mysql',

        'NAME': ‘[database\_name]’,

        'HOST': 'localhost’,

        'PORT': '3331', *Note: may need to change port number*

        'USER': ‘[username]’,

        'PASSWORD': ‘[password]’,

        'OPTIONS': {'charset': 'utf8mb4',

                    'use\_unicode': True},

    }

}

*The default port number is 3306.*

1. Open a terminal and navigate to inside the meercat folder. Next, run pip install -r requirements.txt

***Note*:** *If you face an error in running the above command try re-running using:*

python3 -m pip install -r requirements.txt

1. There are possibilities that you might face issues running the requirement files based on the versions of your operating system and or python. Python 3.10 or higher is recommended. Please install the required libraries separately if required or if you encounter any errors.
2. As of 9/30/2023, if running on a Mac M1/M2, you may need to remove mysqlclient from the requirements.txt and run the following commands instead:

brew install mysql-client pkg-config

export PKG\_CONFIG\_PATH="/opt/homebrew/opt/mysql-client/lib/pkgconfig"

pip install mysqlclient

1. Next, run the following command from within meercat to tell Django to setup the database tables:

python3 manage.py migrate

1. Next you will need to create a file called meercat.config.json and place in meercat folder. This file contains many of the keys you’ll need if you intend to run a public server and use features such as GitHub oAuth, GitHub account for posting messages, and OpenAI for LLM. Type the following into the file, but replace {value} with your leys.

{

“MEERCAT\_USER\_TOKEN”: “{value}”,

“SECRET\_KEY”: “{value}”

“GH\_CLIENT\_SECRET”: “{value}”

“GH\_CLIENT\_SECRET\_DEV”: “{value}”,

“GL\_CLIENT\_SECRET\_DEV”: “{value}”,

“OPEN\_API\_KEY”: “{value}”,

}

1. Next, to create a Django admin user account, run:

python3 manage.py createsuperuser

Follow the steps in terminal for creating a username, entering your email, and creating a password.

1. To ensure the MeerCat Django app is setup correctly, run:

python3 manage.py runserver

1. This should list a link to the server. Click on the link listed in the terminal, which should take you to the MeerCat home page.
2. In the search bar, add /admin/ to the end of the domain. This is the Django admin login. Use the credentials of the user account you created in step 6 to login. The next page should say “**Welcome, [meercat username]**” in the top right corner.

**Creating a New Project in MeerCat:**

1. Make sure that the Django MeerCat server is running:

python3 manage.py runserver

1. In your browser, open the link to the homepage, go to /admin/ and login.
2. Click the **“+ Add**” link by Projects, which will take you to a new page.
3. In the text box to the right of “**URL**” use the same link as the GitHub clone link and then check “**has github**”. The fields Documentation library and Code quality library can be set to “default”.
4. Press the save button at the bottom of the page to save the new project.

***Note****: The title of the project* ***must*** *be the same as the repo name. For example, if adding the project* [*https://github.com/MILO/myproject.git*](https://github.com/MILO/myproject.git)*, then the title must be set to “myproject”.*

**III. Python scripts:**

1. In the src folder is a script for getting the project/repository data into the MeerCat database. First, amend src/gitutils/update\_database.py to add in the correct host, database name, username, password, and port if not using default.

***Note****: Mac users will also need to remove “2>&1 | tee {…}” from all the commands listed in update\_database.py [approx. 4]*

1. Next, create a file called credentials.ini in the directory that contains the meercat and src folders. This file will contain your GitHub credentials needed for using the GitHub API to access pull requests, issues, and event data. Type the following into the file, but replace {username} and {token} with your GitHub username and token.

[github]

login = {username}

token = {token}

1. Now open a new terminal and move up to ideas-uo folder and run update\_database.py to connect the repo to the MeerCat database.

python3 src/gitutils/update\_database.py <username> <password>

* If you face any issues here, please remove the “nohup” command in all of the 4 python commands in update\_database.py to help check for errors.
* If issues still persist, add the below args to each of the python commands. E.g.

git\_command = f'python -m src.gitutils.db\_interface *--port {PORT}* --username {USERNAME} --password {PASSWORD} --add\_project {source\_url}’

1. The command above is the script used to update the database. Depending on project use, it may be helpful to set the script run automatically nightly.
2. If the commits or pull requests don’t show up, first try making a change in any of your repo files and re-run the python script.
3. Optional: If not using a localhost, webhooks in GitHub can be used to trigger more immediate database updates. On GitHub, go to your repository and click on repo settings on the right. Under “Code and automation” click “Webhooks” and select add new. The content type must be “application/json”, select “Just the push event” as the trigger, and select “Active” before saving. The Payload URL should be the host address + ‘/dashboard/githubbot/’.