MySQL Database + Python Bottle Installation

Update

Start by making sure your system is up to date. For apt based systems, use the command `sudo apt update && sudo apt upgrade -y`. From this point onward, I'm assuming the use of apt. ## MySQL For MySQL we will use mariadb, which is a newer opensource fork of MySQL but still remains fully backward compatible. For more information on mariadb, see their github page here. Follow these steps to install mariadb-server

1. `sudo apt install mariadb-server`
2. If you want mariadb to start automatically when you turn on your computer, use `sudo systemctl enable mariadb`. Otherwise use `sudo systemctl disable mariadb`
3. Start mariadb using `sudo systemctl start mariadb`. You can check that mariadb is running using `systemctl status mariadb`
4. With mariadb running, complete the installation by running `sudo mysql_secure_installation`. Complete the installation process like so:
    - Enter the current password for root: (just press enter)
    - Switch to unix socket authentication: n
    - Change the root password?: n
    - Remove anonymous users?: n
    - Disallow root login remotely?: y
    - Remove test database and access to it?: y
    - Reload privilege tables now?: y

Database Setup

Now we have to configure the database to correspond with what the server is expecting:

1. With mariadb running, enter the database as root using `sudo mysql`
2. Create a user called 'admin' with password 'admin' `grant all on *.* to 'admin' identified by 'admin' with grant option;`
3. Exit MySQL using `quit` or `exit`

Python Setup

1. Install all required dependencies using `sudo apt install python3 python3-pip build-essential pkg-config python3-dev default-libmysqlclient-dev`
2. Install required pip packages `pip install bottle bottle-sqlalchemy sqlalchemy-utils mysqlclient`
    - Note that mysqlclient may fail a few times as it finds the correct version. My system used version 2.1.1

Running the Server

You should now have everything required to run `test-server.py`. With the server running, you can connect to it using `http://127.0.0.1:8080/`. This should return the current state of the test_table.