

## Using flask-migrate to update database for backend application:

Depending on whether the database is already set up and has data in it the steps can vary a little. Since we didn't have any important data I decided to delete our database and start from scratch. Going forward since our database and migration file is already set up we can just run flask db migrate then upgrade (#1). If we have to start over we can reset the entire database using the commands on the bottom (#2).

### #1 Migrating database after structural changes (add/delete - column/table) to existing database:

1. Activate virtual environment in your project folder
2. Set FLASK\_APP = app.py (your app.py filename, gritter.py for us)
3. Run: flask db migrate -m "...migration comment..." (creates migration script for current change)
4. Run: flask db upgrade (implements the current changes)
5. Finally log into heroku and run: heroku run flask db upgrade --app [heroku app name] (gritter-3308 for us)

\*May have to run python3 -m flask db [command] if you have more than one version of python

One caveat: Try not to change the names of existing database columns. The auto-migration feature that we are using cannot tell the difference between a column name change vs a drop/add new column, so it interprets a name change as a drop/add column and will drop existing data with it.

---

### #2 Starting a migration file by deleting existing database and resetting the database (starting over from scratch):

1. Delete migrations folder from workspace (if no migration folder to begin with skip this step)
2. Delete database file (ie app.db) (again skip this step if no database to begin with)
3. Activate virtual environment in your project folder
4. Set FLASK\_APP = app.py (your app.py filename, gritter.py for us)
5. Run: flask db init
6. Run: flask db migrate -m "...migration comment..." (creates migration script for current change)
7. Run: flask db upgrade (implements the current changes)
8. Log into heroku
9. Run: heroku pg:reset --app gritter-3308 (resets postgres database on heroku)
10. Finally, run: heroku run flask db upgrade --app [heroku app name] (gritter-3308 for us)

\*May have to run python3 -m flask db [command] if you have more than one version of python