Project User Guide

Environment

Python	3.7.5
PyMySQL	0.9.3
Mysqlclient	1.4.4
MySQL -connector-python	8.0.18
Django	2.2.7
Flask	1.1.1
MySQL	5.6 or later
Recommend install: MySQL Workbench	6.3 or later
VS Code or PyCharm (need to run Django server and flask separately)	

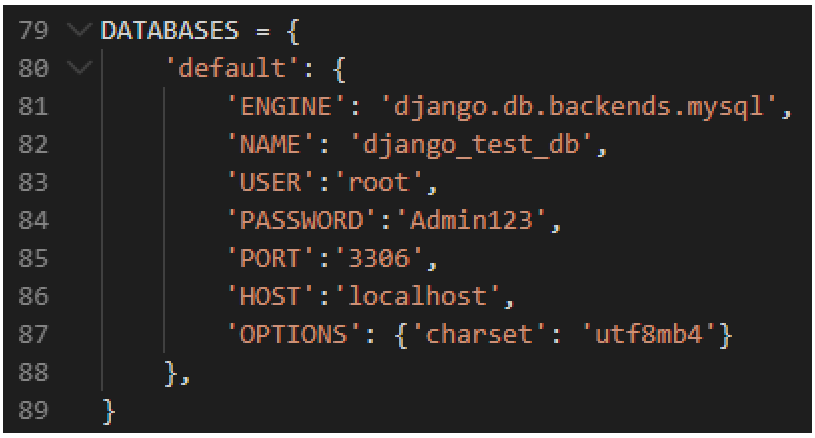
Database Import

>project> database.sql
In your SQL, import this database;

Django Settings

1. **Open VS Code (or other your preferred program), open folder: "Django"**

2. **>Django>hospital>settings.py:**



```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'django_test_db',
        'USER':'root',
        'PASSWORD':'Admin123',
        'PORT':'3306',
        'HOST':'localhost',
        'OPTIONS': {'charset': 'utf8mb4'}
    },
}
```

Change the DATABASES parameters to your database (NAME, USER, PASSWORD, PORT, HOST)

3. >Django (back to Django project folder)

Shell:

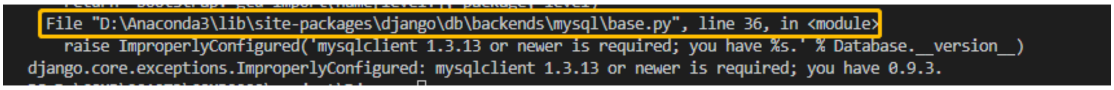
Python (or Python3) manage.py migrate

You'll see a warming message:

django.core.exceptions.ImproperlyConfigured: mysqlclient 1.3.13 or newer is required; you have 0.9.3.

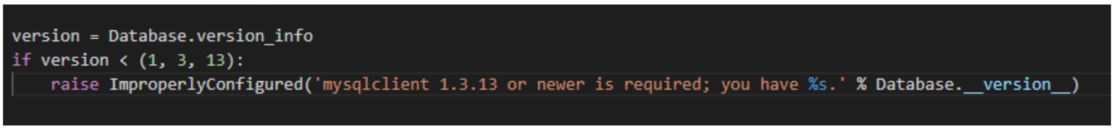
Don't worry, this is because Django's library has some conflict with mysqlclient module (since you have 1.4.4 but Django says you have 0.9.3, which is the version of pysql…)

4. Fix module conflict



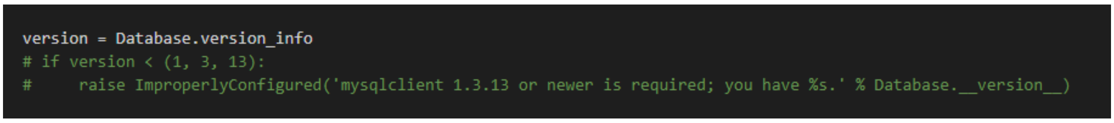
```
  File "D:\Anaconda3\lib\site-packages\django\db\backends\mysql\base.py", line 36, in <module>
    raise ImproperlyConfigured('mysqlclient 1.3.13 or newer is required; you have %s.' % Database.__version__)
django.core.exceptions.ImproperlyConfigured: mysqlclient 1.3.13 or newer is required; you have 0.9.3.
```

Go to this file: base.py, line 36:



```
version = Database.version_info
if version < (1, 3, 13):
    raise ImproperlyConfigured('mysqlclient 1.3.13 or newer is required; you have %s.' % Database.__version__)
```

Comment line 35 and 36:



```
version = Database.version_info
# if version < (1, 3, 13):
#     raise ImproperlyConfigured('mysqlclient 1.3.13 or newer is required; you have %s.' % Database.__version__)
```

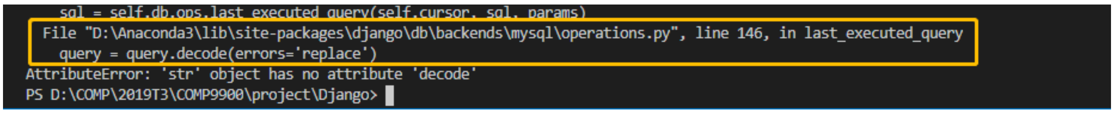
Then, in Shell: **Python (or Python3) manage.py migrate**

You'll see an error:

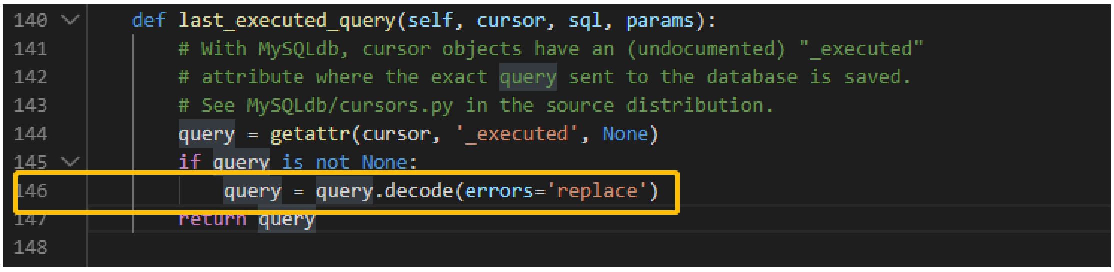
AttributeError: 'str' object has no attribute 'decode'

Don't worry, we are one-step to success;

Go to operation.py line 146:

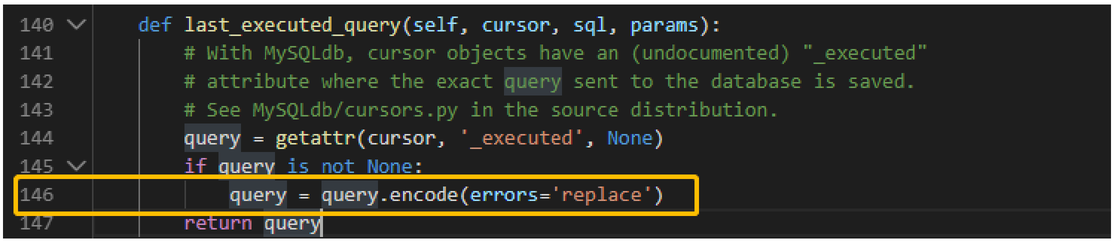


```
    sql = self.db.ops.last_executed_query(self.cursor, sql, params)
  File "D:\Anaconda3\lib\site-packages\django\db\backends\mysql\operations.py", line 146, in last_executed_query
    query = query.decode(errors='replace')
AttributeError: 'str' object has no attribute 'decode'
PS D:\COMP\2019T3\COMP9900\project\Django>
```



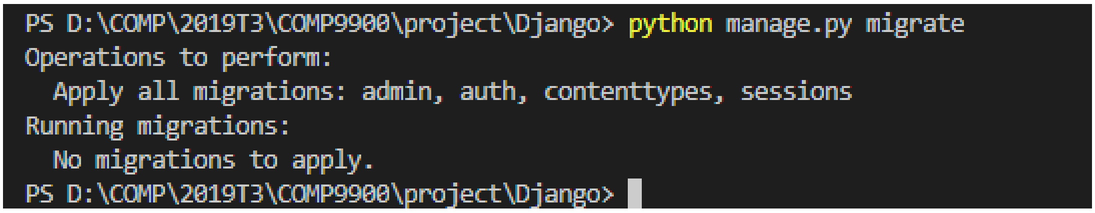
```
140     def last_executed_query(self, cursor, sql, params):
141         # With MySQLdb, cursor objects have an (undocumented) "_executed"
142         # attribute where the exact query sent to the database is saved.
143         # See MySQLdb/cursors.py in the source distribution.
144         query = getattr(cursor, '_executed', None)
145         if query is not None:
146             query = query.decode(errors='replace')
147         return query
148
```

Change decode to encode:



```python
    def last_executed_query(self, cursor, sql, params):
        # With MySQLdb, cursor objects have an (undocumented) "_executed"
        # attribute where the exact query sent to the database is saved.
        # See MySQLdb/cursors.py in the source distribution.
        query = getattr(cursor, '_executed', None)
        if query is not None:
            query = query.encode(errors='replace')
        return query
```

Then migrate again:

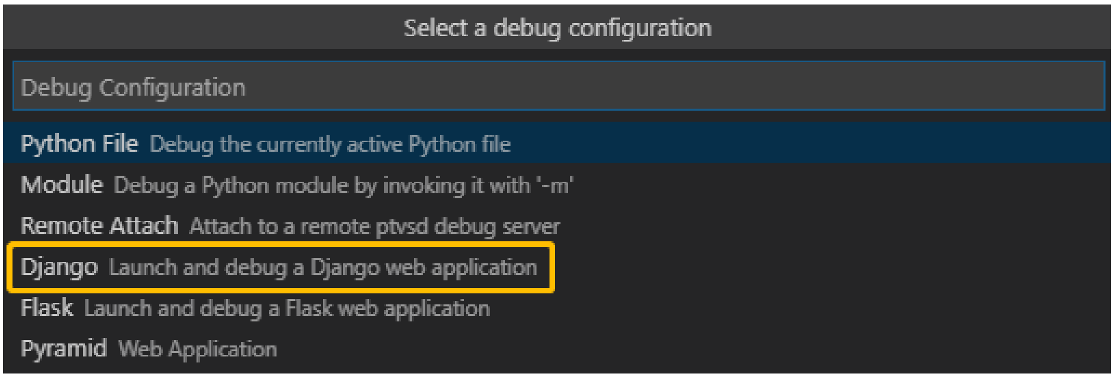


```
PS D:\COMP\2019T3\COMP9900\project\Django> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  No migrations to apply.
PS D:\COMP\2019T3\COMP9900\project\Django>
```

It’s done! (You’ll see some migrate procedures)

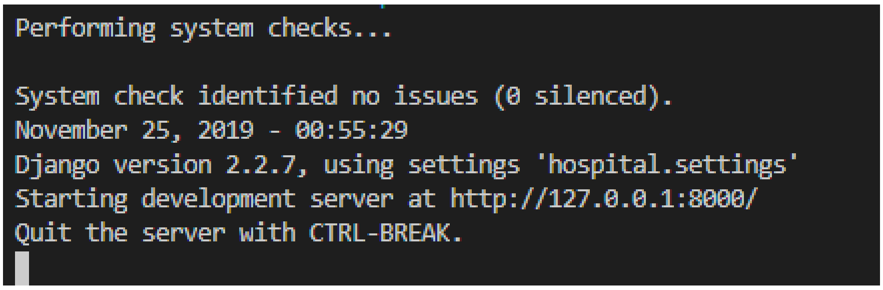
5. Run Django server:

If you are using VS Code, under your opened folder, simply press “F5” and choose:


Select a debug configuration
Debug Configuration
Python File Debug the currently active Python file
Module Debug a Python module by invoking it with '-m'
Remote Attach Attach to a remote ptvsd debug server
Django Launch and debug a Django web application
Flask Launch and debug a Flask web application
Pyramid Web Application

Or in Shell, type: **Python (or Python3) manage.py check**

After checking, in Shell, type: **Python (or Python3) manage.py runserver**



```
Performing system checks...

System check identified no issues (0 silenced).
November 25, 2019 - 00:55:29
Django version 2.2.7, using settings 'hospital.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

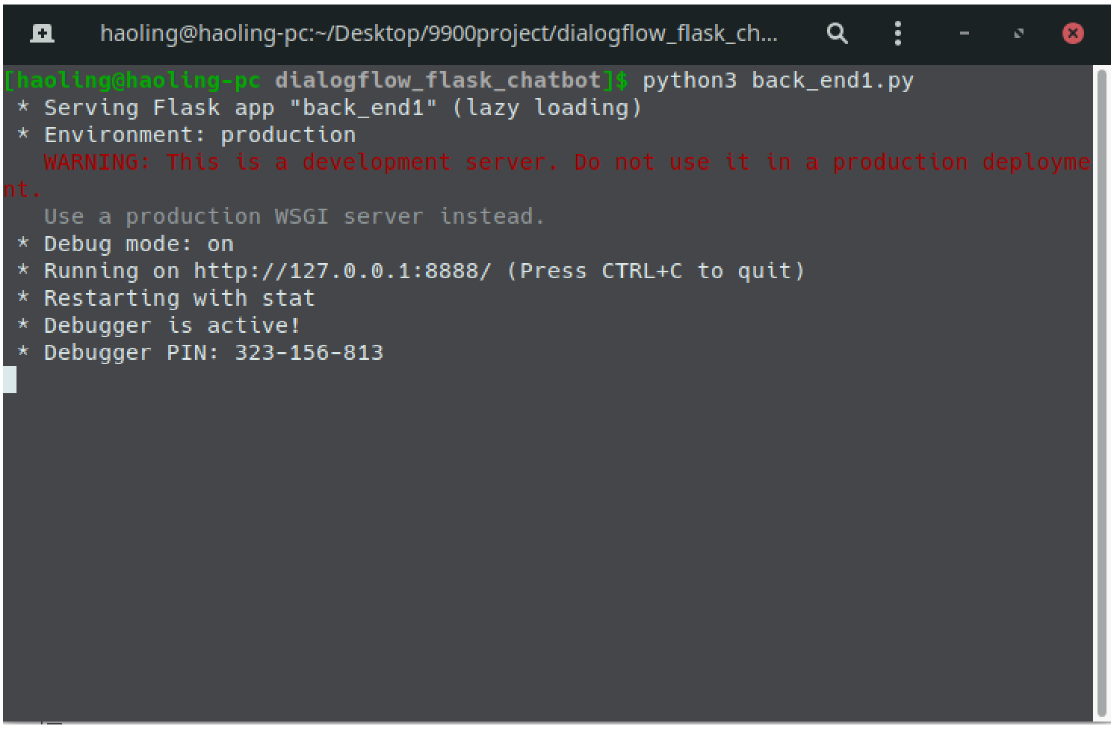
Django server has run successfully!

Flask

1. Open a terminal, and go to “>project>dialogflow_flask_chatbot” directory

2. Set a different port with the port running Django

3. input “python3 backend1.py” in terminal



haoling@haoling-pc:~/Desktop/9900project/dialogflow_flask_ch...

```
[haoling@haoling-pc dialogflow_flask_chatbot]$ python3 back_end1.py
 * Serving Flask app "back_end1" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployme
nt.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:8888/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 323-156-813
```

When you see the response above, you have run flask successfully.