web.py Exercise

Tom Kelliher, CS 317

1 Introduction

Begin this exercise once you've completed the web.py tutorial. In this exercise, you'll add a "todo item complete" function. This function should allow the simultaneous completion of one or more todo items. A todo item will be completed by deleting its row from the database. A possible extension would be to leave the row(s) in the database, but change the `done` attribute from false to true. This would give the capability of showing incomplete and complete todo list items.

This exercise is less of a "follow the recipe in the cookbook" endeavor than the tutorial. I've done this on purpose, to help you make the transition to working on applications on your own through taking advantage of materials on the Internet to find answers to your questions. Remember, however, if you get stuck I'm always just a "Hey, Tom!" away.

Visit `http://phoenix.goucher.edu/~kelliher/wsgi/tutorial.py/` to see a working version of this exercise.

2 Database Structure

It's helpful to keep in mind the structure of your database as you extend the tutorial application. Remember, your `todo` database table has the following structure:

```
Column  |            Type             |                      Modifiers
--------+-----------------------------+--------------------------------
id      | integer                     | not null default \
                                        nextval('todo_id_seq'::regclass)
title   | text                        |
created | timestamp without time zone | default now()
done    | boolean                     | default false
Indexes:
   "todo_pkey" PRIMARY KEY, btree (id)
```

What is the `psql` command to produce this schema table?

Here's an example instance of the table:

```
 id |             title              |          created           | done
----+--------------------------------+----------------------------+------
 17 | Sleep                          | 2017-02-06 15:47:04.927926 | f
 19 | Watch Shrek                    | 2017-02-06 16:00:17.516861 | f
 20 | Learn web.py                   | 2017-02-06 16:00:38.975293 | f
 23 | Extend my Arsenal of CS tools | 2017-02-06 16:01:19.366072 | f
 25 | Learn to love computers        | 2017-02-06 16:04:37.779041 | f
 26 | Find a new favorite number     | 2017-02-06 16:05:00.127227 | f
(6 rows)
```

What is the `psql` command to produce this table listing?

3 Modifying the Template File

Visit the URL above and look at the HTML source code. On most browsers, press `Ctrl-u` to do this. What you need to do is to reverse-engineer this HTML so that you can modify the template file to produce similar HTML.

4 Modifying the Python Application File

1. Open your `code.py` file for editing. Again looking at the HTML source code, what needs to be added to the `URL` tuple in your Python source in order to handle the input from the form when todo items are removed from the todo list?

2. If you change the `URL` tuple, you may have to add a class to your Python code. In this case, you'll need to do so. Looking at the HTML source, what should the name of this class be?

3. Your class will need a method. Again, looking at the HTML source, what should the name of the method be? Look at the existing class methods in your Python source file to determine the parameter list for the method.

4. Now, write the code to process the deletion of the selected todo list items.

   (a) Some debugging will likely be involved here. So, use your development web server rather than phoenix's web server. This will allow you to insert `print` statements into your code for debugging. Make sure you remove these once you're finished with them.

   (b) The web form data passed in to your class method contains multiple inputs. Study web.py's documentation to determine how to handle these.

   (c) To write the line of code to delete an item from your `todo` table, again, study the web.py documentation. (I know, the documentation can be a bit confusing. That's normal; it's time to begin getting used to this.)