

# Project 3: To the Web!

Xuechen Zhang  
03/28/2017

# Outline

- **Run the sample code**
- Examine the code
- Use Transactions in web.py
- Debug web.py
- Project 3

# The Sample Code

- Copy web.py on argo14 [IP: 172.17.0.238]

```
cp -r /usr/class/cs351/project/pa3/web.py/* ~/mini-bay
```

- Copy to an ENCS lab machine

```
scp -r [account]@172.17.0.238:/usr/class/cs351/project/pa3/web.py mini-bay
```

# The Sample Code (Cont.)

- *auctionbase.py*: your “main” application. Responsible for handling requests from the browser.
- *sqlitedb.py*: your database manager. Responsible for interacting with your database.
- *templates/*: directory that contains your template files that correspond to your various URLs. Every web page will require a new template file in this directory.
- *lib/*: directory that contains the library files for web.py and Jinja2

**Do not modify any files in the lib/ directory!!!**

# Generate SQLite Database

- Use your *createDatabase.sh* script from Part 2 of the project
- Generate your SQLite database(.db) binary file
- Move it into your *mini-bay* directory.

# Run the Sample Code

- Modify the *sqlitedb.py* file to use the correct filename of your SQLite database
- Run web.py

*python auctionbase.py*

- Run a web browser (e.g., firefox)

*http://localhost:8080/currtime*

# Outline

- Run the sample code
- **Examine the code**
- Use Transactions in web.py
- Debug web.py
- Project 3

# Query the Database

In `sqlitedb.py`

```
# returns the current time from your database  
def getTime():  
    query_string = 'select time from CurrentTime'  
    results = query(query_string)  
    return results[0].time # alternatively: return results[0]['time']
```

In `auctionbase.py`

```
class curr_time:  
    # A simple GET request, to '/currtime'  
    #  
    # Notice that we pass in `current_time` to our `render_template` call  
    # in order to have its value displayed on the web page  
    def GET(self):  
        current_time = sqlitedb.getTime()  
        return render_template('curr_time.html', time = current_time)
```



# Query the Database (Cont)

```
# returns a single item specified by the Item's ID in the database
def getItemById(item_id):
    q = 'select * from Item where ID = $itemID'
    result = query(q, { 'itemID': item_id })

    try:
        return result[0]
    except IndexError:
        return None
```

Do not directly concatenate the query string using standard Python string operations.  
**Will introduce a security loophole into your website.**

# Add additional URLs

- Mapping from URL to Python class
- Implement GET and/or POST method
- Invoke *render\_template* and return its value

```
# first parameter => URL, second parameter => class name
urls = (
    '/currtime', 'curr_time',
    '/selecttime', 'select_time',
    '/add_bid', 'add_bid',
    '/search', 'search_items',
    '/(.*)', 'view_item'
)
```

# Add additional URLs (Cont.)

```
class curr_time:
    # A simple GET request, to '/currtime'
    #
    # Notice that we pass in `current_time` to our `render_template` call
    # in order to have its value displayed on the web page
    def GET(self):
        current_time = sqlitedb.getTime()
        return render_template('curr_time.html', time = current_time)
```

# Handle Parameters from Users

- `render_template()` arguments:
  - The name of the template file to be rendered
  - A dictionary of key-value pairs as variables

```
class curr_time:
    # A simple GET request, to '/currtime'
    #
    # Notice that we pass in `current_time` to our `render_template` call
    # in order to have its value displayed on the web page
    def GET(self):
        current_time = sqllitedb.getTime()
        return render_template('curr_time.html', time = current_time)
```

# Handle Parameters from Users (Cont.)

In templates/curr\_time.html

```
{% extends "app_base.html" %}
{% block content %}
{# include the first two lines above to extend the app_base template #}
    <h3>Current Time</h3>
    <div class="alert alert-info">Current time is: {{ time }}</div>
{# also make sure to include this very last line as well #}
{% endblock %}
```

1. “time” is set at rendering
2. Webpage inheritance via extends, block, endblock

# Outline

- Run the sample code
- Examine the code
- **Use Transactions in web.py**
- **Debug web.py**
- Project 3

# Transactions in web.py

```
def setTime(new_time):  
    t = db.transaction()  
    try:  
        db.update('CurrentTime', where="time", time = new_time)  
    except Exception as e:  
        t.rollback()  
        print str(e)  
    else:  
        t.commit()
```

Initiate a  
transaction

Print the error  
message

# Debug the Code

- Debug on Argo14
  - SSH –X
  - Firefox &
  - Ctrl-C
- Use print statements
  - Print statements will **interfere** with the rendering of webpages



# Outline

- Run the sample code
- Examine the code
- Use Transactions in web.py
- Debug web.py
- **Project 3**

# Project 3

- Due: 04/24
- Basic functionalities
  - Change the “current time”
  - Enter bids on open auctions
  - Auction closing
  - Browse auctions of interest based on input parameters, e.g., item ID
  - View relevant information related to an auction
  - Add new users

# Project 3 (Cont.)

- Realistic bidding behavior
  - Reject bids that are less than or equal to the current highest bid
  - Bid on closed auctions
  - Bid from users that don't exist
- Group SQL queries into transactions whenever possible
- Submission
  - Remove print statements
  - Test your code on argo14