CSSE1001

05/09/2009

Assignment 3 - Design Document

1. Description

TumblrServ is primarily a tool to help web developers create themes for the popular blogging platform Tumblr (see http://www.tumblr.com). Currently, designers must enter their code into a dialog on a webpage, save it and then attempt to reload the page with the new code.

There are many problems with this approach:

- Images must be hosted on a different server, making testing dependent on the configuration of another server
- Developers often use their own unique development process, Tumblr forces you to use theirs
- Updating the code can take as long as 30 seconds; this is far too long to wait in order to receive feedback
- Developers cannot select the data they wish to use in the test environment
- Offline development is impossible

TumblrServ solves all of these problems by running a small server that listens on the local machine. It parses all theme data and performs operations given data from a flat file to generate a page in the same manner as the parsing engine on the Tumblr servers. This allows developers to immediately see the changes as they work in their preferred development environment. Images are also hosted locally in order to keep complexities to a minimum.

2. User Interface

TumblrServ has no graphical user interface. It is a command line application that is controlled by the switches passed to the application, for example:

tumblrserv.py --local-data
tumblrserv.py --port=8080 --autorefresh
tumblrserv.py --remote-data=username

The server will run on a port specified by a command line switch or configuration file, in that order. TumblrServ follows the principles of Don't Repeat Yourself (DRY), meaning that it will attempt to guess a configuration if none is supplied.

After the server has started, the developer simply needs to point their browser at http://localhost:8080 (the port number could potentially be different) and they will see their theme.

3. Design

The basic functionality of TumblrServ can be broken down into a procedural list:

1. Parse command line switches
2. Load configuration files
3. Load local data or retrieve data from remote location
4. Parse configuration and data into useable format
5. Assign data to classes
6. Load theme data
7. Parse theme data and insert transformed data
8. Generate HTML page and serve via HTTP

CSSE1001

05/09/2009

Assignment 3 - Design Document

Tumblr posts are always given classes inside the web interface, so it makes sense to abstract this data into a heirachy of classes. TumblrServ implements the following classes to do this:

- Post
  - TextPost
  - PhotoPost
  - QuotePost
  - LinkPost
  - ChatPost
  - AudioPost
  - VideoPost

These classes are needed because each post type has a specific way of displaying itself and to attempt to parse all of these types in one function would almost definitely make the code unwieldy and difficult to understand.

4. Supporting Modules

TumblrServ uses Yaml to store all of its configuration data and uses the PyYaml module to parse the data into sets of dictionaries and lists. Similarly, PyJSON is used to extract data from the Tumblr servers. For the webserver component it uses CherryPy, a well known Python server framework.