OST Blog Developer Guide

by Liang Fang

1 Introduction

This project is a blogging platform website written by python and deployed on Google Appengine, using webapp2 framework, Google Appengine datastore and rendering HTML with Jinja2 template.

2 System design

As we know, The Python environment provides rich Python APIs for the datastore, Google Accounts, URL fetch, and email services. App Engine also provides a simple Python web application framework called webapp2 to make it easy to start building applications.

2.1 Overall Design

My project uses Jinja2 template, opened to all users and Google users can operate on this blog.

All HTML files are inside “template” directory.

Model class and handler function are written in “main.py” files.

“app.yaml” file is used to configure system

“index.yaml” is used to configure datastore search indexes.

2.2 HTML Page Architecture

This website has 8 html file plus 1 xml file.

index.html

This is the base html file. Since Jinja2 template supports template inheritance, so some of my subpage html files extends this base file.

bloglist.html

This is the main page of blog system. This page displays all blogs users created, anyone can visit these blogs but only owner can post and edit posts.

createblog.html

This is the page of creating blog.

singleblog.html

After user click blog's title, web browser will lead user to this page, displaying all posts of this specific blog.

singlepost.html

After user click post's title on singleblog page, web browser will turn to this page, displaying the whole content of this post, and with post and edit button.

post.html

If user click post on singlepost page, he will be lead to this page, so that he can write new post.

If a user is not owner of this blog, web browser will lead him to error.html page. If this user is not logged in, page will be directed to Google Account login page.

editpost.html

This is the page of editing posts which are already posted. Besides, if a user want to upload pictures to this post, it has to be down from this page.

error.html

This is a simple error report page for not login or not owner of blog.

rss.xml

This file defines the structure of RSS feed of this blog. Each blog will use this file to generate RSS feed.

2.3 Code Design

My project source code are all written in "main.py". The fist part is importing library functions. The second part is defining Model class and helper function of these classes.

All Model classes are Post, Tag, Blog, Image.

The third part is the definition of handler fucntions. These functions will respond to any request to those specific self defined url:

```
('/', MainPage),
('/createblog', CreateBlog),
('/singleblog/(.*)', BlogPage),
('/post/(.*)', Postblog),
('/singlepost/(.*)', SinglePost),
('/editpost/(.*)', EditPost),
('/tag/(.*)/(.*)', TagHandler),
('/rss/(.*)', RssHandler),
('/image/(.*)', ImageHandler),
('/comment/(.*)', CommentHandler)
```

3 Handler Class

3.1 MainPage

This class display all blogs user created on the bloglist page, render "bloglist.html".

Besides, system will handler Google user login in and log out operations.

3.2 CreateBlog

This class respond to user's click on "Create Blog". Using get method to render "createblog.html", capturing information user write and put into datastore.

3.3 BlogPage

This class render "singleblog.html" and query the datastore to display all posts that belong to selected blog.

3.4 Postblog

This class render "post.html" and get user's input and put into datastore to create a new post on specific blog.

3.5 SinglePost

This class display single post that user select by clicking on post's title.

3.6 EditPost

This class respond to request to "/editpost/(.*)" which should be generated by clicking "edit". It render "editpost.html" and fill blank with original contents and wait for user to change and do posting again to update data in datastore.

3.7 TagHandler

This class respond to user's click on any tags. It will query the datastore to display all posts in this blog that have the same tag, then it will render "singleblog.html"

3.8 RssHandler

This class handle each blog's RSS generation. It will gather each blog's all posts and pass to "rss.xml" to generate RSS feed.

3.9 ImageHandler

This class is used to display image on html pages after user have uploaded image.

3.10 CommentHandler

This class respond to user's operation of submitting comment under any post. It will gather comment's author information and comment's content then put into datastore.