OPEN SOURCE TOOL FINAL PROJECT SUMMARY

Name: Yuan Jia   ID: N14467648

1 Abstract

In this project I completed 10 core features and developed it with Git. All .py and .html files are in /src directory.

2 URL: http://yjstormblog.appspot.com/

3 Feature Explain

3.1 Multiple user and multiple blog

Different users can login with their email account. And each one of them can register Blog by enter new blog's name and press Register button in the page bottom. Notice that blog name must be unique across users.

And they can also switch between blogs by select from dropdown menu and press switch button.

3.2 Select blog and write post

After select a blog, there will be a New Post link at top of the page if current user is the blog's author. Click this link redirect user to edit_post.html where they can write a post and submit.

3.3 View blog with/without login And 10 posts per page

If user is logged in, there will be a welcome word at page bottom. Otherwise it shows anonymous. If posts are more than 10, they will be shown 10 per page and an older post link will display. (Change variable MAX_POST_PER_PAGE in main.py to a smaller value can make this easier to test)

3.4 Max char limitation and permalink

I use jinja2 filter truncate(500) to set the char limit on the blog view. And each post will have a permalink with it.

3.5 Timestamp and post modify

Each post has a create time and modify time. If current user is post's author, they will be able to modify posts in the post view(read_post.html).

3.6 Add tag

If current user is author of the post, they can add tag at the post view.

3.7 Search by tag and display 10 maximum per page

In the Home Page, a list of tags will be generated. Click one of them will search all posts with that tag and display 10 max per page. (Change variable MAX_POST_PER_PAGE in main.py to a smaller value can make this easier to test)

3.8 Link and Pics

Links will be auto generated. And if a link end with .jpg, .png, .gif or it is from Google's Blobstore(eg. http://lh5.ggpht.com/XXX/XXXXX), the picture will be displayed inline. (Note: All links must be separated by space.)

3.9 Blog gallery

Each blog has a gallery, user can upload pictures. If user wants to refer pictures, just right click on the pictures in the gallery and choose copy image's url.

3.10 RSS

In blog view, user can click RSS link. Then the RSS of this blog will be displayed in the browser.

4 Code Explain

4.1 Project Structure:

According to the google app engine tutorial, this project have three types of files.

4.2 app.yaml

This file save the app engine configurations. We can set application name, version, handler and libraries in it. I set main.py as my only handler file. And I choose to use webapp2 framework and jinja2 template.

4.3 main.py

This file has some helper functions in it. I will explain them in the feature part. It also create a jinja environment to help us render html templates with values.

MainPage class checks whether we have selected a blog. If not, the Home Page will be shown and all the posts will be shown in reverse create time order. After selected a blog with 'Switch Blog' button in the bottom, only post in that blog will display.

At the bottom of source code, it defined an application variable to associate handler classed with urls.

4.4 blog_model.py

It defines Blog, BlogPost and Picture three classes. Each class has some ndb properties.

4.5 post_handler.py

PostEditPage class helps to render the edit_post.html file.

AddPost class check whether there is keyurl field in url. If there is, then it fetch the post with that keyurl then modify and save it. If not, it means we are creating new post.

ReadPost class help to render read_post.html

4.6 blog_handler.py

RegisterBlog class first check whether the new blog name exists. If yes, it will prevent user from create new blog. So the blog name will be unique across users.

SwitchBlog just help to switch between blogs.

BlogGallery render the blog_gallery.html. It displays all the pictures in that blog and provide an upload_url if we want to upload new picture.

UploadHandler inherits BlogstoreUploadHandler. It upload the file we selected in line 75 of blog_handler.py and redirect to the gallery page.

RSSHandler takes a blog name, query that blog and posts belong to that blog. Then is generate the blog's RSS in plaintext and display it in the browser.

4.7 tag_handler.py

AddTag first check whether the tag exist in the post. If not, then add the tag to that post. Then redirect back to read_post.html.

SearchTag search all posts with that tag and display them with 10 maximum one page.