rembrant Documentation

Release 0.2.0

Honza Pokorny

May 07, 2012

CONTENTS

Rembrant is a piece of software that helps you organize your photos and create online galleries. You run the software locally to create the gallery, add tags and collections and then you sync it to your online account on AWS S3.

CHAPTER

ONE

ADVANTAGES

- Organize your photos in the browser
- Structure saved in a single JSON file (unlike iPhoto's proprietary format)
- You own your photographs, not Flickr or Picasa
- The website is generated on your computer and not on the server, meaning that it's super fast. All the user has to do is download it.
- Cheap - about $1 for 3GB of images (including bandwidth)
- Flexible
- Familiar Django templates
- Your captions, tags, sets, etc are backed up
- Customizable user interface
- Your library can be version controlled (git, hg, etc)

CHAPTER

TWO

DISADVANTAGES

- You have to run a local server (you need to be handy with the terminal)
- The gallery can only be hosted on a subdomain (e.g. `http://gallery.example.com`, but not `http://example.com`)

CHAPTER

THREE

CONTENTS

3.1 Installation

It's recommended that you install rembrant with virtualenv.

First, clone the project from Github:

```
$ git clone git://github.com/honza/rembrant.git
$ cd rembrant
```

Create your environment and install the requirements:

```
$ virtualenv env --no-site-packages
$ source env/bin/activate
(env) $ pip install -r requirements.txt
```

Initialize your library:

```
$ python rembrant.py init
```

This will create a `library.json` file in the `rembrant/` directory. Next, you will want to symlink your photo directory to `rembrant/photos`. Then, load your photos into the library:

```
$ python rembrant.py load
```

This will add your photos to the `library.json` file. It will assign an *id* to each photo and a *sha* digest to make sure it's unique. It will also create 2 thumbnails for each photo. One that's 100px wide and one that's 800px wide. By default, it will place the thumbnails to the `cache` variable set in your library file.

Next, we will export your gallery to HTML.

```
$ python rembrant.py export
```

This will create a static site representation of your gallery. It will create a detail page for each photo, and a blog-like feed of photos. The HTML generated by this command will be placed in the `build/` directory.

Next, we will deploy this code to AWS S3.

```
$ python rembrant.py deploy
```

This will copy all of the files in `build/` to your S3 bucket.

Note: Before you can deploy code to AWS S3, please make sure that the `aws_key`, `aws_secret` and `aws_bucket` settings are populated.

3.2 Configuration

source
This is the name of the directory where rembrant expects to find image files.

cache
This is the name of the directory where rembrant will place thumbnails.

templates
The name of the directory where your templates are stored.

aws_key
Your AWS access key. This is used for deployment to S3.

aws_secret
Your AWS access secret. This is used for deployment to S3.

aws_bucket
The name of the AWS S3 bucket where you would like your gallery to be deployed.

last_modified
This value is automatically update each time any changes are made to your library.

3.3 Model reference

This is the public API for rembrant.

> **Warning:** This API may change at any time. When rembrant reaches 1.0.0, the API will become stable.

3.3.1 Photo model

class **Photo**
This class contains all the information and metadata about a single photo.

title
The title for the photo.

filename
The photo's filename.

caption
The photo's caption.

sha
A hexdigest of a SHA hash of the original image file.

small_thumb
A filename of the small thumbnail. 100px wide by default.

big_thumb
A filename of the big thumbnail. 800px wide by default.

to_json()
Return a Python dictionary representing the `Photo` instance suitable for serialization into JSON. Related objects are represented as lists of primary keys.