

Journal Tool (jt) acceptance tests

Lars Wirzenius

0.2-64-g5c15f80

Abstract

jt is a tool for managing my personal journal and knowledge base. It creates and manipulates files in a git repository. The journal is rendered to a website using ikiwiki. This document is the automated acceptance test suite for jt.

Contents

1	Introduction	2
2	Basic operation	4
2.1	Make the first journal entry	4
2.2	Edit two drafts	4
2.3	Remove draft	5
2.4	Attach a file to a draft	5
2.5	Create a topic	5

Chapter 1

Introduction

Journal Tool (jt) is a Unix command line tool for maintaining my personal journal. The journal is rendered into a (private) website using ikiwiki. All jt does is create and manipulate some files in a git repository. All the interesting parts of building a useful journal, or personal knowledge base, comes from the user entering information and using ikiwiki constructs to make it accessible. The main constructs I use are blogging, tagging, and categories.

This document forms the automated acceptance test suite for jt.

The journal is structured as a private blog, where jt allows the user to create new entries. The entries are collected by ikiwiki into a timeline. Each entry can be assigned arbitrary tags, and ikiwiki constructs a list of entries with each tag. JT can also create topic pages for specific categories, e.g., for projects, and again ikiwiki will construct a list of entries in each category.

The journal is manipulated as a “source tree”, consisting of files in the markdown format. The files are stored in a git repository, for version control, and for sharing between computers.

How the source tree is laid out mostly doesn't matter for jt, except the following subdirectories in the journal must exist:

- a directory for drafts of new entries (“drafts”)
- a directory for finished journal entries (“blog”)

The names of both directories can be configured for jt. If the directories don't exist, jt will create them as needed. Further, jt will put finished entries into a date-based hierarchy. The minimal tree for jt would look something like below.

```
path/to/journal/  
  .git/  
  drafts/  
  blog/  
    2019/  
      06/  
        15/  
          my_first_journal_entry.mdwn
```

When a new journal is started, the git repository needs to exist, and the user needs to add any category pages, and other structures to it, but jt will help with creating new journal entries and topics.

Chapter 2

Basic operation

2.1 Make the first journal entry

We start a new journal and create the first journal entry. The content of the entry don't matter, only that the draft file is created.

given an empty journal
when I run *jt new* “*random title*”
then there is 1 draft
and draft *0* has title “*random title*”
and there are no journal entries

We have a draft, and at this point we spend some time writing the entry, using the editor that *jt* starts. When we're done, we tell *jt* that we're finished with the entry.

Note that *jt* can't rely on the editor exiting to know this, because the editor might be something like *emacsclient*, which exits at once (after telling a background *emacs* to open the file), but also because we might decide to back out of writing the journal entry in the first place.

when I run *jt finish*
then there are no drafts
and there is 1 journal entry

2.2 Edit two drafts

We can have multiple drafts happening. The *jt* tool will need to be told which draft to operate on.

given an empty journal
when I run *jt new* “*first entry*”
and I run *jt new* “*second entry*”
then there are 2 drafts
and draft *0* has title “*first entry*”
and draft *1* has title “*second entry*”

when I try to run jt finish
then it fails

Drafts can be finished in any order.

when I run jt finish 0
then there is 1 draft
and there is 1 journal entry

when I run jt finish
then there are no drafts
and there are 2 journal entries

2.3 Remove draft

We can decide not to finish a draft and remove it instead. Removal always requires specifying the draft.

given an empty journal
when I run jt new "random title"
then there is 1 draft
when I try to run jt remove
then it fails
when I run jt remove 0
then there are no drafts

2.4 Attach a file to a draft

It's possible to attach files to drafts, for example images. The attachments will be published with the entry.

given an empty journal
and a file pretty.jpg
when I run jt new "title"
and I run jt attach 0 pretty.jpg
then there is 1 draft, with attachment pretty.jpg
when I run jt finish
then there are no drafts
and there is 1 journal entry, with attachment pretty.jpg

2.5 Create a topic

JT can create topic pages as well and mark new entries as belonging to a topic. The topic pages tell ikiwiki collect all entries belonging to the topic.

given an empty journal
when I run jt new-topic topics/cats "Kittens"
then there there is 1 topic, topics/cats, titled "Kittens"

*when I run `jt new -topic topics/cats "New kitten"`
and I run `jt finish`
then there is 1 journal entry, belonging to `topics/cats`*