jt—a journalling tool

Lars Wirzenius and Daniel Silverstone

2022-06-18 08:48

Contents

1 Introduction

The **jt** software (short for "journalling tool") is a helper for maintaining a journal or personal knowledge base. It has been written for the personal use of its authors, but might be useful for others.

The guiding principle for jt is that having longevity for one's data and complete control over it are crucial. Because of these, the approach taken by jt is to build a static web site from source files stored in a version control system. The files will be edited with a text editor chosen by the journal writer, rather than via a custom web, desktop, or mobile application. The built journal is then served with some suitable web server.

The role of jt is to make it easier to create new journal entries (new Markdown files), and to make all the common tasks of maintaining a knowledge base have minimal friction.

1.1 Example

The following example creates a new journal, which will be the default journal for the user, and a new journal entry. The entry is a draft until it's finished.

```
$ jt --dirname ~/Journal init default "My private journal"
$ jt new --tag meta --tag journalling "My first journal entry"
... # text editor is opened so the new entry can be written
$ jt finish 0 first-entry
```

2 Acceptance criteria and their verification

This chapter defines detailed acceptance criteria and how they're verified using *scenarios* for the Subplot[1] tool.

2.1 Configuration file handling

These scenarios verify that jt handles its configuration file correctly.

2.1.1 Shows defaults if no configuration file is present

given an installed jt
when I run **jt config**
then stdout matches regex **dirname:.*/\.local.share/jt**
and stdout matches regex **editor: "/usr/bin/editor"**

2.1.2 Gives error if configuration missing file specified

given an installed jt
when I try to run **jt --config does-not-exist config**
then command fails
and stderr contains "**does-not-exist**"

2.1.3 Accepts empty configuration file

given an installed jt
and file **empty.yaml**
when I run **jt --config empty.yaml config**
then stdout matches regex **dirname:.*/\.local.share/jt**
and stdout matches regex **editor: "/usr/bin/editor"**

File: **empty.yaml**

[1] `https://subplot.liw.fi/`

```
{}
```

2.1.4 Accepts configuration file

Note that the configuration file uses a tilde syntax to refer to the user's home directory.

given an installed jt
and file **config.yaml**
when I run **jt --config config.yaml config**
then stdout matches regex **dirname:.*/.*/journal**
and stdout matches regex **editor: "emacs"**

File: **config.yaml**

```
dirname: ~/journal
editor: emacs
```

2.1.5 Command line options override configuration file fields

given an installed jt
and file **config.yaml**
when I run **jt --config config.yaml --dirname xxx --editor yyy config**
then stdout matches regex **dirname: "xxx"**
and stdout matches regex **editor: "yyy"**

2.1.6 Rejects configuration file with extra entries

given an installed jt
and file **toomuch.yaml**
when I try to run **jt --config toomuch.yaml config**
then command fails
and stderr contains "**unknown_field**"

File: **toomuch.yaml**

```
unknown_field: foo
```

2.2 Create a new local journal repository

`jt` works on a local repository, and it can be created an initialised using the tool.

given an installed jt
when I run **jt --dirname jrnl init default "My test journal"**
then command is successful
and directory **jrnl** exists
and there are no uncommitted changes in **jrnl**
when I run **jt --dirname jrnl is-journal**
then command is successful

when I try to run **jt --dirname bogus is-journal**
then command fails

2.3 Create a new draft, edit it, then publish it

Verify that we can create a new draft entry for the journal.

given an installed jt
when I run **jt --dirname jrnl init default "My test journal"**
then command is successful
and there are no drafts in **jrnl**
and there are no journal entries in **jrnl**
when I run **jt --editor=none --dirname=jrnl new "Abracadabra"**
then command is successful
and there is one draft in **jrnl**
and draft **0** in **jrnl** contains "**Abracadabra**"
and draft **0** in **jrnl** contains "**!meta date=**"
when I run **jt --dirname=jrnl list**
then stdout matches regex **^0 Abracadabra$**
given an executable script **append.sh**
when I run **jt --editor=./append.sh --dirname=jrnl edit 0**
then command is successful
and draft **0** in **jrnl** contains "**Open sesame!**"
when I run **jt --dirname=jrnl finish 0 abra**
then command is successful
and there is one journal entry in **jrnl**, at **FILE**
and file name <**FILE**> ends with **.mdwn**
and journal entry <**FILE**> contains "**Abracadabra**"
and journal entry <**FILE**> contains "**Open sesame!**"
and there are no drafts in **jrnl**
and there are no uncommitted changes in **jrnl**

File: **append.sh**

```
#!/bin/sh
set -eux
echo "Open sesame!" >> "$1"
```

2.4 Create two drafts

Verify that we can create two draft entries at the same time.

given an installed jt
when I run **jt --dirname jrnl init default "My test journal"**
then command is successful
and there are no drafts in **jrnl**
and there are no journal entries in **jrnl**
when I run **jt --editor=none --dirname=jrnl new "Abracadabra"**

then command is successful
and there is one draft in **jrnl**
and draft **0** in **jrnl** contains "**Abracadabra**"
when I run **jt --editor=none --dirname=jrnl new "Simsalabim"**
then command is successful
and there are two drafts in **jrnl**
and draft **0** in **jrnl** contains "**Abracadabra**"
and draft **1** in **jrnl** contains "**Simsalabim**"
given an executable script **append.sh**
when I run **jt --editor=./append.sh --dirname=jrnl edit 0**
then draft **0** in **jrnl** contains "**Open sesame!**"
when I run **jt --editor=./append.sh --dirname=jrnl edit 1**
then draft **1** in **jrnl** contains "**Open sesame!**"
when I run **jt --dirname=jrnl finish 0 abra**
then command is successful
and there is one journal entry in **jrnl**, at **FILE**
and journal entry <**FILE**> contains "**Abracadabra**"
and journal entry <**FILE**> contains "**Open sesame!**"
and there is one draft in **jrnl**
when I run **jt --dirname=jrnl finish 1 sim**
then command is successful
and there are two journal entries in **jrnl**, at **FILE1** and **FILE2**
and journal entry <**FILE1**> contains "**Abracadabra**"
and journal entry <**FILE2**> contains "**Simsalabim**"
and there are no drafts in **jrnl**
and there are no uncommitted changes in **jrnl**

2.5 Remove a draft

Verify that we can remove a draft, and then create a new one.

given an installed jt
when I run **jt --dirname jrnl init default "My test journal"**
then command is successful
and there are no drafts in **jrnl**
and there are no journal entries in **jrnl**
when I run **jt --editor=none --dirname=jrnl new "Hulabaloo"**
then command is successful
and there is one draft in **jrnl**
and draft **0** in **jrnl** contains "**Hulabaloo**"
and draft **0** in **jrnl** contains "**!meta date=**"
when I run **jt --dirname=jrnl remove 0**
then command is successful
and there are no drafts in **jrnl**
and there are no journal entries in **jrnl**
when I run **jt --editor=none --dirname=jrnl new "Abracadabra"**

then command is successful
and there is one draft in **jrnl**
and draft **0** in **jrnl** contains "**Abracadabra**"
and draft **0** in **jrnl** contains "**!meta date=**"

2.6 Override template for new journal entries

Verify that we can have a custom template for new journal entries.

given an installed jt
when I run **jt --dirname jrnl init default "My test journal"**
then command is successful
given file **jrnl/.config/templates/new_entry** from **new_entry_template**
when I run **jt --editor=none --dirname=jrnl new "Abracadabra"**
then command is successful
and there is one draft in **jrnl**
and draft **0** in **jrnl** contains "**custom new entry template**"

File: **new_entry_template**

```
This is a custom new entry template.
```

2.7 Use topic pages

Verify that we can create a new topic page and a new entry referring to that topic page.

given an installed jt
when I run **jt --dirname jrnl init default "My test journal"**
then command is successful
when I try to run **jt --editor=none --dirname=jrnl new --topic foo.bar "Abracadabra"**
then command fails
and stderr contains "**foo.bar**"
when I run **jt --editor=none --dirname=jrnl new-topic topics/foo.bar "Things about Foobars"**
then command is successful
and file **jrnl/topics/foo.bar.mdwn** contains "**Things about Foobars**"
and there are no uncommitted changes in **jrnl**
when I run **jt --editor=none --dirname=jrnl new --topic topics/foo.bar "Abracadabra"**
then command is successful
and there is one draft in **jrnl**
and draft **0** in **jrnl** links to "**topics/foo.bar**"

3 Colophon

This document is meant to be processed with the Subplot[2] program to typeset into HTML or PDF or to generate a program that automatically verifies that all acceptance criteria are met.

[2] `https://subplot.liw.fi/`