What is Nabu?

A Data Publishing System Using ReST

Martin Blais

Furius Enterprise

PyCon 2006

(FACIL, Montreal, Canada, 20th April 2006)

Introduction

Nabu is *not*…

- …a Wiki
- …blogging software
- …a personal information manager
- …a document publishing tool
- …a generic data entry system
- …a desktop search system

It's a little bit of all these things.

So I'm going to take a long winding road to introduce this project via a set of examples using personal information management.

This will be *an ode to the power and elegance of simple text files*…

1993 - Bookmarks

- Using Xmosaic
- Creating lots of bookmarks lots of sites
  (we did not have Google)

A script was born, with typical input like this in a single **text file**:

```
Raymond Hettinger's photography
http://www.knowyourboston.com
photography, boston, sexy girls
```

- Then came Netscape, then came Mozilla, then came IE
  ...with corresponding converters.
- Eventually came Firefox and we were happy ever after...

...or maybe not?

1997 - Address Book

- I was using “paper” technology to store my contact info - little booklets…
- Then I used Netscape to store my contacts in LDIF
- One day, an old `nroff` user showed me this fabulous “ascii” technology to store his contact info:

  ```
  n: Librairie Michel Fortin inc.
  a: 3714 St-Denis
  p: +1.514.849.5719
  ```

  (He was a `vi` user)

Using this and “paragraph-grep” from a shell, I lived happily ever after…

…or have I?

2000 - Blog

Those were the days before blogs were called blogs…

- Pat Jennings/Synaptic - cycling through China
- Phil Greenspun - `photo.net`
- I'm inspired! So I write …

…*another script*

- It takes its input in fashionable XML (it was truly awful)
- So later I converted it to take input from …*simple text files*
- Eventually I discovered ReStructuredText and converted my system to use it

And I was happy with static HTML files forever… …*not!*

2000 - Blog

Those were the days before blogs were called blogs…

- Pat Jennings/Synaptic - cycling through China
- Phil Greenspun - `photo.net`
- I'm inspired!   So I write …

… *another script*

- It takes its input in fashionable XML (it was truly awful)
- So later I converted it to take input from … *simple text files*
- Eventually I discovered ReStructuredText and converted my system to use it

And I was happy with static HTML files forever…   … *not!*

2002 - The Art of Taking Notes

I’m getting a little bit old now, I’m losing bits of memory…

But the older become smarter and now I just **know** in advance that I will forget. When I start a new task, I **invariably** start a new text file to take notes on it.

This is great, because:

- I can grep the files
- I can more easily interrupt my work
  (there is a memory of the task)
- I can put URLs in context, in these files, rather than in a global bookmarks file

2002 - The Art of Taking Notes

Wikis Suck (For This Purpose)

I would like to share many of these short technical documents with other people . . . naturally, the idea of using Wikis come to mind.

But wikis *suck* for jotting down notes. . .

- Anything but the most trivial topic title looks horrible:

  ```
  BrazilTravelNotes
  ```

- The editor capabilities of browsers are inadequate
  - Who has never lost a file being edited in a TEXTAREA?
  - I'm a programmer, I want powerful editing! I live in Emacs
  - I want to be able to save my files without having to submit

2002 - The Art of Taking Notes

Wikis Suck (For This Purpose)

I would like to share many of these short technical documents with other people . . . naturally, the idea of using Wikis come to mind.

But wikis *suck* for jotting down notes. . .

- Anything but the most trivial topic title looks horrible:

  ```
  BrazilTravelNotes
  ```

- The editor capabilities of browsers are inadequate
  - Who has never lost a file being edited in a TEXTAREA?
  - I'm a programmer, I want powerful editing! I live in Emacs
  - I want to be able to save my files without having to submit

Mixed Data Example: Travel Files

Lots of scattered notes files...

One example of these notes files are my travel files, they contains many different types of things:

- They contain a list of things to do for a trip, personal notes, itineraries (**documents**)
- They contain addresses of people and places to visit (**contact infos**)
- They contain URLs of related websites (**bookmarks**)
- They contain references to books and articles (**publications**)

Mixed Data Example: Travel Files

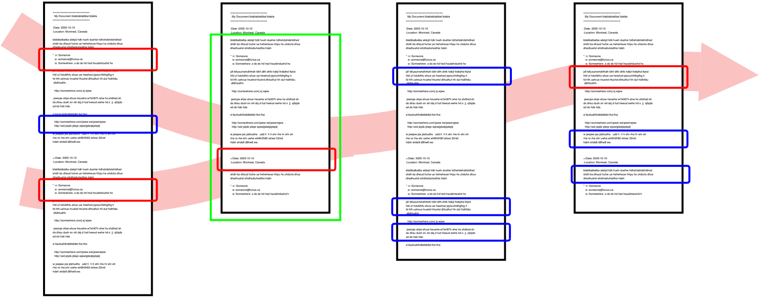
```
====================
   Trip to Brazil
====================

:Id: brazil-trip-notes
:Category: Travel
:Disclosure: public

Itinerary Proposals
===================

  Jan 25
    * Fly to Salvador da Bahia, Brazil

  Jan 26
    * Drink Caipirinhas
```

Mixed Data Example: Travel Files

```
Visa
====

* :n: Consulat général du Brésil À Montréal
  :a: 2000, rue Mansfield, bureau 1700, Montréal (QC) H3A 3A5
  :f: (514) 499-3963
  :e: vistos@consbrasmontreal.org
  :w: http://www.consbrasmontreal.org/

Vaccinations
============
http://www.mdtravelhealth.com/destinations/samerica/brazil.htm

  Routine immunizations
    All travelers should be up-to-date on tetanus-diphtheria,
    measles-mumps-rubella, polio, and varicella immunizations
```

Data Across Documents

My data is scattered *across* the set of all my documents



Data Across Documents

What if I could *identify* and *extract* the meaningful parts from those files and store them appropriately? What could I build with this?

Related idea: the Semantic Web

- In an ideal world, web page authors would identify all relevant parts of their documents with appropriate markup
- You would then be able to collect and use this data, e.g. create an “address book” of the internet
- Search engines are taking a stab at this holy grail

But I want this **now**, and for just my personal corpus of files, even if it’s a restricted version of this idea. I want a database built from the files on my computer to feed a website, like a blog on steroids.

Data Across Documents

What if I could *identify* and *extract* the meaningful parts from those files and store them appropriately? What could I build with this?

Related idea: the Semantic Web

- In an ideal world, web page authors would identify all relevant parts of their documents with appropriate markup
- You would then be able to collect and use this data, e.g. create an “address book” of the internet
- Search engines are taking a stab at this holy grail

But I want this **now**, and for just my personal corpus of files, even if it’s a restricted version of this idea. I want a database built from the files on my computer to feed a website, like a blog on steroids.

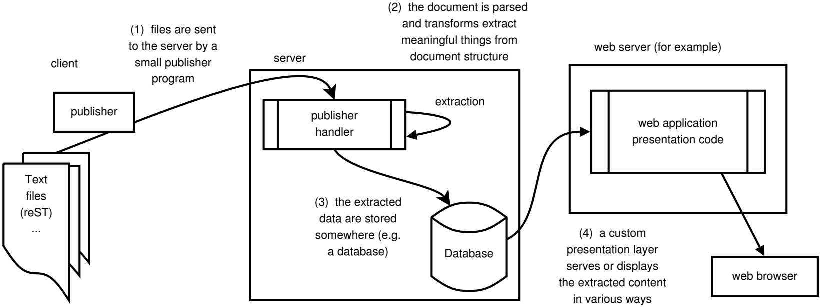
The Goal

Build a system that can extract semantically meaningful informations in my set of personal info files, using weak heuristics and conventions, and store this information in a structured way (in database tables), so I can use this information later and serve it in new, interesting ways.

Nabu is a Python library that allows you to do that.

- It is not tied specifically to PIM info
- You can write extractors for anything, you just have to establish conventions for the docutils structures that you are going to recognize/extract
- On the client it requires only Python to publish text files (minimize dependencies to allow easy deployment)

Components / Overview



Components

- **Nabu Publisher Client**: searches the files on the client side and sends the modified ones to the server
  - It fetches MD5 sums from the server and compares the local files
  - Files are identified by an embedded Id, so file locations don't matter

```
      :Id: 8844db51-36ee-4e2a-8255-84e804f5cbe2
```

- **Nabu Server**: receives the files, parses them through docutils and runs the configured extractors, thereby storing the data
  - All extracted data is tagged with the unique id for the file
  - When a file is uploaded, old data from that file is removed and new data replaces it
- **Storage**: typically, your database server (or files or anything else if you like)
- **Presentation**: your own favourite thing (Nabu does not provide presentation)

Components

- **Nabu Publisher Client**: searches the files on the client side and sends the modified ones to the server
  - It fetches MD5 sums from the server and compares the local files
  - Files are identified by an embedded Id, so file locations don't matter

    ```
    :Id: 8844db51-36ee-4e2a-8255-84e804f5cbe2
    ```
- **Nabu Server**: receives the files, parses them through docutils and runs the configured extractors, thereby storing the data
  - All extracted data is tagged with the unique id for the file
  - When a file is uploaded, old data from that file is removed and new data replaces it
- **Storage**: typically, your database server (or files or anything else if you like)
- **Presentation**: your own favourite thing (Nabu does not provide presentation)

Components

- **Nabu Publisher Client**: searches the files on the client side and sends the modified ones to the server
  - It fetches MD5 sums from the server and compares the local files
  - Files are identified by an embedded Id, so file locations don't matter

```
:Id: 8844db51-36ee-4e2a-8255-84e804f5cbe2
```

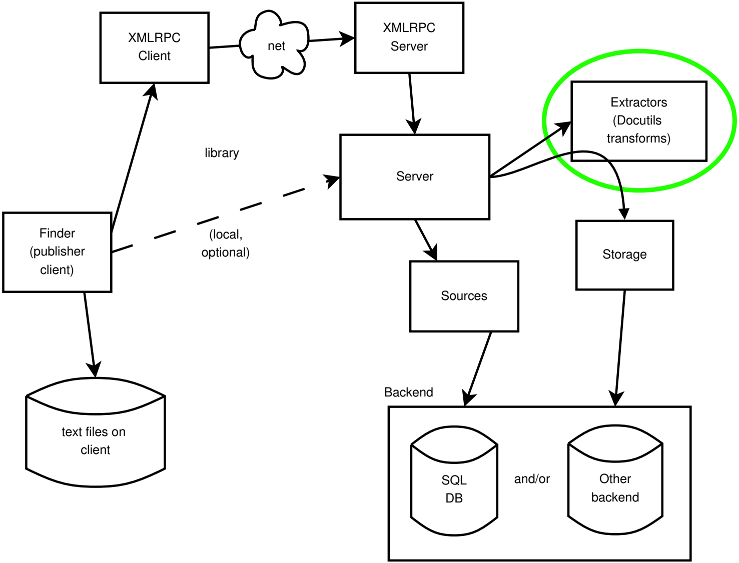
- **Nabu Server**: receives the files, parses them through docutils and runs the configured extractors, thereby storing the data
  - All extracted data is tagged with the unique id for the file
  - When a file is uploaded, old data from that file is removed and new data replaces it
- **Storage**: typically, your database server
  (or files or anything else if you like)
- **Presentation**: your own favourite thing
  (Nabu does not provide presentation)

Components

- **Nabu Publisher Client**: searches the files on the client side and sends the modified ones to the server
  - It fetches MD5 sums from the server and compares the local files
  - Files are identified by an embedded Id, so file locations don't matter

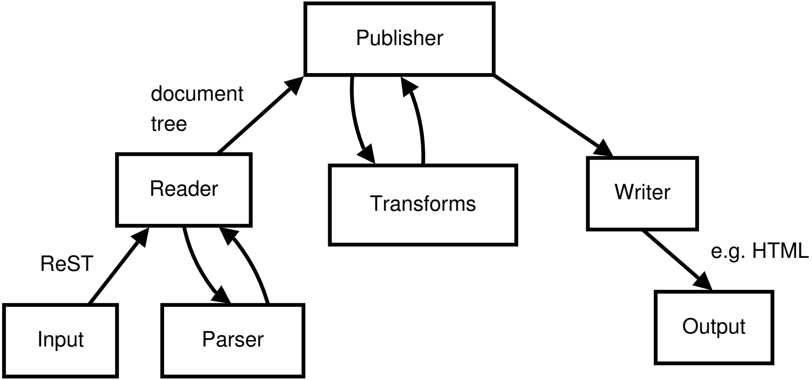
    ```
    :Id: 8844db51-36ee-4e2a-8255-84e804f5cbe2
    ```
- **Nabu Server**: receives the files, parses them through docutils and runs the configured extractors, thereby storing the data
  - All extracted data is tagged with the unique id for the file
  - When a file is uploaded, old data from that file is removed and new data replaces it
- **Storage**: typically, your database server
  (or files or anything else if you like)
- **Presentation**: your own favourite thing
  (Nabu does not provide presentation)

Design


XMLRPC Client
net
XMLRPC Server
Extractors (Docutils transforms)
library
Server
Finder (publisher client)
(local, optional)
Storage
Sources
text files on client
Backend
SQL DB
and/or
Other backend

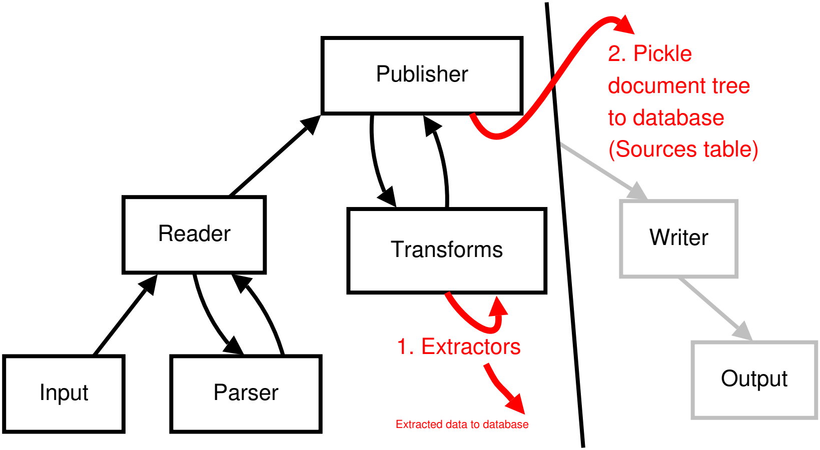
Extractors: Integration with docutils

Here is the complete docutils pipeline:



Extractors: Integration with docutils

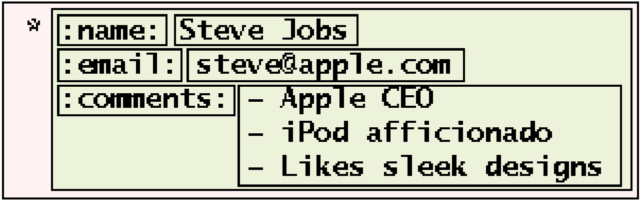
Here is the modified, partial docutils pipeline (no output, just process in order to run the extractors):


Publisher
Reader
Transforms
Input
Parser
Writer
Output
2. Pickle document tree to database (Sources table)
1. Extractors
Extracted data to database

Extractors: docutils document tree

`docutils` is a “2D parser”

- Its structures are recursive boxes of stuff
- For your extractors you have to establish conventions for recognizing the stuff you want to extract from your text files



```
:name:     Steve Jobs
:email:    steve@apple.com
:comments: - Apple CEO
           - iPod afficionado
           - Likes sleek designs
```

e.g. has name and (has email or has address)

Extractors: Viewing the docutils parse tree

You need to write extractors for the stuff that you are interested in, for example:

```
* :name: Bill Gates
  :email: billg@microsoft.com
```

Use `rst2pseudoxml.py` to figure how docutils parses it:

```
<bullet_list bullet="*">
    <list_item>
        <field_list>
            <field>
                <field_name>
                    name
                <field_body>
                    <paragraph>
                        Bill Gates
            <field>
                <field_name>
                    email
                <field_body>
                    <paragraph>
                        <reference refuri="mailto:billg@microsoft.com"
                            billg@microsoft.com
```

Extractors: Implementation

Then implement it:

```
class AddressExtractor(extract.Extractor):

    def apply( self, **kwargs ):
        v = AddressVisitor(self, self.document)
        self.document.walkabout(v)

class AddressVisitor(...):
    ...

class AddressStorage(extract.SQLExtractorStorage):

    def store( self, unid, name, tfields ):
        ... # store the stuff in a database
```

Target Audience

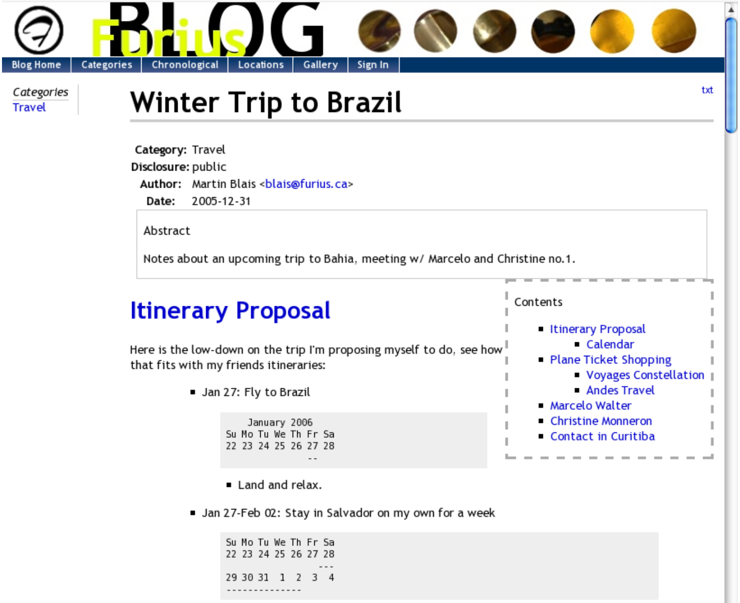
It’s not for your mom!

Nabu is intended only for use by people who have developed the *ability to edit text files carefully* (typically programmers, i.e. you guys).

- We understand indentation
- We know about spacing, justification, filling, etc.
- We are careful about pesky little details
- This is what makes creating ReST files possible

Leverage this ability!

Presentation: Document Example



Categories
Travel

Winter Trip to Brazil

Category: Travel
Disclosure: public
Author: Martin Blais <blais@furius.ca>
Date: 2005-12-31

Abstract

Notes about an upcoming trip to Bahia, meeting w/ Marcelo and Christine no.1.

Contents

Itinerary Proposal

Here is the low-down on the trip I'm proposing myself to do, see how that fits with my friends itineraries:

- Jan 27: Fly to Brazil

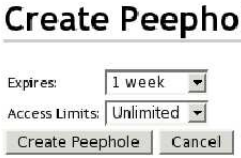
```
    January 2006
Su Mo Tu We Th Fr Sa
22 23 24 25 26 27 28
               --
```

  - Land and relax.

- Jan 27-Feb 02: Stay in Salvador on my own for a week

```
Su Mo Tu We Th Fr Sa
22 23 24 25 26 27 28
                 ---
29 30 31  1  2  3  4
--------------
```

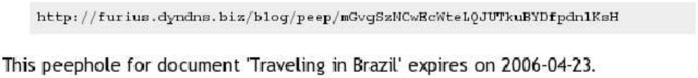
Presentation: Peepholes



Create Peephole for travel-brazil

Expires: 1 week

Access Limits: Unlimited



`http://furius.dyndns.biz/blog/peep/mGvgSzNCwEcWteLQJUTkuBYDfpdnlKsH`

This peephole for document 'Traveling in Brazil' expires on 2006-04-23.

Presentation: Peepholes

Peepholes are passwordless privileged acccess to selected documents in my Nabu store.

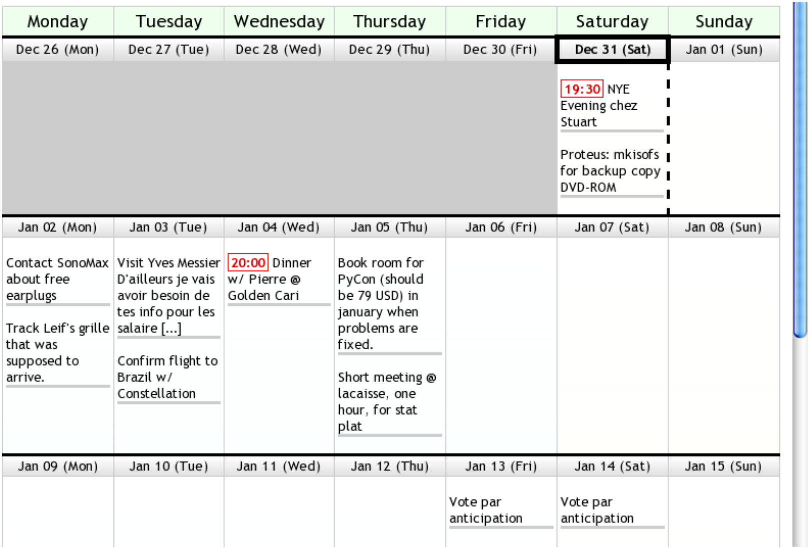
- Useful for sending a link to a file that is work-in-progress
- Does not require users/passwords, relies on the “relative” privacy of email
- Provides access to only some selected resources
- Expires after some time, or after a number of accesses (self-destroying links. . . [insert M.I. music here])

Presentation: Events/Calendar Example

```
sat 2005-12-31 19h30
  - NYE Evening chez Stuart

sat 2005-12-31
  - Proteus: mkisofs for backup copy DVD-ROM

2006-01-02
  - Contact SonoMax about free earplugs
  - Track Leif’s grille that was supposed to arrive.

2006-01-03
  - Visit Yves * D’ailleurs je vais avoir besoin de tes
    salaires de 2005

  - Confirm flight to Brazil w/ Constellation

2006-01-04 20h00
  - Dinner w/ Pierre @ Golden Cari
```

Presentation: Events/Calendar Example

Calendar View



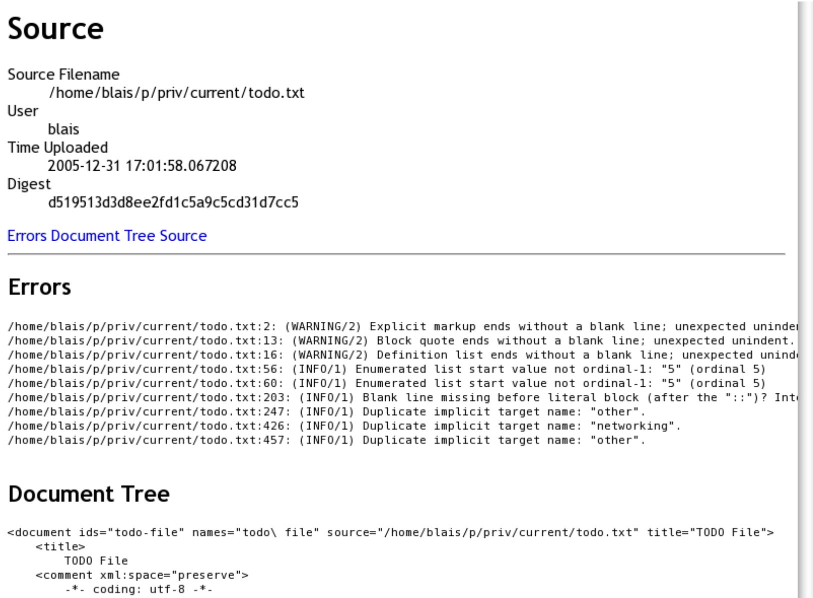
Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Dec 26 (Mon)	Dec 27 (Tue)	Dec 28 (Wed)	Dec 29 (Thu)	Dec 30 (Fri)	**Dec 31 (Sat)**	Jan 01 (Sun)
					19:30 NYE Evening chez Stuart Proteus: mkisofs for backup copy DVD-ROM	
Jan 02 (Mon)	Jan 03 (Tue)	Jan 04 (Wed)	Jan 05 (Thu)	Jan 06 (Fri)	Jan 07 (Sat)	Jan 08 (Sun)
Contact SonoMax about free earplugs Track Leif's grille that was supposed to arrive.	Visit Yves Messier D'ailleurs je vais avoir besoin de tes info pour les salaire [...] Confirm flight to Brazil w/ Constellation	20:00 Dinner w/ Pierre @ Golden Cari	Book room for PyCon (should be 79 USD) in january when problems are fixed. Short meeting @ lacaisse, one hour, for stat plat			
Jan 09 (Mon)	Jan 10 (Tue)	Jan 11 (Wed)	Jan 12 (Thu)	Jan 13 (Fri)	Jan 14 (Sat)	Jan 15 (Sun)
				Vote par anticipation	Vote par anticipation	

Presentation: Other Examples

- Bookmarks: You can serve your extracted bookmarks from a server using RSS format
- Billing info: you could define a simple timesheet format in a text file and have some kind per-task or per-client billing info page that is always up-to-date
- Filter all extracted links to Google Maps and generate a map of all of the locations with links to the original documents
- . . . (insert your own here) . . .
- Dynamic website data: you could use Nabu to feed some data for a web site, this allows you to **avoid having to write input forms**

Debugging: Nabu Contents Browser

View Uploaded File Details



Source

Source Filename
: /home/blais/p/priv/current/todo.txt

User
: blais

Time Uploaded
: 2005-12-31 17:01:58.067208

Digest
: d519513d3d8ee2fd1c5a9c5cd31d7cc5

Errors Document Tree Source

Errors

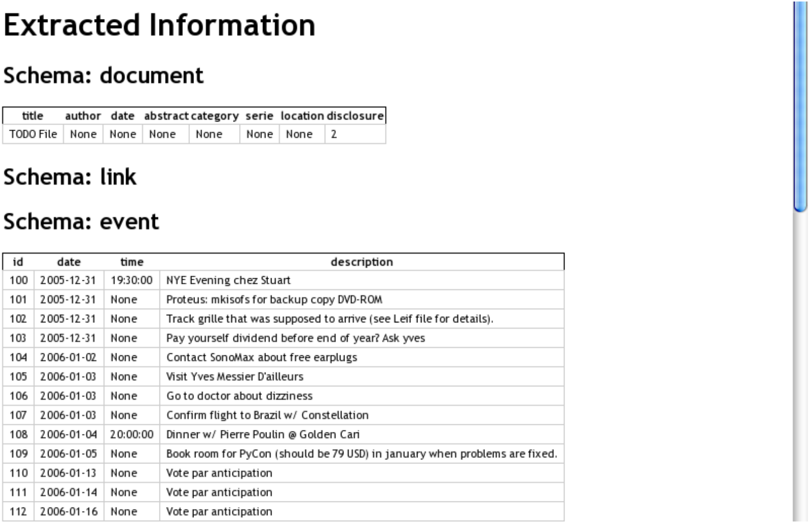
```
/home/blais/p/priv/current/todo.txt:2: (WARNING/2) Explicit markup ends without a blank line; unexpected uninde
/home/blais/p/priv/current/todo.txt:13: (WARNING/2) Block quote ends without a blank line; unexpected unindent.
/home/blais/p/priv/current/todo.txt:16: (WARNING/2) Definition list ends without a blank line; unexpected unind
/home/blais/p/priv/current/todo.txt:56: (INFO/1) Enumerated list start value not ordinal-1: "5" (ordinal 5)
/home/blais/p/priv/current/todo.txt:60: (INFO/1) Enumerated list start value not ordinal-1: "5" (ordinal 5)
/home/blais/p/priv/current/todo.txt:203: (INFO/1) Blank line missing before literal block (after the "::")? Int
/home/blais/p/priv/current/todo.txt:247: (INFO/1) Duplicate implicit target name: "other".
/home/blais/p/priv/current/todo.txt:426: (INFO/1) Duplicate implicit target name: "networking".
/home/blais/p/priv/current/todo.txt:457: (INFO/1) Duplicate implicit target name: "other".
```

Document Tree

```
<document ids="todo-file" names="todo\ file" source="/home/blais/p/priv/current/todo.txt" title="TODO File">
    <title>
        TODO File
    <comment xml:space="preserve">
        -*- coding: utf-8 -*-
```

Debugging: Nabu Contents Browser

View Extracted Info



Extracted Information

Schema: document

title	author	date	abstract	category	serie	location	disclosure
TODO File	None	None	None	None	None	None	2

Schema: link

Schema: event

id	date	time	description
100	2005-12-31	19:30:00	NYE Evening chez Stuart
101	2005-12-31	None	Proteus: mkisofs for backup copy DVD-ROM
102	2005-12-31	None	Track grille that was supposed to arrive (see Leif file for details).
103	2005-12-31	None	Pay yourself dividend before end of year? Ask yves
104	2006-01-02	None	Contact SonoMax about free earplugs
105	2006-01-03	None	Visit Yves Messier D'ailleurs
106	2006-01-03	None	Go to doctor about dizziness
107	2006-01-03	None	Confirm flight to Brazil w/ Constellation
108	2006-01-04	20:00:00	Dinner w/ Pierre Poulin @ Golden Cari
109	2006-01-05	None	Book room for PyCon (should be 79 USD) in january when problems are fixed.
110	2006-01-13	None	Vote par anticipation
111	2006-01-14	None	Vote par anticipation
112	2006-01-16	None	Vote par anticipation

Multiple Users

There are two approches:

1. Multiple users share a single body of files
   - You could update to Nabu on a **Subversion hook** when files are committed
2. Each user has a distinct body of files
   - The Nabu server supports disjoint sets of ids per-user, so that users don't have to manually manage avoiding collisions

Problems

- Creating precise reStructuredText can be fragile from the user's point-of-view
  - You end up having to wrap your head around the docutils document structure and knowing ReST really well in order to generate what you need (but that is ok)
- You need to make sure that your set of ids are unique
  - I like to use UUIDs like this:

    `78d8a600-abb4-49c0-a0fc-1c315cddbc1a`

Future Work

- Stabilize more
    - I need to write more presentations for my data
    - Convert presentations output to microformats
- Support encryption in the publisher client, hidden files
- Support per-document options

Questions

Nabu homepage:
`http://furius.ca/nabu/`

Slides will be posted there.
I will be around during the sprints.

Questions?