Packages sent … and received?

Documenting digital authenticity and creating preservation metadata with BagIt

Assumptions

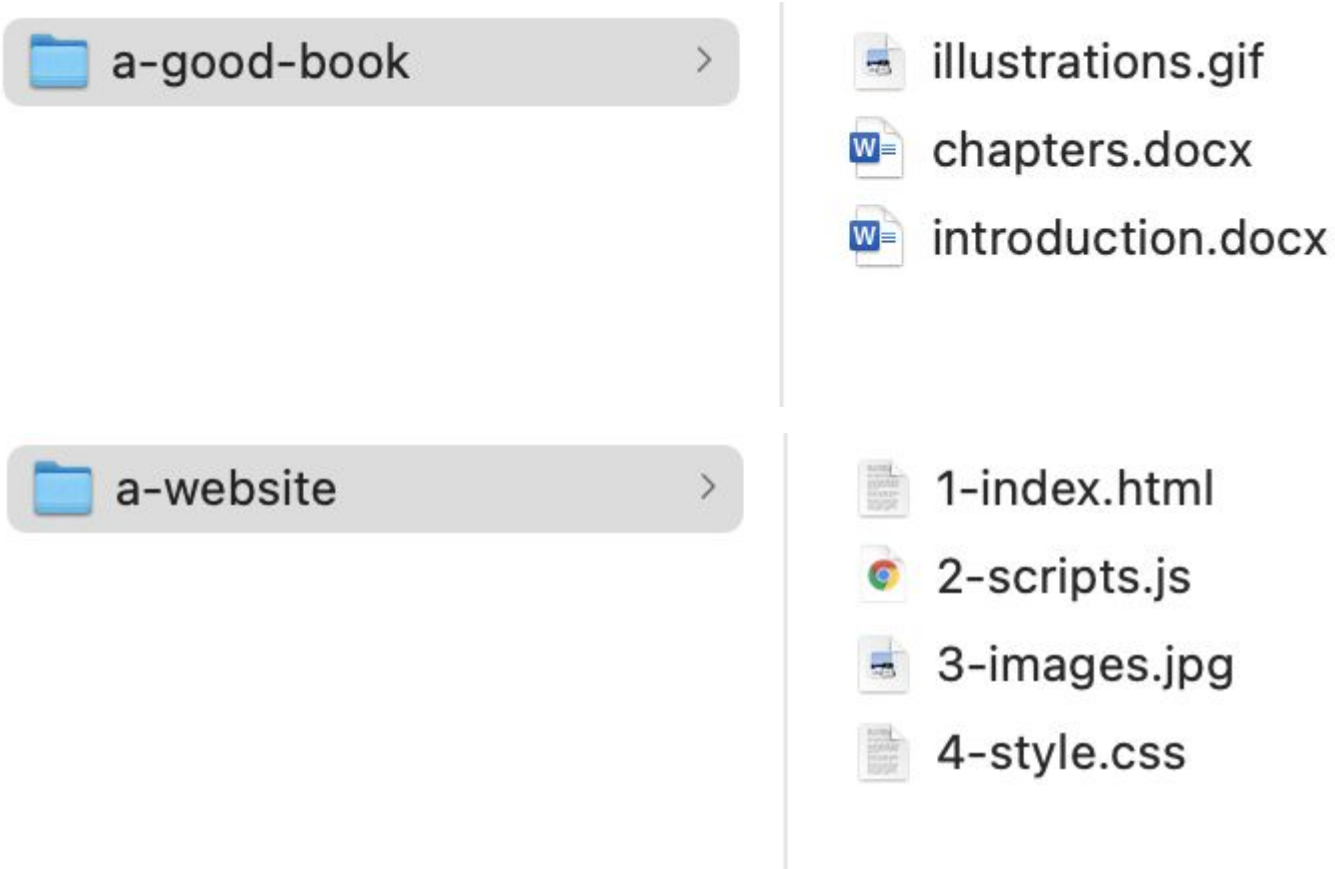
I’m making a few assumptions about what sorts of background you might already have, including that you have some level of comfort with the following:

- working with files on a computer (Windows, Mac, Linux, or other filesystem environments),
- that you can find and open them from an Explorer (Windows) or Finder (Mac) graphical interface,
- that you have done a some basic navigation in a shell environment (e.g., Terminal or Bash shell or GitBash) including finding files and directories,
- you have a familiarity with the Python programming language, including using it from an interactive notebook environment (Jupyter notebooks), and
- downloading and operating Jupyter notebooks and repos from GitHub.

Providing Authentic, Verifiable Digital Content

- “Did you get what I sent you?”
- “Do you have the file ...
  that I sent you last year?
  - “Are you sure that it’s the same?”

It’s more complicated with “complex” digital objects . . .



From the SAA glossary

fixity

n.

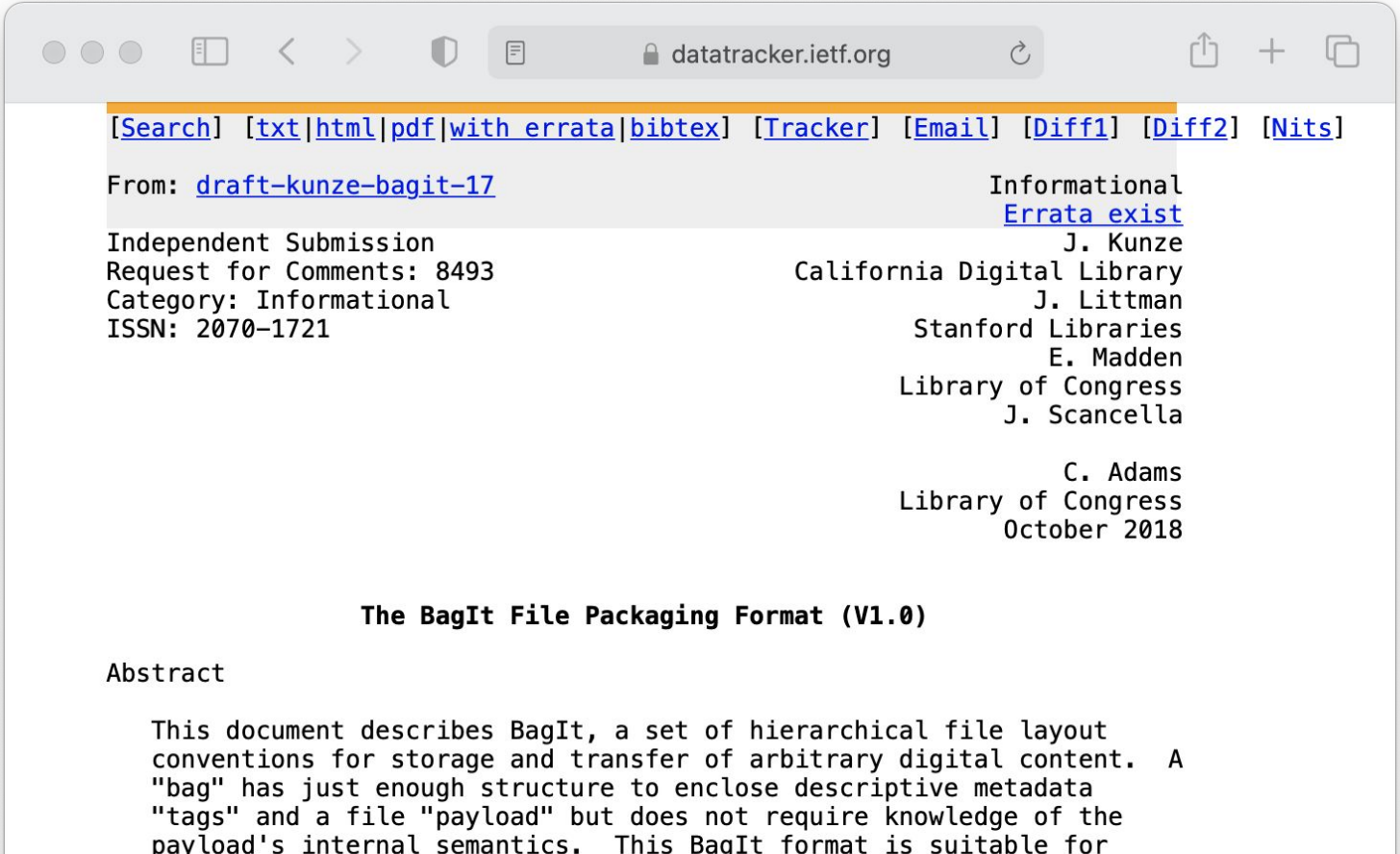
the property of being unchanged

checksum

n.

a unique alphanumeric value that represents the bitstream of an individual computer file or set of files

One Answer: BagIt (A specification, and a tool)



[Search] [txt|html|pdf|with errata|bibtex] [Tracker] [Email] [Diff1] [Diff2] [Nits]

From: draft-kunze-bagit-17 Informational
Errata exist

Independent Submission J. Kunze
Request for Comments: 8493 California Digital Library
Category: Informational J. Littman
ISSN: 2070-1721 Stanford Libraries
E. Madden
Library of Congress
J. Scancella

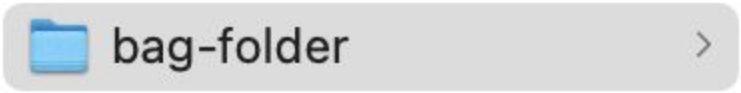
C. Adams
Library of Congress
October 2018

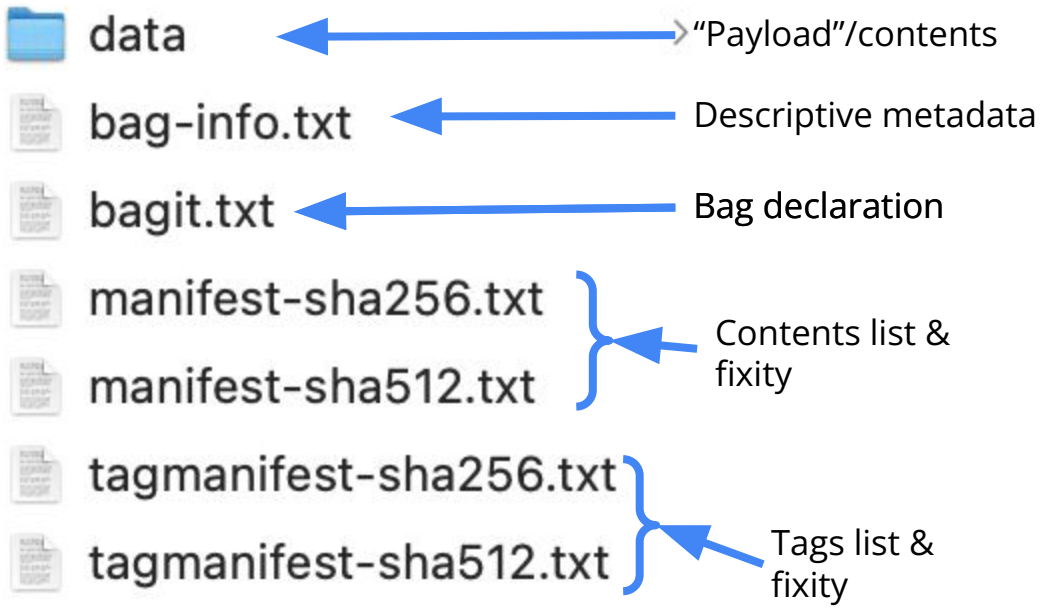
The BagIt File Packaging Format (V1.0)

Abstract

This document describes BagIt, a set of hierarchical file layout conventions for storage and transfer of arbitrary digital content. A "bag" has just enough structure to enclose descriptive metadata "tags" and a file "payload" but does not require knowledge of the payload's internal semantics. This BagIt format is suitable for

What does it look like? How is it structured?





Use “Bag Info” for descriptive information

Data is stored as label-name: value

(this is like a Python dictionary… ‘key’: ‘value’)

Sample BagInfo:

```
Contact-Name: Jesse
External-Description: These are some cool files!
Bagging-Date: 2022-01-08
Payload-Oxum: 26923687.23
```

Manifests & sample bag

```
myfirstbag/
    |
    |   manifest-md5.txt
    |    (49afbd86a1ca9f34b677a3f09655eae9 data/27613-h/images/q172.png)
    |    (408ad21d50cef31da4df6d9ed81b01a7 data/27613-h/images/q172.txt)
    |
    |   bagit.txt
    |    (BagIt-version: 1.0                                          )
    |    (Tag-File-Character-Encoding: UTF-8                          )
    |
   \--- data/
         |
         |   27613-h/images/q172.png
         |    (... image bytes ...                                    )
         |
         |   27613-h/images/q172.txt
         |    (... OCR text ...                                       )
```

Demo

Open up notebook `01b-using-bagit`...

Use Cases

- Transfer and verify large amounts of heterogeneous web archive data (this was the problem in 2005, when California Digital Library and Library of Congress defined BagIt)
- Store large groups of digitized content, such as digitized microfilms (this is what the National Digital Newspaper Program does to preserve millions of pages of digitized newspapers)
- “Bag” local records, transmit them to a state or federal archive, and the archive can confirm they got the authentic files (this is used by the North Carolina State Archives)