Software Heritage

The Great Library of (Python) Source Code

Nicolas Dandrimont, Stefano Zacchiroli

Software Heritage — {olasd,zack}@softwareheritage.org

6 Oct 2018
PyConFr
Lille, France

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Outline

1. The Software Commons
2. Software Heritage
3. The Great Library of Python source code
4. Getting involved

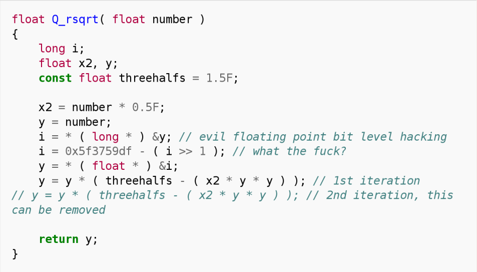
(Free) Software is everywhere

Software source code is *special*

Harold Abelson, Structure and Interpretation of Computer Programs

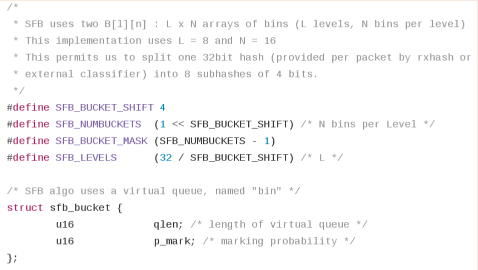
“Programs must be written for people to read, and only incidentally for machines to execute.”

Quake III source code (excerpt)



```c
float Q_rsqrt( float number )
{
    long i;
    float x2, y;
    const float threehalfs = 1.5F;

    x2 = number * 0.5F;
    y = number;
    i = * ( long * ) &y; // evil floating point bit level hacking
    i = 0x5f3759df - ( i >> 1 ); // what the fuck?
    y = * ( float * ) &i;
    y = y * ( threehalfs - ( x2 * y * y ) ); // 1st iteration
// y = y * ( threehalfs - ( x2 * y * y ) ); // 2nd iteration, this
can be removed

    return y;
}
```

Net. queue in Linux (excerpt)



```c
/*
 * SFB uses two B[l][n] : L x N arrays of bins (L levels, N bins per level)
 * This implementation uses L = 8 and N = 16
 * This permits us to split one 32bit hash (provided per packet by rxhash or
 * external classifier) into 8 subhashes of 4 bits.
 */
#define SFB_BUCKET_SHIFT 4
#define SFB_NUMBUCKETS  (1 << SFB_BUCKET_SHIFT) /* N bins per Level */
#define SFB_BUCKET_MASK (SFB_NUMBUCKETS - 1)
#define SFB_LEVELS      (32 / SFB_BUCKET_SHIFT) /* L */

/* SFB algo uses a virtual queue, named "bin" */
struct sfb_bucket {
        u16             qlen; /* length of virtual queue */
        u16             p_mark; /* marking probability */
};
```

Len Shustek, Computer History Museum

“Source code provides a view into the mind of the designer.”

Our Software Commons

Definition (Commons)

The commons is the cultural and natural resources accessible to all members of a society, including natural materials such as air, water, and a habitable earth. These resources are held in common, not owned privately. https://en.wikipedia.org/wiki/Commons

Definition (Software Commons)

The software commons consists of all computer software which is available at little or no cost and which can be altered and reused with few restrictions. Thus *all open source software and all free software are part of the [software] commons.* [...]

https://en.wikipedia.org/wiki/Software_Commons

Our Software Commons

Definition (Commons)

The commons is the cultural and natural resources accessible to all members of a society, including natural materials such as air, water, and a habitable earth. These resources are held in common, not owned privately. https://en.wikipedia.org/wiki/Commons

Definition (Software Commons)

The software commons consists of all computer software which is available at little or no cost and which can be altered and reused with few restrictions. Thus *all open source software and all free software are part of the [software] commons.* [. . .]

https://en.wikipedia.org/wiki/Software_Commons

Source code is *a precious part* of our commons

are we taking care of it?

Software is spread all around

Debian
Sourceforge
CPAN
Gitorious
Maven
Inria
Bitbucket
GitHub
BerliOs
CTAN
Gitlab
CRAN
GoogleCode
Adullact

Fashion victims

- many disparate development platforms
- a myriad places where distribution may happen
- projects tend to migrate from one place to another over time

Software is spread all around

Debian
Sourceforge
CPAN
Gitorious
Maven
Inria
Bitbucket
GitHub
BerliOs
CTAN
Gitlab
CRAN
GoogleCode
Adullact

Fashion victims

- many disparate development platforms
- a myriad places where distribution may happen
- projects tend to migrate from one place to another over time

Where is the place . . .

where we can find, track and search *all* source code?

Software is fragile

Like all digital information, FOSS is fragile

- inconsiderate and/or malicious code loss (e.g., Code Spaces)
- business-driven code loss (e.g., Gitorious, Google Code)
- for obsolete code: physical media decay (data rot)

Software is fragile

Like all digital information, FOSS is fragile

- inconsiderate and/or malicious code loss (e.g., Code Spaces)
- business-driven code loss (e.g., Gitorious, Google Code)
- for obsolete code: physical media decay (data rot)

Where is the archive…

where we go if (a repository on) GitHub or GitLab.com goes away?

Software lacks its own research infrastructure

A wealth of software research on crucial issues…

- safety, security, test, verification, proof
- software engineering, software evolution
- big data, machine learning, empirical studies

Software lacks its own research infrastructure

A wealth of software research on crucial issues…

- safety, security, test, verification, proof
- software engineering, software evolution
- big data, machine learning, empirical studies

If you study the stars, you go to Atacama…

… where is the *very large telescope* of source code?

Outline

1. The Software Commons
2. Software Heritage
3. The Great Library of Python source code
4. Getting involved

The Software Heritage Project

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

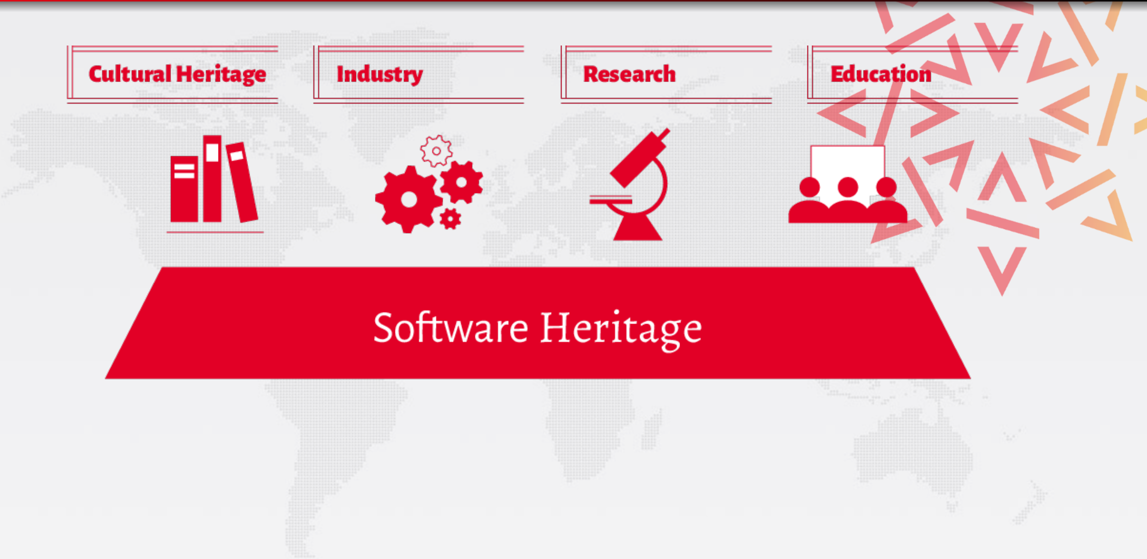
Our mission

Collect, preserve and share the *source code* of *all the software* that is publicly available.

Past, present and future

Preserving the past, *enhancing* the present, *preparing* the future.

Core principles



Cultural Heritage

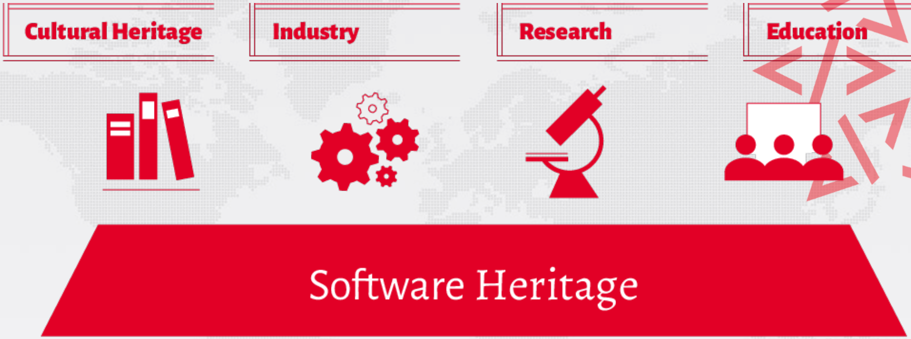
Industry

Research

Education

Software Heritage

Core principles



Cultural Heritage

Industry

Research

Education

Software Heritage

Open approach

- 100% Free Software
- transparency

In for the long haul

- replication
- non profit

Archiving goals

Targets: VCS repositories & source code releases (e.g., tarballs)

We DO archive

- file content (= blobs)
- revisions (= commits), with full metadata
- releases (= tags), ditto
- where (origin) & when (visit) we found any of the above

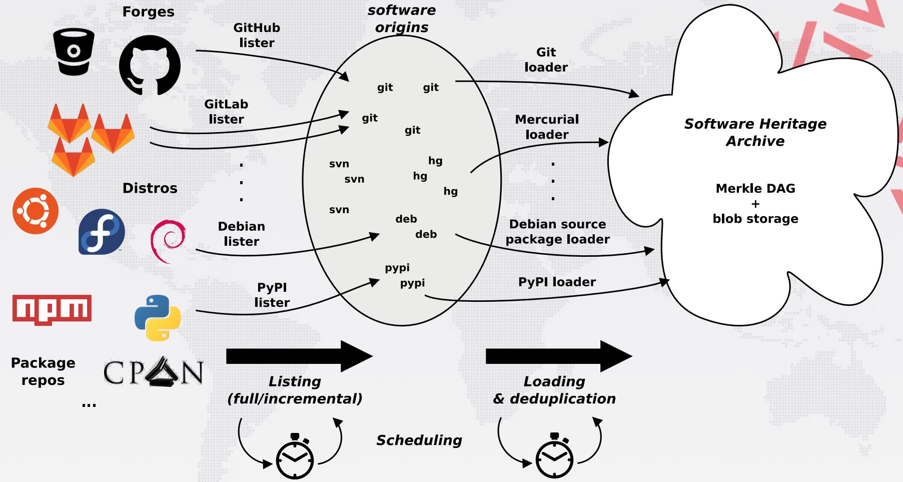
… in a VCS-/archive-agnostic canonical data model

We DON'T archive

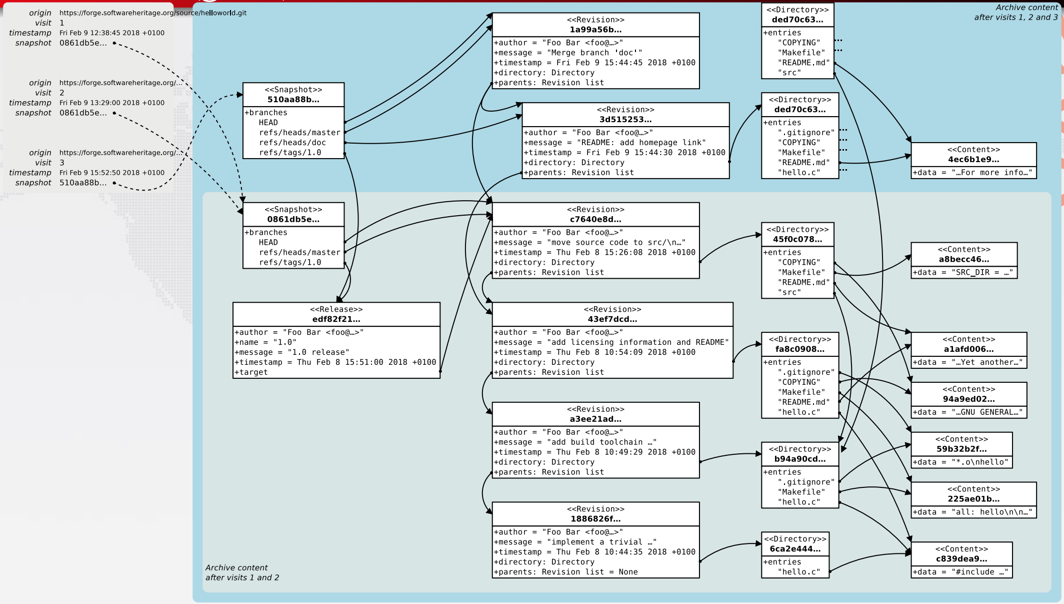
- homepages, wikis
- BTS/issues/code reviews/etc.
- mailing lists

Long term vision: play our part in a *"semantic wikipedia of software"*

Data flow


Forges
GitHub lister
GitLab lister
Distros
Debian lister
PyPI lister
Package repos
CPAN
...
software origins
git
git
git
git
svn
svn
svn
hg
hg
hg
deb
deb
pypi
pypi
Git loader
Mercurial loader
Debian source package loader
PyPI loader
Software Heritage Archive
Merkle DAG
+
blob storage
Listing (full/incremental)
Loading & deduplication
Scheduling

The archive: a (giant) Merkle DAG



origin https://forge.softwareheritage.org/source/helloworld.git
visit 1
timestamp Fri Feb 9 12:38:45 2018 +0100
snapshot 0861db5e...

origin https://forge.softwareheritage.org/...
visit 2
timestamp Fri Feb 9 13:29:00 2018 +0100
snapshot 0861db5e...

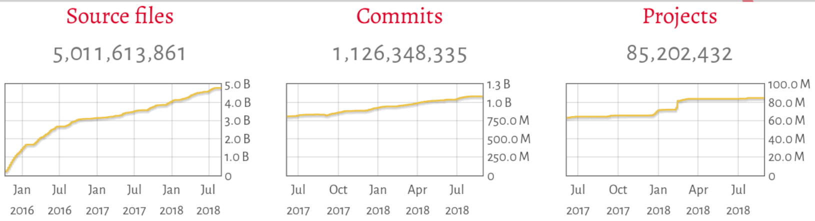
origin https://forge.softwareheritage.org/...
visit 3
timestamp Fri Feb 9 15:52:50 2018 +0100
snapshot 510aa88b...

Archive content after visits 1, 2 and 3

Archive content after visits 1 and 2

```
<<Snapshot>>
510aa88b...
+branches
   HEAD
   refs/heads/master
   refs/heads/doc
   refs/tags/1.0

<<Snapshot>>
0861db5e...
+branches
   HEAD
   refs/heads/master
   refs/tags/1.0

<<Release>>
edf82f21...
+author = "Foo Bar <foo@…>"
+name = "1.0"
+message = "1.0 release"
+timestamp = Thu Feb 8 15:51:00 2018 +0100
+target

<<Revision>>
1a99a56b...
+author = "Foo Bar <foo@…>"
+message = "Merge branch 'doc'"
+timestamp = Fri Feb 9 15:44:45 2018 +0100
+directory: Directory
+parents: Revision list

<<Revision>>
3d515253...
+author = "Foo Bar <foo@…>"
+message = "README: add homepage link"
+timestamp = Fri Feb 9 15:44:30 2018 +0100
+directory: Directory
+parents: Revision list

<<Revision>>
c7640e8d...
+author = "Foo Bar <foo@…>"
+message = "move source code to src/\n…"
+timestamp = Thu Feb 8 15:26:08 2018 +0100
+directory: Directory
+parents: Revision list

<<Revision>>
43ef7dcd...
+author = "Foo Bar <foo@…>"
+message = "add licensing information and README"
+timestamp = Thu Feb 8 10:54:09 2018 +0100
+directory: Directory
+parents: Revision list

<<Revision>>
a3ee21ad...
+author = "Foo Bar <foo@…>"
+message = "add build toolchain …"
+timestamp = Thu Feb 8 10:49:29 2018 +0100
+directory: Directory
+parents: Revision list

<<Revision>>
1886826f...
+author = "Foo Bar <foo@…>"
+message = "implement a trivial …"
+timestamp = Thu Feb 8 10:44:35 2018 +0100
+directory: Directory
+parents: Revision list = None

<<Directory>>
ded70c63...
+entries
   "COPYING"
   "Makefile"
   "README.md"
   "src"

<<Directory>>
ded70c63...
+entries
   ".gitignore"
   "COPYING"
   "Makefile"
   "README.md"
   "hello.c"

<<Directory>>
45f0c078...
+entries
   "COPYING"
   "Makefile"
   "README.md"
   "src"

<<Directory>>
fa8c0908...
+entries
   ".gitignore"
   "COPYING"
   "Makefile"
   "README.md"
   "hello.c"

<<Directory>>
b94a90cd...
+entries
   ".gitignore"
   "Makefile"
   "hello.c"

<<Directory>>
6ca2e444...
+entries
   "hello.c"

<<Content>>
4ec6b1e9...
+data = "…For more info…"

<<Content>>
a8becc46...
+data = "SRC_DIR = …"

<<Content>>
a1afd006...
+data = "…Yet another…"

<<Content>>
94a9ed02...
+data = "…GNU GENERAL…"

<<Content>>
59b32b2f...
+data = "*.o\nhello"

<<Content>>
225ae01b...
+data = "all: hello\n\n…"

<<Content>>
c839dea9...
+data = "#include …"
```

Archive coverage



Source files

5,011,613,861

Commits

1,126,348,335

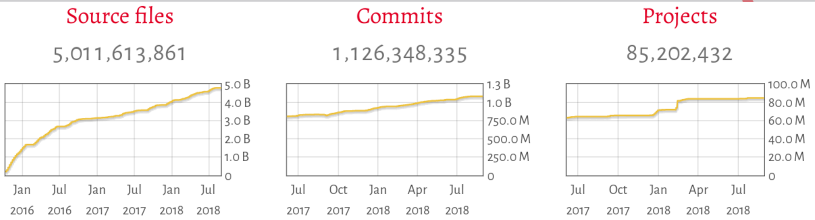
Projects

85,202,432

Current sources

- live: GitHub, Debian, GitLab.com, PyPI
- one-off: Gitorious, Google Code, GNU
- WIP: Bitbucket

Archive coverage

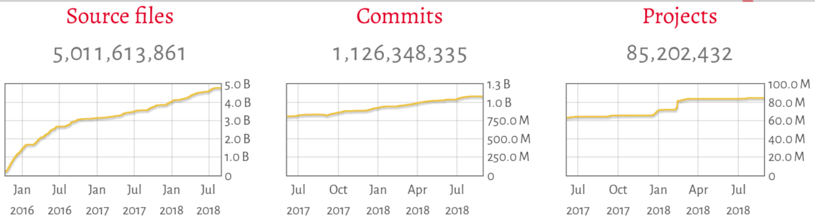


Current sources

- live: GitHub, Debian, GitLab.com, PyPI
- one-off: Gitorious, Google Code, GNU
- WIP: Bitbucket

175 TB (compressed) blobs, 6 TB database (as a graph: 10 B nodes + 100 B edges)

Archive coverage



Current sources

- live: GitHub, Debian, GitLab.com, PyPI
- one-off: Gitorious, Google Code, GNU
- WIP: Bitbucket

175 TB (compressed) blobs, 6 TB database (as a graph: 10 B nodes + 100 B edges)

The *richest* public source code archive, … and growing daily!

Web API

RESTful API to programmatically access the Software Heritage archive
`https://archive.softwareheritage.org/api/`

Features

- pointwise browsing of the archive
  - … snapshots ⟶ revisions ⟶ directories ⟶ contents …
- full access to the metadata of archived objects
- crawling information
  - *when have you last visited this Git repository I care about?*
  - *where were its branches/tags pointing to at the time?*

Endpoint index

`https://archive.softwareheritage.org/api/1/`

Bulk download

Vault service

- source code is thoroughly deduplicated within the Software Heritage archive
- bulk download of large artefacts (e.g., a Linux kernel release) requires collecting millions of objects
- the Software Heritage Vault cooks and caches source code bundles for bulk download needs

Tech bits

- RESTful API to request downloads, notifications, and monitoring
- `docs.softwareheritage.org/devel/swh-vault`

Web user interface

Browser-based interface to browse the Software Heritage archive
`https://archive.softwareheritage.org/browse/`

Features

- all REST API features, but good looking :-)
  - browsing: snapshots ⟶ revisions ⟶ directories ⟶ contents …
  - access to metadata and crawling information
- origin search, as full text indexing of origin URLs
- bulk download, via integration with the Vault

Outline

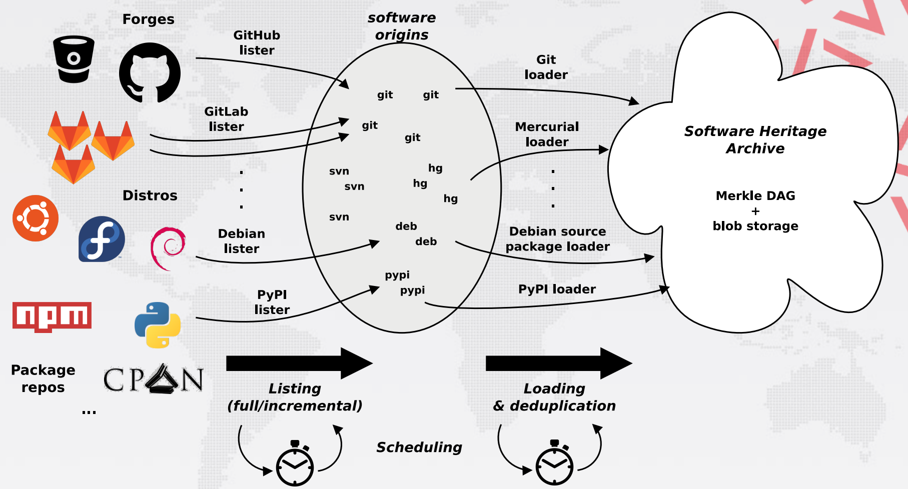
1 The Software Commons

2 Software Heritage

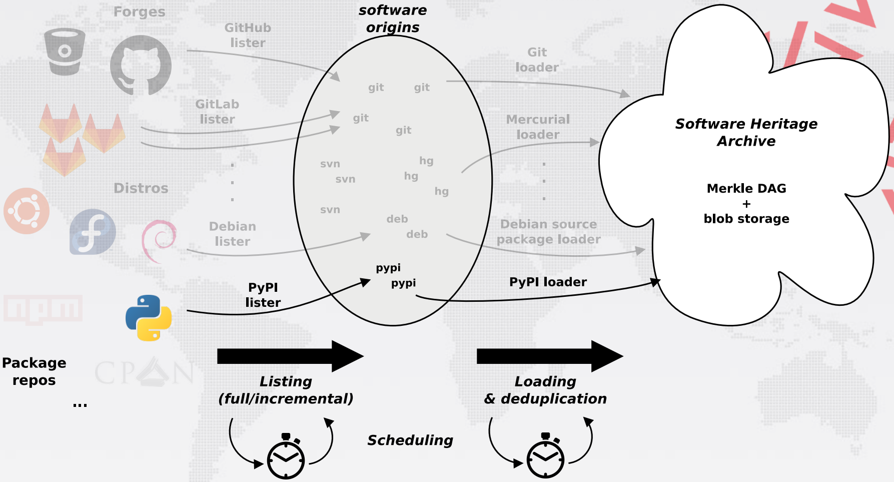
3 The Great Library of Python source code

4 Getting involved

Data flow redux



Our focus


Forges
GitHub lister
GitLab lister
Distros
Debian lister
Package repos
...
PyPI lister
software origins
git
git
git
git
svn
svn
svn
hg
hg
hg
deb
deb
pypi
pypi
Git loader
Mercurial loader
Debian source package loader
PyPI loader
Software Heritage Archive
Merkle DAG
+
blob storage
Listing (full/incremental)
Scheduling
Loading & deduplication

Listing all Python modules (1/3)

`https://forge.softwareheritage.org/source/swh-lister/`

What does a Software Heritage lister do?

- crawls and parses upstream list of project APIs
- generates origins (records that the project has been detected) and loading tasks

Credits go to Avi Kelman for the lister scaffolding, and to Antoine Dumont for the PyPI implementation

A visit of the Cheese Shop

- A little bit more efficiently than John Cleese
- Uses `https://pypi.org/simple/` (according to the warehouse docs, the only "package listing" API that's not on the way to deprecation)

Listing all Python modules (2/3)

GET https://pypi.org/simple/

```
<!DOCTYPE html>
<html>
  <head>
    <title>Simple index</title>
  </head>
  <body>
  <a href="/simple/0/">0</a>
  <a href="/simple/0-0/">0-._.-._.-._.-._.-._.-._.-0</a>
  [...]
  <a href="/simple/django/">Django</a>
  [...]
  </body>
</html>
```

Listing all Python modules (3/3)

```python
# Origin specification
origin = {
    'type': 'pypi',
    'url': 'https://pypi.org/packages/Django/', # Canonical project URL
}
```

Listing all Python modules (3/3)

```python
# Origin specification
origin = {
    'type': 'pypi',
    'url': 'https://pypi.org/packages/Django/', # Canonical project URL
}
```

```python
# Scheduler task specification
update_task = {
    'type': 'origin-update-pypi',
    'policy': 'recurring',
    'next_run': datetime.now(tz=timezone.utc),
    'arguments': {
        'args': [
            'Django',                                  # Project name
            'https://pypi.org/packages/Django/', # Origin URL
            'https://pypi.org/pypi/Django/json', # Metadata URL
        ],
        'kwargs': {},
    },
    'priority': None,
}
```

Task scheduling (1/2)

https://forge.softwareheritage.org/source/swh-scheduler/

What does the Software Heritage scheduler do?

- Record recurrent and one-shot jobs in a database
- Schedules runs of these jobs, records their results
- Manages retries for transient job failures (remote service unavailable, ...)
- Manages adaptive intervals for recurrent jobs

Task scheduling (2/2)

Builds upon trusted Python tools

- Celery is used as a task queuing middleware, and for its worker management framework
- Workers send task results through the Celery events mechanism

And makes them more useful to us

- The database is the single source of truth
- `swh.scheduler.celery_backend.runner` pulls tasks from the database into Celery, limiting the RabbitMQ queue depth (allows task prioritization)
- `swh.scheduler.celery_backend.listener` fetches task results from Celery events and updates the database
- Archival of elapsed tasks/runs/logs in elasticsearch to keep the database snappy

Loading Python packages (1/4)

What's a Python package anyway?

- Source distributions (`sdists`, currently tarballs or zips)
- Binary distributions (`bdists`, which are mostly wheels these days)

As we're interested in source code, Software Heritage looks at `sdists` exclusively

- The current sdist format is unspecified: you probably get a tarball, which maybe contains a `setup.py` somewhere
- When building a sdist, distutils generates a machine-readable `PKG-INFO` file is generated and puts in the tarball

The long wait for PEP 517 ("A build-system independent format for source trees")

- One uniform transport format: a gzipped tarball with one toplevel directory
- Machine parsable data about the project by default (`pyproject.toml`)

Hopefully soon in your nearest Cheese Shop (go help the folks in PyPA!)

Loading Python packages (2/4)

`https://forge.softwareheritage.org/source/swh-loader-pypi/`

Common loading process

Implemented in `swh.loader.core`

- Fetch metadata about current versions
- Compare to latest loaded versions
- Download and process versions we had never seen
- Load new data

Loading Python packages (2/4)

`https://forge.softwareheritage.org/source/swh-loader-pypi/`

Common loading process

Implemented in `swh.loader.core`

- Fetch metadata about current versions
- Compare to latest loaded versions
- Download and process versions we had never seen
- Load new data

PyPI specifics

Implemented in `swh.loader.pypi`

- Comparison done using the `sdist` digests
- PKG-INFO metadata parsed and saved
- versions with multiple sdists imported separately

Loading Python packages (3/4)

PyPI snapshots

```
pifpaf_snapshot = {
  'id': b'\xc6_\xfe#\x94\xba\x81\xc3\x94\x9b\xeb[\x06\xf5JC\x0f\x19n\xa6',
  'branches': {
    b'releases/0.0.1': {
    b'releases/0.0.2': {
    ...
    b'releases/2.1.2': {
      'target': b'\x8a\xcd\xf31\xee\xe50\xe2\x81]\x08:5\xd9_\xd6\xeff\xc9\xa3',
      'target_type': 'revision',
    },
    b'releases/2.1.2.dev7': {
      'target': b'hGh\x15h|\xf3\xd2v\xf8\xec-\xa7\xfeuB\xda3\x83x',
      'target_type': 'revision',
    },
    b'HEAD': {
      'target': b'releases/2.1.2',
      'target_type': 'alias',
    },
  },
}
```

Loading Python packages (4/4)

PyPI revisions

```
pifpaf_revision = {
   'id': b'\x8a\xcd\xf3l\xee\xe50\xe2\x81]\x08:5\xd9_\xd6\xeff\xc9\xa3',
   'author': {
      'name': b'Julien Danjou',
      ...
   },
   'date': {
      'timestamp': {'seconds': 1538577319, 'microseconds': 0},
   },
   ...
   'type': 'tar',
   'directory': b'\xa4\xf2\xad\xb1\xef\r\xcf\x894::@=\xf9R\x86=\x19"\\',
   'message': b'2.1.2',
```

Loading Python packages (4/4)

PyPI revisions

```
pifpaf_revision = {
  'id': b'\x8a\xcd\xf3l\xee\xe50\xe2\x81]\x08:5\xd9_\xd6\xeff\xc9\xa3',
  'author': {
    'name': b'Julien Danjou',
    ...
  },
  'date': {
    'timestamp': {'seconds': 1538577319, 'microseconds': 0},
  },
  ...
  'type': 'tar',
  'directory': b'\xa4\xf2\xad\xb1\xef\r\xcf\x894::@=\xf9R\x86=\x19"\\',
  'message': b'2.1.2',
```

```
  'metadata': {
    'project': { # Metadata parsed from PKG-INFO
      'name': 'pifpaf',
      'author': 'Julien Danjou',
      'license': None,
      'summary': 'Suite of tools and fixtures to manage daemons for testing',
      'version': '2.1.2',
      ...
      'classifiers': [
        'Intended Audience :: Information Technology',
        ...
      ],
      ...
    },
   'classifiers': [
    'Intended Audience :: Information Technology',
    ...
   ],
   ...
  },
  'original_artifact': {  # The original tarball we downloaded
   'url': 'https://files.pythonhosted.org/packages/cc/ce/2599[...]',
   'date': '2018-10-03T14:35:19',
   'sha1': '00c4efc47580b5c4ad1dcdb5118159f9b057b0fd',
   'size': 192940,
   'sha256': 'a6eef2ae56ac90d02df5f45885973e108c960a2ea113cc76[...]',
   'filename': 'pifpaf-2.1.2.tar.gz',
   'sha1_git': '8ce7e3ddda336dd9edff26ae8efaf4b81439c42c',
   'blake2s256': 'c4f7fcd4324715f4bfb54f8eefb10fde803efb7a02e2[...]',
   'archive_type': 'tar',
  },
 },
 'synthetic': True,
 'parents': [],
}
```

Outline

1. The Software Commons
2. Software Heritage
3. The Great Library of Python source code
4. Getting involved

Roadmap

Features…

- (done) lookup by content hash
- (done) browsing: "wayback machine" for source code (API + UI)
- (early access) deposit of source code bundles directly to the archive
- (early access) save code now, on-demand archive
- (done) download: `wget` / `git clone` from the archive
- (todo) provenance lookup for all archived content
- (todo) full-text search on all archived source code files

Roadmap

Features...

- (done) lookup by content hash
- (done) browsing: "wayback machine" for source code (API + UI)
- (early access) deposit of source code bundles directly to the archive
- (early access) save code now, on-demand archive
- (done) download: `wget` / `git clone` from the archive
- (todo) provenance lookup for all archived content
- (todo) full-text search on all archived source code files

... and much more than one could possibly imagine

all the world's software development history at hand's reach!

You can help!

Coding

★★	Web UI improvements
★★★	loaders for unsupported VCS/package formats
★★★	listers for unsupported forges/package managers

https://forge.softwareheritage.org/
https://docs.softwareheritage.org/devel/

You can help!

Coding

★★	Web UI improvements
★★★	loaders for unsupported VCS/package formats
★★★	listers for unsupported forges/package managers

`https://forge.softwareheritage.org/`
`https://docs.softwareheritage.org/devel/`

Community

★★★	spread the world, help us with sustainability
★★	document endangered source code

`wiki.softwareheritage.org/Suggestion_box`

You can help!

Coding

★★	Web UI improvements
★★★	loaders for unsupported VCS/package formats
★★★	listers for unsupported forges/package managers

`https://forge.softwareheritage.org/`
`https://docs.softwareheritage.org/devel/`

Community

★★★	spread the world, help us with sustainability
★★	document endangered source code

`wiki.softwareheritage.org/Suggestion_box`

Join us

- `www.softwareheritage.org/jobs` — job openings
- `wiki.softwareheritage.org/Internship` — internships

Conclusion

Software Heritage is

- a reference archive of all Free Software ever written
- an international, open, nonprofit, mutualized infrastructure
- now accessible to developers, users, vendors
- at the service of our community, at the service of society

Come in, we're open!

`www.softwareheritage.org` – general information
`wiki.softwareheritage.org` – internships, leads
`forge.softwareheritage.org` – our own code