Provit Documentation

Release 1.0.1

Universitätsbibliothek Leipzig

Jun 17, 2019

Contents:

1 PROVIT - PROVenance Integration Tools **1**
- 1.1 Requirements . . . 1
- 1.2 Installation . . . 1
- 1.3 Usage . . . 2
- 1.4 Roadmap . . . 3
- 1.5 Overview . . . 3

2 Vocabulary **5**
- 2.1 Agents . . . 5

3 provit **7**
- 3.1 provit package . . . 7

4 Indices and tables **13**

Python Module Index **15**

Index **17**

CHAPTER 1

PROVIT - PROVenance Integration Tools

PROVIT is a light, dezentralized data provenance and documentation tool. It allows the user to track workflows and modifications of data-files.

PROVIT works completely decentralized, all information is stored in .prov files (as JSON-LD RDF graphs) along it's corresponding data file in the file system. No additional database or server setup is needed.

A small subset of the W3C PROV-O vocabulary is implemented.

PROVIT aim to provided an easy to use interface for users who have never worked with provenance tracking before. If you feel limited by PROVIT you should have a look at more extensive implementations, e.g.: prov.

Full documentation is available under: provit.readthedocs.io.

1.1 Requirements

This software was tested on Linux with Python 3.5 and 3.6.

1.2 Installation

Installation via pip is recommended for end users. We strongly encourage end users to make use of a virtualenv.

1.2.1 pip

Clone the repository and create a virtual environment (optional) and install into with pip into the virtualenv.

```
$ mkvirtualenv provit
$ pip install provit
```

1.2.2 git / Development

Clone the repository and create a virtualenv.

```
$ git clone https://github.com/diggr/provit
$ mkvirtualenv provit
```

Install it with pip in *editable* mode

```
$ pip install -e ./provit
```

1.3 Usage

PROVIT provides a command line client which can be used to enrich any file based data with provenance information.

PROVIT also includes a (experimental) web-based interface (PROVIT Browser).

1.3.1 Command Line Client

Usage:

Open PROVIT Browser:

```
$ provit browser
```

Add provenace event to a file:

```
$ provit add FILEPATH [OPTIONS]
```

Options:

-a AGENT, --agent AGENT Provenance information: agent (multiple=True)

--activity ACTIVITY Provenance information: activity

-d DESCRIPTION, --desc DESCRIPTION Provenance information: Description of the data manipulation process

-o ORIGIN, --origin ORIGIN Provenance information: Data origin

-s SOURCES, --sources SOURCES Provenance information: Source files (multiple=True)

--help Show this message and exit.

1.3.2 Provenance Class

```
from provit import Provenance

# load prov data for a file, or create new prov for file
prov = Provenance(<filepath>)

# add provenance metadata
prov.add(agents=[ "agent" ], activity="activity", description="...")
prov.add_primary_source("primary_source")
prov.add_sources([ "filepath1", "filepath2" ])
```

(continues on next page)

(continued from previous page)

```
# return provenance as json tree
prov_dict = prov.tree()

# save provenance metadata into "<filename>.prov" file
prov.save()
```

1.4 Roadmap

General roadmap of the next steps in development

- Tests
- Tutorials
- Windows support
- Agent management in PROVIT Browser

1.5 Overview

Authors P. Mühleder muehleder@ub.uni-leipzig.de, F. Rämisch raemisch@ub.uni-leipzig.de

CHAPTER 2

Vocabulary

Provit uses a small subset of the W3C PROV-O vocabulary.

2.1 Agents

Provit implements three different agents:

- Organization
- Person
- SoftwareAgent

The names are specified in PROV-O.

CHAPTER 3

provit

3.1 provit package

3.1.1 Subpackages

provit.browser package

Submodules

provit.browser.backend module

provit.browser.backend.**add_prov_data**()

provit.browser.backend.**agent_list**()

provit.browser.backend.**config**()

provit.browser.backend.**config_update**()

provit.browser.backend.**delete_directories**()

provit.browser.backend.**directories**()

provit.browser.backend.**file_browser**()

provit.browser.backend.**file_list**()

provit.browser.backend.**files_with_prov**(*directory*)

provit.browser.backend.**get_prov_data**()

provit.browser.backend.**index**()

provit.browser.backend.**remove_prov_data**()

provit.browser.backend.**start_backend**(*debug=False*)

provit.browser.backend.**start_provit_browser**(*debug=False*)
Start provit backend and open webbrowser

provit.browser.backend.**start_webbrowser**()

provit.browser.backend.**update_file_list**()

Module contents

provit.tests package

Submodules

provit.tests.test_agent module

provit.tests.test_cli module

provit.tests.test_config module

provit.tests.test_config.**test_get_config**(*tmp_path*)

provit.tests.test_config.**test_load_provit_dir**(*tmp_path*)

provit.tests.test_config.**test_load_provit_dir_from_config**(*tmp_path*)

provit.tests.test_home module

provit.tests.test_prov module

provit.tests.test_utils module

Module contents

3.1.2 Submodules

3.1.3 provit.agent module

Everything agent profile related

1. Classes

- Person
- SoftwareAgent
- Organization

Each class contains :to_json(): and :graph(): functions, returning the data of the classes either as json-dict or rdflib Graph.

2. Helper functions

- load_agent_profile(slug) loads the yaml file of agent :slug: and initiates the respective agent class (depending on type)

- load_agent_profiles() loads all agent yaml files and returns a list of agent classes
- agent_factory(slug, type_) initializes an agent class of type :type_: with id/uri :slug:

class provit.agent.**OrganizationAgent**(*slug*, *name=[]*, *homepage=[]*, *uri=''*)
Bases: object

graph()

to_json()

update(*data*)

class provit.agent.**PersonAgent**(*slug*, *name=[]*, *institution=[]*, *homepage=[]*, *email=[]*, *uri=''*)
Bases: object

graph()

to_json()

update(*data*)

class provit.agent.**SoftwareAgent**(*slug*, *name=[]*, *version=[]*, *homepage=[]*, *uri=''*)
Bases: object

graph()

to_json()

update(*data*)

provit.agent.**agent_factory**(*slug*, *type_*)
return "empty" agent class instance of the specified type

provit.agent.**load_agent_profile**(*slug*)
loads agent yaml profile (if available) and initiates agent class with the values obtained from the yaml file

provit.agent.**load_agent_profiles**()

3.1.4 provit.cli module

PROVIT command line client

Usage:

show provenance file information $ provit [options] FILE_PATH

3.1.5 provit.config module

provit configuration module

provides the Config-class as well as its factory method get_config.

By default, $HOME/.provit is assumed to be provits default config directory. This can be customized.

class provit.config.**Config**(*provit_dir: pathlib.Path*, *person: str = 'Person'*, *software: str = 'SoftwareAgent'*, *organization: str = 'Organization'*, *base_uri: str = 'http://vocab.ub.uni-leipzig.de/provit/{}'*)
Bases: object

agent_profile(*slug*)

agent_profile_exists(*slug*)

agents_dir

base_uri = 'http://vocab.ub.uni-leipzig.de/provit/{}'

directories_file

get_agent_profile(*slug*)

organization = 'Organization'

person = 'Person'

software = 'SoftwareAgent'

provit.config.**get_config**(*provit_dir=None*)
factory method for Config class. can be given a custom provit dir. If no directory is given, the default directory ~/.provit will be chosen.

3.1.6 provit.home module

Helper functions for accessing data in the provit home directory

provit.home.**add_directory**(*directory*)
Add directory to project directories list

provit.home.**load_directories**()
load the list of directories from the directories yaml file

provit.home.**remove_directories**(*directory*)
Remove directories from project directory list

3.1.7 provit.namespaces module

3.1.8 provit.prov module

Provenance class handles provenance metadata information.

Use:

from pit.prov import Provenance

#load prov data for a file, or create new prov for file prov = Provenance(<filepath>)

#add provenance metadata prov.add(agent="agent", activity="activity", description="...") prov.add_primary_source("primary_source", url="http://...", comment="...") prov.add_sources(["filepath1", "filepath2"])

#return provenance as json tree prov_dict = prov.tree()

#save provenance metadata into "<filename>.prov" file prov.save()

class provit.prov.**Provenance**(*filepath*, *namespace=None*, *create_new=True*, *overwrite=False*)
Bases: object

Provenance class handles the provenance metadata graph

add(*agents*, *activity*, *description*, *started_at=''*, *ended_at=''*)
Add new basic provenance information (agent, activity) to file

add_graph(*graph*)

add_primary_source(*primary_source*, *url=None*, *comment=None*)
Adds primary source (+ url and comment) to provenance information

add_sources(*filepaths*, *add_prov_to_source=True*)
Add provenance information from source file (wasDerivedFrom) to provenance graph If source file does not have valid provenance data, a prov graph for the source file is initialized

get_agents(*include_primary_sources=False*)
Returns agent profiles from prov graph

get_current_location()
Returs the file location of root element

get_primary_sources()
Returns the URIs of all primary sources in prov graph

iter_remove(*root_uri*)

remove_last_event()
removes the last provenance event and all sources, if they do not belong to the same file

save()
Serializes prov graph as json-ld and saves it to file

tree()
Returns of dict tree with provenance information

provit.prov.**load_prov**(*filepath*, *namespace=Namespace('http://vocab.ub.uni-leipzig.de/provit/')*)
Loads a Provenance Object from the given file path or returns None if no (valid) provenance file was found. :param filepath: :return:

provit.prov.**load_prov_files**(*directory*)

3.1.9 provit.utils module

provit.utils.**load_jsonld**(*filepath*)
Reads json-ld file and returns (rdfslib) graph and context

provit.utils.**provit_uri**(*slug*)

provit.utils.**walk_up**(*start_dir*)
Walks up directory tree from :start_dir: and returns directory paths

3.1.10 Module contents

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

provit, 11
provit.agent, 8
provit.browser, 8
provit.browser.backend, 7
provit.cli, 9
provit.config, 9
provit.home, 10
provit.namespaces, 10
provit.prov, 10
provit.tests, 8
provit.tests.test_config, 8
provit.utils, 11

Index

A

add() (*provit.prov.Provenance method*), 10
add_directory() (*in module provit.home*), 10
add_graph() (*provit.prov.Provenance method*), 10
add_primary_source() (*provit.prov.Provenance method*), 10
add_prov_data() (*in module provit.browser.backend*), 7
add_sources() (*provit.prov.Provenance method*), 11
agent_factory() (*in module provit.agent*), 9
agent_list() (*in module provit.browser.backend*), 7
agent_profile() (*provit.config.Config method*), 9
agent_profile_exists() (*provit.config.Config method*), 9
agents_dir (*provit.config.Config attribute*), 9

B

base_uri (*provit.config.Config attribute*), 10

C

Config (*class in provit.config*), 9
config() (*in module provit.browser.backend*), 7
config_update() (*in module provit.browser.backend*), 7

D

delete_directories() (*in module provit.browser.backend*), 7
directories() (*in module provit.browser.backend*), 7
directories_file (*provit.config.Config attribute*), 10

F

file_browser() (*in module provit.browser.backend*), 7
file_list() (*in module provit.browser.backend*), 7
files_with_prov() (*in module provit.browser.backend*), 7

G

get_agent_profile() (*provit.config.Config method*), 10
get_agents() (*provit.prov.Provenance method*), 11
get_config() (*in module provit.config*), 10
get_current_location() (*provit.prov.Provenance method*), 11
get_primary_sources() (*provit.prov.Provenance method*), 11
get_prov_data() (*in module provit.browser.backend*), 7
graph() (*provit.agent.OrganizationAgent method*), 9
graph() (*provit.agent.PersonAgent method*), 9
graph() (*provit.agent.SoftwareAgent method*), 9

I

index() (*in module provit.browser.backend*), 7
iter_remove() (*provit.prov.Provenance method*), 11

L

load_agent_profile() (*in module provit.agent*), 9
load_agent_profiles() (*in module provit.agent*), 9
load_directories() (*in module provit.home*), 10
load_jsonld() (*in module provit.utils*), 11
load_prov() (*in module provit.prov*), 11
load_prov_files() (*in module provit.prov*), 11

O

organization (*provit.config.Config attribute*), 10
OrganizationAgent (*class in provit.agent*), 9

P

person (*provit.config.Config attribute*), 10
PersonAgent (*class in provit.agent*), 9
Provenance (*class in provit.prov*), 10
provit (*module*), 11
provit.agent (*module*), 8
provit.browser (*module*), 8

provit.browser.backend (*module*), 7
provit.cli (*module*), 9
provit.config (*module*), 9
provit.home (*module*), 10
provit.namespaces (*module*), 10
provit.prov (*module*), 10
provit.tests (*module*), 8
provit.tests.test_config (*module*), 8
provit.utils (*module*), 11
provit_uri() (*in module provit.utils*), 11

R

remove_directories() (*in module provit.home*), 10
remove_last_event() (*provit.prov.Provenance method*), 11
remove_prov_data() (*in module provit.browser.backend*), 7

S

save() (*provit.prov.Provenance method*), 11
software (*provit.config.Config attribute*), 10
SoftwareAgent (*class in provit.agent*), 9
start_backend() (*in module provit.browser.backend*), 7
start_provit_browser() (*in module provit.browser.backend*), 7
start_webbrowser() (*in module provit.browser.backend*), 8

T

test_get_config() (*in module provit.tests.test_config*), 8
test_load_provit_dir() (*in module provit.tests.test_config*), 8
test_load_provit_dir_from_config() (*in module provit.tests.test_config*), 8
to_json() (*provit.agent.OrganizationAgent method*), 9
to_json() (*provit.agent.PersonAgent method*), 9
to_json() (*provit.agent.SoftwareAgent method*), 9
tree() (*provit.prov.Provenance method*), 11

U

update() (*provit.agent.OrganizationAgent method*), 9
update() (*provit.agent.PersonAgent method*), 9
update() (*provit.agent.SoftwareAgent method*), 9
update_file_list() (*in module provit.browser.backend*), 8

W

walk_up() (*in module provit.utils*), 11