WAVES Documentation

Release 1.1.4

Read the Docs

Feb 21, 2018

Use WAVES

CHAPTER 1

ReadME

1.1 Features

- Create a Dedicated web app based on waves-core reusable app http://www.github.com/lirmm/waves-core
- Demonstrate how to override WAVES-core app to fit specific needs

1.2 Installation

- WAVES web app comply to standard reusable project layout for Django. You can use it as a stand alone platform.
- Simply clone or download sources, follow the few configurations steps, and her you are, you can put your tools online !
- See “Details Installation” for more info.

1.3 Documentation

See doc on https://waves-demo.readthedocs.io

1.4 Support

If you are having issues, (or just want to say hello) : we have a mailing list located at: waves-webapp@googlegroups.com

1.5 License

License (GPLv3)

CHAPTER 2

Authors

- Marc Chakiachvili (LIRMM - UMR 5506 CNRS / UM - France),
- Vincent Lefort (LIRMM - UMR 5506 CNRS / UM - France),
- Anne-Muriel Arigon Chiffoleau (LIRMM - UMR 5506 CNRS / UM - France),

2.1 Version

1.1.4 (1.1.4) of 2016/09/25

CHAPTER 3

License (GPLv3)

WAVES are free software: you can redistribute it and/or modify it under the terms of the GNU General Public License version 3 as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

For more specific details see http://www.gnu.org/licenses, the Quick Guide to GPLv3. in the codebase.

The GNU operating system which is under the same license has an informative FAQ here.

CHAPTER 4

Note to developers

We very much appreciate you using our code to do fun and interesting things with. We hope that while doing this you may find and fix bugs or make enhancements that could be useful for the greater community and will makes the developers aware of them by emailing to waves-webapp@googlegroups.com so they can be considered to be added to the original code base.

CHAPTER 5

Installation

GET a WAVES demo website following the next few steps. WAVES can run with Apache or Nginx with uWSGI

Warning: To run WAVES, it is strongly recommended to setup a dedicated user. Indeed, WAVES uses saga-python. This module will try to create directories (.radical and .saga) for which another user (such as www-data) might not have the rights to do so.

5.1 0. Prerequisites

Note:

In order to install WAVES you will need:

- python 2.7.X (WAVES is not yet compatible with python 3.5)
- pip package manager
- A web server: Apache or NGINX
- A database backend (Mysql or Postgres) but by default WAVES runs with sqlite

0.1 Install From sources, clone our repository:

```
git clone https://github.com/lirmm/waves-demo/ [waves_dir]
```

or download archive at https://github.com/lirmm/waves-demo/ and uncompress the archive in your destination directory ([waves_dir])

0.2 Install requirements:

- `$ cd [waves_dir]`
- `[waves_dir]$ virtualenv .venv`

- `[waves_dir]$ source .venv/bin/activate`
- `(.venv)[waves_dir]]$ pip install -r requirements.txt`
- `(.venv)[waves_dir]]$ pip install -r requirements/production.txt` (for production)
- `(.venv)[waves_dir]]$ pip install -r requirements/mysql.txt` (for mysql DB layer)

5.2 1. Install WAVES-Demo

1.1 Configuration files:

- WAVES env configuration file:
  - `(.venv)[waves_dir]$ cd src/waves_demo/settings/`
  - `(.venv)[waves_dir]/src/waves_demo/settings$ cp local.sample.env local.env`
  - minimal setup requires these parameters:
    - SECRET_KEY=your-secret-key-to-keep-secret
    - REGISTRATION_SALT=generate-your-key
    - ALLOWED_HOSTS=your-host-name (Ex: localhost, 127.0.0.1 for testing purpose)
  - you can set up as well your db connection params here (or in classical "DJANGO way" settings if you want)

1.2 Set up database:

- Check parameters with: `(.venv)[waves_dir]/src/$ ./manage.py check` (pip install any missing dependencies)
- See your configuration with: `(.venv)[waves_dir]/src/$ ./manage.py waves config`
- If no setup, default database is used ~/[waves_dir]/waves.sample.sqlite3

1.2.1: Setup your database connection string (if not using sqlite default)

- Create local.env (copy from local.env.sample located in waves_demo/settings/)
- Setup line corresponding to your needs (DATABASE_URL)

1.2.2: Check everything is well.

```
(.venv)[waves_dir]/src/$ ./manage.py check
```

1.2.3: Create migration files:

```
(.venv)[waves_dir]/src/$ ./manage.py makemigrations
```

Note: If you see this message:

```
You are trying to add a non-nullable field 'service' to␣
↪demowavessubmission without a default; we can't do that␣
↪(the database needs something to populate existing rows).
Please select a fix:
```

```
1) Provide a one-off default now (will be set on all existing␣
↪rows with a null value for this column)
2) Quit, and let me add a default in models.py
```

=> Select 1) option and set up default value to '1' (this problem is due to swapping of default Service and Submission)

- Create your database: `(.venv)[waves_dir]/src/$ ./manage.py migrate`
- Create Superadmin user: `(.venv)[waves_dir]/src/$ ./manage.py createsuperuser`

1.2.4: Load sample data into your database (optional):

- Demo database is initially setup from:

```
(.venv)[waves_dir]/src/$ ./manage.py loaddata demo/
fixtures/init.json
```

1.3 Test your server:

- `(.venv)[waves_dir]/src/$ ./manage.py runserver [ServerIP:ServerPort] --insecure`

1.4 Start WAVES daemons:

- `(.venv)[waves_dir]/src/$ ./manage.py waves queue start`
- `(.venv)[waves_dir]/src/$ ./manage.py waves purge start`

waves queue command allow you to control daemon, available commands are start|stop|status

5.3 2. Configure the web server:

2.1 Production settings:

- Create staticfiles `(.venv)[waves_dir]/src/$ ./manage.py collectstatic`
- Setup your server following instruction Django Docs

See also:

UWSGI configuration at http://uwsgi-docs.readthedocs.io/

See also:

APACHE http://uwsgi-docs.readthedocs.io/en/latest/Apache.html

See also:

NGINX http://uwsgi-docs.readthedocs.io/en/latest/tutorials/Django_and_nginx.html

Warning: You might experience some troubles with directories permissions when installing WAVES-demo on your web server on some directories: logs, jobs, data, binaries directories must be writable both from web user/group (www-data or apache) and by the user which launch the queue.

CHAPTER 6

Contributing

You can contribute to WAVES project with following repositories:

- Git source code: https://github.com/lirmm/waves-demo
- Issue tracker: https://github.com/lirmm/waves-demo/issues
- Mailing list: waves-webapp@googlegroups.com

CHAPTER 7

CHANGELOG

7.1 Version 1.1.3 - 2017-02-07

- [RELEASE] - First production release
- [Updated] - Dependency to WAVES-core 1.1.6
- [Updated] - Dependency to WAVES-galaxy 1.1.2
- **[Added] Poc for :**
  - extending Service model
  - extending Submission model
  - extending service form pages
- [Added] REST form API used for specific service
- [UPDATED] - Demo adaptors are fully functional for configuration, but not really execute jobs for obvious security reason

7.2 Version 1.1.2 - 2017-10-18

- Added Waves-Galaxy adaptors by default in app
- Support Galaxy tools import

7.3 Version 1.1.1 - 2017-09-30

- Integrated modification issued from waves-core 1.1.2
- Added changelog

7.4 Version 1.1.0 - 2017-07-07

First Ready to play version - detached from waves-core

CHAPTER 8

Source Documentation

8.1 Waves Demo classes

WAVES Demo extends base Waves-Core functionality

See also:

Waves-Core Documentation http://waves-core.readthedocs.io/en/latest/modules/source.html

8.1.1 Service override example

class demo.models.**DemoWavesService**(*id*, *created*, *updated*, *description*, *short_description*, *api_name*, *binary_file*, *runner*, *name*, *authors*, *citations*, *version*, *status*, *email_on*, *partial*, *created_by*, *remote_service_id*, *edam_topics*, *edam_operations*, *category*, *to_delete*)

Parameters

- **id** – Id
- **created** – Creation timestamp
- **updated** – Last update timestamp
- **description** – Description (HTML)
- **short_description** – Short description (Text)
- **api_name** – App short code, used in url, leave blank for automatic setup
- **binary_file** (ForeignKey to waves.wcore.models.binaries.ServiceBinaryFile) – If set, 'Execution parameter' param line:'command' will be ignored
- **runner** (ForeignKey to waves.wcore.models.runners.Runner) – Service job runs configuration

- **name** – Service displayed name
- **authors** – Tools authors
- **citations** – Citation link (Bibtex format)
- **version** – Service displayed version
- **status** – Service online status
- **email_on** – This service sends notification email
- **partial** – Set whether some service outputs are dynamic (not known in advance)
- **created_by** (ForeignKey to `authtools.models.User`) – Created by
- **remote_service_id** – Remote service tool id
- **edam_topics** – Comma separated list of Edam ontology topics
- **edam_operations** – Comma separated list of Edam ontology operations
- **category** (ForeignKey to `demo.models.ServiceCategory`) – Category
- **to_delete** – Automatic deletion
- **restricted_client** (ManyToManyField to `authtools.models.User`) – Public access is granted to everyone, If status is 'Restricted' you may restrict access to specific users here.
- **adaptor_params** (*GenericRelation*) – Adaptor params

submissions
ManyToOneRel to *demo.models.DemoWavesSubmission*

metas
ManyToOneRel to `demo.models.ServiceMeta`

8.1.2 Submission override example

class demo.models.**DemoWavesSubmission**(*id*, *created*, *updated*, *order*, *slug*, *api_name*, *binary_file*, *runner*, *service*, *availability*, *name*, *to_delete*)

Parameters

- **id** – Id
- **created** – Creation timestamp
- **updated** – Last update timestamp
- **order** – Order
- **slug** – Slug
- **api_name** – App short code, used in url, leave blank for automatic setup
- **binary_file** (ForeignKey to `waves.wcore.models.binaries.ServiceBinaryFile`) – If set, 'Execution parameter' param line:'command' will be ignored
- **runner** (ForeignKey to `waves.wcore.models.runners.Runner`) – Service job runs configuration
- **service** (ForeignKey to *demo.models.DemoWavesService*) – Service

- **availability** – Availability
- **name** – Label
- **to_delete** – Automatic deletion
- **adaptor_params** (*GenericRelation*) – Adaptor params

outputs
ManyToOneRel to `waves.wcore.models.services.SubmissionOutput`

exit_codes
ManyToOneRel to `waves.wcore.models.services.SubmissionExitCode`

submission_groups
ManyToOneRel to `waves.wcore.models.inputs.RepeatedGroup`

inputs
ManyToOneRel to `waves.wcore.models.inputs.AParam`

service_jobs
ManyToOneRel to `waves.wcore.models.jobs.Job`

8.1.3 Adapters override example

Demo adapters are not meant to execute anything for real, but are intended to demonstrate how to configure them in backoffice So every standard Wcore adapter is overriden in order to mock their execution on Demo platform

class demo.adaptors.**WavesDemoAdaptor**(*command=None*, *protocol=u'http'*, *host=u'localhost'*, ***kwargs*)

get_command_line(*obj*)
Retrieve command line normally executed on remote platform

_connect()
Fake connection when running jobs.

__init__(*command=None*, *protocol=u'http'*, *host=u'localhost'*, ***kwargs*)
Force command and add a job member

_job_status(*job*)
Mocking job status

_run_job(*job*)
Mocking job launch

_prepare_job(*job*)
Mocking job preparation

_job_run_details(*job*)
Mocking job run details retrieval

_job_results(*job*)
Mocking job results, add basic command line to standard output, Randomly set stderr output to see warnings happen

class demo.adaptors.**GalaxyJobAdaptor**(*command=None*, *protocol=u'http'*, *host=u'localhost'*, *port=u''*, *api_base_path=u''*, *api_endpoint=u''*, *app_key=None*, *library_dir=u''*, ***kwargs*)

_prepare_job(*job*)
Mocking job remote preparation

class demo.adaptors.**GalaxyJobAdaptor**(*command=None*, *protocol=u'http'*, *host=u'localhost'*, *port=u''*, *api_base_path=u''*, *api_endpoint=u''*, *app_key=None*, *library_dir=u''*, ***kwargs*)

_prepare_job(*job*)
Mocking job remote preparation

See also:

WAVES CORE documentation : http://waves-core.readthedocs.io/en/latest/

CHAPTER 9

Indices and tables

- genindex
- modindex
- search

Index

Symbols

D

E

G

I

M

O

S

W