pyrs Documentation

Release 0.4.0

Csaba Palankai

August 31, 2015

Contents

CHAPTER 1

Contents

1.1 Python MicroService framework

Project homepage: https://github.com/palankai/pyrs
Documentation: http://pyrs.readthedocs.org/
Issues: https://github.com/palankai/pyrs/issues

1.1.1 What is this package for

This package basically an umbrella for other packages. Ensure basic helpers for them like configuration, context, decorators, etc.

1.1.2 The ecosystem

Schema validation, serialization

With this module you can define schemas then you can validate and serialize your data.

Project homepage: https://github.com/palankai/pyrs-schema
Documentation: http://pyrs-schema.readthedocs.org/

Restful web framework

This module ensure a routing and dispatching through defined resources. Baased on schema you can define interface which would be validate. It's an independent solution could work with Flask, Django or even Odoo.

Project homepage: https://github.com/palankai/pyrs-resource
Documentation: http://pyrs-resource.readthedocs.org/

Swagger builder

If you know Swagger then you know in python world there is not that easy to use that. You have to make your documentation by hand. This package aim to make it much more easier.

Project homepage: https://github.com/palankai/pyrs-swagger
Documentation: http://pyrs-swagger.readthedocs.org/

1.1.3 Installation

```
$ pip install pyrs
```

Note: You have to install the other packages independently.

1.1.4 Extending the exist functionality

The *pyrs.ext* and the *pyrs* itself are namespace packages. If you would like to extend an exist functionality of a package check that given package documentation for further information.

If you would like to implement a new funcionality, just create a new package inside *pyrs* package. Make sure the *pyrs* package still remain namespace package.

1.1.5 Dependencies

See requirements.txt. But The goal is less dependency as possible. Aim to work with python 2.7, 3.3, 3.4+ (tested agains them).

1.1.6 Important caveats

This code is in beta version. I working hard on write stable as possible API in the first place but while this code in 0.x version you should expect some major modification on the API.

1.1.7 Contribution

I really welcome any comments! I would be happy if you fork my code or create pull requests. I've already really strong opinions what I want to achieve and how, though any help would be welcomed.

Feel free drop a message to me!

1.2 Default configuration

This module contains the global configurations of the framework

`pyrs.conf.`**`meta_field = '_pyrsmeta'`**
This name suppose to be used for general meta decoration for functions, methods or even classes

1.3 Env module

Give environment and tools for the packages based on pyrs.

1.3.1 Annotations

pyrs.env.annotations.**annotate**(*_func=None*, ***kwargs*)

pyrs.env.annotations.**ensure_meta**(*func*, **args*, ***kwargs*)

pyrs.env.annotations.**get_meta**(*func*, *name=None*, *default=None*)

1.3.2 Configuration module

class pyrs.env.configuration.**Configuration**(**bases*)

Bases: object

get(*name*, *default=None*)

items()

keys()

upgrade(*base*)

1.3.3 Context module

Context usage:

```
ctx = Context(value=1)
with ctx:
    call_a_function(ctx=ctx)
```

A bit more readable version:

```
ctx = Context(value=1)
with ctx.copy(value=2, new_value=3) as c:
    call_a_function(ctx=c)
```

class pyrs.env.context.**Context**(**args*, ***kwargs*)

Bases: thread._local

This is a thread safe stack based implementation of context.

_pop()

_push()

clear(**args*, ***kwargs*)

copy(**args*, ***kwargs*)

get(*name*, *default=None*)

items()

keys()

update(**args*, ***kwargs*)

pyrs.env.context.**ctx = <pyrs.env.context.Context object>**
pyrs context

1.4 Code styling

This document would describe how should look like any part of this framework. I tend to update this document frequently, specify it as precise as possible.

1.4.1 Language

The language of the comments, documentation, variable names, function names have to be English. Use syntax and spell checker against any of them. (It's a bit controversial, because I made lots of spelling mistakes, but I welcome any improvements.)

1.4.2 PEP8

Every python code should be checked with some kind of pep8 checking tool. Please enable all validation (by default some of the disabled). None of the PEP8 errors or warning allowed in release version.

The following thumb rules are PEP8 conform.

1.4.3 Use single quote

Use single quote if it's possible. Double quote only allowed if you can avoid escaping with it. If you have escaping both ways you should use single quote.

1.4.4 Importing

There should be alphabetically sorted import sections. The sections are:

- Future
- Python Standard Library
- Third Party
- Current Python Project
- Explicitly Local (from . import x)

You should import modules and packages rather than classes or functions if it could cause a conflict use `as` to make an alias. (In some cases importing a class or a function allowed but try to avoid)

```
from __future__ import absolute_import

import json
import os

import psycopg2
import werkzeug

from pyrs.swagger import tests
```

```
from . import base
from . import conf
```

1.5 Contribution

There are severe ways how can you help to make this work better. I welcome any kind of contribution. Please read the following chapters about how can you raise an issue, ask a new feature or actively improve the code.

1.5.1 Reporting a bug

You can report bugs or request any new feature on the official issue tracking of this project.

If you open an issue please give me detailed information about the bug. First of all I need to know which environment where you are using it. The most important think what I have to know which version what are you use. If it's the master branch, please mention the commit hash. Second important think is the python version. Please describe how can I reproduce, give me a small example or a gist. If you can attach traceback would be really appreciated. Would help if you send me the `pip freeze` output and maybe the operating system (I hope the different OS cannot cause any problem).

Milestones

Please select the milestone when you open an issue related to your version the code. If you couldn't find that version (for example you are using 0.3.1 then you can find only 0.4, 0.5, 1.0) means that version is not supported. In that case I suppose you should test it against the recent version and report it then if it's still an issue.

I'm planning to fix any real reported issue.

If you would like to open a feature request, please leave empty the milestone. I'll will decide which version would contain this improvement.

Keep in mind, this project has it's goals. I'm working on this project just in my spare time, so I cannot promise any work on any feature request.

Issue labels

I use almost the basic given configuration of github. Please follow that way. The bug should be marked as bug. A feature request should be enhancement.

1.5.2 Create pull request

I hope soon somebody help me to reach the project goals. If you would like to help me work on this project, please contact me first, we should agree about the planned work and how we should work together. I'll review any pull request, though.

Any kind pull request, improvement should follow the coding standard of this framework. I plan to reach the %100 test coverage against the whole project. I'm using PEP8 validation against the whole code. Just tested and PEP8 conform code can be released.

Please see the Code styling documentation.

1.6 License

1.7 Indices and tables

- genindex
- modindex
- search

p

Symbols

A

C

E

G

I

K

M

P

U