
wsgisubdomain Documentation

Release 0.1

Steve Leonard

October 08, 2013

CONTENTS

1	Installation	3
2	Example	5
3	API	7
4	Subdomain Dispatcher	9
	Python Module Index	11

wsgisubdomain is a WSGI application dispatcher. A WSGI application is fetched based on a request's target subdomain, creating one if needed.

wsgisubdomain supports Python 2.6, 2.7, 3.2, 3.3 and PyPy.

The code in this module is adapted from: <http://flask.pocoo.org/docs/patterns/appdispatch/#dispatch-by-subdomain>

INSTALLATION

```
$ pip install wsgisubdomain
```


EXAMPLE

```
# app.wsgi
from wsgisubdomain import SubdomainDispatcher
from app import create_application
application = SubdomainDispatcher(create_application)
```


API

SUBDOMAIN DISPATCHER

class `wsgisubdomain.SubdomainDispatcher` (*create_application*)

A WSGI application that gets or creates other WSGI applications based on the subdomain.

Adapted from: <http://flask.pocoo.org/docs/patterns/appdispatch/#dispatch-by-subdomain>

Parameters `create_application` – A function that accepts ‘subdomain’ as a keyword argument and returns a WSGI application. Subdomain will be either an empty string for the bare domain, *None* if the request is for an IP address, or a full subdomain (e.g. ‘www’ or ‘en.dl’)

`__call__` (*environ*, *start_response*)

WSGI application interface

Parameters

- **environ** – WSGI environ
- **start_response** – WSGI start_response

`__delattr__`

`x.__delattr__('name')` <==> `del x.name`

`__format__` ()

default object formatter

`__getattr__`

`x.__getattr__('name')` <==> `x.name`

`__hash__`

`x.__hash__()` <==> `hash(x)`

static `__new__` (*S*, ...) → a new object with type *S*, a subtype of *T*

`__reduce__` ()

helper for pickle

`__reduce_ex__` ()

helper for pickle

`__repr__`

`x.__repr__()` <==> `repr(x)`

`__setattr__`

`x.__setattr__('name', value)` <==> `x.name = value`

`__sizeof__` () → int

size of object in memory, in bytes

__str__

x.__str__() <==> str(x)

static __subclasshook__()

Abstract classes can override this to customize issubclass().

This is invoked early on by abc.ABCMeta.__subclasscheck__(). It should return True, False or NotImplemented. If it returns NotImplemented, the normal algorithm is used. Otherwise, it overrides the normal algorithm (and the outcome is cached).

__weakref__

list of weak references to the object (if defined)

get_application(*environ*)

Retrieve an application for a wsgi environ

Parameters **environ** – The environ object sent by wsgi to an application

PYTHON MODULE INDEX

W

wsgisubdomain, 1