

Django中国社区 » 翻译

[翻译]python的logging模块配置文件的格式

By Django中国社区 at 2013-05-06 07:17



翻译原文: <http://docs.python.org/2/library/logging.config.html>

配置文件的格式是由fileConfig()函数负责解析的,这个函数基于ConfigParser.

该文件中必须包含一下几段: [loggers], [handlers] 和 [formatters],用来标识文件定义的几种类型的实体.对每个这样的实体,都有一个特定的段来标识这个实体是如何配置的.比如说, [loggers] 段里面定义了一个叫log01的logger,那么关于这个logger的详细配置信息应该写在 [logger_loger01] 里面.同样的,如果 [handlers] 里面定义了一个名为hand01的handler ,那么它的详细配置信息应该放在 [handler_hand01] 里面,而 [formatters] 里面定义的form01就应该用一个名为 [formatter_form01] 的段来定义它的详细信息.对root logger来说,他的定义信息则必须放在名为 [logger_root] 的端中.

下面是一个关于配置文件中这几个段的一个示例::

```
[loggers]
keys=root,log02,log03,log04,log05,log06,log07
```

```
[handlers]
keys=hand01,hand02,hand03,hand04,hand05,hand06,hand07,hand08,hand09
```

```
[formatters]
keys=form01,form02,form03,form04,form05,form06,form07,form08,form09
```

root logger必须包含一个标识logging等级的level和一个包含handlers的列表.下面给出一个root logger的示例 ::

```
[logger_root]
level=NOTSET
handlers=hand01
```

level可以是 DEBUG, INFO, WARNING, ERROR, CRITICAL 或 NOTSET 中的一个. NOTSET 标识用在root logger时,也只有用在root logger时,表示所有的信息都会被记录.Level变量会在logging包命名空间的上下文中被赋值.

handlers项可以接受一个由逗号分隔的handler名字列表,这些个handler的名字必须出现在 [handlers] 段中,而且配置文件中还需要有相关的详细描述这些handler的字段.

对于除root logger外的loggers 来说,有一些额外的信息需要提供.下面是一个这方面的例子::

```
[logger_parser]
level=DEBUG
handlers=hand01
propagate=1
qualname=compiler.parser
```

level和handlers项跟root logger的完全一样,如果一个非root logger的level项定义为 NOTSET ,系统会根据该logger的继承关系向上寻找一个有效的level值来作为该logger的level值. propagete 项如果为1,则表示该logger必须处理它继承的父类logger的log信息;如果为0则不必. qualname 项表示logger的分层结构的通道名,应用程序中就是用这个名字来获得这个logger的.

Django中国社区
专注于Django中国技术发展的社区

现在注册

或者 登录

或者 用微博帐号登录

sov5.com

Sov5问答搜索

知乎、statckoverflow全都有

了解更多 ▶

标签



相关主题推荐

能所之迹 «Django Essentials»

django 中文文档协作翻译寻求翻译小能手

能所之迹 翻译中 «Web Development with Django Cookbook»

[翻译]自己动手写Django app[第二部分], (Writing your first Django app, part 2)

[翻译]Python Logging HOWTO

下面是一个定义**handler**的示例::

```
[handler_hand01]
class=StreamHandler
level=NOTSET
formatter=form01
args=(sys.stdout,)
```

class 项定义了**handlers**的类(通过**eval()**函数在**logging**包的命名空间中实例化).**level**项被用做**logger**的**logging**等级信息, **NOTSET** 表示记录所有的**log**信息.

formatter 项定义用于该**handler**的**formatter**,如果为空则用默认的**formatter**(**logging._defaultFormatter**).如果不为空,那么相应的**formatter**名字也应该出现在**[formatters]** 段中,并且应该有相应的定义段.

args 项中定义的是用于**handler**类实例化时传给构造函数的参数.查阅相应的**handler**的构造函数,或者参考以下示例来看一下典型的配置项::

```
[handler_hand02]
class=FileHandler
level=DEBUG
formatter=form02
args=('python.log', 'w')

[handler_hand03]
class=handlers.SocketHandler
level=INFO
formatter=form03
args=('localhost', handlers.DEFAULT_TCP_LOGGING_PORT)

[handler_hand04]
class=handlers.DatagramHandler
level=INFO
formatter=form04
args=()
```

formatter 项用到的**formatter**定义如下::

```
[formatter_form01]
format=F1 %(asctime)s %(levelname)s %(message)s
datefmt=
class=logging.Formatter
```

format 项定义了整个的格式字符串, **datefmt** 项可以接受跟**strftime()**函数兼容的时间/日期格式字符串,如果为空,则默认用**ISO8601**格式的日期/时间格式,**ISO8601**格式也定义了微秒,可以用一个逗号隔开添加到上面的格式字符串之后,**ISO8601**格式的一个示例是 **2003-01-23 00:29:50,411**.

class 项是可选的,它指出了 **formatter** 的类名(模块名和类名通过点来分隔),这个选项在需要一个**Formatter**子类时很有用,**Formatter**的子类可以通过一种可伸缩的格式来展现更多的异常信息.

注解: 笔者的一个完整的例子可见如下 (**ptailfile**的源码) :

```
[loggers]
keys=root, tailfile, tt2-client

[handlers]
keys=consoleHandler, fileHandler

[formatters]
keys=formatter01, formatter02, formatter03, formatter04
```

	翻译中 《Django设计模式与最佳实践》
	[翻译]自己动手写Django app[第一部分], (Writing your first Django app, part 1)
	[翻译]python的logging模块配置文件的格式
	[翻译]Python打包工具
	自己动手写django app

社区所有版块导航	
Django	Django Python DjangoApp pycharm
DATA	docker Elasticsearch
分享	问与答 闲聊 招聘 翻译 创业 分享发现 分享创造 求职
WEB开发	MongoDB Redis DATABASE NGINX 其他Web框架 web工具 zookeeper tornado NoSql Bootstrap js peewee Git bottle IE MQ
Django-China.CN	反馈 公告 社区推广

```
keys=richFormatter

[logger_root]
level=DEBUG
handlers=consoleHandler

[logger_tailfile]
```

Last edited by Django中国社区 (2013-05-11 17:08)

0

4689 次点击

文章 [1] | 最新文章 2013-05-06 11:21



paomian 2013-05-06 11:21

1楼

nice

关于

专注于Django中国技术发展的社区

Copyright © 2013

