



Django Python exploration

Wednesday, June 22, 2011

Django: Logging Quickstart in Django 1.3

Logging is always one of those things that seems like there is a lot of boilerplate code that I tend to always have to lookup online each time. Though logging is very customizable, most of the time I need simply to only need log to a file or files. Therefore, I'm writing this in hopes of compiling a simple configuration that can be dropped into a project. My understanding is still rudimentary so feel free to drop a comment.

A logging framework is built into Django 1.3 which makes it easier to integrate into a given project. The logging framework contains three main components: formatters, handlers, and loggers. Formatters determine how the output will be displayed when it is logged. Handlers handle where output is logged, whether that be a file, the console, or an email. Loggers are associated with handlers and are the objects that one interacts with when logging something. As in the example below, more than one formatter, handler or logger can be defined in a project.

In settings.py:

```
import os

# choose a path that is your virtual environment root
ENV_ROOT = os.path.join('/', 'web', 'myvenv')

LOGGING = {
    'version': 1,
    'disable_existing_loggers': True,
    'formatters': {
        'verbose': {
            'format': '%(levelname)s %(asctime)s %(module)s %(process)d %(thread)d %(message)s'
        },
        'simple': {
            'format': '%(levelname)s %(message)s'
        },
    },
    'handlers': {
        'file_userlogins': {
            # define and name a handler
            'level': 'DEBUG',
            'class': 'logging.FileHandler', # set the logging class to log to a file
            'formatter': 'verbose',        # define the formatter to associate
            'filename': os.path.join(ENV_ROOT, 'log', 'userlogins.log') # log file
        },
        'file_usersaves': {
            # define and name a second handler
            'level': 'DEBUG',
            'class': 'logging.FileHandler', # set the logging class to log to a file
            'formatter': 'verbose',        # define the formatter to associate
            'filename': os.path.join(ENV_ROOT, 'log', 'usersaves.log') # log file
        },
    },
    'loggers': {
        'logview.userlogins': {
            # define a logger - give it a name
            'handlers': ['file_userlogins'], # specify what handler to associate
            'level': 'INFO',                 # specify the logging level
            'propagate': True,
        },
        'logview.usersaves': {
            # define another logger

```

Blog Archive

- 2015 (1)
- ▼ 2011 (12)
 - July (2)
 - ▼ June (2)
 - [Django: Logging Quickstart in Django 1.3](#)
 - [Django: Executing manage.py from Anywhere](#)
 - May (1)
 - April (3)
 - March (4)

About Me

Joe Jasinski[View my complete profile](#)

```
'handlers': ['file_usersaves'], # associate a different handler
'level': 'INFO',               # specify the logging level
'propagate': True,
},
}
```

In application code such as in views.py:

```
import logging # import the required logging module
logger_logins = logging.getLogger('logview.userlogins') # logger from settings.py
logger_logins.info('Log info')
logger_logins.debug('Log Debug information %s' % ("can pass in variables"))

# we can log to a different file using the other logger
logger_saves = logging.getLogger('logview.usersaves')
logger_saves.error('log an error')
```

Posted by Joe Jasinski at 12:50 AM

 +4 Recommend this on Google

Labels: [Django](#), [logging](#), [Python](#), [quickstart](#)

2 comments:



reinout June 25, 2011 at 5:06 PM

What I've started doing in every file is a simple "logger = logging.getLogger(__name__)". That __name__ translates into the module name (so something like "yourproject.yourmodule"). Consistent naming everywhere!

I personally haven't had any use for separate loggers in one file, yet.

[Reply](#)



Joe Jasinski June 26, 2011 at 10:01 AM

Thanks Reinout. That seems like a nice convention.

[Reply](#)

Enter your comment...

Comment as: Unknown (Goc ▾)

[Sign out](#)

[Publish](#)

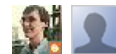
[Preview](#)

☐ Notify me

Join this site

with Google Friend Connect

Members (2)



Already a member? [Sign in](#)

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)

Powered by [Blogger](#).