

PYTHON ДЛЯ ИЗУЧЕНИЯ ПРОГРАММИРОВАНИЯ

Элисбаров Сергей Спартакovich

*Студент ФГБОУ ВО «Армавирский государственный педагогический
университет», г. Армавир*

*Научный руководитель, к.т.н., доцент кафедры
информатики и ИТО Давиденко А.Н.*

PYTHON TO STUDY PROGRAMMING

Elisbarov Sergei Spartakovich

Student of Armavir state pedagogical University, Armavir

АННОТАЦИЯ

В этом тексте мы рассмотрим некоторые передовые методы настройки стандартной библиотеки ведения журналов Python для создания контекстно-насыщенных журналов, отслеживания исключений и маршрутизации журналов в соответствующие места назначения.

ABSTRACT

In this text, we'll go over some best practices for configuring the Python standard logging library to generate context-rich logs, track exceptions, and route the logs to appropriate destinations.

Ключевые слова: Python; JSON; basicConfig; Приложения.

Keywords: Python; JSON; basicConfig; App.

Встроенный в Python модуль ведения журнала разработан, чтобы дать вам критически важную информацию о ваших приложениях с минимальной настройкой. Если вы только начинаете или уже используете модуль ведения журнала Python, это руководство покажет вам, как настроить этот модуль для регистрации всех необходимых данных, направления их в желаемые пункты назначения и централизации

журналов для более глубокого понимания вашего Python. Приложения. В этом посте мы покажем вам, как:

- Настройте уровень приоритета и место назначения ваших журналов
- Настройте индивидуальную настройку, включающую несколько регистраторов и мест назначения
- Включите обработку исключений и трассировку в свои журналы
- Отформатируйте свои журналы в JSON и централизуйте их для более эффективного устранения неполадок.

Основы модуля журналирования Python

Logging Модуль входит в стандартной библиотеке языка Python, который означает, что вы можете начать использовать его без установки. Метод модуля регистрации `basicConfig()` - это самый быстрый способ настроить желаемое поведение вашего регистратора. Однако документация Python рекомендует создать регистратор для каждого модуля в вашем приложении - и может быть сложно настроить установку регистратора для каждого модуля, используя его `basicConfig()` отдельно. Поэтому большинство приложений (включая веб-фреймворки, такие как Django) вместо этого автоматически используют конфигурацию журналирования на основе файлов или словарей. Если вы хотите начать работу с одним из этих методов, мы рекомендуем сразу перейти к этому разделу.

Три основных параметра `basicConfig()`:

- **уровень**: минимальный уровень приоритета сообщений для регистрации. Доступные уровни журнала в порядке возрастания серьезности: **ОТЛАДКА**, **ИНФОРМАЦИЯ**, **ПРЕДУПРЕЖДЕНИЕ**, **ОШИБКА** и **КРИТИЧЕСКИЙ**. По умолчанию установлен уровень

WARNING, что означает, что модуль ведения журнала Python будет отфильтровывать любые сообщения DEBUG или INFO.

- обработчик: определяет, куда направлять ваши журналы. Если не указано иное, библиотека ведения журнала будет использовать StreamHandler для направления сообщений журнала sys.stderr(обычно на консоль).

- формат: По умолчанию библиотека протоколирования будет регистрировать сообщения в следующем формате: <LEVEL>:<LOGGER_NAME>:<MESSAGE>. В следующем разделе мы покажем вам, как настроить это, чтобы включить временные метки и другую информацию, полезную для устранения неполадок.

Поскольку модуль регистрации по умолчанию регистрирует только WARNING и журналы более высокого уровня, вам может не хватать видимости журналов с более низким приоритетом, которые могут быть полезны для проведения анализа первопричин. Модуль ведения журнала также передает журналы в консоль вместо того, чтобы добавлять их в файл. Вместо того, чтобы использовать a StreamHandler или SocketHandler для потоковой передачи журналов непосредственно на консоль или во внешнюю службу по сети, вы должны использовать a FileHandler для входа в один или несколько файлов на диске.

Одним из основных преимуществ ведения журнала в файл является то, что вашему приложению не нужно учитывать возможность возникновения сетевых ошибок при потоковой передаче журналов во внешнее место назначения. Если возникнут какие-либо проблемы с потоковой передачей журналов по сети, вы не потеряете доступ к этим журналам, поскольку они будут храниться локально на каждом сервере. Ведение журнала в файл также позволяет вам создать более индивидуальную настройку ведения журнала, в которой вы можете

направлять различные типы журналов в отдельные файлы, а также отслеживать и централизовать эти файлы с помощью службы мониторинга журналов.

В следующем разделе мы покажем вам, насколько легко настроить `basicConfig()` регистрацию сообщений с более низким приоритетом и их направление в файл на диске.

Пример `basicConfig()`

В следующем примере используется `basicConfig()` для настройки приложения для записи в журнал `DEBUG` и сообщений более высокого уровня в файл на диске (`myapp.log`). Он также указывает, что журналы должны соответствовать формату, который включает отметку времени и уровень важности журнала:

```
import logging

def word_count(myfile):

    logging.basicConfig(level=logging.DEBUG, filename='myapp.log', format='%(asctime)s %(levelname)s: %(message)s')
    try:
        # count the number of words in a file and log the result
        with open(myfile, 'r') as f:
            file_data = f.read()
            words = file_data.split(" ")
            num_words = len(words)
            logging.debug("this file has %d words", num_words)
            return num_words
    except OSError as e:
        logging.error("error reading the file")
```

Рисунок 1. Логирование в Python

Список литературы:

1. Буйначев, С. К. Основы программирования на языке Python : учебное пособие / С. К. Буйначев, Н. Ю. Боклаг ; под редакцией Ю. В. Песин. — Екатеринбург : Уральский федеральный университет, ЭБС АСВ, 2014. — 92 с. — ISBN 978-5-7996-1198-9. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/66183.html>
2. Рик, Гаско Простой Python просто с нуля / ГаскоРик. — Москва : СОЛОН-Пресс, 2019. — 256 с. — ISBN 978-5-91359-334-4. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/94940.html>
3. Шелудько, В. М. Основы программирования на языке высокого уровня Python : учебное пособие / В. М. Шелудько. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2017. — 146 с. — ISBN 978-5-9275-2649-9. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/87461.html>