**Краткий отчёт по интеграции логгера в приложение на Python**

**1. Описание экосистемы**

Для выполнения задания было выбрано простое приложение на Python, использующее Flask как веб-фреймворк. Flask – это легковесный фреймворк для создания веб-приложений, который поддерживает различные расширения, включая логирование.

**2. Выбор логгера**

Для интеграции логгирования была выбрана библиотека logging, встроенная в стандартную библиотеку Python. Она позволяет удобно настраивать уровни логирования и форматы сообщений. Также выбрана отправка логов через UDP с использованием библиотеки socket.

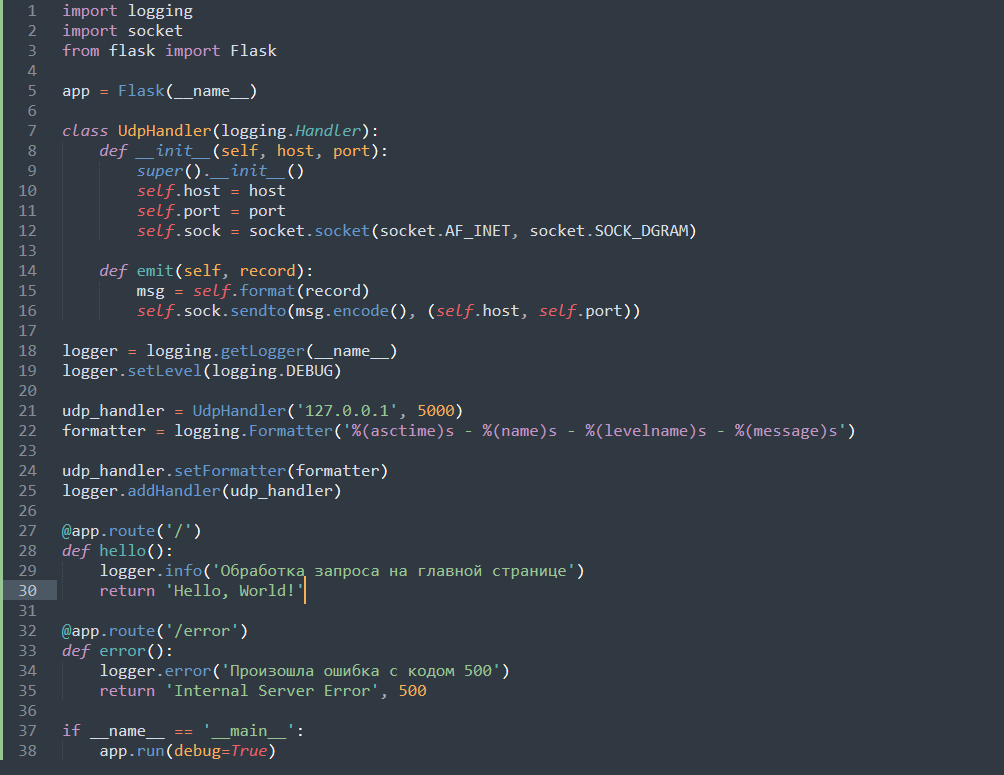
**3. Обоснование выбора логгера**

Выбор logging обоснован следующими факторами:

* Удобство использования: библиотека уже встроена в Python, что исключает необходимость установки дополнительных пакетов.
* Гибкость настройки: позволяет легко управлять уровнями логирования и форматами сообщений.
* Поддержка различных обработчиков: возможность отправки логов на удалённые сервера.

**4. Реализация**

Пример реализации логгирования в приложении Flask с отправкой логов по UDP:



**5. Уровни логирования**

В примере используются различные уровни:

*info: для логирования обычных событий (например, запросы).*

*error: для фиксирования ошибок.*

**6. Результаты работы**

При запуске приложения и обращении к URL, соответствующие логи отправляются на указанный адрес в локальной сети (UDP). Ожидаются сообщения в следующем формате:

*2023-09-13 14:30:00 - \_\_main\_\_ - INFO - Обработка запроса на главной странице*

*2023-09-13 14:30:01 - \_\_main\_\_ - ERROR - Произошла ошибка с кодом 500*

***Заключение***

*Интеграция логгирования в приложение на Python с использованием простой библиотеки logging и отправка логов по UDP позволяет существенно упростить процесс отладки и мониторинга. Такой подход делает приложения более надежными и удобными в обслуживании.*