4.8 Логирование

Логирование

В этом уроке мы разберемся, как контролировать уровень логов, генерируемых библиотекой Flet и ее базовыми компонентами. Возможно, вам потребуется включить детальное логирование для устранения проблем с библиотекой Flet или при отправке новой проблемы в Flet (Flet issue).

# Python

Модули Flet на Python предоставляют именованные регистраторы:

flet\_core

и flet .

Для включения детального / подробного Flet логирования в вашей программе добавьте этот код перед вызовом ft.app() :

import logging

logging.basicConfig(level=logging.DEBUG)

Это позволит включить логгеры во всех модулях Flet ( flet\_core и flet ).

Для уменьшения многословности логов, вы можете подавить логи из модуля flet\_core , добавив:

logging.getLogger("flet\_core").setLevel(logging.INFO)

Отладочное логирование обычно требуется для целей устранения проблем, при отправке новой проблемы в Flet (Flet issue).

В большинстве случаев вам будет достаточно уровня логирования INFO .

# Сервер Fletd

Вы можете контролировать логирование Fletd - встроенного веб-сервера Flet.

Когда уровень логгера Python Flet настроен, его значение неявно передается на сервер Fletd, и вы можете видеть его логи в выводе.

Тем не менее, вы также можете использовать переменную

среды Fletd:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | FLET\_LOG\_LEVEL | | | | |  |
| debug | | , | info | , | warning | |

для переопределения уровня логов

, debug , panic , fatal .

Кроме того, чтобы перенаправить вывод логов Flet в файл, вы можете установить

переменную среды в значение true . Логи Flet будут

FLET\_LOG\_TO\_FILE

сохраняться в /tmp/flet-server.log на macOS и Linux и в на Windows.

%TEMP%\flet-

server.log