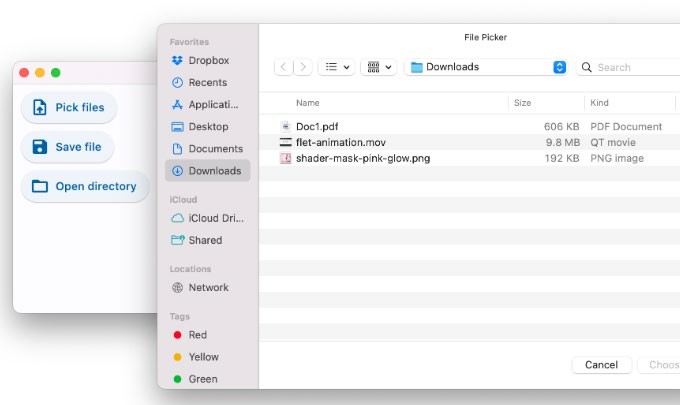
1.9 Выбор и загрузка файлов

Выбор и загрузка файлов

Контроллер выбора файлов открывает родное диалоговое окно ОС для выбора файлов и директорий.

Он работает на всех платформах: Web, macOS, Windows, Linux, iOS и Android.



Выбор файлов позволяет открывать три диалога:

* **Выбор файлов** - один или несколько, любые файлы или только определенные типы.
* **Сохранить файл** - выбрать директорию и имя файла.
* **Получить директорию** - выбрать директорию.

При запуске приложения Flet в браузере доступен только вариант "Выбор файлов", и он используется только для загрузки, так как, очевидно, не возвращает полный путь к выбранному файлу.

Где сборщик файлов действительно блистает, так это в десктопном варианте! Все три диалога возвращают полные пути к выбранным файлам и каталогам - отличная помощь вашим пользователям!

# Использование выбора файлов в вашем приложении

Рекомендуется добавить выбор файлов в коллекцию page.overlay.controls , чтобы это не влияло на макет вашего приложения. Несмотря на то, что выбор

файлов имеет размер , он все еще считается элементом управления при

0x0

помещении в или Column .

Row

import flet as ft

file\_picker = ft.FilePicker()

page.overlay.append(file\_picker)

page.update()

Чтобы открыть диалог выбора файлов, вызовите один из трех методов:

|  |  |  |
| --- | --- | --- |
| pick\_files() | |  |
| save\_file() |  | |
| get\_directory\_path() | | |

Для этого хорошо подходит лямбда:

ft.ElevatedButton("Выбрать файлы...",

on\_click=lambda \_: file\_picker.pick\_files(allow\_multiple=True))

Когда диалог закрывается, вызывается обработчик

событий FilePicker.on\_resul t, объект события которого имеет одно из следующих свойств:

- только диалог "Выбор файлов", список выбранных файлов или None, если диалог был отменен.

files

- диалоги "Сохранить файл" и "Получить директорию", полный путь к файлу или директории или None , если диалог был отменен.

path

import flet as ft

def on\_dialog\_result(e: ft.FilePickerResultEvent):

print("Выбранные файлы:", e.files)

print("Выбранный файл или директория:", e.path)

file\_picker = ft.FilePicker(on\_result=on\_dialog\_result)

Последний результат всегда доступен в свойстве FilePicker.result .

# Загрузка файлов

*File picker* имеет встроенные возможности загрузки, которые работают на всех платформах и в Интернете.

Для загрузки одного или нескольких файлов вы должны сначала

вызвать FilePicker.pick\_files() . При выборе пользователем файлов они никуда автоматически не загружаются, а вместо этого их ссылки сохраняются в состоянии file picker.

Чтобы выполнить фактическую загрузку, вы должны вызвать

метод и передать список файлов, которые необходимо

FilePicker.upload()

загрузить, вместе с их URL для загрузки и методом загрузки ( PUT или POST ):

import flet as ft

def upload\_files(e):

upload\_list = []

if file\_picker.result != None and file\_picker.result.files !=

None:

for f in file\_picker.result.files:

upload\_list.append(

FilePickerUploadFile(

f.name,

upload\_url=page.get\_upload\_url(f.name, 600),

)

)

file\_picker.upload(upload\_list)

ft.ElevatedButton("Загрузить", on\_click=upload\_files)

*ПРИМЕЧАНИЕ*

*Если вам нужно разделить загрузки для каждого пользователя, вы можете указать имя файла, предварительно указав любое количество директорий в*

*вызове page.get\_upload\_url() , например:*

upload\_url = page.get\_upload\_url(f"/{username}/pictures/{f.name}",

600)

*Директории будут автоматически созданы*

*внутри upload\_dir , если их не существует.*

*/{username}/pictures*

# Хранилище загрузок

Обратите внимание на использование метода

- он

генерирует заранее назначенный URL-адрес загрузки для внутреннего хранилища загрузки Flet.

page.get\_upload\_url()

*ИСПОЛЬЗОВАНИЕ ЛЮБОГО ХРАНИЛИЩА ДЛЯ ЗАГРУЗКИ ФАЙЛОВ*

*Сгенерировать заранее назначенный URL для хранилища AWS S3 можно с помощью библиотеки boto3.*

*Аналогичная техника должна работать для Wasabi, Backblaze, MinIO и любых других провайдеров, имеющих S3-совместимый API.*

Чтобы позволить Flet сохранять загруженные файлы в директорию, укажите полный или относительный путь к этой директории при вызове flet.app() :

ft.app(target=main, upload\_dir="uploads")

Вы даже можете поместить загрузки внутрь директории "assets", чтобы загруженные файлы например, изображения, документы или другие медиафайлы, были доступны непосредственно из клиента Flet:

ft.app(target=main, assets\_dir="assets", upload\_dir="assets/uploads")

и где-то в вашем приложении вы можете отобразить загруженное изображение с помощью:

page.add(ft.Image(src="/uploads/<some-uploaded-picture.png>"))

# Прогресс загрузки

Как только вызывается метод FilePicker.upload() , клиент Flet асинхронно начинает загрузку выбранных файлов по одному и сообщает о прогрессе через обратный вызов FilePicker.on\_upload .

FilePickerUploadEvent

Объект события следующими полями:

on\_upload

- это экземпляр класса с

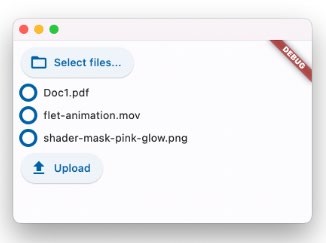


file\_name





- значение от 0.0 до 1.0.

Обратный вызов (callback) вызывается как минимум дважды для каждого загруженного файла: с прогрессом 0 перед началом загрузки и с прогрессом 1.0, когда загрузка завершена. Для файлов больше 1 МБ прогресс дополнительно сообщается каждые 10% от загруженного.

error

progress