How to download file?

Download file manually

First, you must get the *file_id* of the file you want to download. Information about files sent to the bot is contained in Message.

For example, download the document that came to the bot.

```
file_id = message.document.file_id
```

Then use the getFile method to get *file_path*.

```
file = await bot.get_file(file_id)
file_path = file.file_path
```

After that, use the download_file method from the bot object.

download_file(…)

Download file by *file_path* to destination.

If you want to automatically create destination (`io.BytesIO`) use default value of destination and handle result o this method.

class aiogram.client.bot.**Bot(token: str, session: BaseSession | None = None, parse_mode: str | None = None, disable_web_page_preview: bool | None = None, protect_content: bool | None = None)** [source

async download_file(file_path: str, destination: BinaryIO | Path | str | None = None, timeout: int = 30, chunk_size: int = 65536, seek: bool = True) → BinaryIO | None [source

Download file by file_path to destination.

If you want to automatically create destination (`io.BytesIO`) use default value of destination and handle result of this method.

PARAMETERS:

- **file_path** – File path on Telegram server (You can get it from `aiogram.types.File`)
- **destination** – Filename, file path or instance of `io.IOBase`. For e.g. `io.BytesIO`, defaults to None
- **timeout** – Total timeout in seconds, defaults to 30
- **chunk_size** – File chunks size, defaults to 64 kb
- **seek** – Go to start of file when downloading is finished. Used only for destination with `typing.BinaryIO` type, defaults to True

There are two options where you can download the file: to **disk** or to **binary I/O object**.

Download file to disk

To download file to disk, you must specify the file name or path where to download the file. In this case, the function will return nothing.

```
await bot.download_file(file_path, "text.txt")
```

Download file to binary I/O object

To download file to binary I/O object, you must specify an object with the `typing.BinaryIO` type or use the defau (`None`) value.

In the first case, the function will return your object:

```
my_object = MyBinaryIO()
result: MyBinaryIO = await bot.download_file(file_path, my_object)
# print(result is my_object)  # True
```

If you leave the default value, an `io.BytesIO` object will be created and returned.

```
result: io.BytesIO = await bot.download_file(file_path)
```

Download file in short way

Getting *file_path* manually every time is boring, so you should use the download method.

download(…)

Download file by *file_id* or *Downloadable* object to destination.

If you want to automatically create destination (`io.BytesIO`) use default value of destination and handle result o this method.

class aiogram.client.bot.**Bot(token: str, session: BaseSession | None = None, parse_mode: str | None = None, disable_web_page_preview: bool | None = None, protect_content: bool | None = None)** [source

async download(file: str | Downloadable, destination: BinaryIO | Path | str | None = None, timeout: int = 30, chunk_size: int = 65536, seek: bool = True) → BinaryIO | None [source

Download file by file_id or Downloadable object to destination.

If you want to automatically create destination (`io.BytesIO`) use default value of destination and handle result of this method.

PARAMETERS:

- **file** – file_id or Downloadable object
- **destination** – Filename, file path or instance of `io.IOBase`. For e.g. `io.BytesIO`, defaults to None
- **timeout** – Total timeout in seconds, defaults to 30
- **chunk_size** – File chunks size, defaults to 64 kb
- **seek** – Go to start of file when downloading is finished. Used only for destination with `typing.BinaryIO` type, defaults to True

It differs from download_file **only** in that it accepts *file_id* or an *Downloadable* object (object that contains the *file_id* attribute) instead of *file_path*.

You can download a file to disk or to a binary I/O object in the same way.

Example:

```
document = message.document
await bot.download(document)
```