



discord.py



[↑ to top](#)

 [v: stable](#) ▼

New in version 0.6.0.

# Setting Up Logging

`discord.py` logs errors and debug information via the `logging` python module. In order to streamline this process, the library provides default configuration for the `discord` logger when using `Client.run()`. It is strongly recommended that the logging module is configured, as no errors or warnings will be output if it is not set up.

The default logging configuration provided by the library will print to `sys.stderr` using coloured output. You can configure it to send to a file instead by using one of the built-in `logging.handlers`, such as `logging.FileHandler`.

This can be done by passing it through `Client.run()`:

```
import logging

handler = logging.FileHandler(filename='discord.log', encoding='utf-8', mode='a')

# Assume client refers to a discord.Client subclass...
client.run(token, log_handler=handler)
```

You can also disable the library's logging configuration completely by passing `None`:

```
client.run(token, log_handler=None)
```

Likewise, configuring the log level to `logging.DEBUG` is also possible:

```
import logging

handler = logging.FileHandler(filename='discord.log', encoding='utf-8', mode='a')

# Assume client refers to a discord.Client subclass...
client.run(token, log_handler=handler, log_level=logging.DEBUG)
```

This is recommended, especially at verbose levels such as `DEBUG`, as there are a lot of events logged and it would clog the stderr of your program.

If you want to setup logging using the library provided configuration without using `Client.run()`, you can use `discord.utils.setup_logging()`:

[↑ to top](#)

```
import discord

discord.utils.setup_logging()
```

 v: stable ▼

```
# or, for example
discord.utils.setup_logging(level=logging.INFO, root=False)
```

More advanced setups are possible with the [logging](#) module. The example below configures a rotating file handler that outputs DEBUG output for everything the library outputs, except for HTTP requests:

```
import discord
import logging
import logging.handlers

logger = logging.getLogger('discord')
logger.setLevel(logging.DEBUG)
logging.getLogger('discord.http').setLevel(logging.INFO)

handler = logging.handlers.RotatingFileHandler(
    filename='discord.log',
    encoding='utf-8',
    maxBytes=32 * 1024 * 1024, # 32 MiB
    backupCount=5, # Rotate through 5 files
)
dt_fmt = '%Y-%m-%d %H:%M:%S'
formatter = logging.Formatter('[{asctime}] [{levelname:<8}] {name}: {message}')
handler.setFormatter(formatter)
logger.addHandler(handler)

# Assume client refers to a discord.Client subclass...
# Suppress the default configuration since we have our own
client.run(token, log_handler=None)
```

For more information, check the documentation and tutorial of the [logging](#) module.

*Changed in version 2.0:* The library now provides a default logging configuration.

© Copyright 2015-present, Rapptz. Created using [Sphinx](#) 4.4.0.

[↑ to top](#)

 [v: stable](#) ▼