

Configuring the SDK

API keys and clients

By default, the SDK looks for the `OPENAI_API_KEY` environment variable for LLM requests and tracing, as soon as it is imported. If you are unable to set that environment variable before your app starts, you can use the `set_default_openai_key()` function to set the key.

```
from agents import set_default_openai_key

set_default_openai_key("sk-...")
```

Alternatively, you can also configure an OpenAI client to be used. By default, the SDK creates an `AsyncOpenAI` instance, using the API key from the environment variable or the default key set above. You can change this by using the `set_default_openai_client()` function.

```
from openai import AsyncOpenAI
from agents import set_default_openai_client

custom_client = AsyncOpenAI(base_url="...", api_key="...")
set_default_openai_client(custom_client)
```

Finally, you can also customize the OpenAI API that is used. By default, we use the OpenAI Responses API. You can override this to use the Chat Completions API by using the `set_default_openai_api()` function.

```
from agents import set_default_openai_api

set_default_openai_api("chat_completions")
```

Tracing

Tracing is enabled by default. It uses the OpenAI API keys from the section above by default (i.e. the environment variable or the default key you set). You can specifically set the API key used for tracing by using the `set_tracing_export_api_key` function.

```
from agents import set_tracing_export_api_key

set_tracing_export_api_key("sk-...")
```

You can also disable tracing entirely by using the `set_tracing_disabled()` function.

```
from agents import set_tracing_disabled

set_tracing_disabled(True)
```

Debug logging

The SDK has two Python loggers without any handlers set. By default, this means that warnings and errors are sent to `stdout`, but other logs are suppressed.

To enable verbose logging, use the `enable_verbose_stdout_logging()` function.

```
from agents import enable_verbose_stdout_logging

enable_verbose_stdout_logging()
```

Alternatively, you can customize the logs by adding handlers, filters, formatters, etc. You can read more in the [Python logging guide](#).

```
import logging

logger = logging.getLogger("openai.agents") # or openai.agents.tracing for the
Tracing logger

# To make all logs show up
logger.setLevel(logging.DEBUG)
# To make info and above show up
logger.setLevel(logging.INFO)
# To make warning and above show up
logger.setLevel(logging.WARNING)
# etc

# You can customize this as needed, but this will output to `stderr` by default
logger.addHandler(logging.StreamHandler())
```

Sensitive data in logs

Certain logs may contain sensitive data (for example, user data). If you want to disable this data from being logged, set the following environment variables.

To disable logging LLM inputs and outputs:

```
export OPENAI_AGENTS_DONT_LOG_MODEL_DATA=1
```

To disable logging tool inputs and outputs:

```
export OPENAI_AGENTS_DONT_LOG_TOOL_DATA=1
```