```python
"""
{filename}
==============================================================================

| Example of how to generate a text
|
| The resulting text will be directed to the standard error output (stderr)
"""

import asyncio
from typing import List, Optional

from example.boilerplate import API
from novelai_api.BanList import BanList
from novelai_api.BiasGroup import BiasGroup
from novelai_api.GlobalSettings import GlobalSettings
from novelai_api.Preset import PREAMBLE, Model, Preset
from novelai_api.Tokenizer import Tokenizer
from novelai_api.utils import b64_to_tokens


async def main():
    async with API() as api_handler:
        api = api_handler.api
        logger = api_handler.logger

        # model = Model.Sigurd
        # model = Model.Euterpe
        # model = Model.Krake
        model = Model.Clio

        # NOTE: plain text prompt
        prompt = PREAMBLE[model]
        # NOTE: prompt encoded in tokens
        # prompt = Tokenizer.encode(model, PREAMBLE[model])

        # NOTE: empty preset
        preset = Preset("preset", model, {})
        # NOTE: instantiation from default (presets/presets_6B_v4/default.txt)
        # preset = Preset.from_default(model)
        # NOTE: instantiation from official file (presets/presets_6B_v4)
        # preset = Preset.from_official(model, "Storywriter")
        # NOTE: instantiation from file
        # preset =
Preset.from_file("novelai_api/presets/presets_6B_v4/Storywriter.txt")
        # NOTE: instantiation of a new reset
        # preset = Preset("new preset", model)
        # NOTE: modification of the preset
        preset.min_length = 1
        preset.max_length = 20

        # NOTE: instantiate with arguments
        global_settings =
GlobalSettings(num_logprobs=GlobalSettings.NO_LOGPROBS)
```

```python
        # NOTE: change arguments after instantiation
        global_settings.bias_dinkus_asterism = True
        global_settings.rep_pen_whitelist = True

        # NOTE: no ban list
        bad_words: Optional[BanList] = None
        # NOTE: empty ban list
        # bad_words = BanList()
        # NOTE: ban list with elements in it
        # bad_words = BanList(" cat", " dog", " boy")
        # NOTE: disabled ban list with elements in it
        # bad_words = BanList(" cat", " dog", " boy", enabled = False)
        # NOTE: add elements to the bias list
        if bad_words is not None:
            bad_words.add(" man", " Man", " father")
            bad_words += " Father"

        bias_groups: List[BiasGroup] = []
        # NOTE: bias group with a strength of 0.15
        bias_group1 = BiasGroup(0.15)
        # NOTE: bias group with a strength of 0.05
        bias_group2 = BiasGroup(0.05)
        # NOTE: add the bias groups to the bias group list
        # bias_groups.extend([bias_group1, bias_group2])
        # NOTE: add tokens to the bias groups
        if bias_groups:
            bias_group1.add("very", " very", " slightly", " incredibly", "
enormously", " loudly")
            bias_group1 += " proverbially"
            bias_group2 += " interestingly"
            bias_group2 += " brutally"

        # NOTE: no module
        module = None
        # NOTE: Official module - CrossGenre module (module names can be found
in the network tab)
        #       A full list can be found
[here](docs/source/novelai_api/Full_list_of_modules.md)
        # module = "general_crossgenre"
        # NOTE: Custom module (Sage's Mass Effect v2) - ids can be retrieved
from download_user_modules() or a scenario
        # module =
"6B-v4:c6021aaa523e2dcb8588848b5fd4e2516dd4bb7107268aaa6050b5430c3a4b47:"
        #
"b764a71f139d0d829ed0f3077f026db43fdb25bc6b45ac508e85dd4c405a2fae"

        # NOTE: no stop sequence
        stop_sequence = None
        # NOTE: stop sequence as strings
        # stop_sequence = ["The End", "THE END", "\n"]
        # NOTE: stop sequence as tokens
        # stop_sequence = Tokenizer.encode(model, ["The End", "THE END", "\n"])

        # NOTE: normal generation
```

```python
        gen = await api.high_level.generate(
            prompt,
            model,
            preset,
            global_settings,
            bad_words=bad_words,
            biases=bias_groups,
            prefix=module,
            stop_sequences=stop_sequence,
        )
        # NOTE: b64-encoded list of tokens ids
        logger.info(gen["output"])
        # NOTE: list of token ids
        logger.info(b64_to_tokens(gen["output"]))
        # NOTE: decoded response
        logger.info(Tokenizer.decode(model, b64_to_tokens(gen["output"])))

        # NOTE: streamed generation
        async for token in api.high_level.generate_stream(
            prompt,
            model,
            preset,
            global_settings,
            bad_words=bad_words,
            biases=bias_groups,
            prefix=module,
            stop_sequences=stop_sequence,
        ):
            logger.info(
                "%s  %s  '%s'",
                # NOTE: b64-encoded token id
                token["token"],
                # NOTE: token id
                b64_to_tokens(token["token"]),
                # NOTE: decoded token
                Tokenizer.decode(model, b64_to_tokens(token["token"])),
            )

        # ... and more examples can be found in tests/test_generate.py


if __name__ == "__main__":
    asyncio.run(main())
```

```
"""
{filename}
==============================================================================

Example of how to login on the provided account
"""

import asyncio

from example.boilerplate import API


async def main():
    async with API() as api_handler:  # noqa: F841  # pylint: disable=W0612
        print(api_handler.api.headers)


if __name__ == "__main__":
    asyncio.run(main())
```