

Тестовое задание на позицию Python стажёра

Описание задачи

Необходимо разработать CLI-программу, которая будет являться клиентом к условному сервису отправки СМС-сообщений.

Спецификация API сервиса находится в приложенном файле `sms-platform.yaml`. В спецификации указан необходимый формат запросов, а также возвращаемые сервисом ответы. Данная спецификация написана в формате **OpenAPI**, для её визуализации можно воспользоваться онлайн сервисами.

Требования к реализации и логика работы программы

1. Для конфигурации программы должен использоваться файл конфигурации в формате `TOML`. Минимальный набор параметров конфигурации:
 - **Адрес** сервиса отправки СМС сообщений.
 - **Имя пользователя и пароль**, которые будут использоваться для отправки СМС сообщений.
2. Программа должна принимать через параметры командной строки:
 - Номер **отправителя** СМС сообщения;
 - Номер **получателя** СМС сообщения;
 - **Текст** СМС сообщения.
3. После обработки параметров командной строки программа должна сформировать HTTP запрос в соответствии с спецификацией сервиса и отправить его на **Адрес** из файла конфигурации, используя при этом **Имя пользователя и пароль**, которые также указаны в файле конфигурации.
4. После получения ответа программа должна вывести в **стандартный поток вывода** следующую информацию: **код ответа, тело ответа**.
5. В программе должны быть реализованы классы, представляющие HTTP-запрос и HTTP-ответ. Данные классы, как минимум, должны включать в себя следующие методы:
 - `.to_bytes()` -> `bytes` - создает HTTP запрос/ответ из данных, сохранённых в экземпляре класса.
 - `.from_bytes(binary_data: bytes)` -> `Self`: **метод класса**, преобразующий последовательность байт в объект HTTP-запроса или ответа.

6. Объекты HTTP-запроса и ответа реализуются в минимальном требуемом виде, без учёта всех тонкостей HTTP-протокола.
7. Отправка сообщения должна быть реализована с использованием библиотеки `socket` или `asyncio`. Использование `urllib`, `requests`, `aiohttp`, `httpx` запрещено.

Дополнительные рекомендации

- **Логирование:** С целью отслеживания запуска программы, реализуйте логирование информации о переданных параметрах командной строки и полученных от сервиса отправки СМС-сообщений ответов в лог-файл.
- **Модульность:** Разделите код на модули для улучшения читаемости и поддержки, если считаете нужным.
- **Аннотация типов:** Используйте аннотации типов для улучшения поддержки вашего кода инструментами статического анализа кода.
- **Тестирование:** Добавьте автоматические тесты, чтобы проверить работу кода и его отдельных компонентов. Рекомендуется использовать `pytest`, но возможно использование и `unittest`.

Самопроверка

Вашу реализацию программы можно протестировать с мок-сервером:

1. Скачать **Prism** для своей платформы:
 - Windows / Linux / macOS: <https://github.com/stoplightio/prism/releases>
2. Запустить мок-сервер:
 - **Linux:** `./prism-cli-linux mock sms-platform.yaml`
 - **macOS:** `./prism-cli-macos mock sms-platform.yaml`
 - **Windows:** `./prism-cli-win.exe mock sms-platform.yaml`

После запуска API будет доступно по адресу: http://localhost:4010/send_sms

Как сдавать задание?

Разместите свой код на любом общедоступном хостинге Git-репозитория и приложите ссылку. В `README.md` проекта должна содержаться информация по запуску проекта и вся информация, которую вы считаете нужным указать в описании.

Заключение

Это задание направлено на проверку ваших навыков в области сетевого программирования и работы с HTTP-протоколом. Удачи в реализации вашего HTTP-клиента! 