

Improvado Back-end test task for Junior Developer: VK get friends report

Задание

1. Изучить API VK;
2. Разработать систему, позволяющую получить список друзей из ВКонтакте;

Требования к системе

1. Система должна быть выполнена в виде консольного приложения;
2. Система должна генерировать отчет в файл, в одном из нескольких форматов:
 1. CSV
 2. TSV
 3. JSON
3. В систему должны входить такие входные данные:
 1. Авторизационный токен;
 2. ID пользователя, для которого генерируем отчет;
 3. Формат выходного файла. По умолчанию — CSV;
 4. Путь к выходному файлу. По умолчанию — файл с именем report в текущей директории;
4. В отчете должны присутствовать такие поля (и именно в таком порядке):
 1. Имя;
 2. Фамилия;
 3. Страна;
 4. Город;
 5. Дата рождения в ISO формате;
 6. Пол;
5. Строки отчета должны быть отсортированы по имени, в алфавитном порядке;
6. Ошибки не должны маскироваться. Пользователю надо сообщать об ошибках как можно более дружелюбно;

Требования к заданию

1. Готовое задание загрузить на Github. Ссылку на репозиторий отправить в письме;
2. Код должен работать корректно;
3. В репозитории должен быть файл readme с инструкцией для пользователя. С приложением инструкции по авторизации;
4. Код задания должен быть в соответствии с PEP8;

Бонусы

1. Реализована пагинация. Друзей может быть много, они могут не влезть на одну страницу, и тогда отчет будет неполным;
2. Учет большого количества данных. Если друзей слишком много, мы можем не влезть в ограничения по оперативной памяти. Надо стараться расходовать память с умом;

3. Расширяемость кода. Необходимо подумать о том, в какую сторону код может расширяться. Что, если отчет потребуется отдать в файле формата YAML? На сколько болезненны будут такие изменения?
4. Код хорошо читается. При чтении кода быстро становится понятно, что тут происходит;
5. У кода есть организация. При изучении репозитория должно быть понятно, о чем этот репозиторий, глядя на структуру файлов и каталогов;
6. В файлике readme кроме инструкции лежит описание того, как работает скрипт, а так же описание API энд-поинтов, которые задействованы в скрипте;
7. Код покрыт тестами. Что будет, если придет другой человек и изменит код так, что он перестанет работать? Как ему сразу понять, что он изменил лишнего?
8. Пользователю должно быть удобно. Если считаете интерфейс входных данных неудачным, можно предложить свой, но нужно аргументировать;
9. В коде есть комментарии, там, где они требуются. Например, для узких мест, костылей и прочего;
10. В системе присутствуют логи, по которым можно отследить состояние системы во время выполнения;

Полезные ссылки

- <https://vk.com/dev/manuals> — API документация