

## Тестовое задание для кандидата на Backend-практикум «От SQL до DWH»

Разработать загрузчик данных в БД PostgreSQL на языке Python.

### Описание требований к приложению

Приложение должно подключаться к какому-либо доступному источнику данных (REST-службе), скачивать оттуда данные и сохранять в экземпляре PostgreSQL.

Для выполнения задания должны быть выполнены следующие **ключевые действия**:

1. Развертывание экземпляра PostgreSQL в Docker-контейнере и минимальная конфигурация, включающая создание базы данных (БД), создание учетной записи с надлежащими правами доступа, настройку сетевого доступа для обращения к БД при помощи современных UI-инструментов (например, DBeaver или аналогичных сред).
2. Изучение работы с источником данных (формат данных, требования к авторизации, особенности обращения и использования), разработка программного кода для считывания данных.
3. Разработка схемы БД для хранения скачиваемых данных (определение необходимых таблиц).
4. Развертывание окружения для Python с требуемыми компонентами (библиотеками), например, для взаимодействия с PostgreSQL, обращения к REST-службам., журнализации и т.д. Python-окружение также рекомендуется развертывать в контейнере.
5. Организация подключения в БД PostgreSQL и запись в соответствующие таблицы данных источника.

### Дополнительные требования и рекомендации

1. Рекомендуется при выборе источника данных ограничиться источниками, выдающими данные в форматах Json или CSV. Если данных доступны в CSV, то для данных следует разработать структуру (таблицу) соответствующего формата; следует обращать внимание на типы полей, формат чисел и дат, и прочие особенности. Крайне рекомендуется использовать существующие библиотеки для работы с CSV-форматом, если в этом возникнет необходимость.

2. Если данные доступны в Json, можно либо записывать «как есть», то есть помещать все значение json в одно поле, либо (что будет плюсом), раскладывать поля json-объекта по полям таблицы, если это возможно. Не рекомендуется декомпонировать json-запись на вложенные подобъекты и сохранять их в отдельные таблицы.
3. При создании таблицы следует задать РК-ключ; формат и способ создания ключа нужно разработать самостоятельно.
4. Рекомендуется разработать и реализовать какую-либо систему журнализации процесса загрузки данных, способом по усмотрению разработчика.
5. Необходимо каким-либо рациональным способом обрабатывать и журнализировать возможные ошибки и корректно завершать сеанс работы приложения способом по усмотрению разработчика.

**Отчет** должен содержать краткое описание функциональных свойств и архитектуры системы, краткое описание разработанных **автором** артефактов, в том числе:

1. Схему объектов БД.
2. Программный код Python.
3. Конфигурационные файлы для развертывания докеров.
4. Скриншоты, демонстрирующие работу приложения.
5. Прочие документы, представляющие важность и интерес с точки зрения объяснения особенностей работы приложения.

Не следует прикладывать фрагменты документации и материалы из общеизвестных источников.

## **Стек технологий и требований к ним для реализации**

- СУБД PostgreSQL (через отдельный Docker-образ).
- Python 3.
- Фреймворки (по усмотрению разработчика).

## **Условия выполнения**

Не забудь оставить инструкции в README репозитория по развертыванию окружения.

Следующее обязательное условие — работа приложения должна соответствовать описанию приложения, а само приложение должно быть

реализовано с использованием технологий, описанных в пункте «Стек технологий и требований к ним для реализации веб-приложения».

Если ты видишь, как можно улучшить работу приложения и сделать его более удобным для использования, разрешается вносить доработки сверх того, что указаны в описании приложения. Но не забудь рассказать об этих улучшениях в описании приложения в репозитории.

Кроме того, если у тебя появится какое-то интересное решение при реализации приложения, например, ты захочешь добавить дополнительную технологию или библиотеку, то используй смело ее применяй! 😊 Но мы бы очень хотели увидеть аргументы для ее использования в README репозитория.

Выполненное задание разместить в публичном репозитории на любом из этих сервисов: **GitHub, GitLab, Bitbucket**.

*Источник данных:*

- <https://cloud.mail.ru/public/L1xB/nvgHGYJz5>
- *Либо выбираем произвольно.*