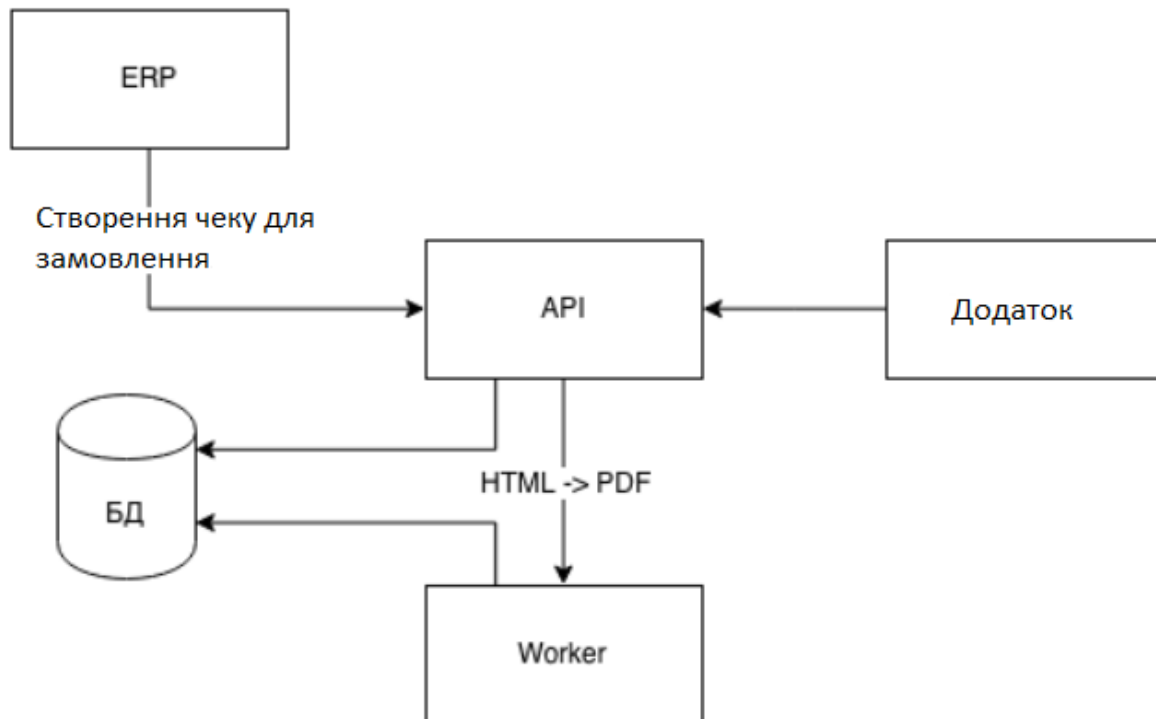


Мережа ресторанів доставки "жращка топ" має безліч точок, на яких готуються замовлення для клієнтів. Кожен клієнт хоче разом із замовленням отримати чек, що містить детальну інформацію про замовлення. Співробітники кухні також хочуть чек, щоб у процесі приготування та упаковки замовлення не забути покласти все що потрібно. Наше завдання допомогти і тим, і іншим, написавши сервіс для генерації чеків.

Схема роботи сервісу:



- 1) Сервіс отримує інформацію про нове замовлення, створює в БД чеки для всіх принтерів точки, зазначеної в замовленні, і ставить асинхронні завдання на генерацію PDF-файлів для цих чеків. Якщо точка не має жодного принтера - повертає помилку. Якщо чеки для цього замовлення вже були створені – повертає помилку(передається номер замовлення).
- 2) Асинхронний воркер за допомогою wkhtmltopdf генерують PDF-файл із HTML-шаблону. Ім'я файлу повинно мати такий вигляд <ID замовлення>_<тип чека>.pdf (123456_client.pdf). Файли повинні зберігатися у папці media/pdf у корені проекту.
- 3) Програма опитує сервіс на наявність нових чеків. Опитування відбувається наступним шляхом: спочатку запитується список чеків, які вже згенеровані для конкретного принтера, після скачується PDF-файл для кожного чека і відправляється на друк. (це типу принтер, який міняє статус в друці чеку)

Технічні вимоги:

- 1) Сервіс повинен бути написаний на python та Django
- 2) База даних - PostgreSQL

- 3) Всі інфраструктурні речі, необхідні для сервісу (PostgreSQL, Redis, wkhtmltopdf) запускати в docker за допомогою docker-compose, сам проект не потрібно обертати в docker.
- 4) Крім API, має бути адмінка для обох моделей, з можливістю фільтрувати чеки за принтером, типом і статусом

Примітки:

- 1) Верстка HTML-шаблонів для чеків лежить у репозиторії у папці templates
- 2) Для простоти роботи з wkhtmltopdf варто використовувати docker-контейнер [wkhtmltopdf](#)

Моделі:

- 1) Принтер (Printer). Кожен принтер друкує лише свій тип чеків. Поле `api_key` набуває унікальних значень, по ньому однозначно визначається принтер. Для цієї моделі повинні бути fixtures (принтери для обох типів чеків для кількох точок).

Поле	Тип	Значення	Опис
<code>name</code>	<code>CharField</code>		назва принтеру
<code>api_key</code>	<code>CharField</code>		ключ доступу до API
<code>check_type</code>	<code>CharField</code>	<code>kitchen client</code>	тип чеку який друкує принтер
<code>point_id</code>	<code>IntegerField</code>		точка до якої привязаний принтер

- 2) Чек (Check). Інформація про замовлення кожного чека зберігається в JSON, немає необхідності робити окремі моделі.

Поле	Тип	Значение	Описание
<code>printer_id</code>	<code>ForeignKey</code>		принтер
<code>type</code>	<code>CharField</code>	<code>kitchen client</code>	тип чеку
<code>order</code>	<code>JSONField</code>		інформація про замовлення
<code>status</code>	<code>CharField</code>	<code>new rendered printed</code>	статус чеку
<code>pdf_file</code>	<code>FileField</code>		посилання на створений PDF-файл

API:

Опис доступних методів міститься у файлі `api.yml` (swagger-специфікація).

Готове тестове можна відіслати на телеграм менторув телеграм: <https://t.me/vjxzebra>