

Тестове завдання - Python Backend Developer

Задача:

Розробити REST API для створення та перегляду чеків з реєстрацією та авторизацією користувачів.

Опис

Реєстрація користувача

API має надавати можливість користувачам зареєструвати свої облікові записи, вказавши обов'язкові дані, такі як ім'я, логін та пароль.

Авторизація користувача

За допомогою свого логіну та паролю користувач повинен мати можливість отримати унікальний JWT-токен для доступу до API.

Створення чеку продажу

Авторизованим користувачам має бути дозволено створювати чеки продажу товарів. Чек повинен містити наступну інформацію: інформацію про товари, їх кількість, ціну та інші додаткові дані (за бажанням). Для кожного чеку має бути збережений ідентифікатор, дата створення та ідентифікатор користувача, який створив чек.

Метод повинен приймати в тілі запиту:

- Список товарів
 - Назву
 - Вартість за одну одиницю товару
 - Кількість товару або його вагу
- Інформацію про сплату за покупку товарів
 - Тип (готівка cash / картка card)
 - Сума
- додаткові дані, за бажанням

У відповідь необхідно повернути додатково розраховану вартість по кожній позиції в чеку, загальну суму чеку та суму решти, яку необхідно видати покупцю

Наприклад, тіло запиту може мати вигляд:

```
{
  "products": [
    {
      "name": string,
      "price": decimal,
      "quantity": int,
    },
    ...
  ]
}
```

```

    ],
    "payment": {
        "type": "cash" / "cashless",
        "amount": decimal,
    },
}

```

тоді відповідь повинна містити наступну інформацію

```

{
    "id": ...,
    "products": [
        {
            "name": string,
            "price": decimal,
            "quantity": decimal,
            "total": decimal,
        }
    ],
    "payment": {
        "type": "cash" / "cashless",
        "amount": decimal,
    },
    "total": decimal,
    "rest": decimal,
    "created_at": datetime,
}

```

Перегляд та фільтрація списку власних чеків

Після авторизації, користувачі повинні мати можливість переглядати список своїх чеків. Список має включати інформацію про кожен чек, таку як ідентифікатор, дата створення, загальна сума та інші дані, що присутні у чеку.

Окремо у користувача повинна бути можливість отримати інформацію про чек за його унікальним ідентифікатором отриманим на етапі виконання запиту створення чеку.

Користувачі повинні також мати можливість застосовувати фільтри до списку чеків, щоб зручно знаходити конкретні чеки. Приклади фільтрів можуть включати:

- Фільтрація за датою створення чеку (наприклад, відфільтрувати чеки, створені впродовж останнього місяця).
- Фільтрація за загальною сумою чеку (наприклад, показати лише чеки зі сумою покупки вище певної величини).
- Фільтрація за типом оплати (наприклад, показати лише чеки сплачені готівкою)
- Фільтри можуть бути реалізовані як параметри запиту до API.

Також додати можливість пагінації результатів пошуку чеків за допомогою параметрів кількості записів на сторінці та номеру сторінки або кількості перших записів що потрібно пропустити у вибірці (offset).

Перегляд чеків покупцем

Чек повинен мати посилання, що має бути доступне неавторизованим користувачам для перегляду чеку у текстовому форматі, даний метод може використовуватись продавцем для передачі чеку покупцю (надіслати повідомленням в месенджері, надіслати email або показати QR-код з посиланням на своєму екрані або інше. Реалізовувати механізм відправки чека за допомогою email або повідомленням в месенджері - не потрібно). Реалізувати можливість конфігурування кількості символів у рядку.

Приблизний вигляд чеку:

```

      ФОП Джонсонюк Борис
=====
3.00 x 298 870.00
Mavic 3T                      896 610.00
-----
20.00 x 31 000.00
Дрон FPV з акумулятором
6S чорний                     620 000.00
=====
СУМА                          1 516 610.00
Картка                        1 516 610.00
Решта                          0.00
=====
      14.08.2023 14:42
      Дякуємо за покупку!
```

Технічні вимоги

- Використати мову програмування Python актуальної версії
- Використати один із веб-фреймворків: FastAPI, aiohttp, BackSheep
- Збереження даних повинно виконуватись в PostgreSQL
- API повинно бути задокументовано, включаючи опис доступних ендпоінтів та необхідні параметри для використання API
- Описати інструкцію з запуску даного сервісу (README) з описом того що необхідно для запуску проєкту

Це завдання вимагає від розробника розуміння створення REST API, роботи з базовими операціями створення, читання та авторизації користувачів, а також здатності генерувати текстової візуалізації для перегляду чеків.

Вихідний код проєкту опублікувати на GitHub або GitLab і надіслати посилання на репозиторій

Додаткове завдання

Опціонально, але рекомендовано реалізувати набір тестів за допомогою `pytest` для перевірки працездатності та надійності вашого REST API. Ці тести мають включати різні сценарії

- Тести авторизації та реєстрації: Переконайтеся, що користувачі можуть успішно реєструватися, авторизовуватися та отримувати доступні токени для використання в API.
- Тести створення чеків: Перевірте, чи можуть користувачі успішно створювати нові чеки і отримувати підтвердження про успішне створення.
- Тести перегляду чеків: Переконайтеся, що користувачі можуть відображати свої чеки та використовувати параметри пагінації та фільтрації.
- Тести публічного перегляду текстового представлення чеку: Переконайтеся, що будь-хто знаючи `id` чеку може переглянути чек у текстовому вигляді.
- Тести недопустимих дій: Впевніться, що API правильно обробляє недопустимі або некоректні запити, і користувачам надається зрозуміле повідомлення про помилку.