Техническое Задание

1. Скачивание товаров

1.1. Реализовать функцию забора товаров с WB в модуле wb/api.py (по аналогии с Ozon product\_list), результатом котогого является генератор с "сырыми" данными в формате JSON. Каждый вызов функции next() возвращает 1 объект.

1.2. Реализовать функцию вставки "сырых" данных полученных через api в базу данных.

Информация о полях:

* sku артикул товара в системе продавца.
* created\_at дата и время создания товара, если WB предоставляет информацию, иначе timezone.localtime(timezone.now()).
* updated\_at дата и время обновления.
* status статус товара. Необходимо мне предоставить список статусов для синхронизации с текущими статусами.
* image основние изображение товара. Первичный ключ на объект в модели core.ProductCardImage.
* raw\_data данные полученные с WB API.
* Необходимо заполнить дополнительные поля, если имеются таковые: wb\_product\_id, wb\_category\_id (аналогия с Ozon).
* Поле moderated\_at не ипользуется.

1.3. Предотвращать создание дубликатов.

1. 4. Не производить обновление, если данные полученные не изменились. Отбрасывать разного рода метки времени.

1.5. Сохранять ссылки на изображения в поле remote\_file\_url, а также удалять не используемые изображения.

2. Скачивание заказов

2.1. Реализовать функцию забора заказов с WB в модуле wb/api.py (по аналогии с Ozon fbs\_posting\_list), результатом котогого является генератор с "сырыми" данными в формате JSON. Каждый вызов функции next возвращает 1 объект.

2.2. Реализовать функцию вставки "сырых" данных полученных через api в базу данных.

Информация о полях:

* posting\_number не используется
* number уникальный номер заказа в системе WB. На Ozon используется posting\_number (1 заказ / несколько отправлений). Если будут вопросы, расскажу подробнее.
* created\_at
* updated\_at
* in\_process\_at когда заказ был принят в обработку. На Ozon created\_at != in\_process\_at. in\_process\_at вероятнее всего отсечка после которой заказ можно считать поступившим в магазин, подтвержденный покупателем заказ.
* posting\_type не используется.
* status статус заказа. Необходимо мне предоставить список статусов для синхронизации с текущими статусами.

2.3. Предотвращать создание дубликатов.

2.4. Не производить обновление, если данные полученные не изменились. Отбрасывать разного рода метки времени.

2.5 Добавить к заказу позиции.

2.6. Информация о полях:

* wb\_sku артикул в системе WB.
* quantity количество проданного товара.
* price цена продажи.
* commission комиссия маркетплейса.
* delivery\_cost стоимость достваки.

2.7. Предотвращать создание дубликатов.

2.8. Не производить обновление, если данные полученные не изменились. Отбрасывать разного рода метки времени.

2.9. Создавать product\_card на каждую позицию заказа, в случае если такой карточки нет в базе данных. Полную или урезанную версию в зависимости от количества данных внутри заказа.

Название полей и реализация могут незначительно отличатся от описанной структуры.

Реализовывать функционал необходимо максимально приближенно к реализации с Ozon.