

Тестовое задание: Разработка процесса интеграции сообщений

Проблема текущей реализации:

При интеграции сообщений с почты процесс загрузки происходит довольно долго так как сообщений много. Сначала происходит поиск последнего импортированного сообщения, а затем от него до самого нового добавление их в систему.

Цель:

Реализация полосы чтения -> загрузки сообщений с почты. Необходимо продумать гибкий функционал для импортирования в систему сообщений из "yandex.ru", "gmail.com", "mail.ru".

Задачи:

1. Реализация моделей:

Необходимо реализовать модели, работающие на PostgreSQL для хранения логина и пароля от почты и хранения информации о сообщениях, полученных с почты.

Минимальные поля для модели хранения сообщений:

- 1) id
- 2) Тема сообщения (наименование)
- 3) Дата отправки
- 4) Дата получения
- 5) Описание или текст сообщения
- 6) Поле для хранения списка прикрепленных файлов к письму

2. Создание страницы списка сообщений:

Сверстайте страницу (дизайн не важен) для отображения списка (желательно в виде таблицы) сообщений. Помимо таблицы должен быть где-то наверху страницы реализован прогресс-бар, где будет сначала надпись "чтение сообщений" а затем "получение сообщений".

3. Логика процесса получения:

После внесения логина и пароля в БД необходимо зайти на страницу списка сообщений и должен начаться процесс их получения. Пока последнее добавленное будет искаться на почте, в прогресс-баре мы должны видеть, сколько писем проверено. Как только данное сообщение найдено, полоса загрузки должна начать

визуально загружаться с обратным отсчётом. В этот момент в таблицу на странице должны добавляться строки с информацией о сообщениях. (в полях таблицы выводить все поля из модели. Поле “описание” или “текст сообщения” можно выводить вкратце.)

Требования:

1. Django 4.2+
2. DjangoRESTframework=3.14.0 (если понадобится)
3. Channels==4.0.0
4. Channels-redis==4.1.0
5. Daphne==4.0.0
6. html/css
7. JavaScript (jQuery)
8. Библиотеки для чтения с почты по желанию исполнителя

Результат:

Реализовано корректное WebSocket-соединения, чтение сообщений с почты без артефактов с правильным декодированием информации и сохранением файлов, прикрепленных к письму, а также стабильно и правильно работающий прогресс-бар.

Оценочные критерии:

1. Корректность реализации модели
2. Корректность реализации socket-соединения
3. Соответствие кода стандарту pep8
4. Корректность работы
5. Отсутствие артефактов при импортировании данных