

Задача

Нужно повысить качество передаваемого кода на ревью, чтобы ускорить процесс разработки и поставки более качественного кода в проекты. Для этого создайте сервис проверки файлов с выполненными задачами.

Основные модули, из которых состоит система:

1. Модуль авторизации и регистрации пользователя.

Пользователь должен иметь возможность зарегистрироваться в системе по паре почта и пароль, а также войти по этим данным.

2. Модуль загрузки файлов с исходным кодом.

Авторизованный пользователь должен иметь возможность загрузить файл в систему, при этом информация о загруженном файле должна сохраниться в базу данных с пометкой о том, что файл новый. Помимо этого, пользователь должен иметь возможность удалить файл или перезаписать, таким образом, в базе данных должны быть соответствующие пометки. Очень важно не давать загружать файлы, у которых расширение не равно “.py”.

3. Модуль проверки соответствия кода общепринятым правилам.

По расписанию выполняется задача на автоматическую проверку кода для новых загруженных или перезагруженных файлов. По итогу проверки необходимо сформировать отложенную задачу на отправку письма пользователю с информацией о проведенной проверке. Важно хранить лог каждой проверки для каждого файла, который находится в списке файлов у пользователя в системе.

4. Модуль отправки письма с уведомлением пользователю.

Необходимо реализовать обработку задач из очереди на отправку уведомлений пользователю о результате проверки его файла. При этом, важно в логах проверки отмечать факт отправки сообщения пользователю.

5. Модуль отчета о проведенных проверках

В системе также должен быть интерфейс, в котором пользователь может просмотреть результаты выполненных проверок с пометками об отправке отчета пользователю на почту.

Рекомендации:

- реализовать интерфейс можно с помощью uikit Bootstrap
- для работы с очередями лучше использовать celery в связке с redis
- для запуска периодических задач можно использовать celery beat

Критерии решения:

- Интерфейс системы должен содержать следующие экраны: вход, регистрация, список загруженных файлов, отчет проверки по каждому файлу отдельно, изменение файла, удаление файла
- Реализованы все 5 модулей
- Для разных сервисов созданы отдельные контейнеры (django, postgresql, redis, celery, при необходимости список можно самостоятельно расширять), все оформлено в docker-compose файле, при необходимости можно создавать вспомогательные Dockerfile
- Проект готов быть размещен на удаленном сервере
- Интерфейс понятен и соответствует базовым требованиям системы
- Решение выложено на github.com