# Тестовое задание (ТЗ)

Требуется написать REST FULL API сервис со следующим функционалом:

1.1 Публичные маршруты (доступны всем не авторизованным пользователям):

1.2 Регистрация пользователя

1.3 Авторизация пользователя

1.4 Сброс пароля пользователя

1.5 Приватные маршруты (доступны только после авторизации пользователя):

1.5.1 Просмотр списка пользователей

1.5.2 Просмотр пользователя

1.5.3 Редактирование пользователя

1.5.4 Удаление пользователя в корзину

1.5.6 Просмотр списка пользователей в корзине (удаленные в корзину)

1.5.7 Восстановление пользователя из корзины

1.5.8 Полное удаление пользователя из БД (база данных)

1.5.9 Групповое удаление пользователей в корзину

1.5.10 Групповое удаление пользователей из БД

1.5.11 Групповое восстановление пользователей из корзины

3. Вести полное логирование действий с пользователями в БД (таблица histories)

# Требования

2.1 Сервис должен быть написан с использованием фреймворка Laravel версии не ниже 10.0

2.2 Использовать php версии не ниже 8.2

2.3 Применить подход тонких контроллеров.

2.4 Максимально декомпозировать код (выделение возможного повторяющегося кода в трейты и абстрактные классы).

2.5 В качестве id (идентификатора записи в БД) использовать строковое представление uuid6

2.6 Использовать Redis для кеширования объектов и снижения нагрузки на БД

2.7 Обеспечить целостность данных БД путем обработки транзакций и внешних ключей (по возможности)

2.8 Написать фабрики для наполнения БД тестовыми (фейковыми) данными

2.9 Вынести бизнес-логику в сервисный слой

2.10 Ответ от сервиса возвращать в формате json либо xml (будет плюсом реализация всех вариантов ответа)

2.11 При просмотре листинга пользователей реализовать возможность фильтров и сортировки по всем полям таблицы

2.12 Допускается использование встроенных библиотек фреймворка

# Структура таблиц

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | users | | | id | uuid | | last\_name | varchar(40) | | name | varchar(40) | | middle\_name | varchar(40) | | email | varchar(80) | | phone | varchar(20) | | deleted\_at | timestamp | | created\_at | timestamp | | updated\_at | timestamp | | |  |  | | --- | --- | | histories | | | id | uuid | | model\_id | uuid | | model\_name | varchar(250) | | before | json | | after | json | | action | enum | | deleted\_at | timestamp | | created\_at | timestamp | | updated\_at | timestamp | |

# Информация по проекту для кандидата

4.1 Максимальное время на выполнение ТЗ - 12 часов

4.2 После инициализации проекта в локальной среде, его необходимо залить в публичный репозиторий github.com в своем аккаунте и сразу после этого предоставить ссылку в telegram на данный репозиторий нашему тим-лиду проекта

4.3 В случае невозможности выполнения ТЗ за один день, сообщать нашему тим-лиду время и что удалось сделать за данный промежуток времени, после того как залили код в репозиторий, для возможности отслеживания изменений по коммитам.

# Дополнительное задание (будет плюсом)

Реализовать приватные маршруты по аналогии с частью маршрутов из пункта 1.5

5.1 Реализовать листинг, просмотр сущности, удаление в корзину, полное удаление записи из БД с сущностями таблицы histories

5.2 Возможность восстановления модели к состоянию из таблицы histories

5.3 Реализовать консольные команды по пунктам 5.1 и 5.2

# Рекомендации (будет плюсом)

6.1 Код должен быть само-документируемый с применением тайпхинтов и PSR.

6.2 Уделить большое внимание чистоте и качеству кода

6.3 Контейнеризация проекта (docker), для быстрого развертывания

6.4 К ТЗ применить паттерны DRY, KISS, YANGI

6.5 Ведение документации в Swager

6.6 Применить в проекте возможность версионирования API

6.7 При форматировании кода рекомендуется использование библиотеки laravel/pint