



Pastebin [hard]

<http://pastebin.com> Он позволяет заливать куски текста/кода и получать на них короткую ссылку, которую можно отправить другим людям. Загружать данные можно как анонимно, так и зарегистрировавшись.

Основная функциональность:

- Возможность загрузить кусок текста ("пасту") из названия и текста
 - загружать можно как анонимно, так и залогинившись
 - можно выбрать срок в течение которого "паста" будет доступна по ссылке (expiration time)
 - 10мин, 1час, 3часа, 1день, 1неделя, 1месяц, без ограничения
 - после окончания срока получить доступ к "пасте" нельзя
 - можно указать ограничение доступа:
 - public -- доступна всем, видна в списках
 - unlisted -- доступна только по ссылке
 - private -- доступна только отправившему (только одному авторизовавшемуся пользователю -- автору)
 - для "пасты" можно выбрать язык, тогда при выводе синтаксис выбранного языка должен подсвечиваться
 - для загруженной пасты выдается короткая ссылка вида <http://my-awesome-pastebin.tld/{какой-то-рандомный-хэш}>, например, <http://my-awesome-pastebin.tld/ab12cd34>
- Авторизация/регистрация

- по логину паролю
 - по соцсетям
 - Возможность просмотра
 - по ссылке
 - на всех страницах блок с последними 10 public пастами
 - на всех страницах залогиненный пользователь видит доп. блок с последними 10 своими пастами
 - зарегистрированный пользователь имеет отдельную страницу, где видит список всех своих паст с пагинацией (например, по 10)
 - все пасты, у которых вышел срок доступности, не видны никому, в том числе и автору
 - Возможность пользователю пожаловаться на пасту
 - Администрирование
 - Прикрутить к проекту админку voyager, orchid
 - Добавить возможность просмотра списка пользователей, паст и жалоб
 - Добавить возможность бана пользователя и удаления паст
 - API для сторонних приложений
 - Добавить возможность взаимодействия с приложением при помощи API запросов
 - Добавить запросы для получения того, что описано выше
-

Требования к проекту:

- Backend
 - Framework: Laravel
 - DB: MySQL, MariaDB, Postgres

- Использовать паттерн Repository и паттерн Service
- Строгая типизация
- Везде писать phpDoc'и
- Прикрутить и настроить phpStan, исправить ошибки на минимум 6 уровне
- Развернуть на docker compose
- Frontend - без ограничений (верстка не оценивается в рамках задания)
- Код в репозитории (git) на github
 - Крайне желательно пользоваться репозиторием, а не залить что получится в конце одним коммитом вместе с фреймворком
 - Описание коммитов должно следовать соглашению **Conventional Commits**