

Student Project: URL Shortener Weekly sprints

Основной технологический стек:

- Java17
- JavaDoc
- SpringBoot
- Maven
- MySQL / PostgreSQL / SQLite
- Docker
- JUnit5
- GitHub

Идея:

Creating a service provides short URL instead of original one

До того, как начнёте:

Think about how many approaches of this service you can describe

Первоочередные задачи (спринты #1-#5):

- необходима веб-страница с возможностью сгенерировать короткую ссылку на основе имеющейся у пользователя
- под генерацией понимается:
 - проверка на уже имеющуюся ссылку в БД, если такая ссылка уже есть, просто вывести имеющуюся
 - генерация новой ссылки (либо нового "ключа")
 - запись в БД новых данных
- переход по короткой ссылке должен перенаправлять нас на оригинальную (исходную) ссылку в БД
- создание таймера для удаления короткой ссылки (1 месяц / 6 месяцев / 1 год от начала создания)
- сервис может работать без регистрации
- определить и постоянно улучшать показатели скорости конвертации короткой ссылки в оригинальную (исходную), чем меньше, тем лучше. Задуматься над redis, если скорость критична

Второстепенные задачи (спринты #5+:

- создание возможности регистрации пользователей, определение ролей (пользователь, администратор)
- для зарегистрированных пользователей (всех ролей) разработать опцию, где таймер для удаления ссылки будет считаться не от создания, а от последнего использования
- разработать возможность подсчёта статистических параметров:
 - количество созданий новых коротких ссылок в день за последние 1 / 7 / 30 дней
 - подсчет использования ссылок в БД, определение наиболее популярных
- для администраторов разработать dashboard, вывести на dashboard все статистические данные по сервису
- добавить возможность для администраторов удалять / замораживать пользователей
- добавить возможность для администраторов удалять имеющиеся ссылки в БД
- добавить возможность для администраторов определять перманентную ссылку (корректировать время таймера)
- добавить возможность проверки коротких ссылок в БД на их работоспособность (какие-то уже устарели и их можно удалить из БД)

Интересные моменты, связанные с монетизацией сервиса (возможно придётся отключить доступ без регистрации)

- определить какое количество коротких ссылок по-умолчанию можно сгенерировать (платная генерация ссылок выше определенного порога, например 50 штук в неделю)
- определить необходимость создания зарезервированных коротких ссылок

Week #	Tasks	Use	As minimal result
#1	Описать в деталях работу прототипа в текстовом виде, отрисовка схем взаимодействия	Word, Notepad, MindMaps, Violet UML Editor	описание проекта в текстовом (+схематичном) формате
#1	Описать схему БД	Word, Notepad, MindMaps, Violet UML Editor	детальная схема взаимосвязей в БД
#1	Описать REST API	Word, Notepad, MindMaps, Violet UML Editor	техническое описание функционала проекта и его работы (объекты, их методы)

Week #	Tasks	Use	As minimal result
#1	Создать репозиторий и проект для работы	GitHub, IntelliJ IDEA	As minimal result в работе репозиторий (использование GitHub проекта как kanban доску), созданный проект
#1	Создать карточки на kanban-доске для мониторинга хода выполнения разработки	GitHub, Trello	наполненная kanban-доска задачами
#2	Создание проекта SpringBoot, подключение необходимых библиотек	GitHub, IntelliJ IDEA	сформированный проект, готовый pom.xml
#2	Реализация функционала по генерации короткой ссылки, сохранению её в БД	GitHub, IntelliJ IDEA	готовая веб-страничка, где можно сгенерировать короткую ссылку
#2	Реализация функционала перенаправления (короткая ссылка -> исходная ссылка)	GitHub, IntelliJ IDEA	готовый функционал, который получает короткую ссылку и перенаправляет пользователя по исходной ссылке
#3	Реализация функционала по добавлению таймера к ссылке, например таймер = 30 дням	GitHub, IntelliJ IDEA	готовый функционал для добавления таймера к уже имеющимся коротким ссылкам в БД
#4	Написание тестов	GitHub, IntelliJ IDEA, JUnit	покрытие тестами всех публичных методов
#4	Написание документации	GitHub, IntelliJ IDEA, JavaDoc	описание всех методов в документации
#5	Упаковка решения в docker контейнер	GitHub, IntelliJ IDEA, docker	сформированный сценарий сборки docker образа
#5	Тестирование развернутого решения	GitHub, IntelliJ IDEA, docker	тестирование сборки образа и его запуск
#5+	Реализовать возможность зарегистрироваться на веб-сайте, определить роли (администратор, пользователь)	GitHub, IntelliJ IDEA	рабочий функционал по регистрации пользователей (сформированы и определены роли)
#5+	Реализовать dashboard администратора	GitHub, IntelliJ IDEA	отдельная веб-страничка в виде dashboard'a для администраторской роли
#5+	Реализация функционала по ручному удалению ссылки из БД (опция для зарегистрированных пользователей)	GitHub, IntelliJ IDEA	готовый функционал с возможностью удаления короткой ссылки из БД
#5+	Реализовать возможность корректировки таймера ссылок для других пользователей; создание перманентной ссылки (dashboard администратора)	GitHub, IntelliJ IDEA	готовый функционал по корректировке таймера для короткой ссылки
#5+	Реализовать возможность удаления / заморозки пользователей (доступно на dashboard администратора)	GitHub, IntelliJ IDEA	управление заморозкой (разморозкой) или удалением пользователей
#5+	Реализовать возможность удаления ссылок других пользователей (доступно на dashboard администратора)	GitHub, IntelliJ IDEA	управление короткими имеющимися ссылками в БД (удаление)
#5+	Реализовать статистику проекта (количество созданий новых коротких ссылок в день за последние 1 / 7 / 30 дней) (dashboard администратора)	GitHub, IntelliJ IDEA	отображение статистики по количеству созданных коротких ссылок в 1 / 7 / 30 дней
#5+	Реализовать статистику проекта (счет использования ссылок в БД, определение наиболее популярных) (dashboard администратора)	GitHub, IntelliJ IDEA	отображение статистики по использованию ссылок