**Цель работы:**

Составление тестовой документации для проекта на Node.js с использованием NestJS, TypeScript, JWT авторизацией, Mongoose и Redis.

**Описание тестируемого приложения:**

Приложение представляет собой бэкенд-сервер, разработанный на Node.js с использованием NestJS. В приложении реализована аутентификация и авторизация пользователей с помощью JWT токенов. Для хранения пользовательских данных используется MongoDB с помощью Mongoose. Также в приложении используется Redis для кэширования данных.

**Тестовая документация:**

**Тест-сьюты:**

**Таблица 1. Список тест-сьютов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Автор | Приоритет | Заголовок | Список тест-кейсов |
| 1 | user | 1 | Аутентификация | 1.1 Авторизация |
|  |  |  |  | 1.2 Регистрация |
| 2 | user | 1 | Сервис авторизации | 2.1 Авторизация пользователя |
|  |  |  |  | 2.2 Ошибка авторизации |
| 3 | user | 1 | Создание ссылки | 3.1 Позитивный сценарий |
|  |  |  |  | 3.2 Негативный сценарий |
| 4 | user | 1 | Получение ссылки | 4.1 Позитивный сценарий |
|  |  |  |  | 4.2 Негативный сценарий |
| 5 | user | 1 | Получение списка ссылок пользователя | 5.1 Позитивный сценарий |
|  |  |  |  | 5.2 Негативный сценарий |
| 6 | user | 1 | Поиск пользователя | 6.1 Поиск по ID |
|  |  |  |  | 6.2 Поиск по имени |
|  |  |  |  | 6.3 Поиск всех пользователей |
| 7 | user | 1 | Управление пользователями | 7.1 Регистрация нового пользователя |
|  |  |  |  | 7.2 Изменение данных пользователя |
|  |  |  |  | 7.3 Удаление пользователя |
| 8 | user | 1 | Управление ссылками | 8.1 Добавление короткой ссылки |
|  |  |  |  | 8.2 Получение списка ссылок пользователя |
|  |  |  |  | 8.3 Добавление аватара пользователя |
|  |  |  |  | 8.4 Получение аватара пользователя |
|  |  |  |  | 8.5 Добавление почты пользователю |
|  |  |  |  | 8.6 Подтверждение почты пользователя |

**Тест-кейсы:**

**Таблица 2. Список тест-кейсов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Описание | Предусловия | Шаги | Ожидаемый результат |
| 1.1 | Авторизация (позитивный) | Пользователь не авторизован в системе | 1. Пользователь вводит свой логин и пароль | Пользователь успешно авторизован |
| 1.2 | Регистрация (негативный) | Пользователь не зарегистрирован в системе | 1. Пользователь вводит существующий в системе email | Выводится сообщение о существующем пользователе |
| 2.1 | Авторизация пользователя (позитивный) | Пользователь зарегистрирован в системе | 1. Пользователь вводит свой логин и пароль | Пользователь успешно авторизован |
| 2.2 | Ошибка авторизации (негативынй) | Пользователь вводит неверные учетные данные | 1. Пользователь вводит неправильный логин и/или пароль | Выводится сообщение об ошибке аутентификации |
| 3.1 | Создание ссылки (позитивный) | Пользователь хочет создать короткую ссылку | 1. Пользователь предоставляет длинную ссылку и свой ID | Ссылка успешно создана |
| 3.2 | Создание ссылки (негативный) | Пользователь хочет создать короткую ссылку | 1. Пользователь предоставляет некорректные данные | Возвращается сообщение об ошибке |
| 4.1 | Получение ссылки (позитивный) | Ссылка существует в системе | 1. Пользователь запрашивает короткую ссылку | Возвращается длинная ссылка |
| 4.2 | Получение ссылки (негативный) | Ссылка не существует в системе | 1. Пользователь запрашивает короткую ссылку | Возвращается сообщение об отсутствии ссылки |
| 5.1 | Получение списка ссылок пользователя (позитивный) | Пользователь имеет созданные короткие ссылки | 1. Пользователь запрашивает список своих коротких ссылок | Возвращается список коротких ссылок пользователя |
| 5.2 | Получение списка ссылок пользователя (негативный) | Пользователь не имеет созданных коротких ссылок | 1. Пользователь запрашивает список своих коротких ссылок | Возвращается сообщение о отсутствии ссылок у пользователя |
| 6.1 | Поиск пользователя по ID | Пользователь хочет найти пользователя по ID | 1. Пользователь предоставляет ID пользователя | Возвращается пользователь с указанным ID |
| 6.2 | Поиск пользователя по имени | Пользователь хочет найти пользователя по username | 1. Пользователь предоставляет имя пользователя | Возвращается пользователь с указанным именем |
| 6.3 | Поиск всех пользователей | Пользователь хочет найти всех пользователей | 1. Пользователь запрашивает список всех пользователей | Возвращается список всех пользователей |
| 7.1 | Регистрация нового пользователя | Пользователь хочет зарегистрироваться | 1. Пользователь предоставляет данные для регистрации | Создается новый пользователь |
| 7.2 | Изменение данных пользователя | Пользователь хочет изменить данные профиля | 1. Пользователь предоставляет новые данные пользователя | Данные пользователя успешно изменены |
| 7.3 | Удаление пользователя | Пользователь хочет удалить пользователя | 1. Пользователь предоставляет ID пользователя | Пользователь успешно удален |
| 8.1 | Добавление короткой ссылки | Пользователь создал короткую ссылку | 1. Пользователь предоставляет длинную ссылку и свой ID | Короткая сслыка добавляется к записи пользователя в БД |
| 8.2 | Получение списка ссылок пользователя | Пользователь имеет созданные короткие ссылки | 1. Пользователь запрашивает список своих коротких ссылок | Возвращается список коротких ссылок пользователя |
| 8.3 | Добавление аватара пользователя | Пользователь хочет добавить аватар-картинку | 1. Пользователь предоставляет файл аватара и свой ID | Аватар успешно добавлен и связан с пользователем |
| 8.4 | Получение аватара пользователя | Пользователь имеет установленный аватар | 1. Пользователь запрашивает свой аватар | Возвращается файл аватара пользователя |
| 8.5 | Добавление почты пользователю | Пользователь хочет добавить почту | 1. Пользователь предоставляет свою почту и свой ID | Почта успешно добавлена |
| 8.6 | Подтверждение почты пользователю | Пользователь предоставил почту и верификацию | 1. Пользователь предоставляет код верификации и свой ID | Почта пользователя успешно подтверждена |

**Выводы по работе:**

В ходе разработки тестовой документации были составлены подробные тест-кейсы, охватывающие как позитивные, так и негативные сценарии использования функциональности. Эти тест-кейсы были структурированы в соответствии с требованиями к документации и обеспечивают полное покрытие основных функций и возможных вариантов использования.

Важным аспектом является проверка корректности авторизации и аутентификации пользователей, а также управление пользователями, включая регистрацию, изменение данных, удаление и т. д. Все эти функции были протестированы как на позитивные, так и на негативные сценарии, чтобы обеспечить надежную работу системы и защитить её от возможных уязвимостей.

Также была описана интеграция с внешними сервисами, такими как Redis, MongoDB и AWS S3, что позволяет убедиться в правильной работе взаимодействия с ними.

**Список использованных источников:**

1. Официальная документация NestJS.
2. Официальная документация TypeScript.
3. Документация по тестированию NestJS.
4. Документация библиотеки для тестирования Jest
5. Документация по MongoDB и Mongoose.
6. Документация по AWS S3.
7. Документация по Redis.