**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Образовательная программа «Программная инженерия»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| СОГЛАСОВАНО  C:\Users\user\Pictures\adb-sign.jpgШтатный преподаватель, Доцент ДПИ ФКН  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Д. Брейман  «16» мая 2025 г. | |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»  \_\_\_\_\_\_\_\_\_\_\_\_\_ Н.А. Павлочев  «16» мая 2025 г. | |
| |  |  | | --- | --- | | *Подп. и дата* |  | | *Инв. № дубл.* |  | | *Взам. инв. №* |  | | *Подп. и дата* |  | | *Инв. № подл* | RU.17701729.06.12-01 51 01-1 | | **СЭД С РАСШИРЕННЫМ ВЕРСИОНИРОВАНИЕМ**  **Программа и методика испытаний**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.06.12-01 51 01-1-ЛУ** | | | | | | |
|  | |  | | | | |
| Исполнитель:  студент группы БПИ 219  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Н.В. Артемов /  «16» мая 2025 г. | | | | |
|  | | | | | | |
|  | | | | |  | |

**Москва 2025**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| УТВЕРЖДЕН  RU.17701729.06.12-01 51 01-1-ЛУ |  | |  | |
| |  |  | | --- | --- | | *Подп. и дата* |  | | *Инв. № дубл.* |  | | *Взам. инв. №* |  | | *Подп. и дата* |  | | *Инв. № подл* | RU.17701729.06.12-01 51 01-1 | | **СЭД С РАСШИРЕННЫМ ВЕРСИОНИРОВАНИЕМ**  **Программа и методика испытаний**  **RU.17701729.06.12-01 51 01-1**  **Листов 25** | | | | |
|  | |  | | |
|  | | |
|  | | | | |
|  | | | |  |

**Москва 2025**

ОГЛАВЛЕНИЕ

[1 ОБЪЕКТ ИСПЫТАНИЙ 4](#_Toc198239447)

[1.1. Наименование программы 4](#_Toc198239448)

[1.2. Краткая характеристика области применения программы 4](#_Toc198239449)

[2 ЦЕЛЬ ИСПЫТАНИЙ 5](#_Toc198239450)

[3 ТРЕБОВАНИЯ К ПРОГРАММЕ 6](#_Toc198239451)

[3.1. Требования к функциональным характеристикам 6](#_Toc198239452)

[3.1.1. Требования к серверной части 6](#_Toc198239453)

[3.1.2. Требования к составу выполняемых функций 6](#_Toc198239454)

[3.1.3. Организация входных данных 8](#_Toc198239455)

[3.1.4. Организация выходных данных 8](#_Toc198239456)

[3.2. Требования к временным характеристикам 8](#_Toc198239457)

[3.3. Требования к интерфейсу 8](#_Toc198239458)

[3.4. Требования к надежности 8](#_Toc198239459)

[4 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ 9](#_Toc198239460)

[4.1. Предварительный состав программной документации 9](#_Toc198239461)

[4.2. Специальные требования к программной документации 9](#_Toc198239462)

[5 СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ 10](#_Toc198239463)

[5.1. Технические средства, используемые во время испытаний 10](#_Toc198239464)

[5.2. Порядок проведения испытаний 10](#_Toc198239465)

[6 МЕТОДЫ ИСПЫТАНИЙ 11](#_Toc198239466)

[6.1. Испытание выполнения требований программной документации 11](#_Toc198239467)

[6.2. Испытание выполнения требований к функциональным характеристикам 11](#_Toc198239468)

[6.2.1. Требования к серверной части 11](#_Toc198239469)

[6.2.2. Требования к составу выполняемых функций 11](#_Toc198239470)

[6.2.3. Организация входных и выходных данных 23](#_Toc198239471)

[6.3. Испытание выполнения требований к временным характеристикам 23](#_Toc198239472)

[6.4. Испытание выполнения требований к надежности 23](#_Toc198239473)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 24](#_Toc198239474)

# ОБЪЕКТ ИСПЫТАНИЙ

## 1.1. Наименование программы

**Наименование темы разработки:** «СЭД с расширенным версионированием»

**Наименование темы разработки на английском языке:** «EDM With Advanced Versioning»

## 1.2. Краткая характеристика области применения программы

«СЭД с расширенным версионированием» - веб приложение, система электронного документооборота. Основная область применения программы – ведение документов, что включает: создание, редактирование, согласование, подписание и дальнейший ввод в действие. Включает в себя функционал гибкой настройки всех процессов ведения документов. Отличается расширенной системой версионирования шаблонов, матриц согласования и документов, а также кросс-доменной функциональностью.

# ЦЕЛЬ ИСПЫТАНИЙ

Целью проведения испытаний данной программы является тестирование корректности работы приложения, выявление ошибок и проверка программы на соответствие определенным требованиям.

# ТРЕБОВАНИЯ К ПРОГРАММЕ

## 3.1. Требования к функциональным характеристикам

Предоставляется Api состоящее из серверной части.

### 3.1.1. Требования к серверной части

1. Серверная часть должна состоять из backend сервисов и api-gateway.
2. Синхронное взаимодействие между микросервисами должно быть реализовано через протокол gRPC, а асинхронное через Apache Kafka.
3. Должно быть реализовано взаимодействие с базой данных PostgreSQL для хранения всей необходимой информации.
4. Все сервисы должны быть реализованы с использованием чистой архитектуры.
5. Серверная часть должна реализовывать функционал настройки и ведения документооборота с поддержкой расширенного версионирования и конфигурации.
6. Серверная часть должна хранить и обрабатывать любые данные, связанные с функционалом системы в безопасном и надежном виде.

### 3.1.2. Требования к составу выполняемых функций

1. **Создание и настройка шаблонов документов**

В приложении должна быть возможность создать шаблон и настроить следующие его составляющие:

1. Основные данные, такие как: наименование, классификация и спецификация;
2. Макет, включая атрибутивный состав (в том числе атрибутивный состав участвующий в согласовании) и схему состояний;
3. Валидаторы на переходы состояний и значения атрибутов;
4. Нумератор для генерации регистрационного номера.
5. **Создание и настройка матриц согласования**

В приложении должна быть возможность создать матрицу согласования и настроить следующие ее составляющие:

1. Графы-деревья с простыми вершинами и вершинами с условными операторами для определения финального маршрута согласования на основании значений атрибутов и настроек;
2. Группы согласующих с условиями для выбора их в маршрут согласования на основании значений атрибутов.
3. **Версионирование матриц согласования**

Версии матриц согласования должны быть связаны с версиями шаблонов документов, к которым они привязаны и по возможности мигрироваться на новый атрибутивный состав при обновлении шаблона.

Должна быть возможность создания различных версий матриц под определенный шаблон с возможностью активировать и деактивировать при необходимости.

1. **Создание документов**

Должна быть реализована возможность создания документов напрямую из конкретного шаблона или по выбранной классификации, которая соответствует шаблону. Документ должен создаваться с незаполненными значениями атрибутов и в начале своей схемы состояний.

1. **Редактирование документов**

Должен быть реализован функционал по редактированию значений атрибутов.

Должен быть реализован функционал по переводу документов по состояниям, которые определены в шаблоне, в том числе перевод в согласование и подписание.

1. **Согласование документов**

Должен быть реализован функционал перевода документа в согласование, при котором будет строиться маршрут согласования на основе настроенной матрицы. Далее, должны быть назначены согласующие, которые далее могут согласовать документ.

Должна быть предусмотрена возможность согласования в несколько этапов с несколькими согласующими параллельно или последовательно.

Должен быть реализован степпер для просмотра этапов согласования с всеми подробностями.

1. **Подписание документов**

Должна быть реализована возможность подписания документов, посредством ручного перевода состояния при предоставлении скана подписанного документа в бумажном виде.

Должна быть реализована возможность электронного подписания документов, посредством интеграции с микросервисом компании, который будет инкапсулировать логику отправки документа в ЭЦП.

1. **Аудит**

Должно быть предусмотрено сохранение всех операций изменений шаблонов, матриц согласования и документов для последующего аудита.

1. **Версионирование документов**

Должно быть реализовано расширенное версионирование документов по версиям и настройкам их шаблонов. Должна быть возможность создания документов из любой версии шаблонов и поддержка их работы после изменений шаблонов.

### 3.1.3. Организация входных данных

Входные данные для серверной части со стороны клиентской должны быть в виде json в теле Http запросов. Файлы должны передаваться в несжатом виде в form запроса.

### 3.1.4. Организация выходных данных

Выходные данные для клиентской части со стороны серверной должны быть в виде json в теле ответов Http запросов. Файлы должны передаваться в несжатом состоянии в форме ответа Http запроса.

## 3.2. Требования к временным характеристикам

Время отклика приложения при стабильном подключении к интернету не должно превышать 1 секунды.

## 3.3. Требования к интерфейсу

Требования к интерфейсу не предъявляются.

## 3.4. Требования к надежности

1. Приложение не должно аварийно завершаться при любом наборе входных данных.
2. Программа не должна давать сбой при некорректных действиях пользователя.

# ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

## 4.1. Предварительный состав программной документации

1. «СЭД с расширенным версионированием». Техническое задание (ГОСТ 19.201-78 [7]);
2. «СЭД с расширенным версионированием». Программа и методика испытаний (ГОСТ 19.301-79 [10]).

## 4.2. Специальные требования к программной документации

1. Все документы к программе должны быть выполнены в соответственни с ГОСТ 19.106-78[6] и ГОСТ к этому виду документа (см п. 4.1.).
2. Текст ВКР должен быть загружен в систему Антиплагиат через ЛМС НИУ ВШЭ. Лист, подтверждающий загрузку текста ВКР, сдается в учебный офис вместе со всеми материалами не позже, чем за день до защиты ВКР.
3. Техническое задание и текст ВКР сдаются в печатном виде, при этом их листы утверждения и листы утверждения других документов (Руководство программиста, Программа и методика испытаний, Текст программы) должны быть обязательно подписаны академическим руководителем образовательной программы 09.03.04 «Программная инженерия», руководителем разработки и исполнителем перед сдачей ВКР в учебный офис не позже одного дня до защиты.
4. Вся документация и программа также сдается в электронном виде в формате .pdf или .docx, или в архиве формата .rar или .zip.
5. Все документы перед защитой ВКР должны быть загружены в информационно образовательную среду НИУ ВШЭ LMS (Learning management system) в личном кабинете, дисциплина – «Защита выпускной квалификационной работы. СЭД с расширенным версионированием», одним архивом.

# СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ

## 5.1. Технические средства, используемые во время испытаний

Во время испытаний использовался ноутбук Apple macbook pro 14 со следующими техническими характеристиками:

* Чип apple m4 pro
* 18 ГБ объединенной оперативной и видео памяти
* Операционная система macOS Sequoia 15.3.2
* Docker engine v27

## 5.2. Порядок проведения испытаний

1. Проверка требований к программной документации,
2. Проверка требований к функциональным характеристиками,
3. Проверка требований к временным характеристиками,
4. Проверка требований к надежности.

# МЕТОДЫ ИСПЫТАНИЙ

## 6.1. Испытание выполнения требований программной документации

Выполнение всех требований программной документации проверяется визуально по загруженным в Smart LMS файлам и подписям на них.

Все документы соответствуют требованиям.

## 6.2. Испытание выполнения требований к функциональным характеристикам

### 6.2.1. Требования к серверной части

Все описанные требования выполнены:

* Основная бизнес-логика серверной части находится в backend cервисах, а за логику взаимодействия с UI отвечают сервисы api-gateway.
* Синхронное взаимодействие между микро-сервисами реализовано через gRPC протокол, а асинхронное взаимодействие происходит через брокер сообщений Apache Kafka. Взаимодействие api-gateway с UI происходит по протоколу http1 через restful api.
* Все данные хранятся и обрабатываются через PostgreSQL.
* Все сервисы реализованы с использованием чистой архитектуры с разделением на слои.
* Серверная часть полностью поддерживает функционал настройки и ведения документооборота с поддержкой расширенного версионирования и конфигурации.
* Серверная часть хранит и обрабатывает любые данные, связанные с функционалом системы в безопасно и надежном виде. Дополнительный уровень защиты обеспечивает хранение всех данных в бинарном виде.

### 6.2.2. Требования к составу выполняемых функций

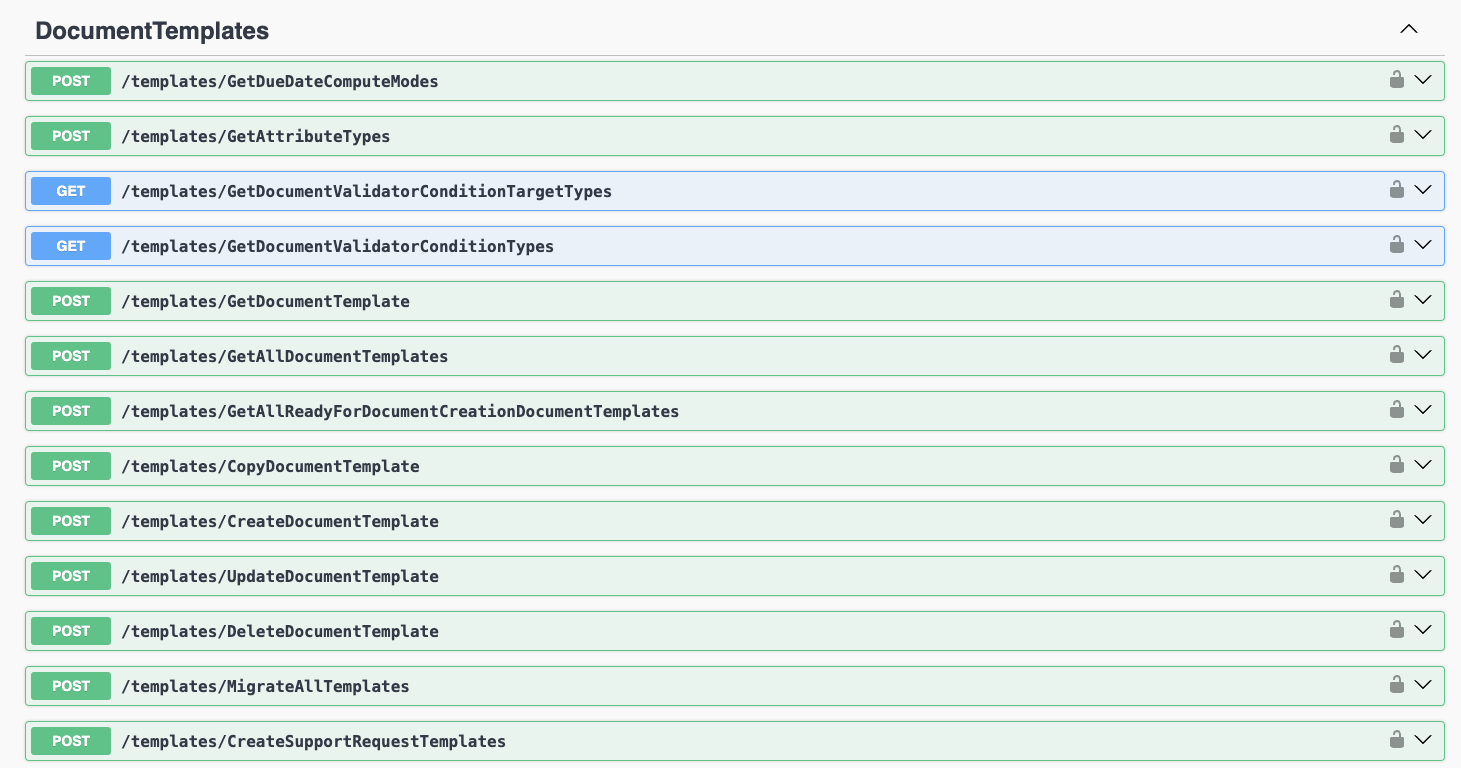
Тестирование основных модулей сервисов и их работы в формате черного ящика проводилось с помощью юнит и e2e тестирования с помощью xUnit, следственно ручное тестирование функционала производилось в формате взаимодействия пользователя с системой через api-gateway без прямых обращений к backend серверной части.

1. **Создание и настройка шаблонов документов**

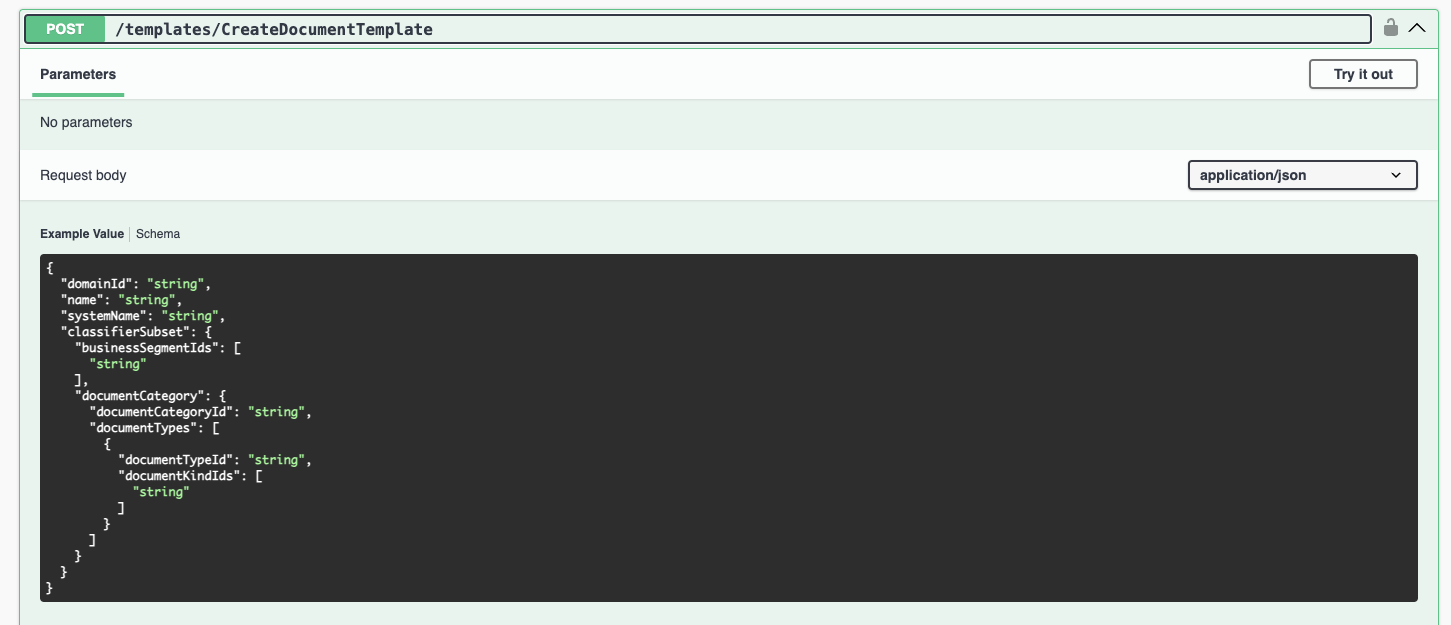
В приложении реализована возможность создавать и настраивать шаблоны документов.

Тестирование функционала проводилось в полном объеме с передачей некорректных данных форматом и бизнес значением, в том числе проводилось тестирование функционала:

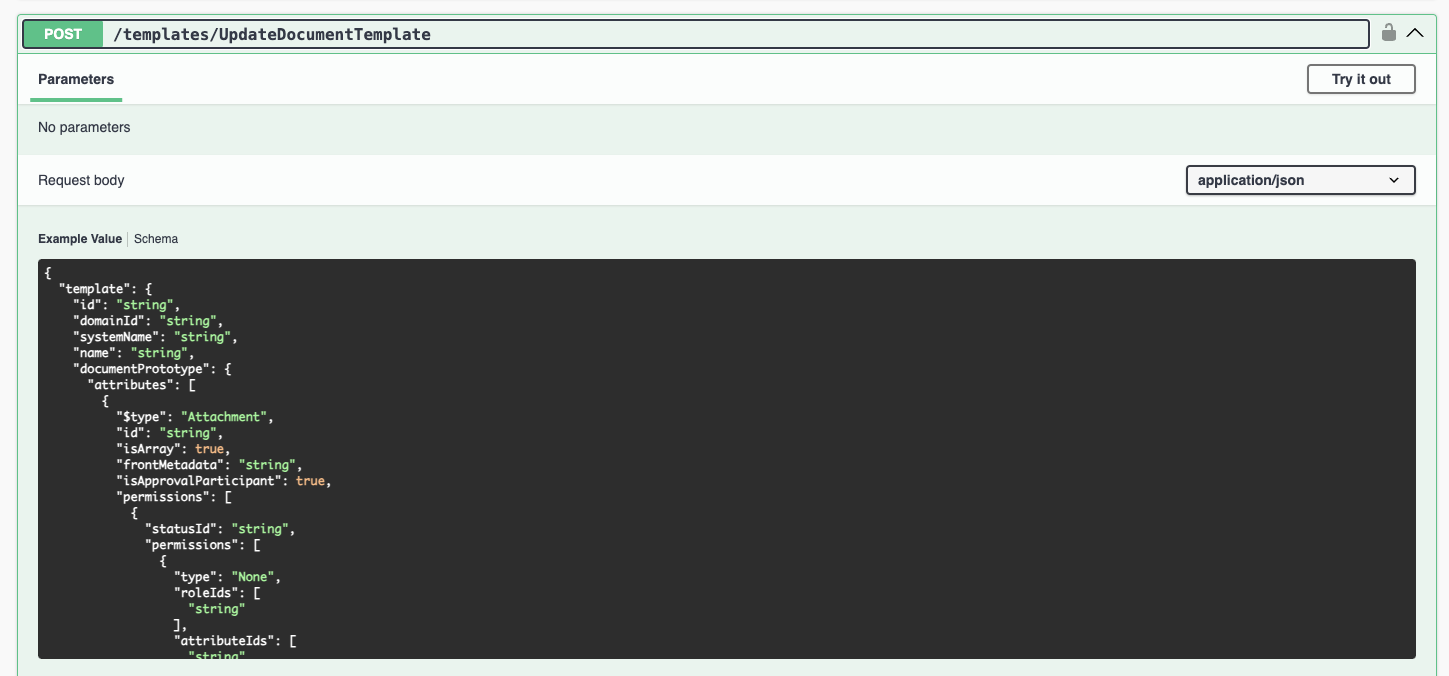
* Доменов шаблонов;
* Наименований и идентификации шаблонов;
* Прототипов и макетов шаблонов, в том числе: атрибутов, переходов между состояниями, уведомлений, регистрационных номеров, валидаторов, иных параметров;
* Состояний шаблонов.



Приложение 1. Иллюстрация документации Swagger. Ручки для создания и настройки шаблонов документов



Приложение 2. Иллюстрация документации Swagger. Ручка для создания шаблонов документов



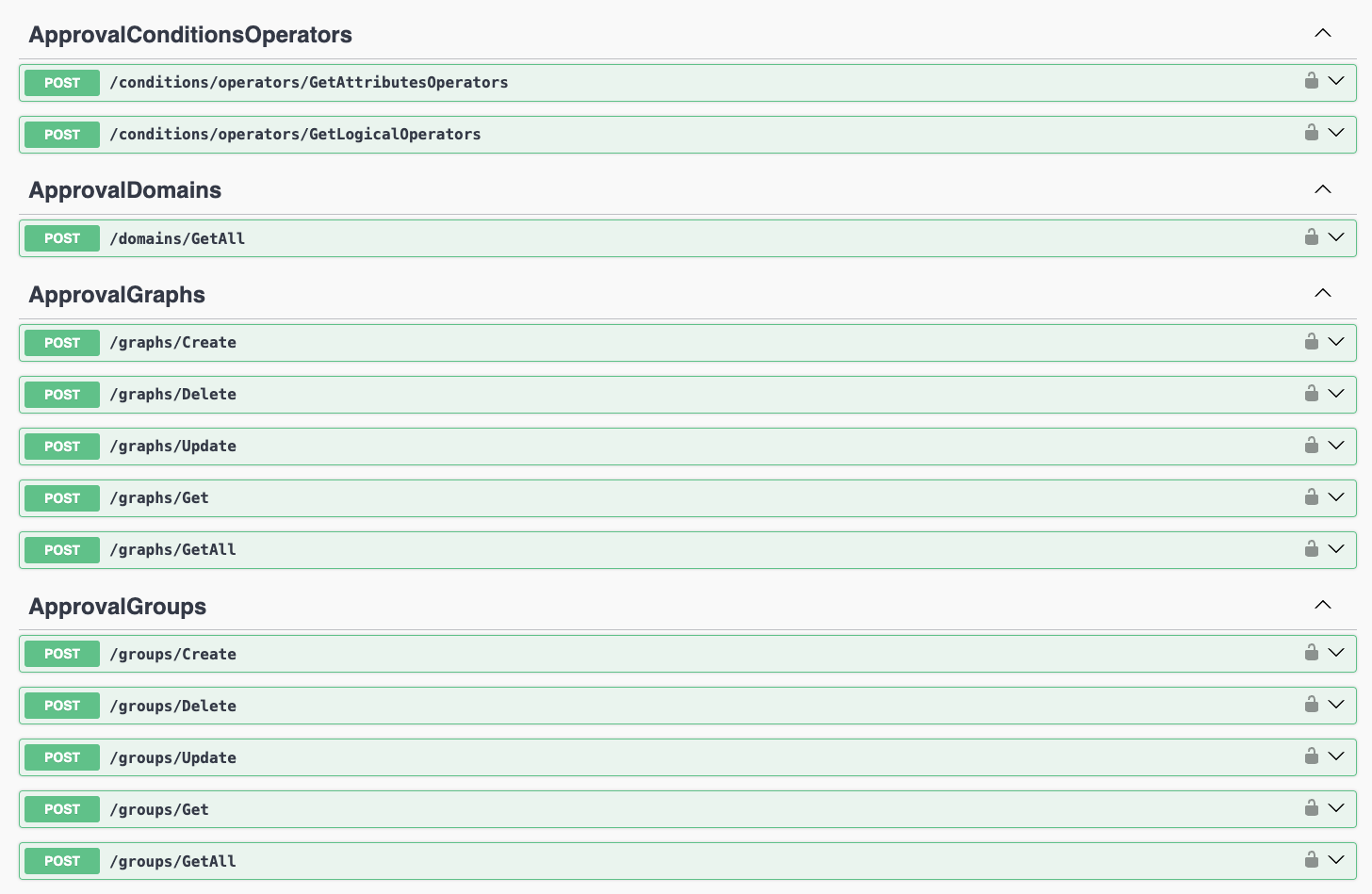
Приложение 3. Иллюстрация документации Swagger. Ручка настройки шаблонов документов

1. **Создание, настройка и версионирование матриц согласования**

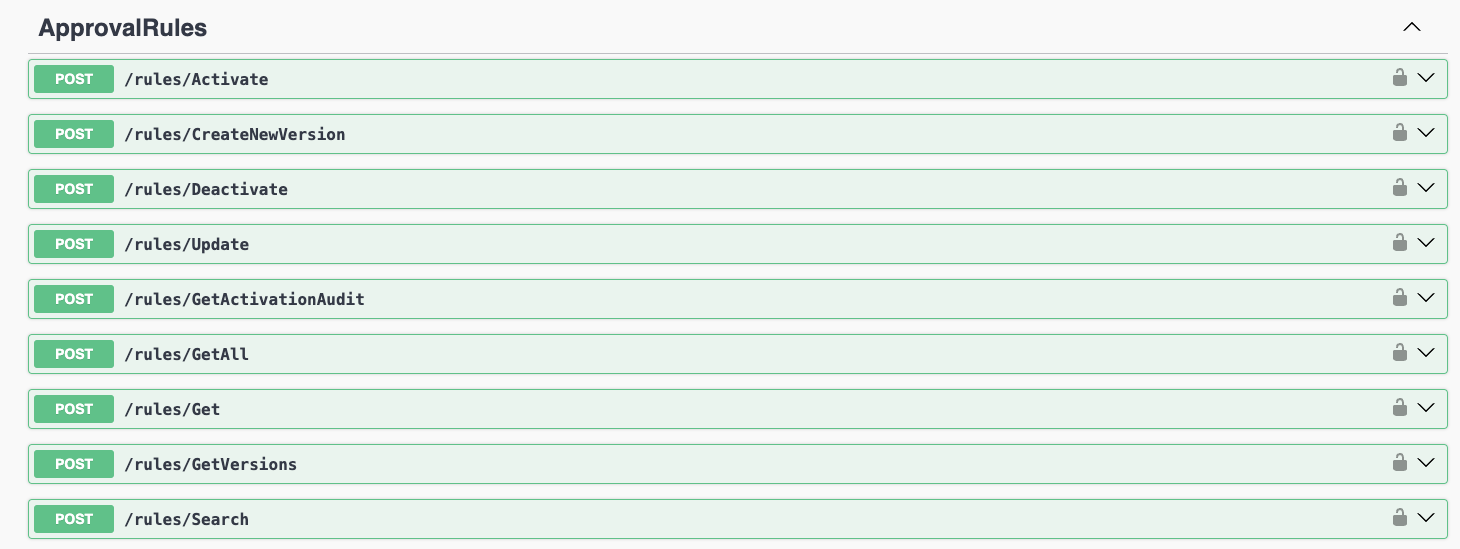
В приложении реализована возможность создавать и настраивать матрицы согласования документов.

Тестирование функционала проводилось в полном объеме с передачей некорректных данных форматом и бизнес значением, в том числе проводилось тестирование функционала:

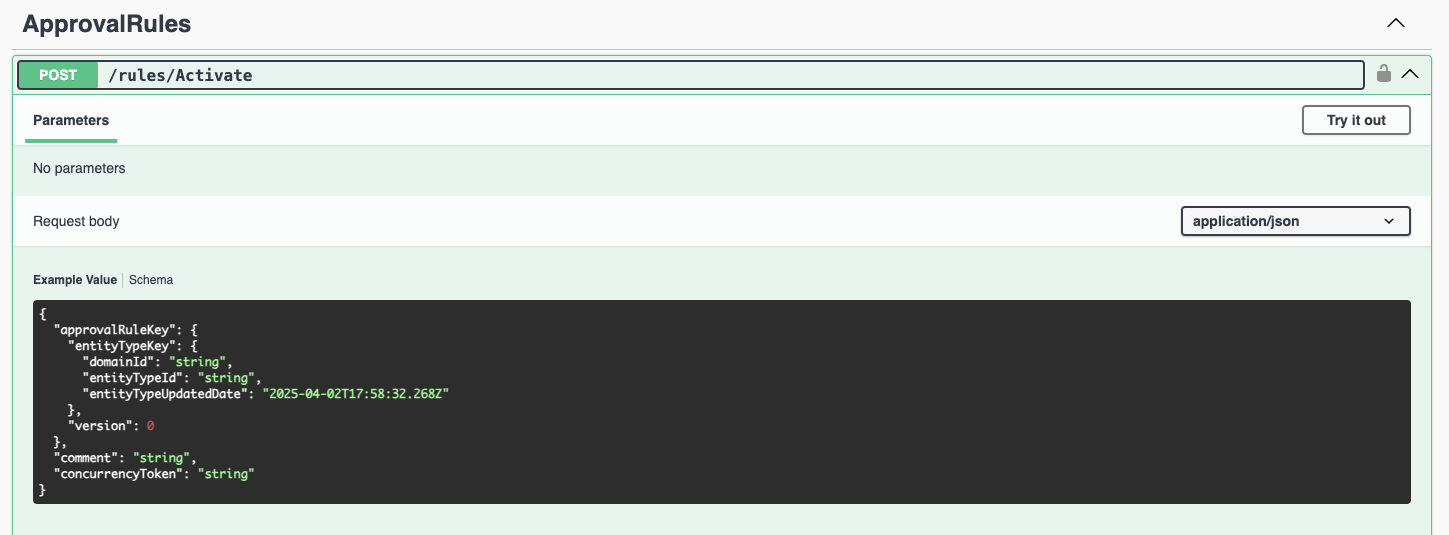
* Создания, обновления и удаления графов согласования;
* Создания, обновления и удаления групп согласования;
* Создания, обновления и удаления условных операторов для настройки ветвления и условий при определении маршрута согласования по графу и группам;
* Создания, обновления и миграции правил согласования, в том числе их расширенный аудит и функционал активации и деактивации версий, а также их миграции между версиями, как глобальными при изменении шаблона, так и своими внутренними;
* Построения маршрута согласования.



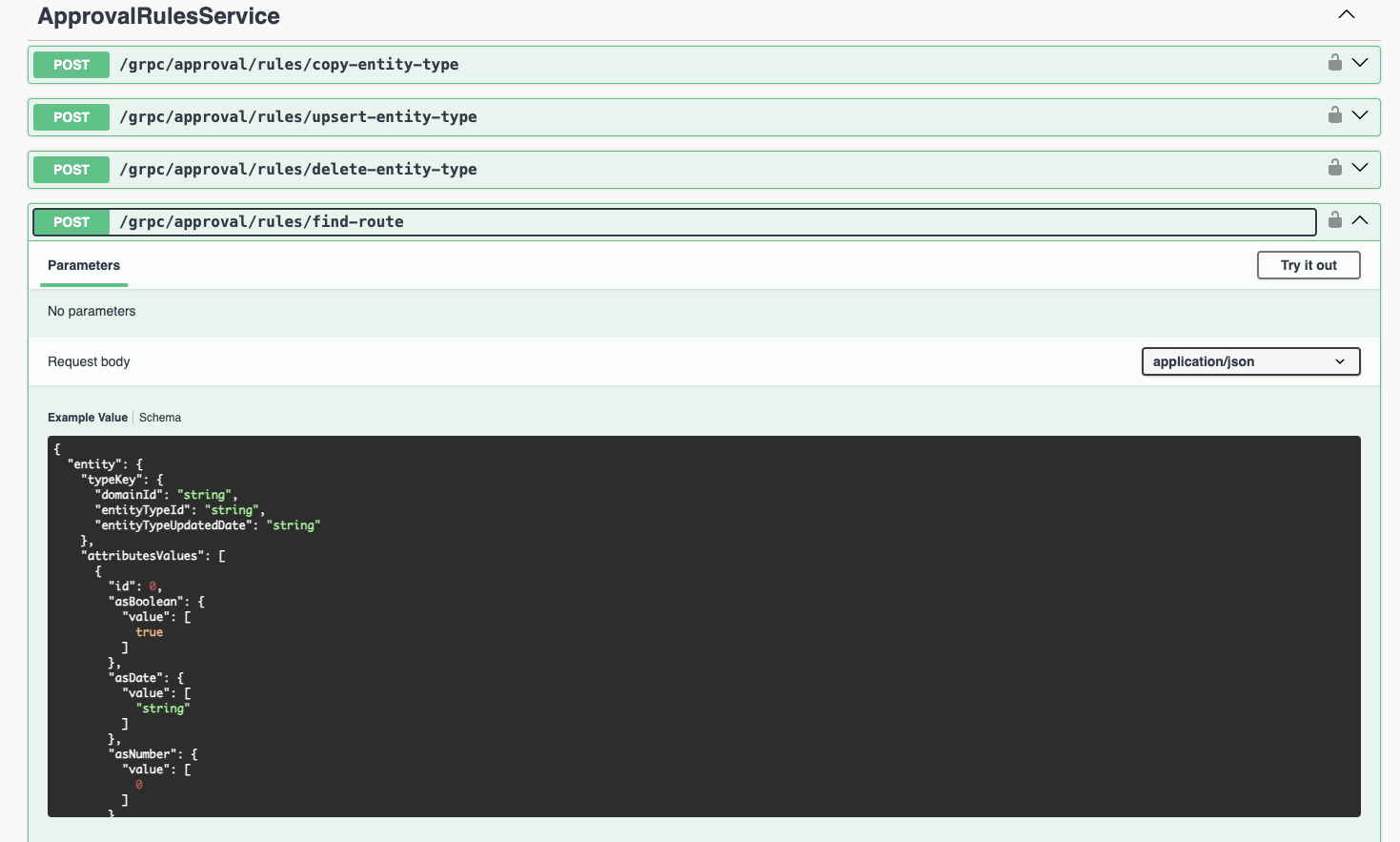
Приложение 4. Иллюстрация документации Swagger. Ручки для взаимодействия по доменам, условным операторам, группам и графам согласования



Приложение 5. Иллюстрация документации Swagger. Ручки для взаимодействия по матрицам согласования



Приложение 6. Иллюстрация документации Swagger. Ручка для активации матрицы согласования



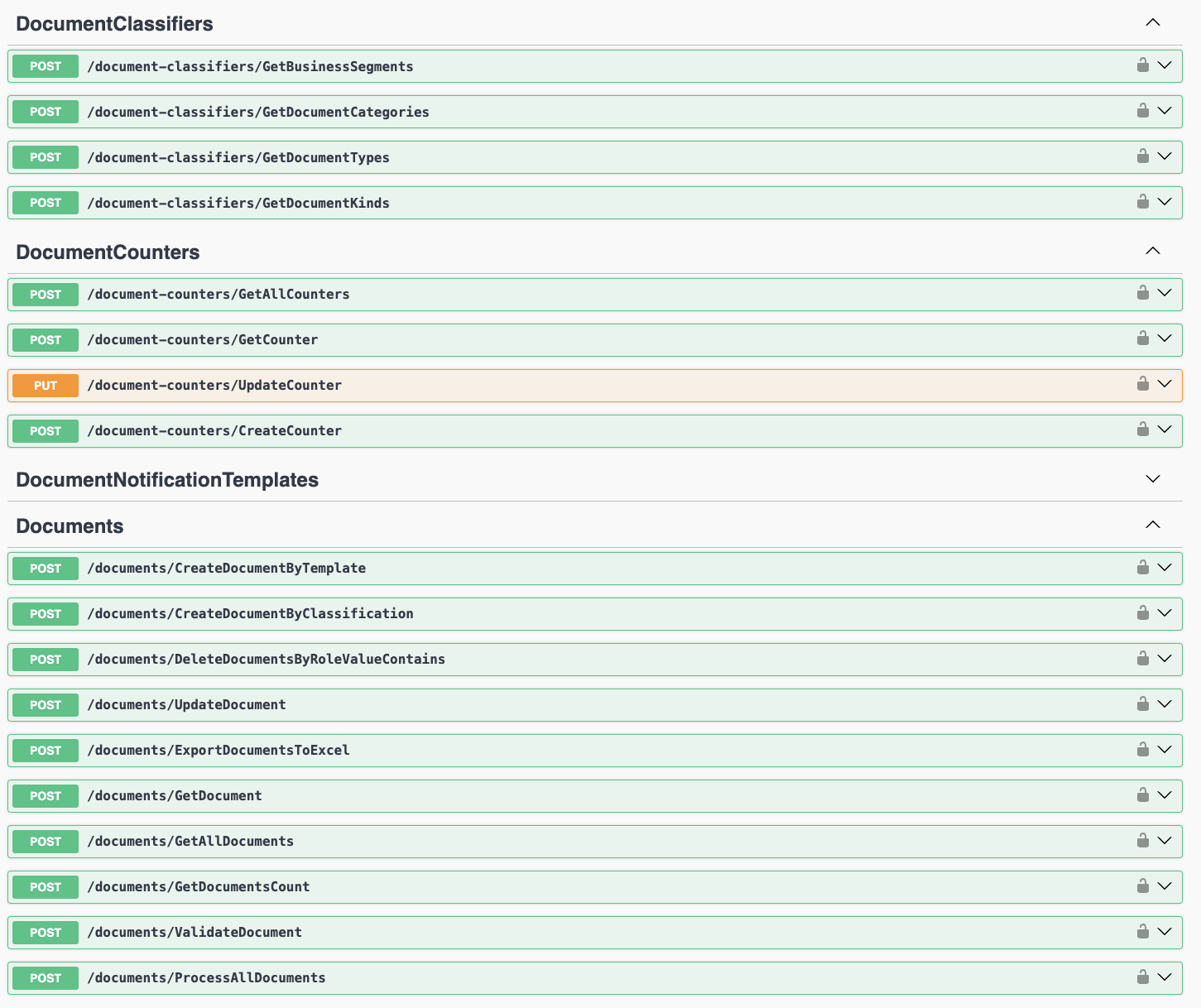
Приложение 7. Иллюстрация документации Swagger. Ручка для поиска маршрута согласования по матрице и значениям атрибутов документа

1. **Создание и работа с документами, согласование, подписание**

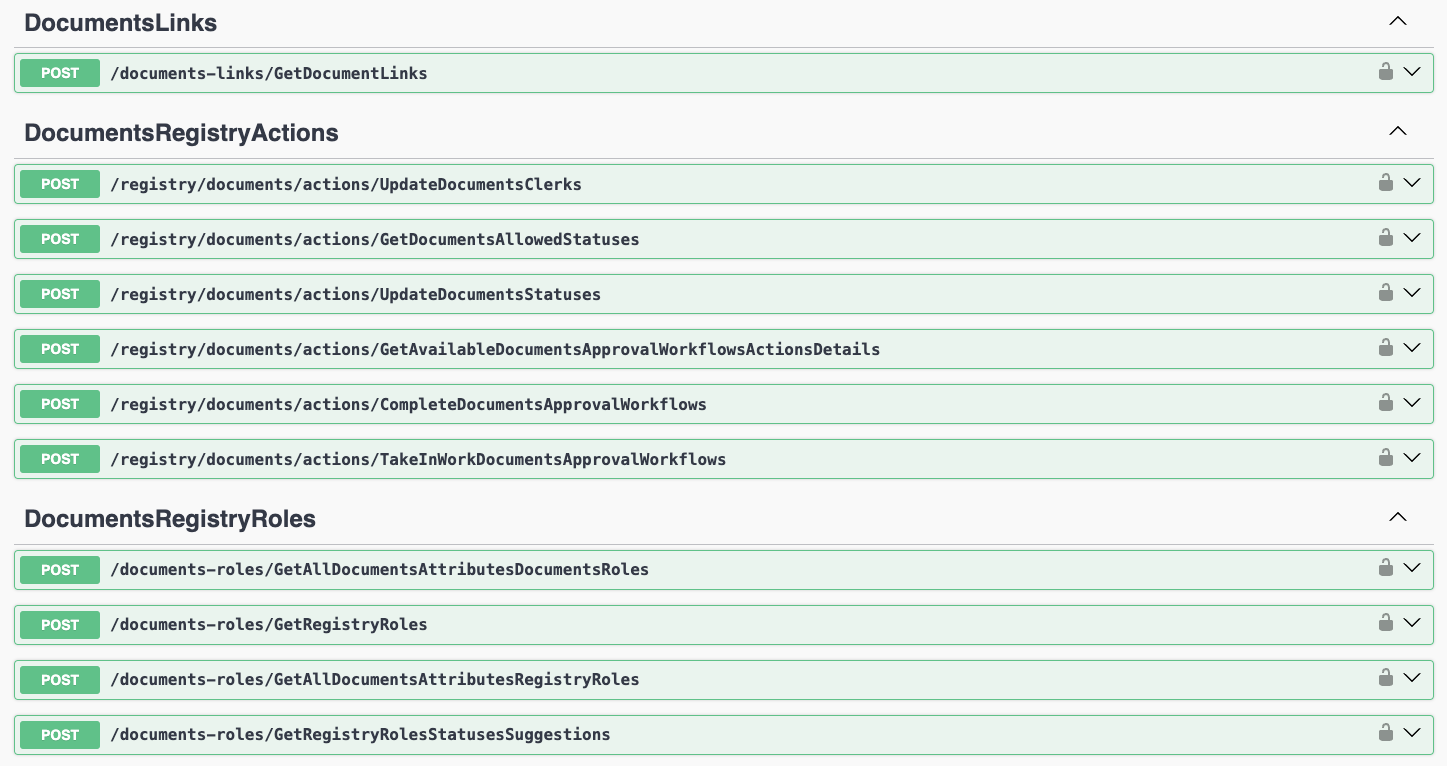
В приложении реализована возможность создавать, редактировать и в цедом проводить документы по бизнес флоу, в том числе согласовывать, подписывать и активировать их. Предусмотрен расширенный аудит и расширенное версионирование по версиям шаблонов документов.

Тестирование функционала проводилось в полном объеме с передачей некорректных данных форматом и бизнес значением, в том числе проводилось тестирование функционала:

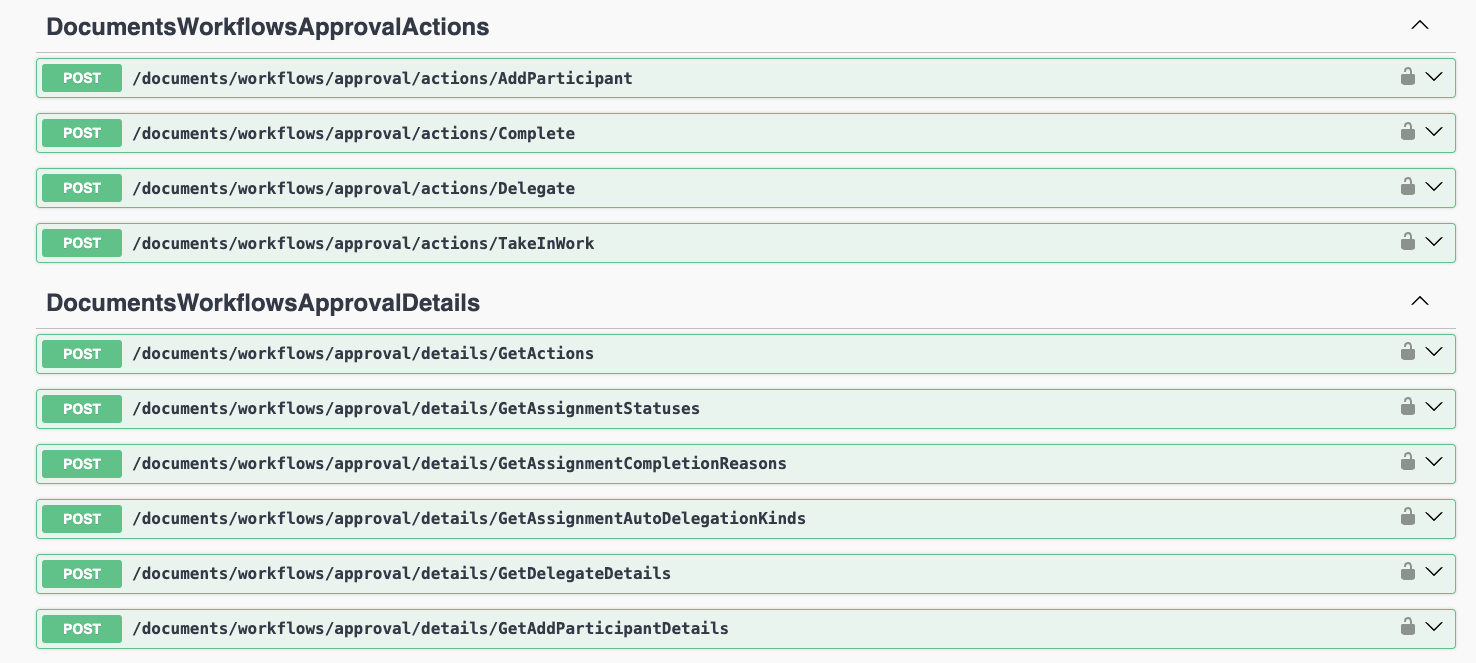
* Добавление, удаление и редактирование счетчиков. Генерация регистрационного номер с использованием счетчика;
* Получение информации по действиям, ролям в реестре и в документах по домену;
* Создание документов из шаблонов и с помощью классификаций. Редактирование и удаление документов. Смена статусов и работа валидаторов, уведомлений.
* Расширенный аудит и версионирование при изменении шаблонов и документов;
* Расширенный реестр документов с функционалом полноценной сортировки и фильтрации по любым определенным для домена ролям в документах;
* Процесс согласования документа, в том числе построения маршрута из матрицы, все варианты обработки этапов: делегация, согласование, отклонение, взятие в работу, добавление новых согласующих. Получение подробной информации по текущему состоянию согласования, в том числе в степпере документа;
* Процесс подписания документа, в том числе процесс электронного подписания с заглушкой на автоматический статус подписания из провайдера подписи. Получение подробной информации по текущему состоянию подписания, в том числе в степпере документа.



Приложение 8. Иллюстрация документации Swagger. Ручки для работы с классификациями, счетчиками и документами



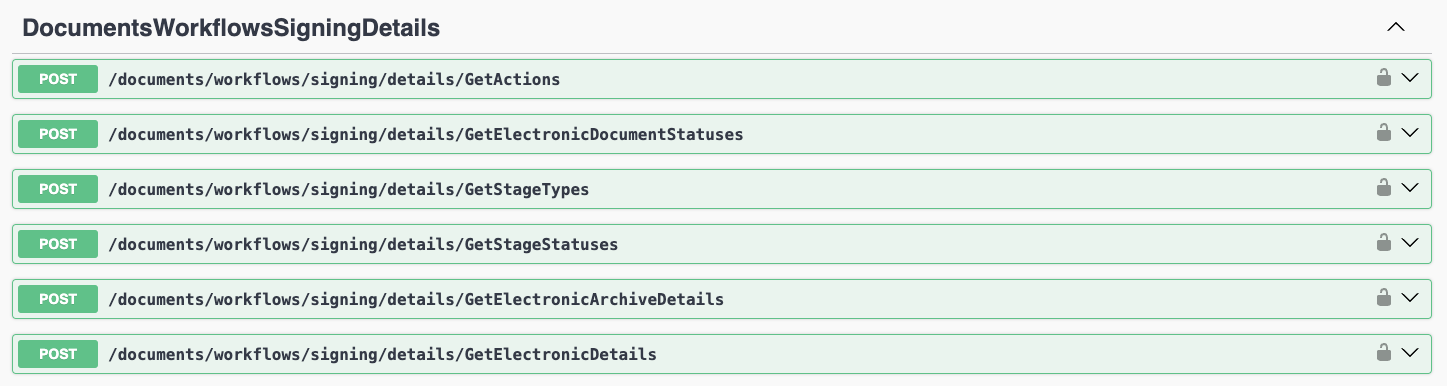
Приложение 9. Иллюстрация документации Swagger. Ручки для работы с ссылками, ролями и действиями с документами в реестре



Приложение 10. Иллюстрация документации Swagger. Ручки для работы с согласованием документов и получения всей необходимой информации



Приложение 11. Иллюстрация документации Swagger. Ручки для работы с подписанием документов (БДО/ЭДО)



Приложение 12. Иллюстрация документации Swagger. Ручки для получения информации по подписанию документов (БДО/ЭДО)

1. **Примеры тестов и тестовых данных**

Файлы в формате json с тестовыми запросами расположены в корневой директории исходного кода приложения, в папке test\_jsons. Их можно найти в github репозитории по ссылке: <https://github.com/nktrtmv/edm/tree/master/test_jsons>

В примерах тестов и тестовых данных можно найти:

1. Создание шаблона с классификацией через ручку `/templates/CreateDocumentTemplate` и проверка созданного шаблона через ручку `/templates/GetDocumentTemplate` (файлы 1-2)
2. Обновление шаблона с указанием атрибутивного состава (включая атрибутивный состав согласования), схемы состояний, нумератора для генерации регистрационного номера, валидаторов через ручку `/templates/UpdateDocumentTemplate` и проверка обновленного шаблона через ручку `/templates/GetDocumentTemplate` (файлы 3-4)
3. Создание двух групп согласующих через ручку `groups/Create` и их проверка через ручку `groups/Get` (файлы 5-6, 9-10)
4. Обновление двух групп согласующих через ручку `groups/Update` с указанием разных типов и основных видов условных операторов (в том числе вложенных) и их проверка через ручку `groups/Get` (файлы 7-8, 11-12)
5. Создание графа(дерева) согласования и его проверка через ручки `graphs/Create` и `graphs/Get` (файлы 13-14)
6. Обновление графа согласования с указанием вершины с условным оператором для разветвления дерева на две ветки согласования с разными группами согласующих и его активация через ручку `graphs/Update` и проверка корректности обновления через ручку `graphs/Get` (файлы 15-16)
7. Активация матрицы согласования и проверка активности версий матриц согласований для текущей версии шаблона через ручки `rules/Activate` и `rules/GetVersions` (файлы 17-18)
8. Активация шаблона документа (перевод в состояния готовности к созданию документов) через ручку `templates/UpdateDocumentTemplate` и проверка состояния через ручку `templates/GetDocumentTemplate` (файлы 19-20)
9. Создание документа из настроенного шаблона через ручку `documents/CreateDocumentByTemplate`, и проверка атрибутивного состава, схемы состояний и регистрационного номера документа через ручку `documents/GetDocument` (файлы 21-22)
10. Заполнение значений атрибутов документа и проверка их заполненности после через ручки `documents/UpdateDocument` и `documents/GetDocument` (файлы 23-24)
11. Перевод состояния документа в «ожидает обработки» и его проверка через ручки `documents/UpdateDocument` и `documents/GetDocument` (файлы 25-26)
12. Перевод состояния документа в «ожидает обработки» и его проверка через ручки `documents/UpdateDocument` и `documents/GetDocument` (файлы 25-26)
13. Перевод состояния документа в «согласование» через ручку `documents/UpdateDocument`, и проверка статуса документа с доступными действиями через ручку `documents/GetDocument`. Здесь же проверяется логика построения маршрутов согласования по указанным условным операторам в матрице согласования и значениям атрибутов в текущем документе (файлы 27-28)
14. Взятие документа в работу в процессе согласования через ручку `documents/workflows/approval/actions/TakeInWork`, и дальнейшая проверка состояния согласования через степпера документа через ручку `documents/GetDocument` (файлы 29-30)
15. Согласование документа через ручку `documents/workflows/approval/actions/Complete`, и дальнейшая проверка состояния согласования и документа через ручку `documents/GetDocument` (файлы 31-32)
16. Отправка документа на подписание обновлением состояния докумета через ручку `documents/UpdateDocument`, и последующая проверка документа ручкой `documents/GetDocument` (файлы 33-34)
17. Подписание документа стороной «собственной» компании через ручку `documents/workflows/signing/actions/SendToContractor` и проверка состояния подписания документа на степпере через ручку `documents/GetDocument` (файлы 35-36)
18. Подписание документа стороной контрагента и автоматический перевод состояния документа в «Действует» через ручку `documents/workflows/signing/actions/PutIntoEffect`, и последующая проверка состояния подписания и документа через ручку `documents/GetDocument` (файлы 37-38)

### 6.2.3. Организация входных и выходных данных

Интерфейс приложения соответствует требованиям и предоставляет http контроллеры, которые принимают запросы с query параметрами, json объектами в теле запроса и файлами в form запроса.

Выходные данные имеют схожий формат и так же соответствуют требованиям.

## 6.3. Испытание выполнения требований к временным характеристикам

Время отклика приложения замерялось с помощью утилиты в приложении Jetbrains Rider, а время ответа приложения замерялось с помощью логов приложения. Время отклика при тестировании не превышало 0.87 секунды, в том числе при нагрузочном тестировании на 50 RPS (requests per second – запросов в секунду).

Временные характеристики приложения соответствуют требованиям.

## 6.4. Испытание выполнения требований к надежности

При тестировании приложение не завершалось аварийно при любом наборе входных, в том числе при некорректных данных и форматом и наполнением.

Программа не дает сбой при некорректных действиях пользователей.

Стоит отметить, что при нахождении пользовательского сценария, который может привести к аварийному завершению или сбою приложения, предусмотрена система автоматического перезапуска приложения.

Таким образом, надежность приложения соответствует требованиям.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19.101-77 Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.;
2. ГОСТ 19.102-77 Стадии разработки. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.;
3. ГОСТ 19.103-77 Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.;
4. ГОСТ 19.104-78 Основные надписи. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.;
5. ГОСТ 19.105-78 Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.;
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.;
7. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.;
8. ГОСТ 19.603-78 Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.;
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
11. Docker. [Электронный ресурс] – URL: <https://www.docker.com/> режим доступа: свободный (дата обращения: 13.03.25).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**