

Техническое задание
маршрутизатор ЕМИАС

СОДЕРЖАНИЕ

1 ОБЩИЕ СВЕДЕНИЯ	2
1.1 Полное наименование Сервиса и его условное обозначение	2
1.2 Термины и определения	2
2 НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СЕРВИСА	3
2.1 Назначение Сервиса	3
2.2 Цели создания Сервиса	3
3 ТРЕБОВАНИЯ К СЕРВИСУУ	4
3.1 Общие требования	4
3.2 Требования к интеграции с внешними Сервисами	4
3.3 Требования к надежности	4
3.4 Требования по обеспечению возможности развития и модернизации Сервиса	4
3.5 Требования к защите информации от несанкционированного доступа	4
3.6 Требования к функциям	5
3.7 Требования к программному обеспечению	5
4 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ И ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ	6

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Полное наименование Сервиса и его условное обозначение

Полное наименование: сервис маршрутизации

Условное обозначение: СМ.

1.2 Термины и определения

Термин	Описание
СМ	Сервис Маршрутизации - итоговый программный результат, описанные в данном документе
ВСЕ	Внутренняя система ЕМИАС
Документ 230	Единая точка доступа. Реализация функций (версия 0.1 от 01.02.2021)

2 НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СЕРВИСА

2.1 Назначение Сервиса

Сервис предназначен осуществлять следующее:

- Прием и обработка параметров исходного url, переданного клиентом
 - Предоставление целевого url клиенту
- Настройка параметров маршрутизации пользователем. Подробнее см. документ 230
 - Хранение текущих параметров маршрутизации

2.2 Цели создания Сервиса

Целью создания Сервиса является выполнение технического тестового задания

3 ТРЕБОВАНИЯ К СЕРВИСУ

3.1 Общие требования

Сервис должна интегрироваться с существующими информационными Системами ЕМИАС, не нарушая рабочие процессы, сетевую и вычислительную инфраструктуру ЕМИАС.

3.2 Требования к интеграции с внешними Сервисами

Сервис не должен напрямую взаимодействовать ни с какими внешними системами и/или другими сервисами

3.3 Требования к надежности

Компоненты Сервисы должны быть рассчитаны на функционирование в режиме круглосуточной работы, и позволять осуществлять выполнение процедур резервного копирования данных и конфигураций, восстановления Сервисы после сбоев.

3.4 Требования по обеспечению возможности развития и модернизации Сервиса

Реализация Сервиса должна обеспечивать возможность обновления используемых программных продуктов в случае их развития, а также расширение функциональности Сервиса в случае необходимости. При этом должна сохраняться непрерывность работы, обеспечена непрерывность поступления данных и их сохранность.

Должны учитываться возможности масштабирования Сервиса, связанные с увеличением числа защищаемых объектов, выделением новых сетевых сегментов, появлением локальных или удаленных сетей.

3.5 Требования к защите информации от несанкционированного доступа

В рамках данного задания не должны быть реализованы меры защиты информации от несанкционированного доступа, обеспечения целостности и доступности указанной информации. Аутентификация и авторизация пользователей не предусмотрена

3.6 Требования к функциям

Сервис должен предоставлять клиентам следующие функциональные возможности:

- прием и обработка параметров исходного url, переданного клиентом
 - предоставление целевого url клиенту
- хранение текущих параметров маршрутизации

3.7 Правила маршрутизации

Данные правила описывают бизнес логику согласно документу 230.

3.7.1 Базовое описание правил

У администратора системы должна быть возможность редактировать (добавление, удаление, изменение) правила маршрутизации. Правила могут быть составлены только из параметров, присутствующих в настройках сервиса маршрутизации

Наименование правила	Параметр 1	Параметр 2	Параметр N	Целевой URL
Старый интерфейс	medicalEntityID = all	roleID = 4, 5		
Новый интерфейс	medicalEntityID = 1,2	roleID = 1		
Листы ожидания				

Создание правил:

- В одном правиле не допускается наличие двух одинаковых параметров (данное ограничение необходимо добавить так же на уровне базы данных)
- Правила могут быть составлены только из параметров, присутствующих в настройках сервиса маршрутизации
 - **При создании правила, оно не должно конфликтовать с уже существующими*** (данное условие не является обязательным при выполнении технического задания)

При настройке правил маршрутизации должна быть возможность задать:

- несколько значений для одного параметра (перечисление)
- любое значение для параметра (в виде значения “all”)
- Целевой URL для каждого правила, как обязательный параметр

Исполнение правил:

- Система сравнивает параметры переданные в запросе клиента с правилами указанными в настройках сервиса.
- Для одного запроса маршрутизации может быть исполнено только одно правило.
 - В ответ на запрос клиента, сервис может вернуть только один целевой url.
 - Параметры в рамках одного запроса могут быть переданы сервису маршрутизации в любом порядке.
- Правило складывается из объединения всех ключей «Параметр – Значения» логическим «И».
- Если система определила в запросе клиента параметр, для которого в правиле указано значение all, при этом нет других правил, в которых задано частное значение для данного параметра, тогда выполняется правило, где указано all.

Пример запроса клиента:

<https://service.com/v1/rule?medicalEntityID=2,role=4>

Пример определения правила:

Наименование правила	Параметр 1	Параметр 2	Целевой URL
Старый интерфейс	medicalEntityID = all	roleID = 4, 5	urlOLDUI
Новый интерфейс	medicalEntityID = all	roleID = 6, 7	urlNEWUI

Если система определила в запросе клиента параметр, для которого в правиле указано значение all, при этом, и есть хотя бы одно другое правило, в котором задано частное значение для данного параметра, то вначале система выполняет правило, где значение клиента совпадает с частным значением параметра. Иначе, выполняется правило, где указано all.

Пример запроса клиента:

<https://service.com/v1/rule?medicalEntityID=2,role=4>

Пример определения правила:

Наименование правила	Параметр 1	Параметр 2	Целевой URL
Старый интерфейс	medicalEntityID = all	roleID = 4, 5	urlOLDUI
Новый интерфейс	medicalEntityID = 2	roleID = 4, 5	urlNEWUI

Новый интерфейс	medicalEntityID = 2	roleID = 9, 8	urlOLDUI
-----------------	---------------------	---------------	----------

3.7.2 Параметры маршрутизации

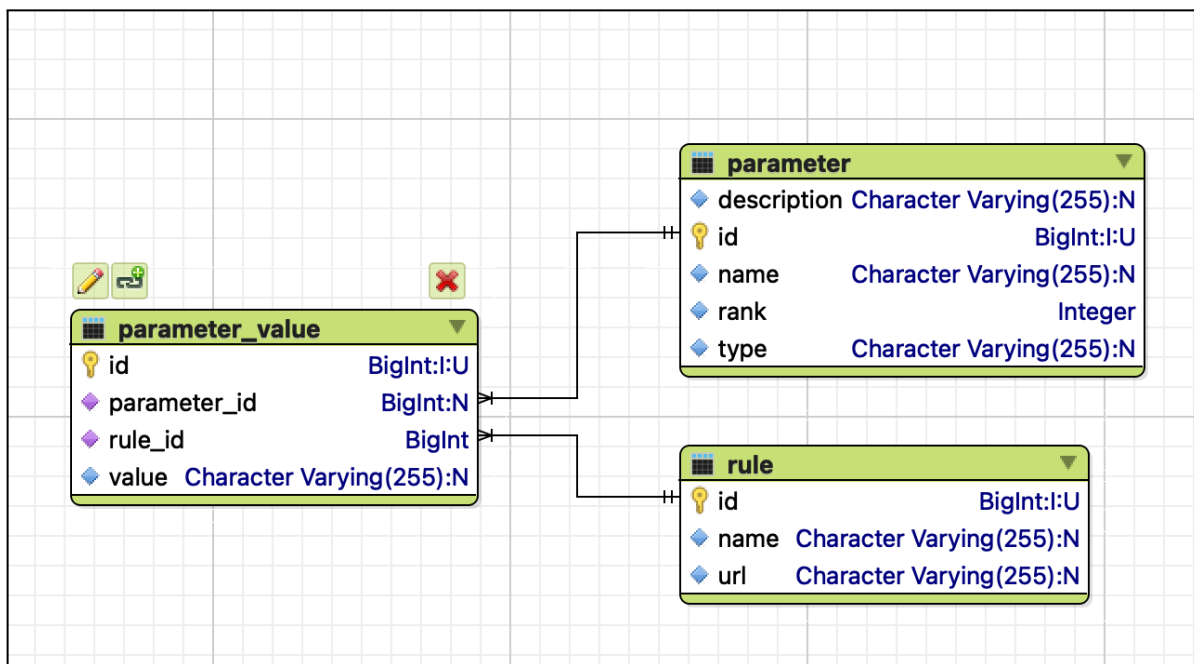
Каждый параметр описывается следующим набором атрибутов:

- Наименование параметра, т.е. имя которое должно передать на вход маршрутизатору клиентское приложение
- Тип – тип данных параметра, возможные варианты: текст (Рус, Англ, возможны спец. символы), логический (true/false), числовой (целые числа). Тип задан условно, фактический тип будет зависеть от реализации.
- Описание – текстовое описание параметра
- Приоритет (rank) – то в каком порядке параметр будет выставлен (обработан в правиле) в таблице маршрутизации. Наименьший rank – должен быть обработан в первую очередь. Если rank не задан, то обработка правил выполняется в порядке, в котором значения параметров были переданы на вход методу клиентом.

Наименование	Тип	Описание	Приоритет обработки (rank)
KPIMenulvl1	текст	Навигационное меню КПИ первого уровня	2
KPIMenulvl2	текст	Навигационное меню КПИ второго уровня	3
medicalEntityID	текст	ИД ЛПУ	1
roleID	текст	ИД роли пользователя	4
orgId	текст	ИД подразделения пользователя	5
medicalEmployeeJobInfoID	текст	ИД исполнения должности пользователя	6
availableResourceId	текст	ИД Доступного ресурса	
complexResourceId	текст	ИД Комплексного ресурса	
SourceLink	текст	Анализируется адрес с, которого был выполнен переход к маршрутизатору	

3.7.3 Схема данных сервиса

Физическая схема хранения правил:



3.7.4 Требования к оптимизации

Для достижения наилучшего быстродействия, следует добавить сохранение успешных результатов ответа сервиса (со статусом 200) в кэш.

В качестве кэш хранилища использовать Redis.

В качестве ключа записи использовать сериализованные в виде строки параметры запроса. Если в запросе было несколько параметров, то перед сериализацией они должны бы отсортированы.

При создании нового либо изменении уже существующего правила - все кэшированные результаты должны быть инвалидированы* (данное условие не является обязательным при выполнении технического задания)

3.7.5 Версионирование AOU

3.8 Требования к программному обеспечению

3.8.1 Требования к совместимости

Программное обеспечение должно функционировать на базе ОС Debian 7 или выше, Ubuntu 18.04 или выше.

Программное обеспечение должно быть реализовано на базе стека Java, Spring Boot.

Для запуска на заявленных ОС, других платформах, а так же для локальной разработки итоговое решение должно иметь Dockerfile для каждого элемента системы, а также docker compose файл для запуска всех элементов системы разом.

4 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ И ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ

N	Вопрос / задача	Описание реализации
1.	Производительность	Система должна обрабатывать не менее 1000 запросов в секунду
2.	Безопасность	Все пользовательские данные, передаваемые по сети, должны быть зашифрованы с использованием надежного протокола шифрования, такого как SSL/TLS, чтобы предотвратить несанкционированный доступ к конфиденциальной информации.
3.	Масштабируемость	Система должна быть способна легко масштабироваться для обработки увеличения нагрузки. Например, она должна поддерживать горизонтальное масштабирование, позволяя добавлять дополнительные серверы для распределения нагрузки.
4.	Надежность	Система должна иметь высокую доступность и минимальное время простоя. Например, система должна автоматически переключаться на резервные серверы в случае сбоя основного сервера, чтобы обеспечить непрерывную работу.
5.	Совместимость	Система должна быть совместима с различными операционными системами, браузерами и устройствами. Например, она должна поддерживать работу на Windows, macOS, Linux и мобильных платформах.
6.	Поддержка	Должна быть предоставлена документация, руководства пользователя и возможность получения технической поддержки в случае возникновения проблем.
7.	Тестируемость	Приложение должно быть легко тестируемым, с возможностью автоматического тестирования и регрессионного тестирования.