



Сервисы Smart Services

Версия 2022.7

Содержание

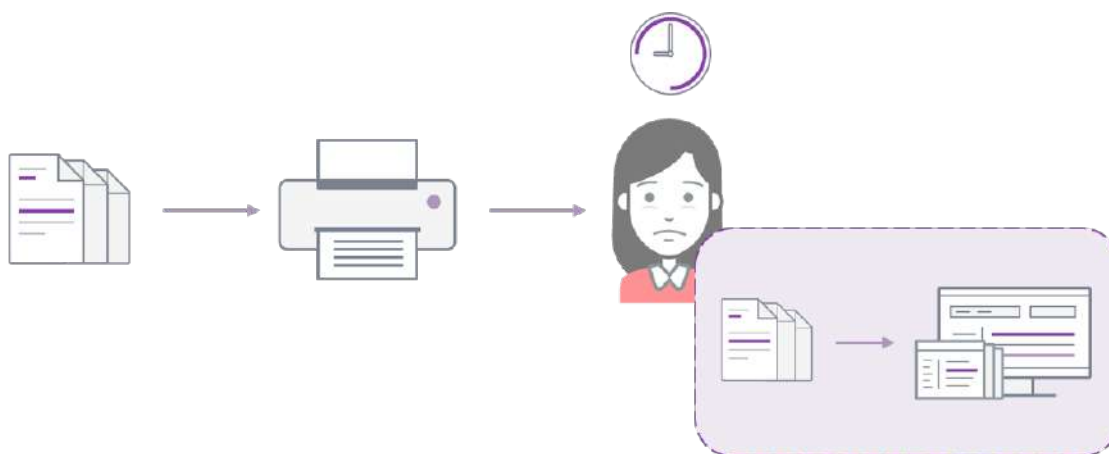
Введение.....	4
Основные понятия.....	6
Архитектура.....	9
Схема обработки документов.....	11
Системные требования.....	13
Требования к ПО серверов с сервисами.....	16
Развертывание сервисов.....	17
Подготовка к развертыванию.....	17
Установка Docker Engine.....	17
Установка Docker Compose.....	20
Развертывание.....	21
Подключение дополнительных серверов.....	24
Завершение развертывания.....	25
Запуск и настройка инструмента checker.html.....	27
Установка инструментов Elasticsearch.....	28
Установка поисковой системы Elasticsearch.....	28
Установка сервиса Filebeat.....	32
Обновление сервисов.....	34
Сервисы установлены на одном сервере.....	34
Сервисы установлены на разных серверах.....	34
Удаление сервисов.....	35
Настройки для обработки документов.....	36
Установка и настройка Postman.....	36
Импорт коллекции запросов.....	36
Настройка коллекции Postman.....	38
Принципы отправки запросов через Postman.....	38
Получение токена для аутентификации пользователей.....	40
Настройка классификации документов.....	41
Подготовка документов для обучения классификатора.....	43
Создание классификатора.....	45
Обучение классификатора.....	46
Анализ результатов обучения.....	49
Тестирование классификатора.....	50
Дообучение классификатора.....	51
Информационные запросы.....	53
Иерархическая классификация.....	56
Импорт и экспорт прикладной конфигурации.....	59
Сопровождение работы.....	61
Конфигурирование настроек.....	61
Smart Service.....	68
Identity Service.....	71
Publishing Service.....	72
Text Extractor Service.....	74
Text Classifier Service.....	77
Image Classifier Service.....	80

Fact Extractor Service	81
Fact Extractor Rules Service.....	82
Fact Extractor Learn Service	84
Object Detection Service	87
Storage Service.....	88
Elastic APM Service	88
InfluxDB	90
Grafana	90
Telegraf	91
Общие параметры.....	91
Устранение неисправностей.....	92
Анализ логов сервисов Smart Services.....	94
Анализ трассировок запросов.....	94
Примеры анализа и устранения ошибок	97
Справочная информация.....	99
Виды обрабатываемых документов.....	99
Список извлекаемых фактов.....	100
Ограничения при извлечении.....	108
Печатей и подписей.....	108
Фактов из скан-копии документа.....	108
Список распознаваемых языков.....	109
Сторонние компоненты.....	111

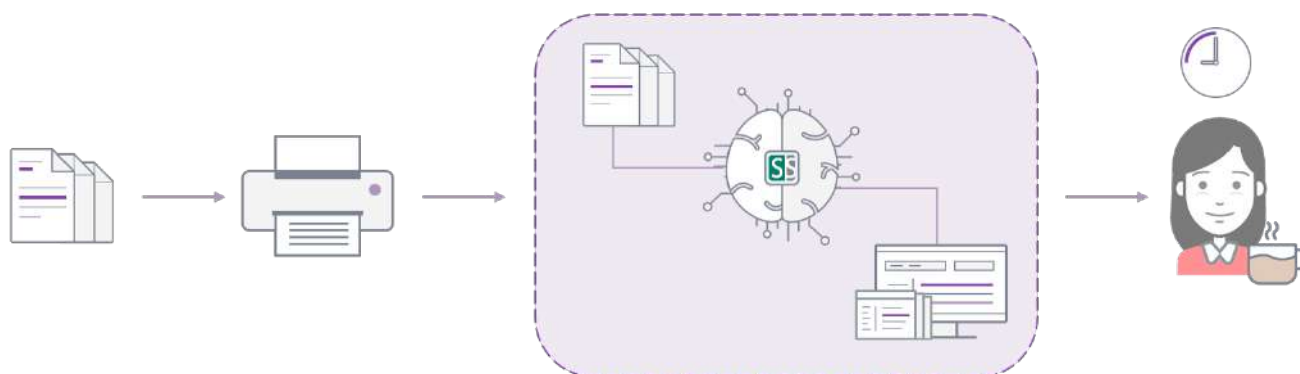
Введение

Сервисы Smart Services основаны на интеллектуальных инструментах, использующих технологии компьютерного зрения и обработки естественного языка, а также алгоритмы машинного обучения. Сервисы уменьшают количество рутинных операций в информационных системах компании. Например, облегчают обработку входящего потока: за человека классифицируют документы и извлекают из них значимую информацию.

Обработка **без Smart Services**



Обработка **со Smart Services**



Сервисы обрабатывают как объекты, содержащие текст, так и документы без текстового слоя (отсканированные документы, PDF-документы без текстового слоя и т.д.):

- определяют наличие и местоположение печатей и подписей и ориентацию страниц;
- извлекают текстовый слой из изображений и документов основных форматов;
- классифицируют документы с помощью специальных классификаторов, обученных на подготовленном наборе данных: документах, записях справочников и т.п.;
- извлекают из текста значимые данные, факты и приводят их к единому формату. Например, дату и номер документа, ИНН и КПП контрагента;
- рассчитывают вероятность правильного извлечения данных из текста.

Все веб-сервисы независимы и каждый из них выполняет узкоспециализированную задачу. Такая [архитектура](#) обеспечивает стабильность работы и позволяет постепенно добавлять в систему новые интеллектуальные механизмы.

Задействуйте возможности сервисов Smart Services, чтобы автоматизировать обработку входящего документопотока в информационной системе компании. Для этого:

1. [Установите](#) или [обновите](#) сервисы Smart Services, учитывая [требования](#), необходимые для работы сервисов.
2. При помощи Postman [настройте сервисы для обработки документов](#).
3. Используйте [конфигурационные файлы](#), если необходимо выполнить дополнительные или изменить текущие настройки сервисов.
4. [Обеспечьте стабильность работы](#) и производительность сервисов. Например, используйте инструменты для анализа и своевременного устранения ошибок в работе сервисов.

Установку, настройку и сопровождение работы сервисов выполняет администратор.

Подробнее о сервисах также см. в разделах [«Основные понятия»](#), [«Справочная информация»](#). Последний содержит сведения информационного характера, например список обрабатываемых видов документов.

Основные понятия

Атрибут

То, чему соответствует *факт* в разрезе *группы атрибутов*. Например, можно присвоить факту атрибут «Дата» или «№ документа».

Грамматика

Совокупность правил и *моделей извлечения фактов, форматеров*, признаков фактов, а также правил расчета вероятностей для определенного класса документов.

Группа атрибутов

Набор *атрибутов*, объединенных по смыслу. Например, атрибуты «Дата» и «№ документа» можно объединить в группу атрибутов «Данные».

Классификация

Этап обработки документа, при котором сервис по тексту определяет, к какому классу он относится. В качестве классов обычно выделяют виды документов: входящее письмо, договор, акт выполненных работ и т.д. Для правильной классификации документов требуется обучить *классификатор*, предоставив ему хотя бы по 100 документов каждого класса. В результате классификации формируется *модель классификации*.

Классификация может быть:

- двухклассовая, при которой документу присваивается один из двух классов. Например, при определении ориентации страниц документа;
- многоклассовая, при которой документу присваивается один из нескольких классов. Например, при определении вида документа;
- многозначная (нечеткая), при которой документу может быть присвоен один или несколько классов. Например, при определении ответственных за документ;
- [иерархическая](#), при которой документы классифицируются сначала по группам, а затем по классам. Используется для классификации большого количества документов, например 1000.

Классификатор

Программный модуль, выполняющий *классификацию* документов.

Ключевое слово

Слово или словосочетание в тексте документа, по которому сотрудник определяет значимые *факты*. Как правило, ключевое слово предшествует факту. Например, ключевыми словами являются «ИНН» и «КПП».

Коллекция запросов Postman

Проект, который содержит сгруппированные по типам задач связанные запросы к API. С помощью готовых коллекций легче автоматизировать выполнение запросов, так как не нужно писать их с нуля.

Модель извлечения фактов

Совокупность файлов, которые содержат набор значимых данных для каждого класса документов, а также скрипты и алгоритмы, выполняющие извлечение фактов.

Модель классификации

Совокупность файлов, содержащих набор признаков, характерных для каждого класса, управляющих скриптов и алгоритмов, выполняющих *классификацию*.

Область

Выделенный геометрической формой участок документа, который является значимым *фактом*.

Признаки фактов

Свойства текста, на наличие которых проверяется каждое слово в документе при обучении *моделей извлечения фактов*. Модели обучаются определять *факт* на основании особенностей написания и расположения слов, а также на основании проверки их соответствия записям специальных словарей. Например, по признаку **CountriesRus** для слов «Российская» и «Федерация» обнаружится совпадение в словаре названий стран на русском языке, который содержит запись «Российская Федерация».

Прикладная конфигурация Smart Services

Отделяемая часть сервисов Smart Services, которая содержит *грамматику*, библиотеки и состав извлекаемых *фактов*.

Разметка

Совокупность всех *областей* документа. Для каждой области указаны *атрибуты*.

Стиль разметки

Визуальное представление *разметки*. В базовую поставку включены несколько стилей, которые различаются цветом линий размеченных фактов и видом подписи атрибутов.

Факт

Значимые данные документа, которые необходимо извлечь из текста и заполнить в карточку документа.

Форма документа

Совокупность реквизитов документа, расположенных в определенной последовательности.

Форматер

Механизм для постобработки фактов после их извлечения. Форматеры делятся на:

- форматеры *фактов*, которые выполняют постобработку каждого факта. Например, форматеры дат документов, сумм, ФИО персон или темы письма. Используются для удаления лишних символов, изменения регистра и форматирования значений.

- формтеры правил, которые выполняют постобработку на уровне всех фактов, извлеченных из документа. Например, формтеры контрагентов в товарных накладных или адресатов в письме. Используются для объединения нескольких фактов в один и удаления незначимых данных документа.

Шаблон авторазметки

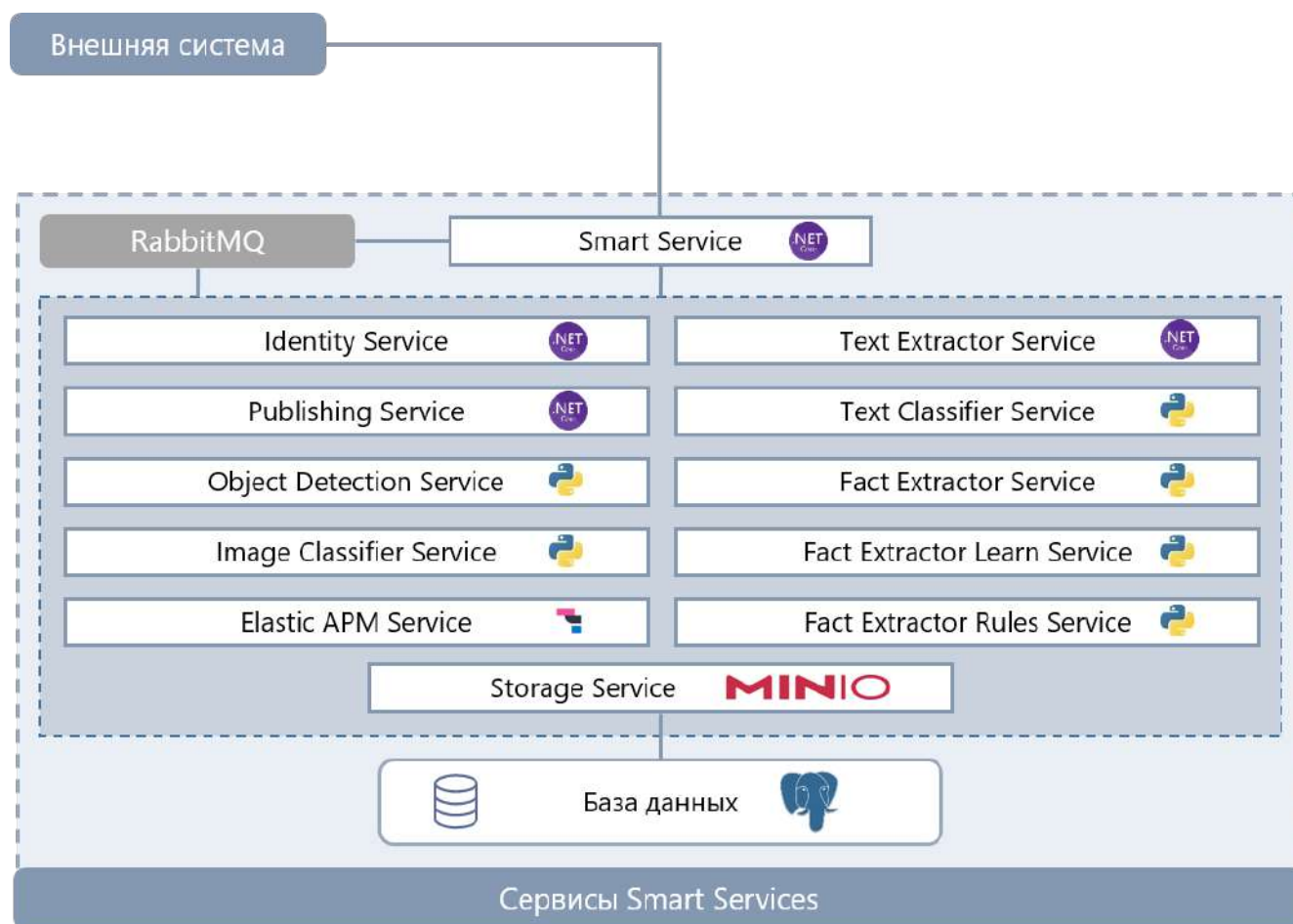
Набор *областей* и *атрибутов* для конкретного класса документов. Например, при автоматической разметке по шаблону «Письмо» документы размечаются в соответствии с типовым расположением *фактов* в письмах.

Архитектура

Smart Services – система сервисов, основанная на инструментах искусственного интеллекта и предназначенная для распознавания значимых фактов в документах. Сервисы Smart Services доступны в облачной поставке по модели SaaS, а также в локальной поставке:

- **Облачная поставка** обеспечивает быстрый старт без большого бюджета и снижает затраты на развертывание и обслуживание сервисов. При этом облачные сервисы сохраняют всю богатую функциональность и готовы к работе с первого дня;
- **Локальная поставка** доступна, если в компании развита внутренняя IT-инфраструктура с поддерживающими ее сотрудниками. Она предоставляет расширенные возможности адаптации сервисов, позволяя вносить изменения в их базовые механизмы.

Сервисы Smart Services созданы при помощи кроссплатформенных инструментов – платформы .Net Core и языка Python. Обработка результатов распознавания выполняется во внешних системах, для интеграции с которыми используется REST API.



Smart Service – управляющий сервис системы, предоставляющий доступ к интеллектуальным механизмам. Он выполняет следующие функции:

- организация взаимодействия сервисов. Взаимодействие реализуется посредством обмена сообщениями через менеджер RabbitMQ;
- обработка запросов на распознавание документов от внешней системы.

Состав сервисов Smart Services

Smart Services включает в себя следующие сервисы:

- **Identity Service** реализует авторизацию и аутентификацию пользователей в Smart Services. Сервис отвечает за создание и хранение учетных записей, а также выдачу и валидацию токенов доступа. Это позволяет управлять доступом к данным и обеспечивает дополнительную безопасность.

При запуске сервисов управляющий сервис запрашивает у Identity Service ключ для валидации токенов. Если сервисы установлены локально или в частном облаке, дополнительно запрашивается токен анонимного пользователя для работы без авторизации.

Авторизация позволяет ограничить права доступа на:

- результаты классификации;
- *классификаторы*;
- результаты извлечения фактов;
- *грамматики*.

Признак использования аутентификации системы устанавливается в [конфигурационном файле](#) сервиса Smart Service;

- **Image Classifier Service** определяет документы жестких форм и ориентацию страниц;
- **Object Detection Service** определяет наличие и местоположение подписей на документе. При обработке документа сервис возвращает информацию о позиции объекта в документе: номер страницы, на которой расположен объект, а также его абсолютную позицию в пикселях относительно размера этой страницы;
- **Text Extractor Service**:
 - извлекает текст из изображений и текстовых документов. Поддерживаемые форматы: JPG, JPEG, PNG, BMP, GIF, TIF, TIFF, PDF, DOC, DOCX, DOT, DOTX, RTF, ODT, OTT, TXT, XLS, XLSX, ODS;
 - выполняет предварительную обработку изображений и постобработку текста: делит файл на несколько документов, конвертирует документы в формат PDF с текстовым слоем для поиска по тексту документов;
 - извлекает метаданные (позиции слов, блоков текста) и сохраняет их в PDF-документ. Извлеченные данные используются повторно, если в документ не добавлялся текст, не менялись ориентация страниц и [язык распознавания](#);
 - определяет местоположение печатей и таблиц;
- **Text Classifier Service** обучает классификаторы, создает модели классификации и классифицирует документы в разных разрезах, например по видам: договоры, служебные записки, акты, приказы и т.д. Создание, обучение и дообучение классификаторов выполняется при помощи набора инструментов [Postman](#);
- **Fact Extractor Service** включает в себя:
 - **Fact Extractor Base Service**, который рассчитывает вероятность правильного извлечения данных из текста и приводит факты к единому формату, например, удаляет лишние символы или изменяет регистр текста;
 - **Fact Extractor Rules Service**, который извлекает факты, в том числе данные таблиц, на основе правил;

- **Fact Extractor Learn Service**, который извлекает факты на основе обучаемых моделей;
- **Storage Service** хранит большие объемы данных, за счет чего уменьшается размер сообщений, которые сервисы передают друг другу. Таким образом ускоряется обработка запросов сервисами;
- **Publishing Service** импортирует и экспортирует прикладную конфигурацию сервисов, облегчая ее доработку и обновление;
- **Elastic APM Service** отслеживает запросы между всеми перечисленными сервисами при обработке документов и измеряет скорость работы. Результаты обработки запросов записываются в [файлы трассировки](#), которые можно [проанализировать с помощью интерфейса Kibana](#).

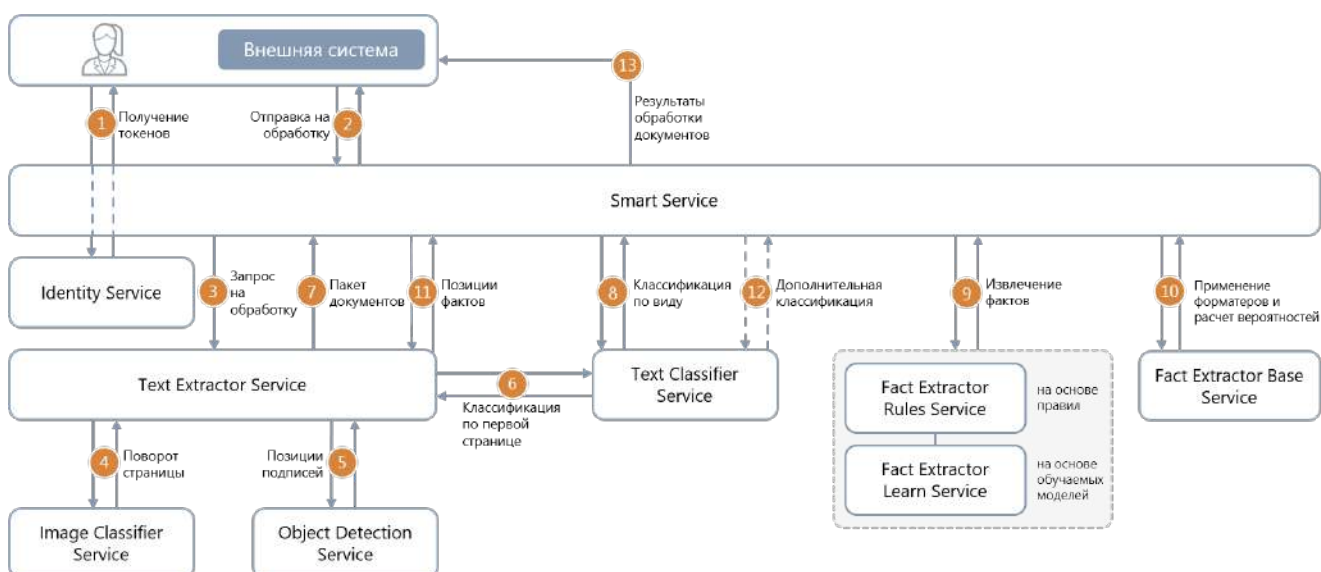
На **операционных системах на базе Linux** каждый сервис разворачивается в отдельном Docker-контейнере. Для отслеживания метрик контейнеров используется веб-приложение [Grafana](#). Собранные метрики передаются для хранения в базу данных [InfluxDB](#) при помощи агента [Telegraf](#).

Для повышения производительности в системах с большим объемом входящего документопотока рекомендуется установить часть сервисов на отдельных серверах. Подробнее см. в системных требованиях, подраздел [«Обработка более 90 000 документов в месяц»](#).

Интеграция с внешними системами

Интеграция сервисов Smart Services с другими системами требует разработки запросов на распознавание документов. Запросы создаются при помощи набора инструментов Postman.

Схема обработки документов



1. Если используется облачная версия сервисов, внешняя система отправляет управляющему сервису Smart Service учетные данные пользователя. Smart Service передает эти данные сервису Identity Service, который выпускает токен для отправки запросов к сервисам. После этого Smart Service передает токен системе.

Срок жизни токена составляет 24 часа, по его истечении запрашивается новый.

2. Внешняя система отправляет сервису Smart Service асинхронный запрос на обработку документов:
 - если используется облачная версия сервисов, утилита передает сервису документы и токен пользователя. Smart Service выполняет валидацию полученного токена. Если он недействительный или истекший, возвращается сообщение с ошибкой;
 - если сервисы установлены локально или в частном облаке, утилита передает сервису только документы. При получении запроса Smart Service использует токен анонимного пользователя.
3. Smart Service передает полученные документы сервису Text Extractor Service, который выполняет предварительную обработку страниц:
 - убирает фон страницы, например изображения или специальные знаки, для более качественного извлечения текстового слоя;
 - поворачивает страницу так, чтобы угол поворота был кратен 90 градусам.

ПРИМЕЧАНИЕ. Если в сервисы Smart Services передан многостраничный документ, обработка выполняется для каждой его страницы отдельно. Число одновременно обрабатываемых страниц настраивается в [конфигурационном файле](#) сервиса Text Extractor Service.
4. Text Extractor Service передает страницу сервису Image Classifier Service, который определяет ее поворот: 0, 90, 180 или 270 градусов. После этого Text Extractor Service поворачивает страницу до корректного положения.
5. Text Extractor Service:
 - передает страницу сервису Object Detection Service, который ищет на ней подписи и возвращает их координаты;
 - ищет на странице печати и таблицы.
6. Text Extractor Service извлекает из страницы текстовый слой, затем передает ее сервису Text Classifier Service, который определяет, является ли страница первой в документе.
7. После обработки всех переданных страниц Text Extractor Service делит их на документы с помощью признака «первая/не первая страница», затем формирует для каждого документа PDF-файл с извлеченным текстовым слоем и передает пакет сервису Smart Service.
8. Smart Service передает пакет документов сервису Text Classifier Service, который классифицирует каждый документ по виду.
9. Smart Service передает документы на параллельное извлечение фактов сервисами:
 - Fact Extractor Rules Service, который извлекает факты на основе правил;
 - Fact Extractor Learn Service, который извлекает факты на основе обучаемых моделей.
10. Результаты извлечения фактов объединяются в Smart Service и передаются сервису Fact Extractor Base Service. Он форматирует факты и рассчитывает вероятности их корректного извлечения, затем передает эти данные обратно в Smart Service.
11. Text Extractor Service определяет позиции извлеченных фактов на страницах.
12. Если указаны дополнительные классификаторы, например по видам договоров, Smart Service передает документы сервису Text Classifier Service для классификации.
13. Smart Service передает обработанный пакет документов во внешнюю систему.

Системные требования

Перед установкой и настройкой сервисов Smart Services ознакомьтесь с типовыми требованиями к аппаратному и программному обеспечению, а также с [ограничениями](#).

Минимальные требования для установки сервисов:

Компонент	Требование
Процессор (Intel/AMD-совместимый x64)	8 и более логических ядер с частотой 2,4 ГГц и поддержкой AVX
Память (ОЗУ)	20 ГБ
Тип жесткого диска	SSD
Свободное место на жестком диске	Установка: 50 ГБ Временные файлы в процессе работы: от 20 ГБ (в зависимости от объема и размера обрабатываемых документов)
Сетевой адаптер	1 Гбит/с и выше

Требования к программному обеспечению серверов, на которые устанавливаются сервисы Smart Services, см. в разделе [«Требования к ПО серверов с сервисами»](#).

В продуктивной среде рекомендуется устанавливать СУБД PostgreSQL и брокер сообщений на разные компьютеры, выделенные только для их работы.

Соблюдение минимальных системных требований позволяет обрабатывать небольшой поток документов, поэтому подходит только для настройки сервисов и демонстрации их работы. Для продуктивных систем требуемая конфигурация оборудования определяется с учетом объема документов, который обрабатывается сервисами Smart Services:

- если объем документопотока в месяц составляет [до 90000](#) документов, все сервисы Smart Services можно установить на один сервер;
- если объем документопотока в месяц [превышает 90000 документов](#), рекомендуется установить Text Extractor Service на отдельный сервер.

ПРИМЕЧАНИЕ. Объем документопотока рассчитан исходя из средних значений: объем одного документа – 2 страницы; время обработки – 8 часов в день, 22 дня в месяц.

Обработка до 90000 документов в месяц

Определите необходимую конфигурацию оборудования по таблице:

Количество документов в месяц	Количество логических ядер на сервере	Объем оперативной памяти на сервере, ГБ
До 25000	8	20
25001–65000	8	26
65001–90000	16	32

ПРИМЕЧАНИЕ. Если количество документов в месяц близко к пограничному значению, рекомендуется использовать конфигурацию для большего объема документопотока. Таким образом обеспечивается надлежащая скорость обработки при пиковых нагрузках, когда за короткий период времени поступает большое количество документов.

Обработка более 90000 документов в месяц

Определите необходимую конфигурацию оборудования по таблице:

Количество документов в месяц	Основной сервер		Сервер Text Extractor Service	
	Количество логических ядер	Объем оперативной памяти, ГБ	Количество логических ядер	Объем оперативной памяти, ГБ
90001–110000	8	16	8	16
110001–130000	8	20	16	32
Более 130000	16	28	32	64

Для обеспечения отказоустойчивости также рекомендуется установить на отдельные виртуальные машины или физические сервера:

- сервис [Storage Service](#). На сервере используйте высокоскоростные жесткие диски HDD или SSD, объединенные в конфигурацию RAID;
- сервис Object Detector Service;
- сервис Image Classifier Service;
- сервис [Text Classifier Service](#);
- сервисы [Fact Extractor Service](#), Fact Extractor Rules Service и Fact Extractor Learn Service.

Подключите виртуальные машины или физические сервера к единому брокеру сообщений RabbitMQ.

Объем системы хранения данных

Объем системы хранения данных рассчитывается по формуле:

$$\boxed{\text{Объем системы хранения данных}} = \boxed{\text{количество обрабатываемых документов в день}} * \boxed{\text{средний размер документа (МБ)}} * \boxed{\text{среднее количество страниц в документе}} * \boxed{\text{период хранения временных документов}}$$

Например, для 5000 обрабатываемых документов в день со средним размером документа 4 МБ и 2 страницами документа при периоде хранения временных документов по умолчанию (3 дня) объем системы хранения составит: 5000 шт * 4 МБ * 2 стр * 3 дн = 120 ГБ.

Объем памяти для обучения моделей классификации и извлечения фактов

Для **обучения моделей** классификации и извлечения фактов рекомендуется установить копии сервисов Text Classifier Service и Fact Extractor Learn Service на отдельные виртуальные машины или физические сервера, чтобы не замедлять работу продуктивной системы. После установки настройте параметры **can_train_models** в соответствующих конфигурационных файлах сервисов.

Для сервиса Text Classifier Service системные требования не меняются в зависимости от количества размеченных документов.

Для сервиса Fact Extractor Learn Service определите необходимую конфигурацию оборудования по таблице:

Количество размеченных документов	Количество логических ядер на сервере	Объем оперативной памяти, ГБ
До 500	4	16
501–1000	4	24
1001–1500	4	32
1501–3000	8	48
3001–6000	8	64

Если планируется **дообучение** моделей базовой поставки, то для определения конфигурации оперативной памяти учитывается количество размеченных документов и вид исходной модели. Объем рассчитывается по формуле:

$$\boxed{\text{Объем памяти для дообучения}} = \boxed{\text{объем памяти для исходного количества документов}} + \boxed{\text{объем памяти для обучения на определенном виде документа}}$$

где:

- объем памяти для исходного количества документов определяется по [таблице выше](#);
- объем памяти для обучения на определенном виде документа, где:
 - Акт выполненных работ – 32 ГБ;
 - Постановление судебных приставов – 16 ГБ;
 - Счет на оплату (англоязычный) – 48 ГБ;
 - Входящее письмо, Счет на оплату (русскоязычный), Счет-фактура, Товарная накладная, УПД – 64 ГБ.

Так, например, для дообучения модели на 3000 актов выполненных работ потребуется 80 ГБ оперативной памяти на сервере (48 ГБ для трех тысяч документов + 32 ГБ для акта выполненных работ).

Для удобства определения необходимой конфигурации памяти для дообучения можно воспользоваться таблицей:

Вид документа	Количество размеченных документов				
	до 500	501–1000	1001–1500	1501–3000	3001–6000
Акт выполненных работ	48	56	64	80	96
Входящее письмо	80	88	96	112	128
Постановление судебных приставов	32	40	48	64	80
Счет на оплату (русскоязычный)	80	88	96	112	128
Счет на оплату (англоязычный)	64	72	80	96	112
Счет-фактура	80	88	96	112	128
Товарная накладная	80	88	96	112	128
Универсальный передаточный документ	80	88	96	112	128

Ограничения

Для работы сервисов Text Extractor Service и Fact Extractor Learn Service требуется дополнительное ограничение объема используемой оперативной памяти соответствующих контейнеров. По достижении установленных лимитов контейнеры автоматически перезапускаются.

По умолчанию установлены следующие лимиты объема оперативной памяти:

- для Text Extractor Service – 16 ГБ;
- для Fact Extractor Learn Service – 16 ГБ.

Итоговое значение определяется с учетом объема обрабатываемых документов по соответствующим таблицам выше.

После расчета величин ограничьте объем оперативной памяти, который будет использоваться контейнерами.

Требования к ПО серверов с сервисами

В разделе зафиксированы требования к программному обеспечению серверов, на которых планируется установка сервисов Smart Services.

Компонент	Требование
ОС	Альт Сервер 9.1
	Astra Linux Common Edition 2.12.40 (Опел)
	Ubuntu 20.04 LTS
Компоненты для установки и настройки	Docker Engine 20 и выше
	Docker Compose 1.29.2
	СУБД PostgreSQL 9.6.3 и выше
	RabbitMQ 3.8.1
	Python 3.7 и выше
Прочее	Portainer 2.1.0
	Portainer Agent 2.1.0
	InfluxDB 1.8
	Grafana 7.5.1
	Telegraf 1.22.1

Развертывание сервисов

Перед установкой ознакомьтесь с [типовыми требованиями](#) к аппаратному и программному обеспечению.

1. [Выполните подготовительные действия](#) для развертывания сервисов.
2. [Разверните сервисы](#) с помощью скриптов.
3. [Выполните завершающие действия](#).
4. [Проверьте состояние сервисов](#) после запуска с помощью инструмента checker.html.
5. [Установите инструменты Elasticsearch](#) для работы с лог-файлами.

Подготовка к развертыванию

1. Скачайте необходимые для работы сервисов программные компоненты, указанные в [типовых требованиях](#).
2. Установите на сервер, на котором планируется развертывание сервисов Smart Services, компоненты в следующем порядке:
 - [Docker Engine](#);
 - [Docker Compose](#);
 - Python, если он не был установлен ранее.

ПРИМЕЧАНИЕ. Если сервисы будут развертываться [на разных серверах](#), то на каждом из них должны быть установлены перечисленные компоненты.

Установка Docker Engine

Порядок установки компонентов Docker Engine зависит от используемого дистрибутива Linux.

Подробнее о порядке установки Docker на Ubuntu 20.04 LTS см. в документации Docker статью [Install Docker Engine](#). Порядок установки на [Альт Сервер 9.1](#) и [Astra Linux](#) Common Edition 2.12.40 (Орел) см. в соответствующих разделах справки.

Альт Сервер 9.1

ПРИМЕЧАНИЕ. Из-за специфических настроек безопасности в системе Альт Сервер 9.1 у пользователей по умолчанию отсутствуют права на выполнение команды sudo. Чтобы получить права, выполните соответствующие [настройки](#).

1. Получите список обновлений. Для этого выполните команды:


```
sudo apt-get update
sudo apt-get install ca-certificates curl
```
2. Подключите репозиторий **Sisyphus**:


```
sudo apt-repo add sisyphus
```
3. Обновите информацию из репозитория:


```
sudo apt-get update
```

4. Убедитесь, что установка Docker будет производиться из репозитория **Sisyphus**, а не из репозитория по умолчанию. Для этого выполните команду:

```
sudo apt-cache policy docker-engine
```

Пример ответа:

```
Установлен: (нет)
```

```
кандидат: 20.10.6-alt1:sisyphus+271055.100.1.1@1619787705
```

ПРИМЕЧАНИЕ. Если вместо ответа возвращается ошибка или версия предлагаемого к установке пакета ниже 20, обратитесь в службу поддержки.

5. Установите Docker:

```
sudo apt-get install docker-engine
```

ПРИМЕЧАНИЕ. Вместе с Docker обязательно должны установиться пакеты **containerd.io** и **docker-ce-cli**.

6. Проверьте установленную версию Docker. Она должна быть не ниже 20. Для этого выполните команду:

```
sudo docker -v
```

7. Отключите репозиторий **Sisyphus**:

```
sudo apt-repo rm sisyphus
```

8. Проверьте, что сервис запустился:

```
sudo systemctl status docker
```

Команда должна вернуть статус «active», выделенный зеленым цветом.

9. Включите сервис вручную, если он не запустился автоматически. Для этого выдайте разрешение на автоматический запуск docker вместе с системой, последовательно выполнив команды:

```
sudo systemctl enable --now docker
```

```
sudo systemctl start docker
```

ПРИМЕЧАНИЕ. Рекомендуется всегда запускать сервис docker с помощью команды sudo. При необходимости можно настроить запуск сервиса docker без использования sudo. Для этого нужно добавить пользователя текущего сеанса в группу docker. При этом следует учитывать, что добавление пользователей в группу docker повышает риск несанкционированного доступа к системным файлам, поэтому выполнять это действие не рекомендуется.

Настройка прав для выполнения команды sudo

1. Переключитесь с текущего пользователя на пользователя root. Для этого выполните команду:

```
su -
```

2. Введите пароль пользователя **root**, заданный при установке операционной системы.
3. Откройте файл /etc/sudoers. Для этого выполните команду:

```
visudo
```

ПРИМЕЧАНИЕ. В результате указанный файл откроется в редакторе vi. Подробнее об использовании редактора см. в статье [Работа с редактором vi](#).

4. Раскомментируйте строку WHEEL_USERS ALL=(ALL) ALL, удалив символ #, и сохраните файл.

5. Проверьте, что пользователь добавлен в группу **wheel**. Пример:

```
cat /etc/group | grep wheel
```

При необходимости включите пользователя в группу:

```
usermod -aG wheel <имя пользователя>
```

6. Перезайдите в ОС под текущим пользователем.

Astra Linux Common Edition 2.12.40

ВАЖНО. В инструкции приводятся готовые примеры кода и командных строк. При копировании текста убедитесь в отсутствии лишних пробелов, переносов строк и спецсимволов.

1. Получите список обновлений. Для этого выполните команды:

```
sudo apt update
```

```
sudo apt install apt-transport-https ca-certificates curl software-properties-common gnupg2
```

2. Добавьте ключ официального репозитория **Docker**:

```
sudo curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
```

3. Добавьте репозиторий **Docker** в систему в зависимости от типа прав пользователя:

- при работе под пользователем с правами **root** выполните команду:

```
echo "deb [arch=amd64] https://download.docker.com/linux/debian buster stable" >> /etc/apt/sources.list
```

- при работе под другим пользователем выполните команду:

```
sudo bash -c "echo 'deb [arch=amd64] https://download.docker.com/linux/debian buster stable' >> /etc/apt/sources.list"
```

4. Обновите информацию из репозитория:

```
sudo apt update
```

5. Убедитесь, что установка Docker будет производиться из репозитория **Docker**, а не из репозитория по умолчанию. Для этого выполните команду:

```
sudo apt-cache policy docker-ce
```

Пример ответа:

```
docker-ce:
```

```
  установлен: (нет)
```

```
  кандидат: 5:20.10.6~3-0~debian-buster
```

ПРИМЕЧАНИЕ. Если вместо ответа возвращается ошибка или версия предлагаемого к установке пакета ниже 20, обратитесь в службу поддержки.

6. Установите Docker:

```
sudo apt install docker-ce
```

ПРИМЕЧАНИЕ. Вместе с Docker обязательно должны установиться пакеты **containerd.io** и **docker-ce-cli**.

7. Проверьте, что сервис запустился:

```
sudo systemctl status docker
```

Команда должна вернуть статус «active», выделенный зеленым цветом.

8. Включите сервис вручную, если он не запустился автоматически. Для этого выдайте разрешение на автоматический запуск docker вместе с системой, последовательно выполнив команды:

```
sudo systemctl enable --now docker
sudo systemctl start docker
```

ПРИМЕЧАНИЕ. Рекомендуется всегда запускать сервис docker с помощью команды sudo. При необходимости можно настроить запуск сервиса docker без использования sudo. Для этого нужно добавить пользователя текущего сеанса в группу docker. При этом следует учитывать, что добавление пользователей в группу docker повышает риск несанкционированного доступа к системным файлам, поэтому выполнять это действие не рекомендуется.

Установка Docker Compose

Для работы сервисов Smart Services установите Docker Compose версии 1.29.2.

ВАЖНО. В инструкции приводятся готовые примеры кода и командных строк. При копировании текста убедитесь в отсутствии лишних пробелов, переносов строк и спецсимволов.

1. Получите права на запуск скриптов. Для этого выполните команду:

```
source change-mod.sh
```

2. С привилегиями суперпользователя выполните команду:

```
./docker-compose-install.sh
```

ВАЖНО. Если на сервере уже использовалась более ранняя версия Docker Compose, то она будет заменена на версию 1.29.2. Существующий исполняемый файл /usr/local/bin/docker-compose будет переименован в /usr/local/bin/docker-compose_backup.

3. Проверьте, что установлена нужная версия Docker Compose, выполнив команду:

```
docker-compose -v
```

Если установка выполнена успешно, команда выведет версию 1.29.2.

Пример ответа:

```
docker-compose version 1.29.2, build 8a1c60f6
```

Развертывание

Для развертывания сервисов Smart Services используются docker-контейнеры. Каждому сервису соответствует отдельный контейнер.

ВАЖНО. В инструкции приводятся готовые примеры кода и командных строк. При копировании текста убедитесь в отсутствии лишних пробелов, переносов строк и спецсимволов.

Порядок развертывания

1. Скопируйте на сервер архив с Docker-образами сервисов и скриптами их развертывания и распакуйте с помощью команды:

```
tar -xf <Имя архива>
```

ПРИМЕЧАНИЕ. Если планируется использование нескольких серверов, скопируйте архив на каждый из них.

2. После распаковки архива перейдите в корневой каталог сервисов. Для этого выполните команду:

```
cd <путь до корневого каталога сервисов>
```

ВАЖНО. Далее все команды выполняются из корневого каталога сервисов, который обозначается как ./

3. Получите права на запуск скриптов. Для этого выполните команду:

```
source change-mod.sh
```

Если сервисы развертываются в операционной системе Альт Сервер, выдайте другим пользователям права на запись в папки, в которых хранятся трейсы, с помощью команды:

```
sudo chmod -R o+w ./etc/elastic-apm
```

4. Загрузите Docker-образы сервисов. Для этого выполните команду:

```
./load-images.sh
```

5. Убедитесь, что все образы загружены. Для этого выполните команду:

```
docker images
```

Количество образов должно быть равно количеству архивов в каталоге ./images.o

6. Инициализируйте инструмент управления взаимодействием контейнеров Docker Swarm, если он не был инициализирован ранее. Для этого выполните команду:

```
./swarm-init.sh
```

В результате текущий сервер станет управляющим по отношению к другим серверам.

Если на сервере используется несколько IP-адресов, то явно задайте адрес, который будет использоваться для обращения к серверу. Пример:

```
./swarm-init.sh 172.16.64.104
```

7. Задайте метку сервера, на котором будут запускаться docker контейнеры с сервисами. Для этого выполните команду:

```
./swarm-add-label-to-node.sh <метка> <имя узла>
```

где:

- **<метка>** – значение параметра **manager_node_label** из конфигурационного файла `.env`. По умолчанию все сервисы запускаются на управляющем сервере, где значение метки – **main**. При необходимости значение метки можно изменить. Если сервисы запускаются на одном сервере, то метка для них должна быть одинаковой;
- **<имя узла>** – имя текущего сервера.

Чтобы узнать имя текущего сервера, выполните команду:

```
hostname
```

Чтобы узнать значение текущей метки, выполните команду:

```
docker node inspect <имя узла>
```

В соответствующем параметре ответа будет указано текущее значение. Пример:

```
"Spec": {
  "Labels": {
    "service": "main"
  },
  "Role": "manager",
  "Availability": "active"
}
```

8. Добавьте рассчитанные лимиты объема оперативной памяти для Text Extractor Service и Fact Extractor Learn Service в конфигурационный файл `deploy.yml`. Для этого измените значение параметра **memory** в соответствующих секциях файла, указав рассчитанный объем в мегабайтах. Сервису Text Extractor Service соответствует секция **dtes**, сервису Fact Extractor Learn Service – **dfesl**:

```
deploy:
  mode: replicated
  replicas: ${dtes_replicas}
  placement:
    constraints:
      - "node.labels.service == ${dtes_node_label}"
    max_replicas_per_node: 1
  restart_policy:
    condition: any
    delay: 15s
    window: 180s
  resources:
    limits:
      memory: 16384m
```

ПРИМЕЧАНИЕ. Если для распознавания и обучения используется два экземпляра сервиса Fact Extractor Learn Service, то в первом случае лимит оперативной памяти имеет постоянное значение – 2 ГБ. Лимит оперативной памяти для обучения определяется с учетом количества обрабатываемых документов.

Если для распознавания и обучения используется один экземпляр Fact Extractor Learn Service, лимит оперативной памяти для распознавания также будет равен 2 ГБ, а итоговый лимит памяти равен объему памяти для обучения согласно [таблице](#), которая используется для расчета объема памяти для обучения моделей классификации и извлечения фактов. Например, при 2 ГБ для распознавания и 16 ГБ для обучения на 400 документах итоговый лимит 16 ГБ. Таким образом, итоговый лимит равен наибольшему значению лимита для экземпляра сервиса.

9. Убедитесь, что в конфигурационном файле `.env`:
 - в параметре **timezone** указан текущий часовой пояс;
 - в параметре **dss_allow_local_path** указано значение **true**;
 - в параметре **dtcs_allow_local_path** указано значение **True**.
10. Запросите ключ активации сервисов в службе поддержки и скопируйте полученный файл `ActivationKey.aak` в каталог `./etc/smart-service/license`. После установки и запуска сервисов активация выполнится автоматически.
11. Запустите контейнеры сервисов Smart Services. Для этого выполните команду:

```
./start-services.sh
```

ПРИМЕЧАНИЕ. На запуск уходит несколько минут.

12. Проверьте, что контейнеры запустились, одним из способов:
 - с помощью команды:


```
docker service ls
```

В ответе в колонке «Replicas» проверьте состояние запущенных сервисов: **0/1** – сервис еще не запущен, **1/1** – сервис запущен;
 - с помощью Portainer. Для этого:
 - a) Войдите в Portainer на сетевой адрес сервера на порт, указанный в конфигурационном файле `.env` в секции **Portainer**. По умолчанию **8999**.
 - b) Задайте логин и пароль для входа.
 - c) На домашней странице выберите текущий Endpoint – **primary**.
 - d) Перейдите в раздел «Stacks» и нажмите на стек **smart-services**.
 - e) В выпадающем пункте **Items per page** внизу страницы установите значение **100**.
 - f) В колонке «Scheduling Mode» отслеживайте состояние запущенных контейнеров:
 - **0/1** – сервис еще не запущен;
 - **1/1** – сервис запущен.

ПРИМЕЧАНИЕ. Чтобы увидеть лог-файлы, разверните строку нужного сервиса по кнопке  и в столбце **Actions** нажмите на кнопку **Logs**.

13. Если необходимо изменить настройки, например, изменить порт подключения к Smart Service, внесите изменения в конфигурационный файл `.env`.
14. После изменения параметров конфигурации перезапустите сервисы. Для этого последовательно:
 - остановите сервисы:


```
./stop-services.sh
```
 - проверьте, что все сервисы остановлены, с помощью команды:


```
docker ps -a
```
 - повторно запустите сервисы:


```
./start-services.sh
```

В результате сервисы запустятся с учетом изменений, внесенных в конфигурационный файл.

15. При необходимости разверните отдельные сервисы на [других серверах](#).

16. Выполните [завершающие работы](#).

В результате развертывания:

- из архива загружаются docker-образы;
- запускаются контейнеры с сервисами Smart Services.

Подключение дополнительных серверов

Если сервисы развертываются на разных серверах, например Text Extractor Service устанавливается на отдельный, то их нужно связать между собой. Связь представляет собой Swarm-кластер, где есть [управляющий](#) сервер и подключаемые к нему дополнительные сервера.

Если подключаются дополнительные сервера, то на управляющем должны развертываться следующие сервисы:

- сервисы Smart Services: Smart Service, Identity Service, Publishing Service;
- Elastic APM;
- Influxdb;
- Grafana;
- Telegraf.

Другие сервисы и компоненты можно разворачивать на дополнительных серверах.

Подключение еще одного сервера выполняется в следующем порядке:

1. Остановите на управляющем сервере сервисы Smart Services. Для этого последовательно выполните команду:

```
./stop-services.sh
```

2. Проверьте, что все сервисы остановлены, с помощью команды:

```
docker ps -a
```

3. Сгенерируйте команду с ключом для подключения дополнительного сервера. Для этого выполните команду:

```
./swarm-get-token.sh
```

4. Выполните полученную команду на подключаемом сервере, чтобы создать связь с управляющим сервером.

5. Проверьте, что сервер подключен успешно. Для этого на управляющем сервере выполните команду:

```
docker node ls
```

6. Измените значение метки дополнительного сервера. Для этого в секции **ServicePlacement** конфигурационного файла .env задайте значения параметра **node_label** для сервиса, развернутого отдельно. Например:

```
dtes_node_label = dtes
```

ПРИМЕЧАНИЕ. Значение параметра **node_label** должно отличаться от [параметра manager_node_label](#) для управляющего сервера.

7. Задайте метку для вновь подключенного сервера. Для этого выполните команду на управляющем сервере:

```
./swarm-add-label-to-node.sh <метка> <имя узла>
```

где:

- **<метка>** – значение параметра **node_label** из конфигурационного файла .env. По умолчанию **main**;
- **<имя узла>** – имя подключенного сервера.

Пример:

```
./swarm-add-label-to-node.sh dtes SMART-SERVER-2
```

8. Повторите на подключаемом сервере [шаги 1-5](#) инструкции по развертыванию сервисов.

ВАЖНО. Путь до корневого каталога сервисов должен быть одинаковым на всех серверах.

9. Запустите [контейнеры](#).

Завершение развертывания

1. Импортируйте *модели извлечения фактов*, если факты должны извлекаться на основе обучаемых моделей. Для этого запустите скрипт `import_models.py` из каталога `./etc/factextractor-learn-service/trained_models` с параметрами:

- **-url.** Путь до сервиса Smart Service. Параметр обязателен для заполнения;
- **-login.** Логин пользователя для аутентификации. Если параметр не указан, используется анонимная аутентификация (при условии, что она включена);
- **-password.** Пароль пользователя для аутентификации. Указывается, только когда указан логин.

Пример команды:

```
python import_models.py -url http://smart:61100
```

Пример команды с дополнительными параметрами:

```
python import_models.py -url http://smart:61100 -login user -password 123
```

ПРИМЕЧАНИЕ. В зависимости от настроек ОС может потребоваться указание конкретной версии Python, например `python3.7`, а вместо имени сервера – его IP-адрес.

В результате работы скрипта в Smart Services инициализируются базовые модели извлечения фактов.

2. Импортируйте базовые *модели классификации* документов.

Для **первичной загрузки моделей в новый классификатор** запустите скрипт `import_models.py` из каталога `./etc/textclassifier-service/trained_models` с параметрами:

- **-url.** Путь до сервиса Smart Service. Параметр обязателен для заполнения;
- **-login.** Логин пользователя для аутентификации. Если параметр не указан, используется анонимная аутентификация (при условии, что она включена);
- **-password.** Пароль пользователя для аутентификации. Указывается, когда указан логин.

Пример команды:

```
python import_models.py -url http://smart:61100
```

Пример команды с дополнительными параметрами:

```
python import_models.py -url http://smart:61100 -login user -password 123
```

ПРИМЕЧАНИЕ. В зависимости от настроек ОС может потребоваться указание конкретной версии Python, например python3.7, а вместо имени сервера – его IP-адрес.

В результате создаются классификаторы по виду документа и по первым страницам, в которые импортируются соответствующие базовые модели классификации.

ПРИМЕЧАНИЕ. Если уже существуют классификаторы с именами **Базовый классификатор по виду документа** и **Базовый классификатор первая/не первая страница**, то в результате работы скрипта обновятся их модели.

Для **загрузки моделей классификации в существующий классификатор** запустите тот же скрипт с теми же параметрами, указав дополнительно:

- **-id_kind.** Идентификатор классификатора документов по виду. Параметр не обязателен для заполнения;
- **-id_first.** Идентификатор классификатора по первым страницам. Параметр не обязателен для заполнения.

Пример команды:

```
python import_models.py -url http://smart:61100 -id_kind 1 -id_first 2
```

Пример команды с дополнительными параметрами:

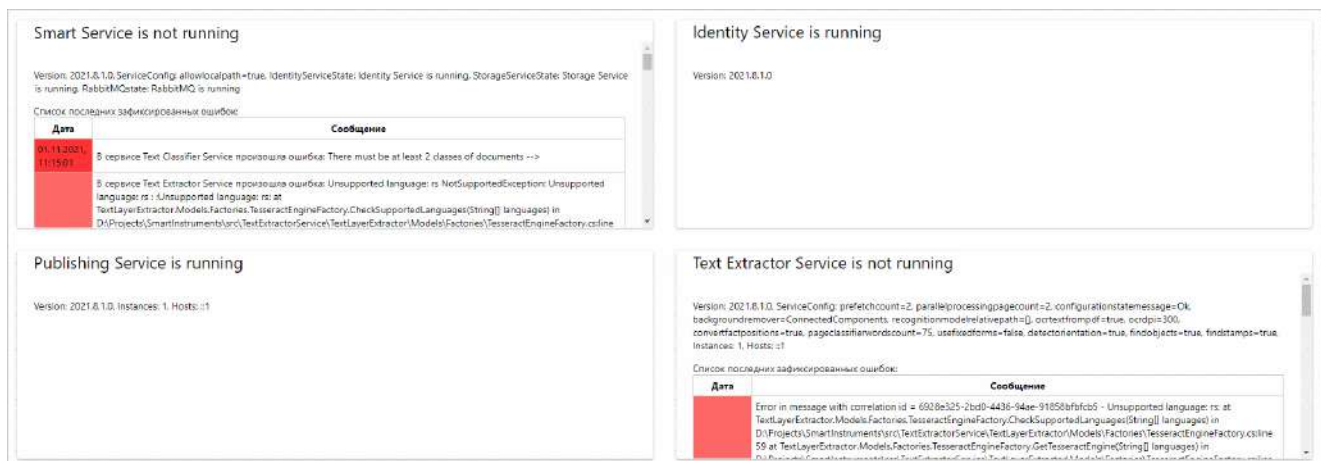
```
python import_models.py -url http://smart:61100 -id_kind 1 -id_first 2 -  
login user -password 123
```

В результате в классификатор по виду документа с идентификатором **1** и в классификатор по страницам с идентификатором **2** импортируются базовые модели классификации.

ПРИМЕЧАНИЕ. Если классификаторов с такими идентификаторами нет, возвратится соответствующая ошибка.

Запуск и настройка инструмента checker.html

Инструмент checker.html позволяет быстро проверить состояние сервисов после запуска, а также проанализировать последние ошибки в работе. Если они есть, то отображается таблица со списком последних 10 зафиксированных ошибок. В таблице фиксируется дата и время ошибки, а также ее полное описание:



Состояние **is running** означает, что сервис запущен и работает.

Запуск

- если все сервисы Smart Services установлены на один сервер, в дистрибутиве из корневого каталога откройте файл checker.html;
- если сервисы Smart Services установлены на разные серверы, то настройте инструмент для распределенной работы и запустите файл checker.html на сервере со Smart Service.

Особенности:

- если используется Linux без графической оболочки, то настройте инструмент для распределенной работы и запускайте checker.html на сервере с графической оболочкой.

Настройка инструмента для распределенной работы

Чтобы настроить инструмент для распределенной работы, на сервере со Smart Service в файле checker.html укажите имя или IP-адрес дополнительного сервера. Например, укажите имя или IP-адрес сервера с Text Extractor Service, если он развернут отдельно.

Особенности:

- в операционной системе на базе Linux рекомендуется указывать IP-адреса сервисов;
- если используется Linux без графической оболочки, скопируйте файл checker.html на сервер с графической оболочкой и внесите в него изменения с помощью любого текстового редактора. При этом измените IP-адреса всех серверов.

Пример настройки утилиты с помощью Notepad++:

```

<body>
  <div id="container">

  </div>

  <script>

    const service_urls = {
      "Smart Service": "http://smart:61100/info",
      "Identity Service": "http://smart:9030/info",
      "Publishing Service": "http://smart:9057/info",
      "Text Extractor Service": "http://intelligent:9027/api/info/state",
      "Text Classifier Service": "http://smart:9007/",
      "Image Classifier Service": "http://smart:9008/",
      "Fact Extractor Service": "http://smart:9017/",
      "Fact Extractor Rules Service": "http://smart:9018/",
      "Fact Extractor Learn Service": "http://smart:9019/",
      "Object Detection Service": "http://smart:9047/"
    };
  
```

Установка инструментов Elasticsearch

Для загрузки лог-файлов в Elasticsearch и их последующего анализа:

- [установите поисковую систему Elasticsearch и панель управления Kibana](#);
- [настройте шаблоны индекса в Kibana](#);
- [установите и настройте сервис Filebeat](#).

Подробнее о работе инструментов см. в документации [Elasticsearch](#), [Kibana](#), [Filebeat](#).

Установка поисковой системы Elasticsearch

В службе поддержки запросите дистрибутивы Elasticsearch и Kibana. Дистрибутивы предоставляются бесплатно, дополнительную лицензию приобретать не нужно.

Чтобы установить Elasticsearch:

1. Распакуйте ZIP-архив с дистрибутивом Elasticsearch в отдельную локальную папку.
2. В командной строке перейдите в папку <Локальная папка с Elasticsearch>\bin и выполните команду:

```
elasticsearch-service.bat install
```

В результате устанавливается служба Elasticsearch.

3. В оснастке **Службы** запустите службу **Elasticsearch (elasticsearch-service-x64)**.
Настройте автозапуск службы, чтобы она сама стартовала после перезагрузки операционной системы. Для этого откройте свойства службы, в раскрывающемся списке **Тип запуска** выберите значение **Автоматически** и нажмите на кнопку **ОК**.
4. В браузере перейдите по адресу <http://localhost:9200>, откроется стартовая страница Elasticsearch.

Пример:

```
{
  "name" : "elastic",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "tdrMX2SyQOm-bT724ViLoA",
  "version" : {
    "number" : "7.16.3",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "4e6e4eab2297e949ec994e688dad46290d018022",
    "build_date" : "2022-01-06T23:43:02.825887787Z",
    "build_snapshot" : false,
    "lucene_version" : "8.10.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

5. [Настройте](#) обратный прокси-сервер (Reverse Proxy) и ограничение по IP-адресам для безопасного подключения к Elasticsearch.
6. [Настройте](#) фильтрацию запросов в IIS для корректного индексирования текстов документов.

Чтобы установить панель управления Kibana:

1. Распакуйте ZIP-архив с дистрибутивом Kibana в отдельную локальную папку.
2. Запустите файл <Папка с Kibana>\bin\kibana.bat. В результате откроется окно консольного приложения Kibana. Чтобы продолжить работу с Kibana, не закрывайте окно. Работу Kibana можно возобновить в любой момент, запустив указанный bat-файл.
3. В браузере перейдите по адресу <http://localhost:5601>, откроется стартовая страница Kibana.

Обратный прокси-сервер

Установка ролей и компонентов

1. На сервере с установленной поисковой системой Elasticsearch запустите оснастку **Диспетчер сервера**.
2. В меню **Управление** выберите действие **Добавить роли и компоненты**. Запустится мастер добавления ролей и компонентов.
3. На странице «Перед началом работы» нажмите на кнопку **Далее**.
4. На странице «Выбор типа установки» выберите пункт **Установка ролей или компонентов** и нажмите на кнопку **Далее**.
5. На странице «Выбор целевого сервера» выберите пункт **Выберите сервер из пула серверов** и укажите сервер, на котором необходимо добавить роли и компоненты. Нажмите на кнопку **Далее**.
6. На странице «Выбор ролей сервера» последовательно раскройте узлы **Веб-сервер (IIS)**, **Веб-сервер**, **Безопасность** и установите флажок **IP-адрес и ограничения домена**. Нажмите на кнопку **Далее**.
7. На странице «Выбор компонентов» нажмите на кнопку **Далее**.

8. На странице «Подтверждение установки компонентов» нажмите на кнопку **Установить**.
9. На странице «Ход установки» убедитесь, что установка роли успешно завершена, и нажмите на кнопку **Заккрыть**.
10. Установите расширение [URL Rewrite](#).

Настройка обратного прокси-сервера

1. В приложении **Диспетчер служб IIS** добавьте веб-сайт. Для этого на панели «Подключения» перейдите к узлу сайты и в контекстном меню выберите пункт **Добавить веб-сайт**.
2. В окне «Добавить веб-сайт» заполните поля:
 - **Имя сайта** – например, ElasticsearchSite;
 - **Физический путь** – расположение сайта Elasticsearch, например, **C:\inetpub\wwwroot\ElasticsearchSite**;
 - **Порт** – **9201**;
 - **Имя узла** – *****.

Значения остальных полей оставьте без изменений. Нажмите на кнопку **ОК**.

3. На начальной странице созданного сайта выберите действие **Переопределение URL-адресов**.
4. На странице «Переопределение URL-адресов» на панели выберите действие **Добавить правила**.
5. В открывшемся окне «Добавить правила» выберите шаблон **Обратный прокси-сервер** и нажмите на кнопку **ОК**.
6. В открывшемся окне «Добавить правила обратного прокси-сервера» настройте правило для входящего трафика. В качестве имени сервера, на который будут перенаправляться HTTP-запросы, укажите **localhost:9200**. Затем установите флажок **Включить загрузку SSL** и нажмите на кнопку **ОК**.
7. На начальной странице сайта выберите действие **Ограничения IP-адресов и доменов**.
8. На странице «Ограничения IP-адресов и доменов» выберите действие **Добавить разрешающий элемент**.
9. В окне «Добавить разрешающее правило ограничения» в поле **Определенный IP-адрес** укажите IP-адрес, к которому должен быть разрешен доступ. Нажмите на кнопку **ОК**.
10. На панели выберите действие **Изменить параметры**. В открывшемся окне в раскрывающемся списке «Доступ для неуказанных клиентов» выберите пункт **Запретить** и нажмите на кнопку **ОК**.
11. На начальной странице сайта перейдите в группу «Управление» и выберите действие **Редактор конфигураций**.
12. На открывшейся странице редактора конфигураций в раскрывающемся списке **Раздел** выберите **system.web/httpRuntime**.
13. В табличной части для атрибута **relaxedUrlToFileSystemMapping** укажите значение **true**.

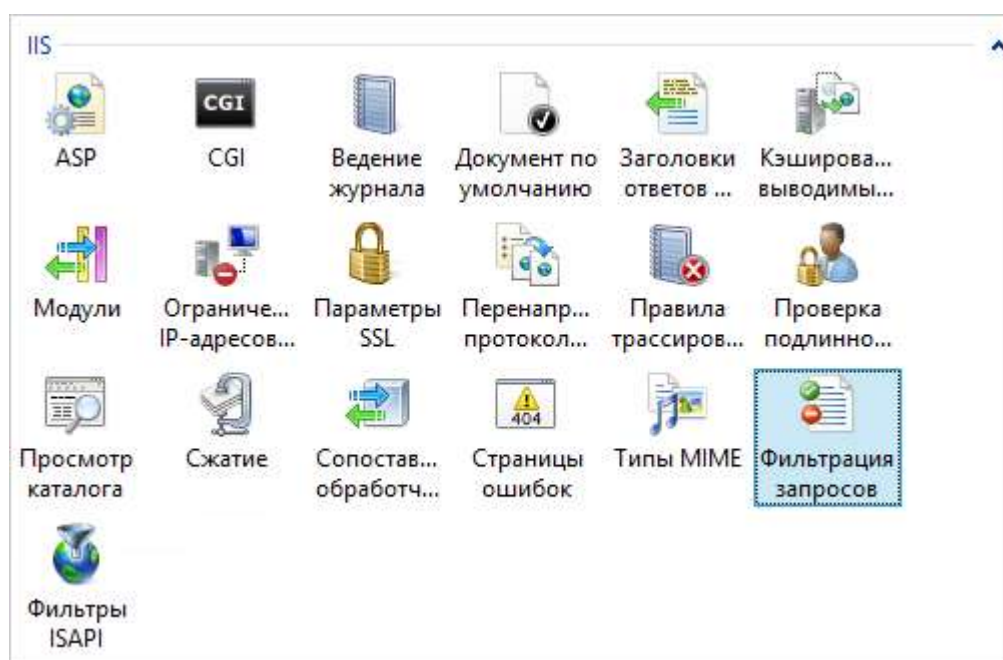
14. В атрибуте **requestPathInvalidCharacters** укажите произвольное значение, переместите курсор на другую строку, затем вернитесь и удалите введенные символы. В результате значение атрибута должно полностью очиститься. Нажмите на кнопку **Применить**.
15. Проверьте заданные настройки. Для этого в браузере на другом компьютере откройте страницу <http://<Имя компьютера с Elasticsearch>:9201>. В результате откроется тестовая страница Elasticsearch.

ПРИМЕЧАНИЕ. Если для доступа в Интернет используется прокси-сервер, то в его исключения добавьте адрес компьютера с установленными компонентами Elasticsearch.

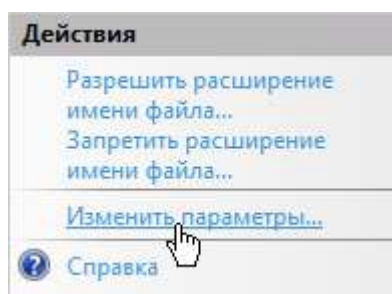
Фильтрация запросов в IIS

По умолчанию IIS обрабатывает запросы, содержимое которых не превышает 30 МБ. Для корректной работы необходимо с запасом увеличить максимально допустимый размер запроса.

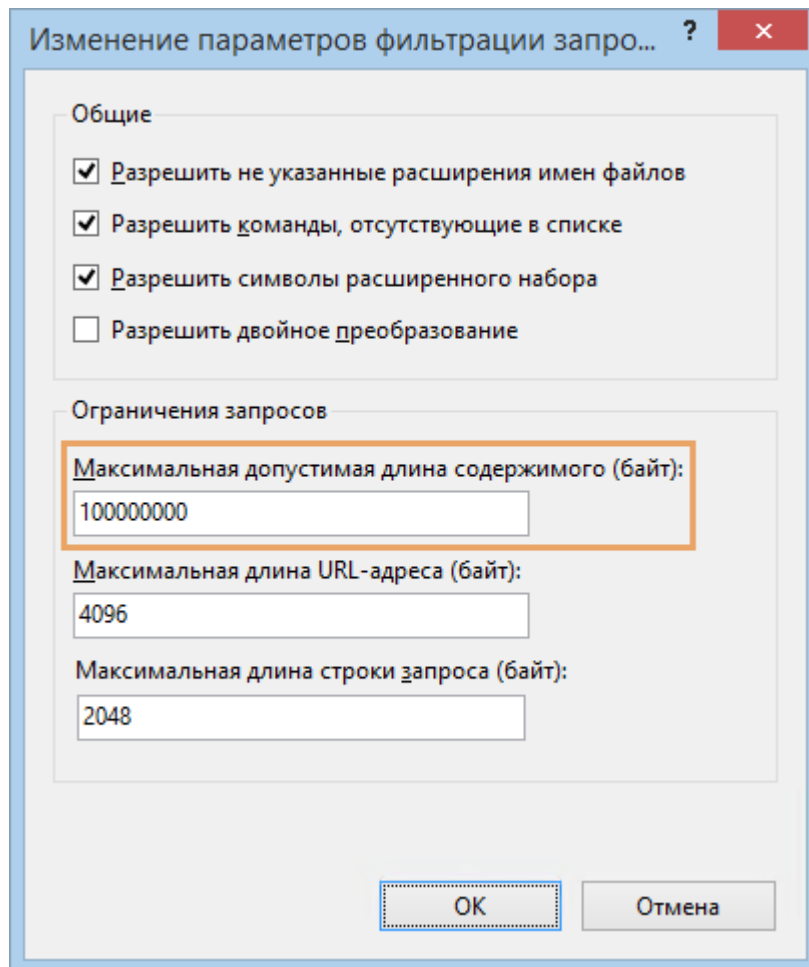
1. В приложении **Диспетчер служб IIS** перейдите к сайту Elasticsearch и на его начальной странице выберите действие **Фильтрация запросов**:



2. На странице «Фильтрация запросов» на панели действий выберите действие **Изменить параметры**:



- В открывшемся окне в поле **Максимальная допустимая длина содержимого (байт)** укажите значение **100 000 000** (100 МБ):



- Нажмите на кнопку **ОК**.

Установка сервиса Filebeat

ВАЖНО. В инструкции приводятся готовые примеры кода и командных строк. При копировании текста убедитесь в отсутствии лишних пробелов и спецсимволов.

Автоматическая загрузка лог-файлов в Elasticsearch выполняется с помощью сервиса Filebeat. Сервис входит в линейку продуктов Elasticsearch и распространяется бесплатно.

Чтобы настроить загрузку лог-файлов с помощью Filebeat:

- Скачайте и установите Filebeat 7.14.0. Для этого выполните команды:

```
curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.14.0-linux-x86_64.tar.gz
tar xzvf filebeat-7.14.0-linux-x86_64.tar.gz
```

- Настройте сервис с помощью конфигурационного файла **filebeat.yml**. Путь до конфигурационного файла по умолчанию: `/etc/filebeat/filebeat.yml`.

В секции **filebeat.inputs** укажите параметры получения и декодировки файлов:

- **paths** – пути до лог-файлов сервисов в формате <Папка с лог-файлами сервиса>/*.log>. Пути до папок с лог-файлами указываются в конфигурационных файлах каждого сервиса;
- **json.keys_under_root** – true;
- **json.overwrite_keys** – true;
- **json.add_error_key** – true;
- **json.message_key** – message.

В секции **output.elasticsearch** укажите параметры загрузки данных в Elasticsearch:

- **hosts** – адрес Elasticsearch.

В остальных секциях оставьте значения по умолчанию.

Пример настройки:

```
filebeat.inputs:
- type: log
  enabled: true
  paths:
    - C:\Smart_Services\SmartService\SmartService\Logs\*.log
    - C:\Smart_Services\TextExtractorService\logs\*.log
    - C:\Smart_Services\PublishingService\Logs\*.log
    - C:\Smart_Services\TextClassifierService\logs\*.log
    - C:\Smart_Services\ImageClassifierService\logs\*.log
    - C:\Smart_Services\FactExtractorServices\Base\logs\*.log
    - C:\Smart_Services\FactExtractorServices\Learn\logs\*.log
    - C:\Smart_Services\FactExtractorServices\Rules\logs\*.log
    - C:\Smart_Services\ObjectDetectionService\logs\*.log
  json.keys_under_root: true
  json.overwrite_keys: true
  json.add_error_key: true
  json.message_key: message

output.elasticsearch:
  hosts: ["elasticsearchhost:9200"]
```

3. Запустите Filebeat с помощью команды:

```
sudo ./filebeat
```

Обновление сервисов

Перед обновлением ознакомьтесь с [типовыми требованиями](#) и убедитесь, что ваше аппаратное и программное обеспечение актуально.

Алгоритм обновления сервисов Smart Services зависит от конфигурации установки:

- сервисы установлены на [одном сервере](#);
- сервисы установлены на [разных серверах](#).

Сервисы установлены на одном сервере

1. [Удалите предыдущую версию](#) сервисов Smart Services.
2. [Разверните новую версию](#) сервисов Smart Services.

ПРИМЕЧАНИЕ. При необходимости при развертывании на [шаге 1](#) замените конфигурационный файл .env на аналогичный конфигурационный файл, сохраненный от предыдущей установленной версии.

Сервисы установлены на разных серверах

1. На основном сервере [удалите предыдущую версию](#) сервисов Smart Services.
2. На дополнительных серверах удалите Docker-образы сервисов средствами Docker. Подробнее см. в документации Docker статью [docker rmi](#).
3. [Разверните новую версию](#) сервисов Smart Services.
ПРИМЕЧАНИЕ. При необходимости при развертывании на [шаге 1](#) замените конфигурационный файл .env на аналогичный конфигурационный файл, сохраненный от предыдущей установленной версии.
4. [Подключите дополнительные серверы](#).

Удаление сервисов

Чтобы удалить сервисы Smart Services с сервера:

1. Сохраните текущий конфигурационный файл `.env`.
2. Остановите сервисы. Для этого выполните команду:
`./stop-services.sh`
3. Проверьте, что все сервисы остановлены, с помощью команды:
`docker ps -a`
4. Удалите Docker-образы сервисов средствами Docker. Подробнее см. в документации Docker статью [docker rmi](#).
5. Удалите корневой каталог, в котором находятся скрипты управления сервисами.
6. При необходимости удалите хранилища данных (volumes) Portainer, Grafana, InfluxDB, Postgres, RabbitMQ и Minio средствами Docker. Например, если не планируете использовать текущие хранилища данных с [новой версией сервисов](#). Подробнее см. в документации Docker статью [docker volume rm](#).

ВАЖНО. Данные из хранилищ удаляются без возможности восстановления.

Настройки для обработки документов

Настройте сервисы Smart Services, чтобы задействовать их возможности в обработке документов организации.

Взаимодействие с сервисами и их настройка для обработки документов осуществляются с помощью HTTP-запросов к REST API управляющего сервиса [Smart Service](#). Для выполнения запросов удобно использовать специальный HTTP-клиент [Postman](#).

Порядок настройки сервисов для обработки документов:

1. [Установите и настройте Postman](#).
2. [Получите токен](#) для аутентификации пользователей в Smart Services, если сервисы установлены в облаке.
ПРИМЕЧАНИЕ. При локальной установке сервисов пропустите шаг.
3. [Настройте классификацию документов](#) с помощью Postman.
4. [Перенесите конфигурацию сервисов](#) в продуктивную систему.

Установка и настройка Postman

Для использования Postman:

1. На официальном сайте разработчика [скачайте](#) и установите десктоп-приложение или воспользуйтесь [веб-версией](#).
Для использования веб-версии требуется установка Postman Desktop Agent. Типовые требования к аппаратному и программному обеспечению и дополнительную информацию об установке см. в [документации Postman](#).
2. [Импортируйте коллекцию](#) с примерами запросов к API Smart Service в Postman.
3. [Настройте коллекцию](#), чтобы отправлять запросы к сервису Smart Service.

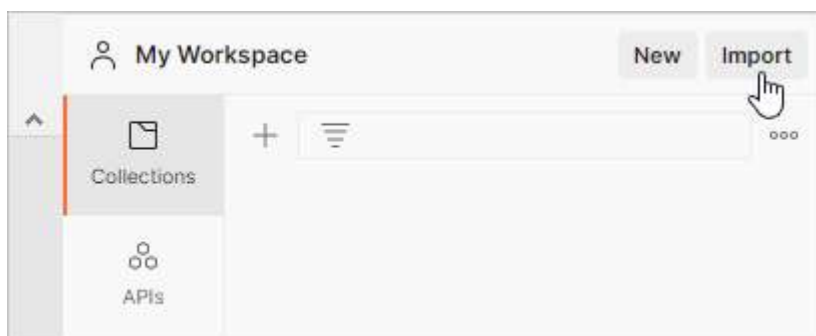
После установки и настройки приложения [ознакомьтесь с принципами](#) отправки запросов через Postman и воспользуйтесь его возможностями, чтобы [настроить](#) сервисы Smart Services.

Импорт коллекции запросов

Чтобы импортировать *коллекцию запросов* в Postman:

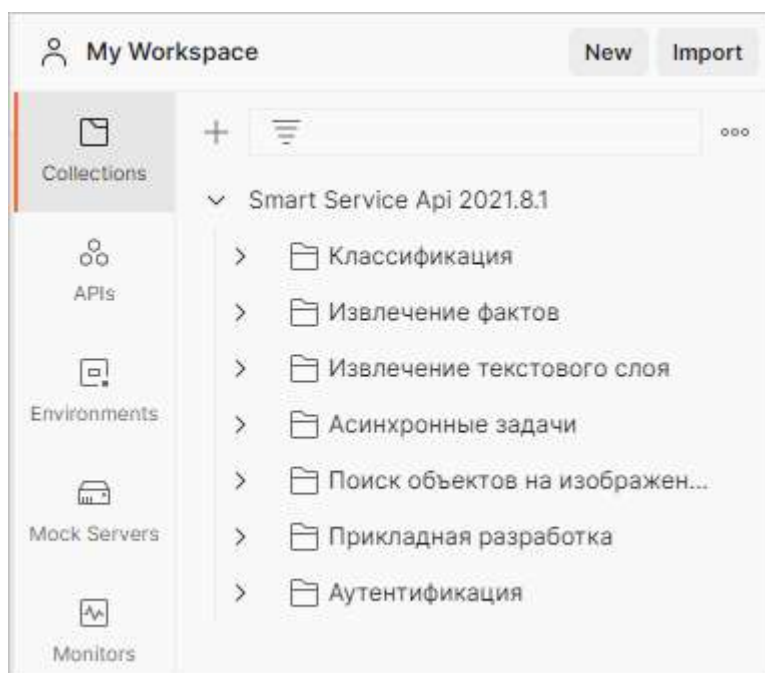
1. Откройте [коллекцию](#).
ПРИМЕЧАНИЕ. Коллекция должна соответствовать установленной версии сервисов Smart Services.
2. Скачайте коллекцию. Для этого щелкните правой кнопкой мыши в любом пространстве открывшегося окна, выберите пункт **Сохранить как...** и сохраните файл в формате JSON в локальной папке.

3. В рабочем пространстве Postman на вкладке **Collections** нажмите на кнопку **Import**:



4. В открывшееся окно загрузите или перетащите с помощью Drag-and-Drop файл сохраненной коллекции.

В результате в рабочем пространстве Postman появится коллекция с примерами запросов:

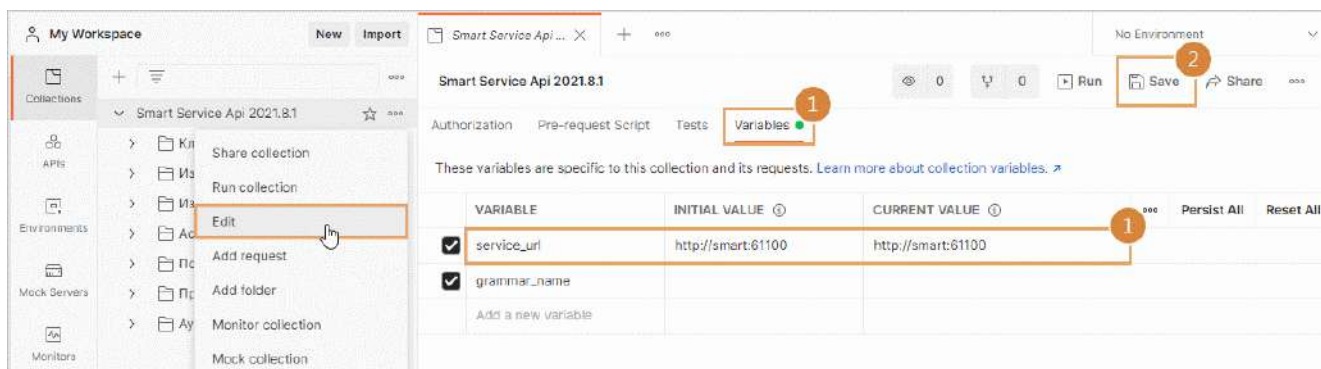


Настройка коллекции Postman

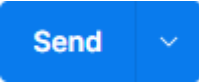
В импортированной коллекции Postman настройте переменную **service_url** для последующего выполнения запросов на развернутом сервисе Smart Service. Это также позволит при отправке запросов не прописывать каждый раз адрес сервиса вручную:

1. В контекстном меню коллекции выберите пункт **Edit** и на закладке **Variables** укажите адрес веб-сайта сервиса Smart Service.

Пример:



2. Сохраните изменения.
3. Проверьте, что коллекция запросов импортирована и настроена корректно. Для этого:
 - а) Выберите в коллекции GET-запрос «Получить список всех активных грамматик извлечения фактов».
 - б) Убедитесь, что адрес веб-сайта сервиса Smart Service в строке запроса соответствует адресу, указанному в шаге 1. Для этого наведите курсор на соответствующую переменную:

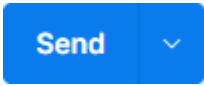
с) Нажмите на кнопку .

Ответ должен содержать список всех активных *грамматик извлечения фактов*.

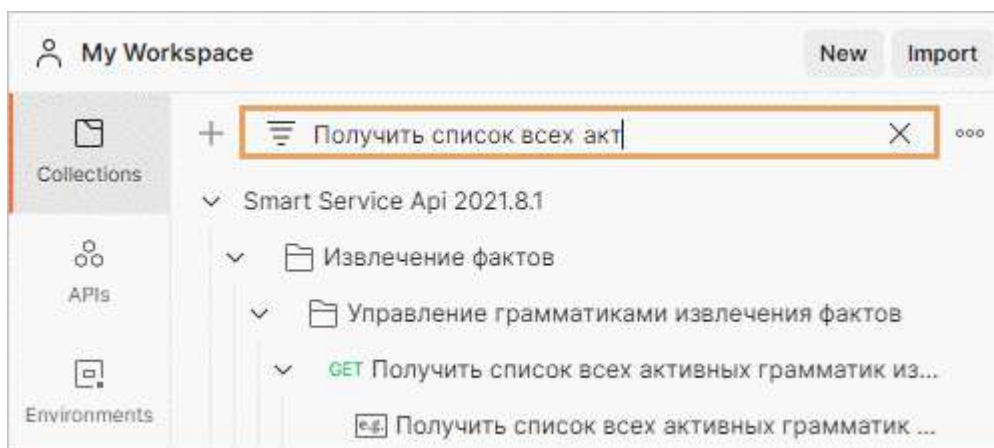
Принципы отправки запросов через Postman

В разделе описаны общие принципы отправки HTTP-запросов через Postman.

1. Перед отправкой запросов проверьте, что адрес веб-сайта сервиса Smart Service в строке запроса соответствует адресу, указанному при настройке коллекции.

2. Отправляйте запросы по кнопке .

- Чтобы быстро находить нужные запросы в коллекции, вводите их в поисковую строку:

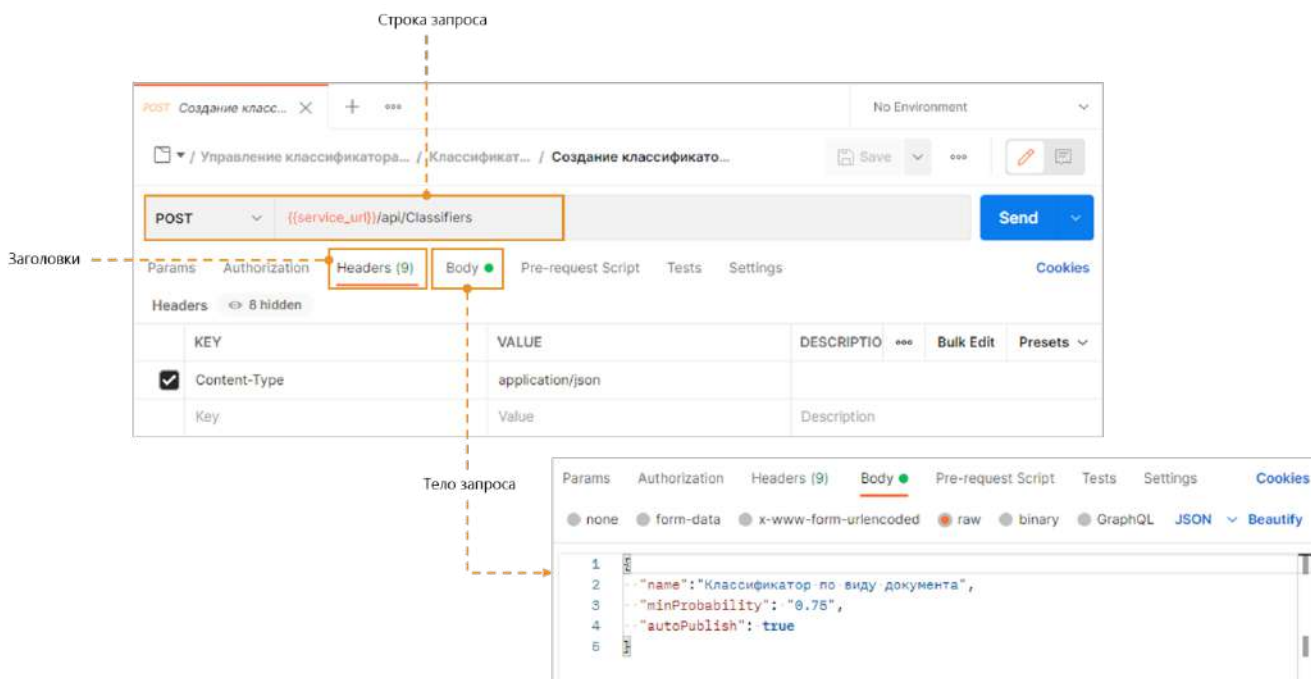


- Структура запроса выглядит так:

- строка запроса** (обязательный элемент). В строке выбирается метод обращения к API и передаются адрес сервиса Smart Service и краткое имя запроса;
- заголовки** (обязательный элемент). Могут указываться заголовки с указанием формата запроса к сервису Smart Service и ответа от него и др. Перечисляются на закладке **Headers**;
- тело запроса** (опциональный элемент). Используются в POST и PUT-запросах для передачи параметров. Перечисляются на закладке **Body**.

ПРИМЕЧАНИЕ. Для GET-запросов параметры передаются не в теле, а в строке запроса.

Пример:



- Если включена [аутентификация и авторизация](#) систем для сервисов Smart Services, установленных в облаке, то во всех запросах должен передаваться [токен](#) для аутентификации пользователей.

Получение токена для аутентификации пользователей

Чтобы безопасно работать с сервисами Smart Services в облаке, необходимо получить аутентификационный токен. Он содержит информацию о пользователе, который выполняет операции в рамках запросов к сервису Smart Service.

Когда пользователь отправляет запрос, Smart Service проверяет достоверность токена и, если он валиден, передаёт запрос дальше. В противном случае возвращает ошибку, а выполнение запроса останавливается.

Для работы в облаке получите токен и настройте коллекцию так, чтобы он передавался в каждом запросе. Для этого:

1. В коллекции Postman выберите POST-запрос «Получить токен».
2. На закладке **Body**:
 - в параметре **clientId** укажите логин пользователя;
 - в параметре **clientSecret** задайте пароль пользователя.

Пример запроса:

```
POST http://smart:61100/api/identity/token
{
  "clientId": {логин пользователя},
  "clientSecret": {пароль пользователя}
}
```

Send

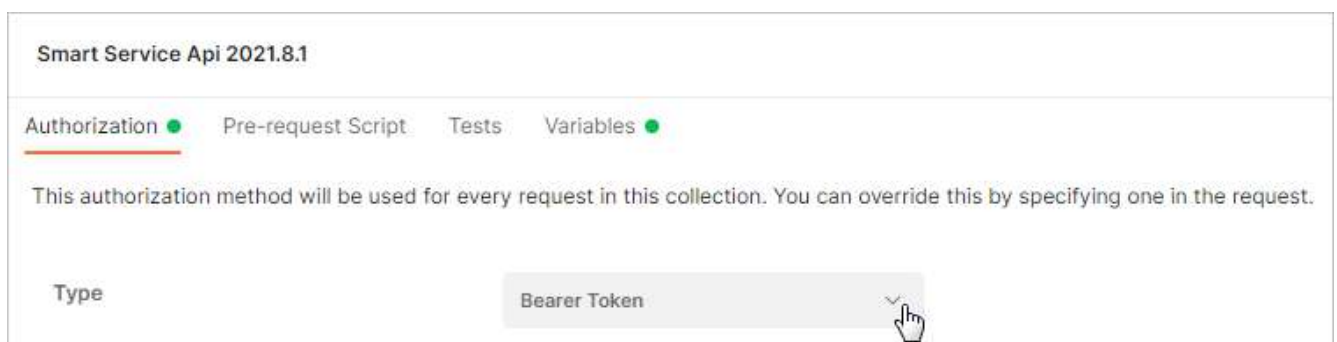
3. Отправьте запрос по кнопке

В параметре **access_token** ответа будет содержаться токен.

Пример ответа:

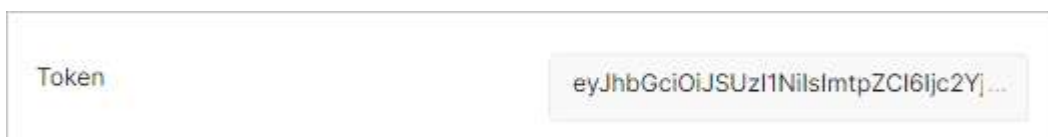
```
{
  "access_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6Ijc2YjZjMzExMzhjNGRhOTI2ODZiZDZhNTUzNmJmODY5IiwiaHR0cDovL2xvY2FsaG9zdDo1MDAwIiwiaXVkiIjpbImh0dHA6Ly9sb2NhbGhvc3Q6NTAwMC9yZXNvdXJzZXMiLCJBCmlvQXBpI10sImNsawVudF9pZCI6IkFyaw8iLCJzY29wZSI6WyJBCmlvQXBpI119.VZJlGy22zFguCOONHVGlDUi5V2tyz7cFGbnBG4GME4q0qsQzvz600sN2pMo5l4masBcluafr7r18LQIsVNNp2McevZQJZX8OD7x8AggV_E9sm6pfYHHV00Se99cDFd7AwsirXvLwU3355RaIEhvnyBW_0zhmk3",
  "expires_in": 86400,
  "scope": "Smart ServicesApi IdentityServerApi"
}
```

4. В контекстном меню коллекции выберите пункт **Edit** и на закладке **Authorization** в выпадающем списке выберите пункт **Bearer Token**:



5. В появившемся поле **Token** укажите полученный токен.

Пример:



6. Сохраните настройки коллекции.

ВАЖНО. Токен действителен 24 часа с момента выдачи. После этого необходимо получить новый. Если токен невалидный, то при выполнении запроса с переданным токеном:

- Status Code имеет значение **401**;
- заголовок WWW-Authenticate содержит сообщение «Bearer error="invalid_token"».

Если токен просрочен, то дополнительно появляется сообщение «Bearer error="invalid_token", error_description="The token is expired"».

Настройка классификации документов

Понятие классификации

Классификация документа – один из этапов обработки, на котором определяется, к какому классу относится документ. Например, в качестве классов могут использоваться виды документов: входящее письмо, договор, акт выполненных работ и т.д. Для классификации необходимо [создать](#) и [обучить](#) классификатор на выборке документов каждого класса. В результате обучения формируется *модель классификации*, на основании которой выполняется *классификация* документов.

Базовые модели классификации

Для начала обработки документов с помощью сервисов Smart Services можно использовать базовые модели классификации, которые входят в поставку:

- **модель классификации по первым страницам** (DocumentFirstPageClassifierModel.zip), которая определяет, является ли страница файла первой в документе. Модель обучена на примерах первых и не первых страниц документов. С ее помощью происходит разделение комплекта на отдельные документы;
- **модель классификации по видам документов** (DocumentKindClassifierModel.zip), которая по тексту документа определяет, к какому виду он относится. Модель обучена на примерах различных видов документов и позволяет классифицировать следующие документы, [оформленные по правилам](#) и рекомендациям российского законодательства:
 - акт выполненных работ;
 - входящее письмо;
 - договор;
 - дополнительное соглашение;
 - корректировочный счет-фактура;
 - счет на оплату (в том числе англоязычный);
 - счет-фактура;

- товарная накладная;
- универсальный передаточный документ;
- универсальный корректировочный документ.

Настройка классификации

Чтобы выполнить настройки классификации, определите потребности организации. Возможны три случая:

1. Для обработки документов достаточно базовых моделей классификации и их доработка не требуется. В этом случае нужно только импортировать базовые модели классификации в классификатор и подключить используемую систему к сервисам Smart Services.
2. Базовые модели классификации в целом закрывают потребности, но нужны доработки: добавление в классификатор еще одного вида документа или обучение классификатора на реальных документах организации для повышения точности классификации. В этом случае дообучите базовые модели классификации.
3. Базовые модели классификации совсем не закрывают потребностей, нужно классифицировать другие виды документов. В таком случае необходимо создать и обучить новый классификатор.

Импорт базовых моделей классификации

Если базовые модели классификации закрывают все или основные потребности организации в обработке документов, то импортируйте их в классификатор одним из способов:

- [скриптами](#);
- с помощью Postman. Для этого:
 - а) [Создайте соответствующий классификатор](#), например по видам документов.
 - б) В коллекции *Postman* выберите POST-запрос «Импорт модели классификатора».
 - в) В строке запроса укажите **Id** созданного ранее классификатора.
Вид строки:
`{service_url}/api/Classifiers/{Id классификатора}/models/import`
Пример:
`http://smart:61100/api/Classifiers/1/models/import`
 - д) На закладке **Body** в параметре **file** выберите zip-архив с нужной [моделью классификации](#) из комплекта поставки, и отправьте запрос.

Пример запроса:

```
POST http://smart:61100/api/Classifiers/1/models/import
{
  "file": "DocumentkindClassifierModel.zip"
}
```

ПРИМЕЧАНИЕ. Описанные способы импорта равнозначны. Выбирайте удобный для вас.

После импорта базовых моделей можно начать классифицировать документы организации с их помощью, поэтому остается только подключить используемую систему к сервисам Smart Services.

Однако под конкретные бизнес-задачи и для повышения точности классификации реальных документов организации рекомендуется либо [дообучить классификатор по виду](#), либо [создать и обучить новый](#).

Дообучение базового классификатора по виду документа

[Дообучение](#) используется, когда перечень классифицируемых базовым классификатором видов и качество классификации реальных документов организации в целом закрывают ожидания и потребности в обработке документов. Но есть необходимость «научить» классификатор определять несколько новых классов и повысить точность классификации документов организации.

Создание и обучение нового классификатора

Создание и обучение нового классификатора необходимо, если:

- организация работает с видами документов, которые не входят в [перечень](#) классифицируемых;
- в организации принято оформление документов, отличное от [общепринятых форматов](#).

В этом случае:

1. [Подготовьте набор](#) документов.
2. [Создайте классификатор](#).
3. [Обучите модель классификации](#).
4. [Проанализируйте результаты обучения](#).
5. [Протестируйте классификатор](#).

Для каждого типа классификации, например по видам документов, по типу финансовой услуги или по предмету спора, нужно создавать и обучать свой классификатор.

ВАЖНО. Из-за особенностей обучения классификатора по первым страницам обучить или дообучить его самостоятельно нельзя. Если необходимо разделять комплекты документов по логике, отличающейся от [стандартной](#), обратитесь в службу поддержки.

В процессе работы сервисов можно также [просматривать информацию](#) о классификаторе и его моделях для выполнения дополнительных действий, например для экспорта и последующего импорта классификатора в прикладную систему.

Подготовка документов для обучения классификатора

Обучить *классификатор* можно одним из способов:

- указать путь к папке с документами. Используется для обучения многоклассового классификатора.
- передать CSV-файл;
- указать путь к папке с CSV-файлом.

Последние два используются при обучении многозначного классификатора, но подходят и для обучения многоклассового.

Проанализируйте, какой способ подходит вам, и в зависимости от этого подготовьте для обучения [папку с документами](#) или [CSV-файл](#).

Структура папки с документами

1. Составьте перечень видов документов, обработку которых нужно автоматизировать.
2. Отберите не менее 100 документов каждого вида той *формы* (оформления), которая используется в организации.

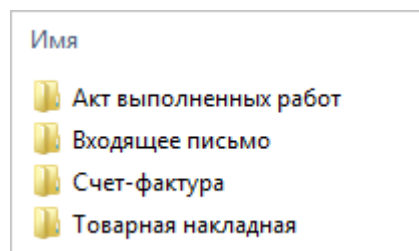
ПРИМЕЧАНИЕ. Обратите внимание, что в выборку должны входить все возможные формы, даже если используется несколько форм одного вида документа.

3. При необходимости разделите комплекты на отдельные документы в соответствии с принятым в организации форматом их хранения и классификации в системе.

Чаще всего комплекты нужно делить так, чтобы один файл содержал один документ одного вида. То есть файл не должен содержать, например, акт выполненных работ и счет на оплату или два счета на оплату.

Однако, если принято хранить документы разного вида как один объект, то исходный файл может содержать разные документы. Например, договор и приложение к нему, если принято хранить их в системе одним документом и классифицировать как «Договор».

4. Создайте в локальной папке на компьютере с установленными сервисами Smart Services или в сетевой папке структуру подпапок, каждая из которых содержит документы одного вида:



Классы, которые будут созданы после обучения, получают названия соответствующих папок.

ВАЖНО. В операционной системе на базе Linux структуру подкаталогов необходимо создавать внутри каталога `./dataset`.

5. В каждой папке разместите документы соответствующего вида. [Форматы документов](#) должны поддерживаться сервисом Text Extractor Service. То есть в папке «Акт выполненных работ» должны лежать только акты выполненных работ. Чем больше документов, тем выше точность классификации, но дольше обучение модели.

Структура CSV-файла

1. Файл должен содержать колонки, указанные в конфигурационном файле сервиса [Text Classifier Service](#):
 - `csv_text_field`;
 - `csv_class_field`;
 - `csv_classes_field`.

Файл может содержать колонку `csv_external_id_field`.

2. Текст в колонке `csv_text_field` заключается в апострофы.
3. Если имя класса содержит запятую, то при многоклассовой классификации его необходимо заключить в апострофы.
4. При многозначной классификации:
 - список классов заключается в апострофы и квадратные скобки;
 - классы в списке перечисляются через запятую;
 - каждый класс заключается в кавычки.
5. Csv файл должен содержать один из параметров: **csv_class_field** или **csv_classes_field**.

Пример секции конфигурационного файла сервиса Text Classifier Service:

```
csv_text_field = Text
csv_class_field = Category
csv_classes_field = Categories
csv_external_id_field = ExternalId
```

Пример csv файла для обучения многоклассового классификатора:

```
Id,Category,Text
1,Акт выполненных работ,`Текст документа`
2,Входящее письмо,`Текст документа`
3,Входящее письмо,`Текст документа`
4,`Входящее, исходящее письмо`,`Текст документа`
```

Пример csv файла для обучения многозначного классификатора:

```
Id,Categories,Text
0,["Performer_0", "Performer_1"],`Текст документа`
1,["Performer_1"],`Текст документа`
2,["Performer_2", "Performer_0"],`Текст документа`
3,["Performer_1"],`Текст документа`
```

Создание классификатора

1. В [импортированной ранее](#) коллекции Postman выберите POST-запрос «Создание классификатора».
2. На закладке **Body** в параметре **name** укажите наименование классификатора.

Пример запроса:

```
POST http://smart:61100/api/Classifiers
Content-Type: application/json
{
  "name": "Классификатор по виду документа"
  "minProbability": "0.75"
  "autoPublish": true
}
```

3. Отправьте запрос.

В результате создается новый классификатор, идентификационный номер которого содержится в параметре **classifierId**.

Пример ответа:

```
{
  "publishedModelId": null,
  "publishedModel": null,
  "classifierModels": [],
  "classifierId": 1,
  "name": "Классификатор по виду документа",
  "status": 0,
  "created": "2021-09-17T14:21:49.243236",
  "published": null,
  "autoPublish": true,
  "minProbability": 0.75,
  "useFixedForms": false
}
```

Обучение классификатора

Для классификации документов необходимо обучить *классификатор*. В результате его обучения формируется *модель классификации*.

Модель представляет собой совокупность признаков для каждого класса документов, по которым классификатор в дальнейшем определяет класс среди множества прочих. Другими словами, модель классификации – это инструмент, с помощью которого классификатор определяет, к какому классу относится документ.

Как правило, классификатор содержит несколько моделей классификации, так как в результате каждого цикла обучения формируется новая, улучшенная модель, но только одна из них может быть активной (действующей).

Для обучения выберите один из способов:

- указание пути к папке с документами;
- передача CSV-файла;
- указание пути к папке с CSV-файлом.

ПРИМЕЧАНИЕ. Время обучения зависит от количества документов.

ВАЖНО. Обучение классификатора возможно при наличии минимум двух новых классов документов.

Указание пути к папке

1. В коллекции Postman выберите синхронный POST-запрос «Обучение через указание пути к папке».
2. В строке запроса укажите **Id** [созданного ранее](#) классификатора.

Вид строки:

{service_url}/api/Classifiers/{Id классификатора}/trainfolder

Пример:

<http://smart:61100/api/Classifiers/1/trainfolder>

3. На закладке **Body** в параметре **path** укажите путь до корневой папки с [подготовленными подпапками](#) каждого класса и отправьте запрос.

ПРИМЕЧАНИЕ. Путь к папке задается через двойной обратный слеш. Например, C:\KindClassifier\Kinds.

Пример запроса:

```
POST http://smart:61100/api/Classifiers/1/trainfolder
Content-Type: application/json
{
  "path": "D:\\Виды документов",
  "languages": ["rus", "eng"]
}
```

В результате обучения в классификаторе создается новая модель классификации, обученная на подготовленной выборке документов.

Обучена модель **classifierModelId** с идентификатором **1**, которая может классифицировать акты выполненных работ:

```
{
  "classifierModelId": 1,
  "created": "2021-09-17T14:21:49.243236+04:00",
  "metrics": {
    "metricsId": 1,
    "trainCount": 180,
    "testCount": 20,
    "accuracy": 0.9,
    "precision": 0.8766666666666666,
    "recall": 0.9,
    "f1Measure": 0.8792063492063493,
    "trainDuration": 1.0480735301971436,
    "testDuration": 0.04298520088195801
  },
  "trainErrors": null,
  "classes": [
    {
      "classId": 1,
      "classifierModelId": 1,
      "displayName": "Акт выполненных работ",
      "name": "Акт выполненных работ",
      "metrics": {
        "metricsId": 2,
        "trainCount": 16,
        "testCount": 4,
        "accuracy": 0,
        "precision": 0.8,
        "recall": 1,
        "f1Measure": 0.8888888888888889,
        "trainDuration": 0,
        "testDuration": 0
      }
    }
  ]
}
```

Передача CSV-файла

1. Подготовьте [CSV-файл](#) для обучения модели и разместите его в локальной папке на компьютере, на котором установлены сервисы Smart Services, или в сетевой папке.
2. В коллекции *Postman* выберите синхронный POST-запрос «Обучение через CSV».
3. В строке запроса укажите **Id** созданного классификатора.

Вид строки:

```
{service_url}/api/Classifiers/{Id классификатора}/traincsv
```

Пример:

```
http://smart:61100/api/Classifiers/1/traincsv
```

4. На закладке **Body** в параметре **file** выберите нужный файл и отправьте запрос.

В результате обучения в классификаторе создается новая модель классификации, обученная на подготовленном файле.

```
{
  "classifierModelId": 2,
  "created": "2021-09-17T14:21:49.243236+04:00",
  "metrics": {
    "metricsId": 23,
    "trainCount": 176,
    "testCount": 20,
    "accuracy": 0.9,
    "precision": 0.8833333333333332,
    "recall": 0.9,
    "f1Measure": 0.8800000000000001,
    "trainDuration": 0.9200775623321533,
    "testDuration": 0.056002140045166016
  },
  "trainErrors": null,
  "classes": [
    {
      "classId": 21,
      "classifierModelId": 2,
      "displayName": "Акт выполненных работ",
      "name": "Акт выполненных работ",
      "metrics": {
        "metricsId": 24,
        "trainCount": 19,
        "testCount": 1,
        "accuracy": 0,
        "precision": 1,
        "recall": 1,
        "f1Measure": 1,
        "trainDuration": 0,
        "testDuration": 0
      }
    }
  ]
}
```

Указание пути к CSV-файлу

1. Подготовьте [CSV-файл](#) для обучения модели и разместите его в каталоге ./dataset
2. В коллекции Postman выберите синхронный POST-запрос «Обучение через путь к CSV».
3. В строке запроса укажите **Id** созданного классификатора.

Вид строки:

{service_url}/api/Classifiers/{Id классификатора}/traincsvpath

Пример:

http://smart:61100/api/Classifiers/1/traincsvpath

4. На закладке **Body** в параметре **file** укажите путь к CSV-файлу.

Пример запроса:

```
POST http://smart:61130/api/Classifiers/1/traincsvpath
Content-Type: application/json
{
  "file": "D:\\train_dataset.csv",
  "isMultilabel": false
}
```

5. Отправьте запрос.

В результате обучения в классификаторе создается новая модель классификации, обученная на подготовленном файле.

```
{
  "classifierModelId": 3,
  "created": "2021-09-17T14:21:49.243236+04:00",
```

```

"metrics": {
  "metricsId": 12,
  "trainCount": 176,
  "testCount": 20,
  "accuracy": 0.85,
  "precision": 0.9333333333333332,
  "recall": 0.85,
  "f1Measure": 0.8549999999999999,
  "trainDuration": 0.9430665969848633,
  "testDuration": 0.04201531410217285
},
"trainErrors": null,
"classes": [
  {
    "classId": 32,
    "classifierModelId": 2,
    "displayName": "Акт выполненных работ",
    "name": "Акт выполненных работ",
    "metrics": {
      "metricsId": 13,
      "trainCount": 18,
      "testCount": 2,
      "accuracy": 0,
      "precision": 1,
      "recall": 1,
      "f1Measure": 1,
      "trainDuration": 0,
      "testDuration": 0
    }
  }
]

```

Анализ результатов обучения

Для оценки качества обучения классификатора по виду документов используйте метрики:

- **precision** (точность). Отображает долю документов, которые отнесены *классификатором* к классу и действительно принадлежат ему. Другими словами, отображает, сколько полученных от классификатора положительных ответов являются правильными. Например, на обработку отправлено 100 договоров и 100 дополнительных соглашений. Предположим, что классификатор определил:
 - 110 документов как «Договор». При этом 100 из них действительно являются договорами, но к ним также ошибочно отнесено десять доп.соглашений;
 - 90 документов как «Дополнительное соглашение». При этом все из них действительно являются доп.соглашениями.

Таким образом, процент точности классификации договоров составляет – 90%, а дополнительных соглашений – 100%.

- **recall** (полнота). Отображает долю документов нужного класса, которые нашел классификатор, по отношению к тем документам, которые он ошибочно пропустил. Другими словами, метрика отображает способность классификатора предсказывать как можно большее число положительных ответов среди ожидаемых. Например, на обработку отправлено 100 договоров и 100 дополнительных соглашений. Предположим, что классификатор определил:
 - 90 документов как «Договор»;
 - 110 как «Дополнительное соглашение».

Таким образом, процент полноты классификации договоров составляет – 90%, а дополнительных соглашений – 100%.

- **f1Measure** (мера успешности обучения). Так как одновременное достижение максимальной точности и полноты классификации в реальности практически невозможно, вводится метрика, которая отображает гармоническое среднее между точностью (precision) и полнотой (recall). По ее значению можно сделать вывод о результатах обучения:
 - 0,8-0,84 – удовлетворительно;
 - 0,85-0,93 – хорошо;
 - 0,94 и выше – отлично.

Максимальное значение каждой метрики **1**. Чем больше значение, тем точнее и полнее результат обучения.

Информация о метриках содержится в соответствующих параметрах ответа.

Пример:

```
{
  "classifierModelId": 1,
  "created": "2021-09-17T14:57:17.5297316+04:00",
  "metrics": {
    "metricsId": 1,
    "trainCount": 180,
    "testCount": 20,
    "accuracy": 0.9,
    "precision": 0.8766666666666666,
    "recall": 0.9,
    "f1Measure": 0.8792063492063493,
    "trainDuration": 1.0480735301971436,
    "testDuration": 0.04298520088195801
  }
}
```

В примере мера успешности обучения (**f1Measure**) больше 0,85, но меньше 0,93 – значит, модель обучена хорошо.

Тестирование классификатора

Для проверки созданного классификатора:

1. В коллекции *Postman* выберите синхронный POST-запрос «Классификация файла».
2. В строке запроса укажите **Id** созданного классификатора.

Вид строки:

{service_url}/api/Classifiers/{Id классификатора}/predictfile

Пример:

<http://smart:61100/api/Classifiers/1/predictfile>

3. На закладке **Body** в параметре **file** выберите файл для тестирования из локальной или сетевой папки и отправьте запрос на классификацию.

ПРИМЕЧАНИЕ. Для объективной оценки работы классификатора рекомендуется выбирать документ, относящийся к одному из видов, на которых обучен классификатор, но при этом сам документ не входил в выборку для обучения.

4. Дождитесь ответа сервиса и изучите секции **classResults** и **predictedClass**:

- в секции **classResults** отражаются все возможные классы, присущие модели, и вероятность, с которой документ относится к тому или иному из них. Как правило, с наибольшей вероятностью документ будет относиться к классу, который указан в секции первым;
- в секцию **predictedClass** попадает класс, вероятность соответствия которому равна или больше значения, указанного в параметре **minProbability** самого классификатора.

Дообучение классификатора

В процессе работы может потребоваться дообучить *модели классификации*. Например, когда:

- в организации появились новые *формы* документов для уже существующих классов;
- нужно обучить *классификатор* распознавать дополнительные классы документов;
- нужно повысить точность классификации, добавив к исходной выборке документов новые документы тех же классов.

Чтобы не переобучать классификатор заново, можно использовать соответствующий запрос на дообучение. Принципиальная разница дообучения и обучения состоит в том, что:

- при дообучении к результатам обучения исходной модели на первичных документах добавляются результаты обучения на вновь подобранных документах;
- при обучении исходная модель заменяет новую.

Дообучение выполняется [способами](#) аналогичными тем, которые используются при обучении, но имеет [ограничения](#), которые необходимо учитывать.

Подробнее рассмотрим способ дообучения через указание пути к папке.

1. [Подготовьте документы](#), на которых будет дообучаться классификатор.
2. В коллекции *Postman* выберите синхронный POST-запрос «Дообучение через указание пути к папке».
3. В строке запроса укажите **Id** классификатора и модели классификации.

Вид строки:

```
{service_url}/api/Classifiers/{Id классификатора}/trainfolder/{Id модели классификации}
```

Пример:

```
http://smart:61100/api/Classifiers/1/trainfolder/3
```

4. На закладке **Body** в параметре **path** укажите путь до корневой папки с [подготовленными подпапками](#) каждого класса и отправьте запрос.

ПРИМЕЧАНИЕ. Путь к папке задается через двойной обратный слеш. Например, C:\KindClassifier\Kinds.

Пример запроса:

```
POST http://smart:61100/api/Classifiers/1/trainfolder/3
Content-Type: application/json
{
  "path": "D:\\виды документов для дообучения",
  "languages": ["rus", "eng"]
}
```

В результате дообучения создается новая модель классификации.

На примере фрагмента ответа ниже рассмотрим, на какие секции и значения параметров нужно обратить внимание в этом запросе. В них содержится:

- в параметре **classifierModelId** – идентификационный номер новой модели классификации;
- в первой секции **metrics** – информация об [основных метриках](#) модели, по которым можно оценить успешность обучения классификатора;
- в секции **classes** – параметры с информацией обо всех классах, входящих в модель классификации (идентификационный номер класса, наименование класса, метрики результатов обучения).

Пример фрагмента ответа:

```
{
  "classifierModelId": 4,
  "created": " 2021-09-21T11:21:49.243236+04:00",
  "metrics": {
    "metricsId": 1,
    "trainCount": 180,
    "testCount": 20,
    "accuracy": 0.9,
    "precision": 0.8766666666666666,
    "recall": 0.9,
    "f1Measure": 0.8792063492063493,
    "trainDuration": 1.0480735301971436,
    "testDuration": 0.04298520088195801
  },
  "trainErrors": null,
  "classes": [
    {
      "classId": 1,
      "classifierModelId": 4,
      "displayName": "Акт выполненных работ",
      "name": "Акт выполненных работ",
      "metrics": {
        "metricsId": 2,
        "trainCount": 16,
        "testCount": 4,
        "accuracy": 0,
        "precision": 0.8,
        "recall": 1,
        "f1Measure": 0.8888888888888889,
        "trainDuration": 0,
        "testDuration": 0
      }
    }
  ]
}
```

5. [Протестируйте работу](#) дообученного классификатора.

Ограничения при дообучении классификатора

1. Дообучение невозможно, если используется версия сервисов Smart Services ниже 2.0.
2. Дообучение многозначного классификатора возможно только через передачу CSV-файла или через указание пути к папке с ним.
3. Количество документов, на котором будет производится дообучение, должно быть примерно равно первичной выборке документов. Например, при дообучении исходной модели на новый класс количество документов этого класса должно быть приблизительно равным количеству документов, на котором обучены остальные классы.

СОВЕТ. Чтобы узнать, на каком количестве документов обучен класс, выполните запрос на [просмотр информации о модели классификации](#).

- Из модели классификации нельзя удалять классы. Например, если в модели есть класс «Договор», а требуется, чтобы вместо него были «Договор расходный» и «Договор доходный», то нельзя удалить или переименовать класс. Необходимо [переобучить](#) модель.

Информационные запросы

Просмотр списка классификаторов

Запрос используется для отображения списка классификаторов.

В коллекции *Postman* выберите GET-запрос «Список классификаторов» и отправьте запрос.

Пример запроса:

```
http://smart:61130/api/classifiers
```

На примере фрагмента ответа ниже рассмотрим, на какие секции и значения параметров нужно обратить внимание в этом запросе. В них содержится:

- в параметре **classifierId** – идентификационный номер классификатора;
- в параметре **name** – наименование классификатора;
- в параметре **status** – статус классификатора: активный (действующий) или неактивный (недействующий). Возможные значения: **0** – неактивный, **1** – активный;
- в параметре **minProbability** – граница минимальной вероятности для определения класса. Если при классификации вероятность будет ниже указанного значения, то определить класс не удастся;
- в секции **publishedModel** – параметры с информацией об активной модели классификации (идентификационный номер модели, метрики);
- в секции **classes** – параметры с информацией обо всех классах, входящих в модель классификации (идентификационный номер класса, наименование класса).

```
{
  "classifierId": 2,
  "name": "Классификатор первых страниц",
  "status": 1,
  "created": "2018-12-14T11:29:00.792031",
  "published": "2018-12-14T11:30:13.189545",
  "autoPublish": true,
  "minProbability": 0.6,
  "publishedModelId": 11,
  "publishedModel": {
    "classifierModelId": 11,
    "created": "2018-12-14T11:30:13.183036",
    "metrics": {
      "metricsId": 103,
      "trainCount": 17264,
      "testCount": 1919,
      "accuracy": 0.9729025534132361,
      "precision": 0.9729434690602494,
      "recall": 0.9729025534132361,
      "f1Measure": 0.9729133640420062,
      "trainDuration": 11.273868083953857,
      "testDuration": 0.55906081199646
    },
    "trainErrors": null,
    "classes": [
      {
        "classId": 93,
```

```

        "classifierModelId": 11,
        "displayName": "Not First",
        "name": "Not First",
        "metrics": null,
        "externalId": ""
      },
      {
        "classId": 94,
        "classifierModelId": 11,
        "displayName": "First",
        "name": "First",
        "metrics": null,
        "externalId": ""
      }
    ]
  }
}

```

Просмотр списка моделей, используемых в классификаторе

Запрос используется для отображения списка моделей классификатора с указанием их идентификаторов.

1. В коллекции Postman выберите GET-запрос «Список моделей классификатора».
2. В строке запроса укажите **Id** нужного классификатора.

Вид строки:

{service_url}/api/Classifiers/{Id классификатора}/models

Пример запроса:

<http://smart:61130/api/Classifiers/4/models>

На примере фрагмента ответа ниже рассмотрим, на какие секции и значения параметров нужно обратить внимание в этом запросе. В них содержится:

- в параметре **classifierModelId** – идентификационный номер модели классификации;
- в секции **metrics** – [метрики](#) модели классификации.

```

{
  "classifierModelId": 5,
  "classes": null,
  "recommendedThreshold": null,
  "withDataset": true,
  "created": "2021-09-11T12:25:30.814221",
  "metrics": {
    "metricsId": 30,
    "trainCount": 1814,
    "testCount": 202,
    "accuracy": 0.995049504950495,
    "precision": 0.9951411807847452,
    "recall": 0.995049504950495,
    "f1Measure": 0.995064303938182,
    "trainDuration": 35.764763832092285,
    "testDuration": 2.624953031539917
  },
  "trainErrors": null
},
{
  "classifierModelId": 38,
  "classes": null,
  "recommendedThreshold": null,
  "withDataset": true,
  "created": "2021-09-20T17:23:31.328956",
  "metrics": {
    "metricsId": 173,
    "trainCount": 3506,
    "testCount": 390,
    "accuracy": 0.9256410256410257,
    "precision": 0.9259273595542283,

```

```

        "recall": 0.9256410256410257,
        "f1Measure": 0.9249776542080481,
        "trainDuration": 9.23331880569458,
        "testDuration": 0.5623910427093506
      },
      "trainErrors": null
    }
  }
}

```

Просмотр информации о модели классификации

Запрос может использоваться для отображения информации:

- о конкретной модели классификатора, например сведений о входящих в ее состав классов;
- о количестве документов, на которых обучен класс;
- о метриках, которые помогают [оценить качество обучения](#) класса.

1. В коллекции Postman выберите GET-запрос «Информация о модели классификатора».
2. В строке запроса укажите **Id** классификатора и нужной модели в нем.

Вид строки:

{service_url}/api/Classifiers/{Id классификатора}/models/{Id модели}

Пример запроса:

http://smart:61130/api/Classifiers/4/models/5

На примере фрагмента ответа ниже рассмотрим, на какие секции и значения параметров нужно обратить внимание в этом запросе. В них содержится:

- в параметре **classifierModelId** – идентификационный номер модели классификации;
- в секции **classes** – параметры с подробной информацией о каждом классе, входящем в модель классификации (идентификационный номер класса, наименование класса, метрики).
- в секции **metrics** – [метрики](#) класса;
- в параметре **trainCount** – количество документов, на котором обучен класс;
- в параметре **testCount** – количество документов, которое откладывается при обучении на тестирование модели.

```

{
  "classifierModelId": 5,
  "classes": [
    {
      "classifierModelId": 5,
      "externalId": "",
      "isFixedForm": null,
      "classId": 26,
      "displayName": "Входящее письмо",
      "name": "Входящее письмо",
      "metrics": {
        "metricsId": 31,
        "trainCount": 541,
        "testCount": 53,
        "accuracy": 0.0,
        "precision": 0.9814814814814815,
        "recall": 1.0,
        "f1Measure": 0.9906542056074767,
        "trainDuration": 0.0,
        "testDuration": 0.0
      }
    }
  ],
  "trainErrors": null
}

```

```

        "classifierModelId": 5,
        "externalId": "",
        "isFixedForm": null,
        "classId": 27,
        "displayName": "Договор",
        "name": "Договор",
        "metrics": {
            "metricsId": 32,
            "trainCount": 1273,
            "testCount": 149,
            "accuracy": 0.0,
            "precision": 1.0,
            "recall": 0.9932885906040269,
            "f1Measure": 0.9966329966329966,
            "trainDuration": 0.0,
            "testDuration": 0.0
        }
    },
    "recommendedThreshold": null,
    "withDataset": false,
    "created": "2018-08-31T12:25:30.814221",
    "metrics": {
        "metricsId": 30,
        "trainCount": 1814,
        "testCount": 202,
        "accuracy": 0.995049504950495,
        "precision": 0.9951411807847452,
        "recall": 0.995049504950495,
        "f1Measure": 0.995064303938182,
        "trainDuration": 35.764763832092285,
        "testDuration": 2.624953031539917
    },
    "trainErrors": null
}

```

Иерархическая классификация

Задача

Механизмы классификации в Smart Services показывают отличные результаты для относительно небольшого количества классов (в среднем до 20-30). Этого достаточно для решения большинства задач по обработке документов.

В отдельных случаях необходимо классифицировать документы по значительно большему количеству классов. Например, классификатор для обращений граждан может включать более 1000 классов.

Как правило, в таких классификаторах есть схожие классы, которые можно объединить в группы. Например, группа «Договорные документы» может включать в себя «Договор подряда», «Договор аренды», «Договор купли-продажи» и т.д. Такие документы отличаются друг от друга значительно меньше, чем от других видов документов, например, писем или счетов-фактур. Из-за этого при их классификации с помощью сервисов Smart Services могут возникать ошибки, в то время как существенно отличающиеся между собой документы классифицируются корректно.

Для решения подобных задач рекомендуется использовать иерархическую классификацию, при которой документы классифицируются в несколько этапов: сначала по группам (верхнеуровневая классификация), затем по классам. За счет этого повышается точность классификации документов.

Решение

Чтобы создать иерархическую классификацию:

1. [Объедините](#) отдельные классы документов в группы.
2. [Обучите](#) классификаторы различных уровней.
3. [Настройте](#) иерархическую классификацию в рамках интеграционного решения.

Группировка классов

Определите, можно ли объединить схожие классы в группы. Используйте один из способов:

- сгруппируйте классы в соответствии с существующей иерархией. Например, для обращений граждан есть утвержденный иерархический классификатор, который состоит из 4 уровней (раздел, тематика, тема, вопрос);
- проанализируйте предметную область документов и выделить группы документов. Например, можно объединить:
 - все виды договоров – в класс «Договоры»;
 - письма, коммерческие предложения и рекламные материалы – в класс «Письма»;
 - счета на оплату, акты выполненных работ, товарные накладные, счета-фактуры, УПД – в класс «Финансовые документы».

СОВЕТ. Для видов документов, которые сложно отнести к какой-либо группе, можно выделить группу «Прочие» или не включать их в группы, оставив на верхнем уровне.

В результате на каждом уровне должно остаться не более 20 групп, которые будут выступать в качестве классификаторов подуровней.

Обучение классификаторов различных уровней

Подготовьте наборы данных для классификаторов и обучите их. Для этого:

1. [Подготовьте набор данных](#) для обучения верхнеуровневого классификатора. Он включает в себя классы, соответствующие классификаторам подуровней. Для каждого из них создайте папку и поместите в нее соответствующие документы, которые будут использоваться для обучения.
Например, чтобы обучить классификатор по видам документов, подготовьте папки с документами следующих видов:
 - «Договоры»;
 - «Письма»;
 - «Финансовые документы» и т.д.
2. Создайте верхнеуровневый классификатор с помощью POST-запроса [«Создание классификатора»](#).

ПРИМЕЧАНИЕ. Ответ на запрос «Создание классификатора» содержит идентификатор классификатора. Используйте этот идентификатор в последующих запросах.

3. Обучите верхнеуровневый классификатор на основе подготовленного набора данных с помощью POST-запроса [«Обучение через указание пути к папке»](#). В строке запроса укажите идентификатор классификатора.

Пример:

```
http://smart:61100/api/Classifiers/1/trainfolder
```

4. Подготовьте набор данных для классификаторов подуровней. Например, для классификатора «Договоры» подготовьте папки с документами следующих видов:
 - «Договор аренды»;
 - «Договор подряда»;
 - «Договор купли-продажи»;
 - «Договор страхования».

Создайте и обучите классификаторы аналогично верхнеуровневому классификатору.

5. Аналогичным образом создайте и обучите классификаторы последующих уровней при их наличии. Количество классификаторов будет равно количеству групп на каждом уровне, за исключением нижнего, плюс один верхнеуровневый классификатор.

Настройка иерархической классификации

В прикладной системе реализуйте сопоставление классов верхнеуровневого классификатора и классификаторов последующих уровней. Это можно сделать одним из способов:

- с помощью [дополнительных классификаторов](#);
- путем [последовательной классификации](#).

В обоих случаях классификация выполняется в несколько запросов на обработку документов.

Рекомендации:

- рассчитывайте итоговую вероятность классификации как произведение вероятностей верхнеуровневого классификатора и выбранного классификатора подуровня;
- в первом запросе документ конвертируется в формат PDF с метаданными. В последующих запросах рекомендуется отправлять на обработку уже сконвертированный PDF-файл, полученный по идентификатору GUID. Обработка такого файла будет выполнена существенно быстрее, чем при отправке исходного документа.

Дополнительные классификаторы

1. Классифицируйте документы с помощью POST-запроса [«Классификация файла несколькими классификаторами»](#), передав классификаторы подуровней в качестве дополнительных. Для этого в теле запроса укажите параметры:

- **ClassifierId** – идентификатор верхнеуровневого классификатора;
- **AdditionalClassifierIds** – идентификаторы классификаторов подуровней.

В результате для классификации документов будут использоваться все классификаторы. Результаты обработки возвращаются в параметрах:

- **classificationResult** – для верхнеуровневого классификатора;
- **additionalClassificationResults** – для классификаторов подуровней.

- В прикладной системе реализуйте логику для сопоставления классов верхнеуровневого классификатора (**classificationResult**) с классификаторами подуровней (**additionalClassificationResults**). Для этого разработайте справочник соответствия. Подробнее см. руководство по разработке для прикладной системы.

Класс документа определяется из ответов, вернувшихся в параметре **additionalClassificationResults**.

- Для полученного класса определите необходимую *грамматику* и выполните запрос на извлечение фактов. Подробнее см. в [описании](#) методов API сервисов Smart Services.

Преимущества способа:

- классификация файла выполняется одним запросом.

Недостатки способа:

- большое количество дополнительных классификаторов может замедлить работу;
- увеличивается размер ответа на запрос. Он будет содержать результаты всех дополнительных классификаторов, из которых нужен только один.

Последовательная классификация

- Классифицируйте документы с помощью POST-запроса [«Классификация файла»](#). В строке запроса укажите идентификатор верхнеуровневого классификатора:

Пример:

```
http://smart:61100/api/Classifiers/1/predictfile
```

- В прикладной системе реализуйте логику для сопоставления полученного класса верхнеуровневого классификатора (**classificationResult**) с классификаторами подуровня. Для этого разработайте справочник соответствия. Подробнее см. руководство по разработке для прикладной системы.
- Выполните еще один POST-запрос «Классификация файла». В строке запроса укажите идентификатор классификатора подуровня, полученного при сопоставлении. В результате параметр **classificationResult** вернет класс документа.
- Для полученного класса определите необходимую *грамматику* и выполните запрос на извлечение фактов. Подробнее см. в [описании](#) методов API сервисов Smart Services.

Преимущества способа:

- при большом количестве классификаторов на втором и последующих уровнях классификация может выполняться быстрее, особенно для документов с большим (более 10) количеством страниц.

Недостатки способа:

- выполняется дополнительный запрос на классификацию файла.

Импорт и экспорт прикладной конфигурации

Работайте с *прикладной конфигурацией Smart Services* отдельно от сервисов, если необходимо:

- добавить или изменить правила извлечения фактов;
- добавить или изменить *модели извлечения фактов*;

- добавить или изменить *форматеры*;
- передать текущую конфигурацию для заказной разработки или устранения ошибок.

Для этого используйте [методы API](#) сервисов:

- при экспорте выполните запрос POST {{service_url}}/api/modules/export
где **service_url** – адрес сервиса Smart Service.

Сервис Publishing Service с помощью RabbitMQ собирает прикладную конфигурацию. По умолчанию конфигурация запрашивается у сервисов Fact Extractor Service, Fact Extractor Rules Service и Fact Extractor Learn Service. Если какой-то из сервисов не установлен, удалите его из списка в параметре **Services** [конфигурационного файла](#). Каждый сервис выгружает разработку в виде zip-архива и JSON-файла с его описанием. Publishing Service объединяет полученные архивы в один;

- при импорте выполните запрос POST {{service_url}}/api/modules/import

Сервис Publishing Service разбивает импортированный архив на отдельные части и передает сервисам с помощью RabbitMQ.

Сопровождение работы

Для обеспечения стабильности и производительности сервисов Smart Services администратор:

- [редактирует конфигурационные файлы](#) при необходимости;
- [анализирует и устраняет ошибки](#), возникающие в процессе обработки документов.

Конфигурирование настроек

Конфигурирование настроек выполняется с помощью единого конфигурационного файла .env, который находится в корневом каталоге сервисов. Параметры каждого сервиса вынесены в отдельные секции:

- [Smart Service](#);
- [Identity Service](#);
- [Publishing Service](#);
- [Text Extractor Service](#);
- [Text Classifier Service](#);
- [Image Classifier Service](#);
- [Fact Extractor Service](#);
- [Fact Extractor Rules Service](#);
- [Fact Extractor Learn Service](#);
- [Object Detection Service](#);
- [Storage Service](#);
- [Elastic APM Service](#);
- [база данных InfluxDB](#);
- [информационная панель Grafana](#);
- [серверный агент Telegraf](#).

Кроме того, в конфигурационном файле есть секции и параметры, [общие](#) для всех сервисов Smart Services.

При конфигурировании настроек учитывайте следующие рекомендации и особенности:

1. Для удаленных серверов вместо имен рекомендуется указывать IP-адреса.
2. В именах и паролях пользователей не используйте символы # @ \ / : * ? " < > |.
3. Для названия сервисов используется аббревиация. Например, сервис Text Classifier Service сокращен до dtcs, а Publishing Service до dps и т.д.
4. Названия параметров для разных сервисов именуются по разным моделям:
 - для сервисов Text Extractor Service, Smart Service, Publishing Service, Identity Service по модели: <аббревиатура сервиса>_<секция конфигурационного файла>_<параметр>. Например, dss_serilog_elastic_apm;
 - для остальных сервисов по модели: <аббревиатура сервиса>_<параметр>. Например, dfesl_debug.

Общая структура конфигурационного файла:

```
smart_services_version = {Версия сервисов Smart Services}

queues_message_ttl = {Время хранения сообщений в RabbitMQ в секундах}

# ContainersTimezone
timezone = {Часовой пояс внутри сервисов}

# Logs
log_path = {Корневой каталог для лог-файлов}
log_level = {Минимальный уровень протоколирования}
log_path_format = {Формат имени лог-файла}

dss_logs = {Подкаталог для лог-файлов Smart Service}
dtes_logs = {Подкаталог для лог-файлов Text Extractor Service}
dis_logs = {Подкаталог для лог-файлов Identity Service}
dps_logs = {Подкаталог для лог-файлов Publishing Service}
dfes_logs = {Подкаталог для лог-файлов Fact Extractor Service}
dfesr_logs = {Подкаталог для лог-файлов Fact Extractor Rules Service}
dfesl_logs = {Подкаталог для лог-файлов Fact Extractor Learn Service}
dods_logs = {Подкаталог для лог-файлов Object Detection Service}
dics_logs = {Подкаталог для лог-файлов Image Classifier Service}
dtcs_logs = {Подкаталог для лог-файлов Text Classifier Service}

# ServicePlacement
manager_node_label = {Label для управляющего сервера}
pg_node_label = {Label для сервера с СУБД PostgreSQL}
rabbit_node_label = {Label для сервера с брокером сообщений RabbitMQ}
storage_service_node_label = {Label для сервера с сервисом хранилищ}
dtes_node_label = {Label для сервера с Text Extractor Service}
dfes_node_label = {Label для сервера с Fact Extractor Service}
dfesr_node_label = {Label для сервера с Fact Extractor Rules Service}
dfesl_node_label = {Label для сервера с Fact Extractor Learn Service}
dods_node_label = {Label для сервера с Object Detection Service}
dics_node_label = {Label для сервера с Image Classifier Service}
dtcs_node_label = {Label для сервера с Text Classifier Service}

# Portainer
portainer_external_port = {TCP-порт Portainer}

# RabbitMQ
rabbit_replicas = {Количество запускаемых копий брокера сообщений}
rabbit_external_port = {TCP-порт брокера сообщений}
rabbit_user = {Имя пользователя брокера сообщений}
rabbit_password = {Пароль пользователя брокера сообщений}
rabbit_url = amqp://{rabbit_user}:{rabbit_password}@{Виртуальный хост}:{Порт}

# PostgreSQL
pg_replicas = {Количество запускаемых копий СУБД PostgreSQL}
pg_external_port = {TCP-порт СУБД PostgreSQL}
pg_host = {Адрес СУБД PostgreSQL}
pg_user = {Имя пользователя для подключения к СУБД PostgreSQL}
pg_password = {Пароль для подключения к СУБД PostgreSQL}
pg_database = {Имя базы данных PostgreSQL для Smart Service}
pg_database_dis = {Имя базы данных PostgreSQL для Identity Service}
```

```
pg_port = {Порт для подключения к СУБД PostgreSQL}
pg_connection = User
ID=${pg_user};Password=${pg_password};Host=${pg_host};Port=${pg_port};Database=${pg_database};Pooling={Использовать пул подключений к БД};
```

[StorageService](#)

```
storage_service_replicas = {Количество запускаемых копий сервиса хранения объектов}
storage_service_external_port = {TCP-порт сервиса хранения объектов}
storage_service_access_key = {Имя пользователя сервиса хранения объектов}
storage_service_secret_key = {Пароль пользователя сервиса хранения объектов}
```

[ElasticApm](#)

```
elastic_apm_replicas = {Количество запускаемых копий сервиса Elastic APM Service}
elastic_apm_external_port = {TCP-порт сервиса Elastic APM Service}
elastic_apm_server_urls = {Адрес сервиса Elastic APM Service}
elastic_apm_log_level = {Минимальный уровень протоколирования сервиса Elastic APM Service}
elastic_apm_transaction_sample_rate = {Регулярность отправки трейсов в сервис Elastic APM Service}
elastic_apm_span_frames_min_duration = {Время обработки документа при возникновении ошибки, при котором в лог-файл записывается подробная информация о ней. В миллисекундах}
elastic_apm_metric_interval = {Интервал сбора метрик в секундах}
elastic_apm_capture_body = {Сообщения, которые передаются в сервис Elastic APM Service}
elastic_apm_traces_path = {Путь до каталога с файлами трассировок}
elastic_apm_traces_file_name = {Базовое имя файлов трассировки}
elastic_apm_output_file_rotate_every_kb = {Максимальный размер файла трассировки}
elastic_apm_output_file_number_of_files = {Максимальное количество файлов трассировки}
```

[InfluxDB](#)

```
influxdb_replicas = {Количество запускаемых копий InfluxDB}
influxdb_user = {Имя пользователя InfluxDB}
influxdb_password = {Пароль пользователя InfluxDB}
```

[Grafana](#)

```
grafana_replicas = {Количество запускаемых копий Grafana}
grafana_external_port = {TCP-порт Grafana}
grafana_user = {Имя пользователя Grafana}
grafana_password = {Пароль пользователя Grafana}
```

[Telegraf](#)

```
telegraf_replicas = {Количество запускаемых копий Telegraf}
```

[SmartService](#)

```
dss_replicas = {Количество запускаемых копий сервиса Smart Service}
dss_address = {Адрес сервиса Smart Service}
dss_external_port = {TCP-порт сервиса Smart Service}
dss_connection_strings_default_connection = ${pg_connection}
dss_serilog_default = ${log_level}
dss_serilog_entity_framework_core = Error
dss_serilog_system = Error
dss_serilog_elastic_apm = Error
dss_serilog_application = SmartService
```

```

dss_serilog_name = RollingFile
dss_serilog_formatter = SmartService.Logger.SmartEcsTextFormatter, SmartService
dss_serilog_path_format = {Путь до каталога для логов Smart Service внутри
контейнера}
dss_identity_service = {Адрес сервиса Identity Service}
dss_allow_local_path = {Разрешить обучение классификаторов из локального каталога}
dss_allow_anonymous_access = {Разрешить анонимную работу}
dss_anonymous_client_id = {Логин анонимного клиента}
dss_anonymous_client_secret = {Пароль анонимного клиента}
dss_storage_endpoint = {Адрес сервиса хранения объектов}
dss_storage_access_key = ${storage_service_access_key}
dss_storage_secret_key = ${storage_service_secret_key}
dss_storage_bucket = {Имя контейнера}
dss_storage_uri_lifetime_seconds = {Время хранения ссылок на объекты в секундах}
dss_storage_file_lifetime_hours = {Время хранения объектов в часах}
dss_storage_retry_connection_count = {Количество повторных попыток подключения к
Storage Service}
dss_elastic_apm_server_urls = ${elastic_apm_server_urls}
dss_elastic_apm_transaction_sample_rate = ${elastic_apm_transaction_sample_rate}
dss_elastic_apm_span_frames_min_duration = ${elastic_apm_span_frames_min_duration}
dss_elastic_apm_metrics_interval = ${elastic_apm_metric_interval}
dss_elastic_apm_capture_body = ${elastic_apm_capture_body}
dss_elastic_apm_capture_body_content_types = "*"
dss_queues_message_ttl = ${queues_message_ttl}
dss_queue_connection = ${rabbit_url}
dss_queue_retry_connection_count = {Количество повторных попыток подключения к
RabbitMQ}

```

[IdentityService](#)

```

dis_replicas = {Количество запускаемых копий сервиса Identity Service}
dis_external_port = {TCP-порт сервиса Identity Service}
dis_serilog_default = ${log_level}
dis_serilog_entity_framework_core = Error
dis_serilog_system = Error
dis_serilog_elastic_apm = Error
dis_serilog_application = IdentityService
dis_serilog_name = RollingFile
dis_serilog_formatter = IdentityService.Logger.IdentityEcsTextFormatter,
IdentityService
dis_serilog_path_format = {Путь до каталога для логов Identity Service внутри
контейнера}
dis_connection_strings_default_connection = User
ID=${pg_user};Password=${pg_password};Host=${pg_host};Port=${pg_port};Database=${p
g_database_dis};Pooling={Использовать пул подключений к БД};
dis_elastic_apm_server_urls = ${elastic_apm_server_urls}
dis_elastic_apm_service_name = ${dis_log_properties_service_name}
dis_elastic_apm_log_level = ${elastic_apm_log_level}
dis_elastic_apm_transaction_sample_rate = ${elastic_apm_transaction_sample_rate}
dis_elastic_apm_span_frames_min_duration = ${elastic_apm_span_frames_min_duration}
dis_elastic_apm_metrics_interval = ${elastic_apm_metric_interval}
dis_elastic_apm_capture_body = ${elastic_apm_capture_body}
dis_certificate_name = {Наименование файла сертификата}
dis_certificate_password = {Пароль сертификата}

```

[PublishingService](#)

```

dps_replicas = {Количество запускаемых копий сервиса Publishing Service}
dps_external_port = {TCP-порт сервиса Publishing Service}
dps_serilog_default = ${log_level}
dps_serilog_entity_framework_core = Error
dps_serilog_system = Error
dps_serilog_elastic_apm = Error
dps_serilog_application = PublishingService
dps_serilog_name = RollingFile
dps_serilog_formatter = PublishingService.Logger.LoggerEcsTextFormatter,
PublishingService
dps_serilog_path_format = {Путь до каталога для логов Publishing Service внутри
контейнера}
dps_elastic_apm_server_urls = ${elastic_apm_server_urls}
dps_elastic_apm_transaction_sample_rate = ${elastic_apm_transaction_sample_rate}
dps_elastic_apm_span_frames_min_duration = ${elastic_apm_span_frames_min_duration}
dps_elastic_apm_metrics_interval = ${elastic_apm_metric_interval}
dps_elastic_apm_capture_body = ${elastic_apm_capture_body}
dps_elastic_apm_capture_body_content_types = "*"
dps_queue_connection = ${rabbit_url}
dps_queues_message_ttl = ${queues_message_ttl}
dps_queues_services = {Список сервисов}
dps_queue_retry_connection_count = {Количество повторных попыток подключения к
RabbitMQ}

```

[TextExtractor](#)

```

dtes_replicas = {Количество запускаемых копий сервиса Text Extractor Service}
dtes_external_port = {TCP-порт сервиса Text Extractor Service}
dtes_serilog_default = ${log_level}
dtes_serilog_microsoft = Warning
dtes_serilog_aspnet_core = Information
dtes_serilog_entity_framework_core = Error
dtes_serilog_system = Error
dtes_serilog_elastic_apm = Error
dtes_serilog_application = TextExtractorService
dtes_serilog_name = RollingFile
dtes_serilog_formatter = TextExtractorService.Logger.SmartEcsTextFormatter,
TextExtractorService
dtes_serilog_path_format = {Путь до каталога для логов Text Extractor Service
внутри контейнера}
dtes_background_remover = {Алгоритм удаления фона}
dtes_extraction_quality = {Качество извлечения текстового слоя}
dtes_ocr_text_from_pdf = {Всегда извлекать текстовый слой из PDF-документов, даже
если текстовый слой уже существует}
dtes_ocr_dpi = {Разрешение, используемое для извлечения текста из PDF-документов
без текстового слоя или из документов, у которых игнорируется существующий
текстовый слой}
dtes_convert_fact_positions = {Учитывать поворот документов при получении
координат фактов}
dtes_page_classifier_words_count = {Количество слов на странице, которые
используются для определения начала нового документа}
dtes_use_fixed_forms = {При обработке изображений применять жесткие формы
документов}
dtes_detect_orientation = {Определять ориентацию страниц}
dtes_find_objects = {Искать подписи}

```

```

dtes_find_stamps = {Искать печати}
dtes_prefetch_count = {Количество одновременно обрабатываемых запросов на
сервисе}
dtes_parallel_processing_page_count = {Количество одновременно обрабатываемых
страниц документа}
dtes_max_pages = {Максимальное число страниц, при котором из документа будет
извлекаться текстовый слой}
dtes_debug_images_enabled = {Режим отладки для сохранения промежуточных
результатов обработки изображений}
dtes_debug_images_path = {Каталог для сохранения промежуточных результатов
обработки изображений внутри контейнера}
dtes_debug_images_path_host = {Каталог для сохранения промежуточных результатов
обработки изображений}
dtes_elastic_apm_server_urls = ${elastic_apm_server_urls}
dtes_elastic_apm_transaction_sample_rate = ${elastic_apm_transaction_sample_rate}
dtes_elastic_apm_span_frames_min_duration =
${elastic_apm_span_frames_min_duration}
dtes_elastic_apm_metrics_interval = ${elastic_apm_metric_interval}
dtes_elastic_apm_capture_body = ${elastic_apm_capture_body}
dtes_elastic_apm_capture_body_content_types = "*"
dtes_queues_message_ttl = ${queues_message_ttl}
dtes_queue_connection = ${rabbit_url}
dtes_queue_retry_connection_count = {Количество повторных попыток подключения к
RabbitMQ}

```

[FactExtractorService](#)

```

dfes_replicas = {Количество запускаемых копий сервиса Fact Extractor Service}
dfes_external_port = {TCP-порт сервиса Fact Extractor Service}
dfes_debug = {Записывать в лог-файл отладочные сообщения}
dfes_area = {Используемый раздел конфигурационного файла}
dfes_queue_connection = ${rabbit_url}
dfes_message_ttl_seconds = ${queues_message_ttl}
dfes_prefetch_count = {Количество одновременно обрабатываемых сообщений}
dfes_server_url = ${elastic_apm_server_urls}
dfes_metrics_interval = ${elastic_apm_metric_interval}
dfes_span_frames_min_duration = ${elastic_apm_span_frames_min_duration}
dfes_suffix = {Формат названия лог-файлов}

```

[FactExtractorRulesService](#)

```

dfesr_replicas = {Количество запускаемых копий сервиса Fact Extractor Rules
Service}
dfesr_external_port = {TCP-порт сервиса Fact Extractor Rules Service}
dfesr_debug = {Записывать в лог-файл отладочные сообщения}
dfesr_area = {Используемый раздел конфигурационного файла}
dfesr_parser_file_name = {Имя файла синтаксического анализатора для извлечения
фактов}
dfesr_parser_debug = {Записывать в лог-файл отладочные сообщения в процессе
извлечения фактов и расчета вероятностей}
dfesr_parsers_count = {Количество экземпляров синтаксического анализатора в пуле}
dfesr_parser_timeout = {Таймаут в секундах, после которого обработка запроса
прерывается}
dfesr_queue_connection = ${rabbit_url}/?heartbeat={Значение}
dfesr_message_ttl_seconds = ${queues_message_ttl}
dfesr_prefetch_count = {Количество одновременно обрабатываемых сообщений}
dfesr_server_url = ${elastic_apm_server_urls}
dfesr_metrics_interval = ${elastic_apm_metric_interval}

```

```
dfesr_span_frames_min_duration = ${elastic_apm_span_frames_min_duration}
dfesr_suffix = {Формат названия лог-файлов}
```

[FactExtractorLearnService](#)

```
dfesl_replicas = {Количество запускаемых копий сервиса Fact Extractor Learn Service}
dfesl_max_replicas_per_node = {Ограничение количества запускаемых копий сервиса Fact Extractor Learn Service на одном сервере}
dfesl_external_port = {TCP-порт сервиса Fact Extractor Learn Service}
dfesl_debug = {Записывать в лог-файл отладочные сообщения}
dfesl_worker_cache_lifetime = {Время хранения результата обучения в кэше сервиса в секундах}
dfesl_can_train_models = {Разрешить использовать сервис как сервис обучения}
dfesl_dbname = ${pg_database}
dfesl_dbuser = ${pg_user}
dfesl_dbhost = ${pg_host}
dfesl_dbpass = ${pg_password}
dfesl_dbport = ${pg_port}
dfesl_queue_connection = ${rabbit_url}///?heartbeat={Значение}
dfesl_message_ttl_seconds = ${queues_message_ttl}
dfesl_prefetch_count = {Количество одновременно обрабатываемых сообщений}
dfesl_model_cache_lifetime = {Время хранения модели в пуле}
dfesl_subprocesses = {Количество подпроцессов}
dfesl_algorithm = {Алгоритм обучения}
dfesl_c1 = {Коэффициент L1-регуляризации}
dfesl_c2 = {Коэффициент L2-регуляризации}
dfesl_epsilon = {Точность вычисления показателя качества модели}
dfesl_delta = {Относительный показатель качества модели}
dfesl_slice_period = {Количество итераций обучения, через которое оценивается относительный показатель качества модели}
dfesl_max_iterations = {Максимальное число итераций обучения}
dfesl_all_possible_transitions = {Генерировать все сочетания классов}
dfesl_logging_period = {Интервал логирования процесса обучения в итерациях}
dfesl_test_size = {Размер выборки для расчета метрик}
dfesl_server_url = ${elastic_apm_server_urls}
dfesl_metrics_interval = ${elastic_apm_metric_interval}
dfesl_span_frames_min_duration = ${elastic_apm_span_frames_min_duration}
dfesl_suffix = {Формат названия лог-файлов}
```

[ObjectDetectionService](#)

```
dods_replicas = {Количество запускаемых копий сервиса Object Detection Service}
dods_external_port = {TCP-порт сервиса Fact Extractor Learn Service}
dods_debug = {Записывать в лог-файл отладочные сообщения}
dods_queue_connection = ${rabbit_url}
dods_message_ttl_seconds = ${queues_message_ttl}
dods_prefetch_count = {Количество одновременно обрабатываемых сообщений}
dods_server_url = ${elastic_apm_server_urls}
dods_metrics_interval = ${elastic_apm_metric_interval}
dods_span_frames_min_duration = ${elastic_apm_span_frames_min_duration}
dods_suffix = {Формат названия лог-файлов}
```

[ImageClassifierService](#)

```
dics_replicas = {Количество запускаемых копий сервиса Image Classifier Service}
dics_external_port = {TCP-порт сервиса Image Classifier Service}
dics_debug = {Записывать в лог-файл отладочные сообщения}
```

```

dics_area = {Используемый раздел конфигурационного файла}
dics_queue_connection = ${rabbit_url}
dics_message_ttl_seconds = ${queues_message_ttl}
dics_prefetch_count = {Количество одновременно обрабатываемых сообщений}
dics_multithreaded_predict = {Использовать многопоточность}
dics_server_url = ${elastic_apm_server_urls}
dics_metrics_interval = ${elastic_apm_metric_interval}
dics_span_frames_min_duration = ${elastic_apm_span_frames_min_duration}
dics_suffix = {Формат названия лог-файлов}

# TextClassifierService
dtcs_replicas = {Количество запускаемых копий сервиса Text Classifier Service}
dtcs_external_port = {TCP-порт сервиса Text Classifier Service}
dtcs_debug = {Записывать в лог-файл отладочные сообщения}
dtcs_area = {Используемый раздел конфигурационного файла}
dtcs_max_models_in_pool = {Максимальное количество моделей классификации в пуле}
dtcs_model_cache_lifetime = {Время хранения модели классификации в пуле}
dtcs_dbname = ${pg_database}
dtcs_dbuser = ${pg_user}
dtcs_dbhost = ${pg_host}
dtcs_dbpass = ${pg_password}
dtcs_dbport = ${pg_port}
dtcs_csv_text_field = {Название поля с текстом для обучения данными в формате CSV}
dtcs_csv_class_field = {Название поля с классами для обучения данными в формате CSV многоклассового классификатора}
dtcs_csv_classes_field = {Название поля с классами для обучения данными в формате CSV многозначного классификатора}
dtcs_csv_external_id_field = {Название поля с внешним идентификатором класса для обучения данными в формате CSV многоклассового классификатора}
dtcs_repository = {Тип репозитория для хранения моделей классификации}
dtcs_can_train_models = {Разрешить использовать сервис как сервис обучения классификаторов}
dtcs_queue_connection = ${rabbit_url}
dtcs_message_ttl_seconds = ${queues_message_ttl}
dtcs_prefetch_count = {Количество одновременно обрабатываемых сообщений}
dtcs_max_text_extraction_requests = {Количество документов для одновременного извлечения текста при обучении классификатора}
dtcs_allow_local_path = {Разрешить обучение из локальной папки}
dtcs_server_url = ${elastic_apm_server_urls}
dtcs_metrics_interval = ${elastic_apm_metric_interval}
dtcs_span_frames_min_duration = ${elastic_apm_span_frames_min_duration}
dtcs_suffix = {Формат названия лог-файлов}

```

Smart Service

Сервис Smart Service настраивается в одноименной секции конфигурационного файла .env.

Структура секции:

```

# SmartService
dss_replicas = {Количество запускаемых копий сервиса Smart Service}
dss_address = {Адрес сервиса Smart Service}
dss_external_port = {TCP-порт сервиса Smart Service}
dss_connection_strings_default_connection = ${pg_connection}

```

```

dss_serilog_default = ${log_level}
dss_serilog_entity_framework_core = Error
dss_serilog_system = Error
dss_serilog_elastic_apm = Error
dss_serilog_application = SmartService
dss_serilog_name = RollingFile
dss_serilog_formatter = SmartService.Logger.SmartEcsTextFormatter, SmartService
dss_serilog_path_format = {Путь до каталога для логов Smart Service внутри
контейнера}
dss_identity_service = {Адрес сервиса Identity Service}
dss_allow_local_path = {Разрешить обучение классификаторов из локального каталога}
dss_allow_anonymous_access = {Разрешить анонимную работу}
dss_anonymous_client_id = {Логин анонимного клиента}
dss_anonymous_client_secret = {Пароль анонимного клиента}
dss_storage_endpoint = {Адрес сервиса хранения объектов}
dss_storage_access_key = ${storage_service_access_key}
dss_storage_secret_key = ${storage_service_secret_key}
dss_storage_bucket = {Имя контейнера}
dss_storage_uri_lifetime_seconds = {Время хранения ссылок на объекты в секундах}
dss_storage_file_lifetime_hours = {Время хранения объектов в часах}
dss_storage_retry_connection_count = {Количество повторных попыток подключения к
Storage Service}
dss_elastic_apm_server_urls = ${elastic_apm_server_urls}
dss_elastic_apm_transaction_sample_rate = ${elastic_apm_transaction_sample_rate}
dss_elastic_apm_span_frames_min_duration = ${elastic_apm_span_frames_min_duration}
dss_elastic_apm_metrics_interval = ${elastic_apm_metric_interval}
dss_elastic_apm_capture_body = ${elastic_apm_capture_body}
dss_elastic_apm_capture_body_content_types = "*"
dss_queues_message_ttl = ${queues_message_ttl}
dss_queue_connection = ${rabbit_url}
dss_queue_retry_connection_count = {Количество повторных попыток подключения к
RabbitMQ}

```

- **dss_replicas**. Количество запускаемых копий сервиса Smart Service. Возможные значения: **0** – сервис DSS не будет запущен при следующем запуске сервисов, **1** – сервис DSS будет запущен при следующем запуске сервисов. Значение по умолчанию **1**.
- **dss_address**. Адрес контейнера с сервисом Smart Service. Значение по умолчанию **http://dss:80**.
- **dss_external_port**. TCP-порт сервиса Smart Service. Значение по умолчанию **61100**.
- **dss_connection_strings_default_connection**. Строка подключения к базе данных PostgreSQL. Подставляется значение, указанное для параметра **pg_connection** в конфигурационном файле .env в секции PostgreSQL.
- **dss_serilog_default**. Параметры ведения лог-файлов.
- **dss_serilog_path_format**. Путь до каталога с лог-файлами Smart Service.
- **dss_identity_service**. Адрес сервиса аутентификации и авторизации пользователей Identity Service. Пример значения **http://localhost:9030**.
- **dss_allow_local_path**. Разрешить обучение классификаторов из локального каталога сервера. Возможные значения: **True**, **False**. Если установлено значение **False**, повышается безопасность работы. Значение по умолчанию **True**.

- **dss_allow_anonymous_access.** Разрешить анонимную работу системы. Возможные значения: **True, False**. Значение **True** устанавливается при локальной поставке сервисов Smart Services и позволяет не использовать аутентификацию. При облачной поставке устанавливается значение **False**, после чего требуется обязательная аутентификация и передача токена в каждом запросе.
- **dss_anonymous_client_id.** Логин анонимного клиента, который взаимодействует с сервисами Smart Services при разрешенной анонимной работе. Значение по умолчанию **anonymous**.
- **dss_anonymous_client_secret.** Пароль анонимного клиента, который взаимодействует с сервисами Smart Services при разрешенной анонимной работе. Значение по умолчанию **AnonymousSecret**.
- **dss_storage_endpoint.** Веб-адрес сервиса хранения объектов. Указывается без протокола http/https. Пример значения: **localhost:9000**.
- **dss_storage_access_key.** Имя пользователя для подключения к сервису хранения объектов. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Storage Service.
- **dss_storage_secret_key.** Пароль для подключения к сервису хранения объектов. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Storage Service.
- **dss_storage_bucket.** Имя каталога для хранения объектов. Значение по умолчанию **smart**.
- **dss_storage_uri_lifetime_seconds.** Время хранения ссылок на объекты в секундах. Значение по умолчанию **3600**.
- **dss_storage_file_lifetime_hours.** Время хранения объектов в хранилище в часах. Значение по умолчанию **72**.
- **dss_storage_retry_connection_count.** Количество повторных попыток подключения к сервису хранения объектов. Значение по умолчанию **5**.
- **dss_elastic_apm_server_urls.** Адрес сервиса Elastic APM Service. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dss_elastic_apm_transaction_sample_rate.** Регулярность отправки трейсов в Elastic APM Service. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dss_elastic_apm_span_frames_min_duration.** Если при обработке документа возникла ошибка и время обработки превышает значение параметра, в лог-файл записывается не только факт ошибки, но и подробная информация о ней. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dss_elastic_apm_metrics_interval.** Интервал сбора метрик о состоянии памяти и CPU. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dss_elastic_apm_capture_body.** Сообщения, которые передаются в сервис Elastic APM Service. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dss_queue_message_ttl.** Время хранения сообщений об обработке документа в RabbitMQ в секундах. Подставляется значение, указанное для данного параметра в конфигурационном файле .env.

- **dss_queue_connection.** Строка подключения к очереди сообщений RabbitMQ. Подставляется значение, указанное для параметра **rabbit_url** в конфигурационном файле .env в секции RabbitMQ.
- **dss_queue_retry_connection_count.** Количество повторных попыток подключения DSS к брокеру сообщений. Значение по умолчанию **5**.

Identity Service

Сервис Identity Service настраивается в одноименной секции конфигурационного файла .env.

Структура секции:

```
# IdentityService
dis_replicas = {Количество запускаемых копий сервиса Identity Service}
dis_external_port = {TCP-порт сервиса Identity Service}
dis_serilog_default = ${log_level}
dis_serilog_entity_framework_core = Error
dis_serilog_system = Error
dis_serilog_elastic_apm = Error
dis_serilog_application = IdentityService
dis_serilog_name = RollingFile
dis_serilog_formatter = IdentityService.Logger.IdentityEcsTextFormatter,
IdentityService
dis_serilog_path_format = {Путь до каталога для логов Identity Service внутри
контейнера}
dis_connection_strings_default_connection = User
ID=${pg_user};Password=${pg_password};Host=${pg_host};Port=${pg_port};Database=${p
g_database_dis};Pooling={Использовать пул подключений к БД};
dis_elastic_apm_server_urls = ${elastic_apm_server_urls}
dis_elastic_apm_service_name = ${dis_log_properties_service_name}
dis_elastic_apm_log_level = ${elastic_apm_log_level}
dis_elastic_apm_transaction_sample_rate = ${elastic_apm_transaction_sample_rate}
dis_elastic_apm_span_frames_min_duration = ${elastic_apm_span_frames_min_duration}
dis_elastic_apm_metrics_interval = ${elastic_apm_metric_interval}
dis_elastic_apm_capture_body = ${elastic_apm_capture_body}
dis_certificate_name = {Наименование файла сертификата}
dis_certificate_password = {Пароль сертификата}
```

- **dis_replicas.** Количество запускаемых копий сервиса Identity Service. Возможные значения: **0** – сервис DIS не будет запущен при следующем запуске сервисов, **1** – сервис DIS будет запущен при следующем запуске сервисов. Значение по умолчанию **1**.
- **dis_external_port.** TCP-порт сервиса Identity Service. Значение по умолчанию **9030**.
- **dis_serilog_default.** Параметры ведения лог-файлов.
- **dis_serilog_path_format.** Путь до каталога с лог-файлами Identity Service.
- **dis_connection_strings_default_connection** – параметры подключения к базе данных PostgreSQL, где:
 - **UserID** и **Password** – имя и пароль пользователя для подключения к СУБД PostgreSQL. Хранятся в зашифрованном виде;
 - **Host** – адрес СУБД PostgreSQL;
 - **Port** – порт для подключения к СУБД PostgreSQL;

- **Database** – имя базы данных сервиса Identity Service;
- **Pooling** – признак использования пула подключений к базе данных. Возможные значения: **True**, **False**. Если установлено значение **True**, то для новых запросов к базе данных используется ранее созданное подключение из пула. Если установлено значение **False**, то каждый раз создается новое подключение. Для ускорения работы сервиса рекомендуется оставить в параметре значение по умолчанию **True**.

ПРИМЕЧАНИЕ. Identity Service имеет собственную базу данных, отличную от базы данных Smart Service.

- **dis_elastic_apm_server_urls**. Адрес сервиса Elastic APM Service. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dis_elastic_apm_service_name**. Имя, под которым сервис Identity Service отображается в лог-файлах Elastic APM Service. Значение по умолчанию **Identity Service**.
- **dis_elastic_apm_log_level**. Уровень логирования библиотеки, которая отправляет в сервис Elastic APM Service информацию о запросах и метриках сервиса Identity Service. Значение по умолчанию **Error**.
- **dis_elastic_apm_transaction_sample_rate**. Регулярность отправки трейсов в Elastic APM Service. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dis_elastic_apm_span_frames_min_duration**. Если при обработке документа возникла ошибка и время обработки превышает значение параметра, в лог-файл записывается не только факт ошибки, но и подробная информация о ней. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dis_elastic_apm_metrics_interval**. Интервал сбора метрик о состоянии памяти и CPU. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dis_elastic_apm_capture_body**. Сообщения, которые передаются в сервис Elastic APM Service. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dis_certificate_name**. Наименование файла сертификата для выдачи токена пользователю.
- **dis_certificate_password**. Пароль пользователя сертификата.

Publishing Service

Сервис Publishing Service настраивается в одноименной секции конфигурационного файла .env.

Структура секции:

```
# PublishingService
dps_replicas = {Количество запускаемых копий сервиса Publishing Service}
dps_external_port = {TCP-порт сервиса Publishing Service}
dps_serilog_default = ${log_level}
dps_serilog_entity_framework_core = Error
dps_serilog_system = Error
```

```

dps_serilog_elastic_apm = Error
dps_serilog_application = PublishingService
dps_serilog_name = RollingFile
dps_serilog_formatter = PublishingService.Logger.LoggerEcsTextFormatter,
PublishingService
dps_serilog_path_format = {Путь до каталога для логов Publishing Service внутри
контейнера}
dps_elastic_apm_server_urls = ${elastic_apm_server_urls}
dps_elastic_apm_transaction_sample_rate = ${elastic_apm_transaction_sample_rate}
dps_elastic_apm_span_frames_min_duration = ${elastic_apm_span_frames_min_duration}
dps_elastic_apm_metrics_interval = ${elastic_apm_metric_interval}
dps_elastic_apm_capture_body = ${elastic_apm_capture_body}
dps_elastic_apm_capture_body_content_types = "*"
dps_queue_connection = ${rabbit_url}
dps_queues_message_ttl = ${queues_message_ttl}
dps_queues_services = {Список сервисов}
dps_queue_retry_connection_count = {Количество повторных попыток подключения к
RabbitMQ}

```

- **dps_replicas.** Количество запускаемых копий сервиса Publishing Service. Возможные значения: **0** – сервис DPS не будет запущен при следующем запуске сервисов, **1** – сервис DPS будет запущен при следующем запуске сервисов. Значение по умолчанию **1**.
- **dps_external_port.** TCP-порт сервиса Publishing Service. Значение по умолчанию **9057**.
- **dps_serilog_default.** Параметры ведения лог-файлов.
- **dps_serilog_path_format.** Путь до каталога с лог-файлами Publishing Service.
- **dps_elastic_apm_server_urls.** Адрес сервиса Elastic APM Service. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dps_elastic_apm_transaction_sample_rate.** Регулярность отправки трейсов в Elastic APM Service. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dps_elastic_apm_span_frames_min_duration.** Если при обработке документа возникла ошибка и время обработки превышает значение параметра, в лог-файл записывается не только факт ошибки, но и подробная информация о ней. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dps_elastic_apm_metrics_interval.** Интервал сбора метрик о состоянии памяти и CPU. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dps_elastic_apm_capture_body.** Сообщения, которые передаются в сервис Elastic APM Service. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dps_queue_connection.** Строка подключения к очереди сообщений RabbitMQ. Подставляется значение, указанное для параметра **rabbit_url** в конфигурационном файле .env в секции RabbitMQ.
- **dps_queue_message_ttl.** Время хранения сообщений об обработке документа в RabbitMQ в секундах. Подставляется значение, указанное для данного параметра в конфигурационном файле .env.

- **dps_queue_services.** Список сервисов, с которыми работает сервис Publishing Service. Если сервис не указан, то он и его дочерние сервисы не обновляются. Значение по умолчанию **Fact Extractor Service, Fact Extractor Rules Service, Fact Extractor Learn Service.**
- **dps_queue_retry_connection_count.** Количество повторных попыток подключения DSS к брокеру сообщений. Значение по умолчанию **5.**

Text Extractor Service

Сервис Text Extractor Service настраивается в одноименной секции конфигурационного файла .env.

Структура секции:

```
# TextExtractor
dtes_replicas = {Количество запускаемых копий сервиса Text Extractor Service}
dtes_external_port = {TCP-порт сервиса Text Extractor Service}
dtes_serilog_default = ${log_level}
dtes_serilog_microsoft = Warning
dtes_serilog_aspnet_core = Information
dtes_serilog_entity_framework_core = Error
dtes_serilog_system = Error
dtes_serilog_elastic_apm = Error
dtes_serilog_application = TextExtractorService
dtes_serilog_name = RollingFile
dtes_serilog_formatter = TextExtractorService.Logger.SmartEcsTextFormatter,
TextExtractorService
dtes_serilog_path_format = {Путь до каталога для логов Text Extractor Service
внутри контейнера}
dtes_background_remover = {Алгоритм удаления фона}
dtes_extraction_quality = {Качество извлечения текстового слоя}
dtes_ocr_text_from_pdf = {Всегда извлекать текстовый слой из PDF-документов, даже
если текстовый слой уже существует}
dtes_ocr_dpi = {Разрешение, используемое для извлечения текста из PDF-документов
без текстового слоя или из документов, у которых игнорируется существующий
текстовый слой}
dtes_convert_fact_positions = {Учитывать поворот документов при получении
координат фактов}
dtes_page_classifier_words_count = {Количество слов на странице, которые
используются для определения начала нового документа}
dtes_use_fixed_forms = {При обработке изображений применять жесткие формы
документов}
dtes_detect_orientation = {Определять ориентацию страниц}
dtes_find_objects = {Искать подписи}
dtes_find_stamps = {Искать печати}
dtes_prefetch_count = {Количество одновременно обрабатываемых запросов на
сервисе}
dtes_parallel_processing_page_count = {Количество одновременно обрабатываемых
страниц документа}
dtes_max_pages = {Максимальное число страниц, при котором из документа будет
извлекаться текстовый слой}
dtes_debug_images_enabled = {Режим отладки для сохранения промежуточных
результатов обработки изображений}
```

```

dtes_debug_images_path = {Каталог для сохранения промежуточных результатов
обработки изображений внутри контейнера}
dtes_debug_images_path_host = {Каталог для сохранения промежуточных результатов
обработки изображений}
dtes_elastic_apm_server_urls = ${elastic_apm_server_urls}
dtes_elastic_apm_transaction_sample_rate = ${elastic_apm_transaction_sample_rate}
dtes_elastic_apm_span_frames_min_duration =
${elastic_apm_span_frames_min_duration}
dtes_elastic_apm_metrics_interval = ${elastic_apm_metric_interval}
dtes_elastic_apm_capture_body = ${elastic_apm_capture_body}
dtes_elastic_apm_capture_body_content_types = "*"
dtes_queues_message_ttl = ${queues_message_ttl}
dtes_queue_connection = ${rabbit_url}
dtes_queue_retry_connection_count = "{Количество повторных попыток подключения к
RabbitMQ}"

```

- **dtes_replicas.** Количество запускаемых копий сервиса Text Extractor Service. Возможные значения: **0** – сервис DTES не будет запущен при следующем запуске сервисов, **1** – сервис DTES будет запущен при следующем запуске сервисов. Значение по умолчанию **1**.
- **dtes_external_port.** TCP-порт сервиса Text Extractor Service. Значение по умолчанию **9027**.
- **dss_serilog_default.** Параметры ведения лог-файлов.
- **dss_serilog_path_format.** Путь до каталога с лог-файлами Text Extractor Service.
- **dtes_background_remover.** Алгоритм, который используется для удаления фона в документе. Возможные значения:
 - **ConnectedComponents.** Основан на методе поиска связанных компонент установленного размера и последующем удалении компонент меньше этого значения. Быстрее и точнее обрабатывает простые и сложные фоны, но иногда может задевать текстовый слой: стирать границы таблиц и удалять знаки препинания.
 - **Sobel.** Основан на методе поиска границ изображения с помощью оператора Собеля. Обрабатывает простые фоны.
 - **None.** Фон не удаляется.

Значение по умолчанию **ConnectedComponents**.

- **dtes_extraction_quality.** Качество извлечения текстового слоя. Возможные значения:
 - **fast.** Высокая скорость извлечения, при этом снижается качество.
 - **quality.** Высокое качество извлечения, при этом снижается скорость.

Значение по умолчанию **fast**.

- **dtes_ocr_text_from_pdf.** Всегда извлекать текстовый слой из PDF-документов, даже если текстовый слой уже существует. Возможные значения: **True**, **False**. Значение по умолчанию **True**.
- **dtes_ocr_dpi.** Разрешение, используемое для извлечения текста из PDF-документов без текстового слоя или из документов, у которых игнорируется существующий текстовый слой. Укажите целое число больше нуля. Значение по умолчанию **300**.
- **dtes_convert_fact_positions.** Учитывать поворот документов при получении координат фактов. Возможные значения: **True**, **False**. Укажите значение **False**, если импортируются PDF-документы без предварительной обработки. Если импортируются обработанные PDF-документы, оставьте значение по умолчанию **True**.

- **dtes_page_classifier_words_count.** Количество слов на странице, которые используются для определения начала нового документа. Укажите целое число больше нуля. Значение по умолчанию **50**.
- **dtes_use_fixed_forms.** При обработке изображений применять жесткие формы документов, например формы паспортов РФ. Возможные значения: **True, False**. Если нет необходимости в распознавании документов с жесткими формами, для ускорения обработки документов рекомендуется оставить значение по умолчанию **False**.
- **dtes_detect_orientation.** Определять ориентацию страниц. Возможные значения: **True, False**. Значение по умолчанию **True**. Если нет необходимости в определении ориентации страниц, для ускорения обработки документов установите значение **False**.
- **dtes_find_objects.** Искать подписи. Возможные значения: **True, False**. Значение по умолчанию **True**. Если нет необходимости в поиске, для ускорения обработки документов установите значение **False**.
- **dtes_find_stamps.** Искать печати. Возможные значения: **True, False**. Значение по умолчанию **True**. Если нет необходимости в поиске, для ускорения обработки документов установите значение **False**.
- **dtes_prefetch_count.** Количество одновременно обрабатываемых запросов на сервисе. Значение по умолчанию **0**. Если установлено такое значение, то оптимальное количество одновременно обрабатываемых документов подбирается автоматически. Оно зависит от доступного ресурса системы.

Доступный ресурс представляет собой минимальное значение, выбранное между числом логических ядер процессора (за вычетом одного ядра на работу ОС) и значением доступной оперативной памяти, разделенной на 3 Гб – объем памяти для обработки одного документа. Формула для расчета: **Min(CPU-1, RAM/3)**.

Значение самого параметра вычисляется как половина от доступного ресурса с округлением вниз. Формула для расчета: **Floor(Min(CPU-1, RAM/3)/2)**.

- **dtes_parallel_processing_page_count.** Количество одновременно обрабатываемых страниц документа. Значение по умолчанию **0**. Если установлено такое значение, то оптимальное количество одновременно обрабатываемых страниц подбирается автоматически. Оно зависит от доступного ресурса системы, который рассчитывается по формуле, аналогичной расчету ресурса для параметра **dtes_prefetch_count**. При этом, если значение доступного ресурса:
 - больше 1, то количество одновременно обрабатываемых страниц равно 2;
 - меньше или равно 1, то количество одновременно обрабатываемых страниц равно 1.
- **dtes_max_pages.** Максимальное число страниц, при котором из документа извлекается текстовый слой. Если количество страниц превышает установленный лимит, то текстовый слой не извлекается.

Если из документа уже был извлечен текстовый слой, то при конвертации в формат PDF он сохранится. Исключение составляют многостраничные файлы в формате TIFF, при конвертации которых в PDF текстовый слой не сохраняется. Также такие файлы не разделяются на несколько документов.

Значение по умолчанию **100**. Если указать в параметре **0** или отрицательное число, то лимит будет отключен.

- **dtes_debug_images_enabled.** Включить режим отладки для сохранения промежуточных результатов обработки изображений. Возможные значения: **True, False**. Если указано значение **True**, то во временный каталог, указанный в параметре **dtes_debug_images_path**, сохраняются промежуточные результаты обработки изображений на каждом из этапов. Значение по умолчанию **False**.
- **dtes_debug_images_path.** Адрес временного каталога для сохранения промежуточных результатов обработки изображений. Значение по умолчанию **tmp/debug**. Не рекомендуется менять значение параметра.
- **dtes_debug_images_path_host.** Каталог для сохранения промежуточных результатов обработки изображений на сервере. Значение по умолчанию **./etc/textextractor-service/debug**. Не рекомендуется менять значение параметра.
- **dtes_elastic_apm_server_urls.** Адрес сервиса Elastic APM Service. Подставляется значение, указанное для данного параметра в конфигурационном файле **.env** в секции Elastic APM Service.
- **dtes_elastic_apm_transaction_sample_rate.** Регулярность отправки трейсов в Elastic APM Service. Подставляется значение, указанное для данного параметра в конфигурационном файле **.env** в секции Elastic APM Service.
- **dtes_elastic_apm_span_frames_min_duration.** Если при обработке документа возникла ошибка и время обработки превышает значение параметра, в лог-файл записывается не только факт ошибки, но и подробная информация о ней. Подставляется значение, указанное для данного параметра в конфигурационном файле **.env** в секции Elastic APM Service.
- **dtes_elastic_apm_metrics_interval.** Интервал сбора метрик о состоянии памяти и CPU. Подставляется значение, указанное для данного параметра в конфигурационном файле **.env** в секции Elastic APM Service.
- **dtes_elastic_apm_capture_body.** Сообщения, которые передаются в сервис Elastic APM Service. Подставляется значение, указанное для данного параметра в конфигурационном файле **.env** в секции Elastic APM Service.
- **dtes_queue_message_ttl.** Время хранения сообщений об обработке документа в RabbitMQ в секундах. Подставляется значение, указанное для [данного параметра](#) в конфигурационном файле **.env**.
- **dtes_queue_connection.** Строка подключения к очереди сообщений RabbitMQ. Подставляется значение, указанное для параметра **rabbit_url** в конфигурационном файле **.env** в секции RabbitMQ.
- **dtes_retry_connection_count.** Количество повторных попыток подключения к брокеру сообщений. Значение по умолчанию **5**.

Text Classifier Service

Сервис Text Classifier Service настраивается в одноименной секции конфигурационного файла **.env**.

Структура секции:

```
# TextClassifierService
dtcs_replicas = {Количество запускаемых копий сервиса Text Classifier Service}
dtcs_external_port = {TCP-порт сервиса Text Classifier Service}
```

```

dtcs_debug = {Записывать в лог-файл отладочные сообщения}
dtcs_area = {Используемый раздел конфигурационного файла}
dtcs_max_models_in_pool = {Максимальное количество моделей классификации в пуле}
dtcs_model_cache_lifetime = {Время хранения модели классификации в пуле}
dtcs_dbname = ${pg_database}
dtcs_dbuser = ${pg_user}
dtcs_dbhost = ${pg_host}
dtcs_dbpass = ${pg_password}
dtcs_dbport = ${pg_port}
dtcs_csv_text_field = {Название поля с текстом для обучения данными в формате CSV}
dtcs_csv_class_field = {Название поля с классами для обучения данными в формате CSV многоклассового классификатора}
dtcs_csv_classes_field = {Название поля с классами для обучения данными в формате CSV многозначного классификатора}
dtcs_csv_external_id_field = {Название поля с внешним идентификатором класса для обучения данными в формате CSV многоклассового классификатора}
dtcs_repository = {Тип репозитория для хранения моделей классификации}
dtcs_can_train_models = {Разрешить использовать сервис как сервис обучения классификаторов}
dtcs_queue_connection = ${rabbit_url}
dtcs_message_ttl_seconds = ${queues_message_ttl}
dtcs_prefetch_count = {Количество одновременно обрабатываемых сообщений}
dtcs_max_text_extraction_requests = {Количество документов для одновременного извлечения текста при обучении классификатора}
dtcs_allow_local_path = {Разрешить обучение из локальной папки}
dtcs_server_url = ${elastic_apm_server_urls}
dtcs_metrics_interval = ${elastic_apm_metric_interval}
dtcs_span_frames_min_duration = ${elastic_apm_span_frames_min_duration}
dtcs_suffix = {Формат названия лог-файлов}

```

- **dtcs_replicas.** Количество запускаемых копий сервиса Text Classifier Service. Возможные значения: **0** – сервис DTCS не будет запущен при следующем запуске сервисов, **1** – сервис DTCS будет запущен при следующем запуске сервисов. Значение по умолчанию **1**.
- **dtcs_external_port.** TCP-порт сервиса Text Classifier Service. Значение по умолчанию **9007**.
- **dtcs_debug.** Записывать в лог-файл отладочные сообщения. Возможные значения: **True**, **False**. Значение по умолчанию **True**.
- **dtcs_area.** Используемый в данный момент раздел конфигурационного файла. Используется для быстрого переключения между настройками окружения. Пример значения: **develop**.
- **dtcs_max_models_in_pool.** Максимальное количество моделей классификации, хранящихся в невыгружаемом пуле. Если модель классификации находится в пуле, обращение к ней выполняется быстрее, таким образом ускоряется классификация текстов. Укажите целое число больше нуля. Значение по умолчанию **15**.
- **dtcs_model_cache_lifetime.** Время хранения модели классификации в пуле в минутах. Модели, к которым не обращались в течение этого промежутка времени, выгружаются из пула. Укажите целое число больше нуля. Значение по умолчанию **60**.
- **dtcs_dbname.** Имя базы данных PostgreSQL. Подставляется значение, указанное для параметра **pg_database** в конфигурационном файле .env в секции PostgreSQL.
- **dtcs_dbuser.** Имя пользователя для подключения к СУБД PostgreSQL. Подставляется значение, указанное для параметра **pg_user** в конфигурационном файле .env в секции PostgreSQL.

- **dtcs_dbhost**. Адрес СУБД PostgreSQL. Подставляется значение, указанное для параметра **pg_host** в конфигурационном файле .env в секции PostgreSQL.
- **dtcs_dbpass**. Пароль для подключения к СУБД PostgreSQL. Подставляется значение, указанное для параметра **pg_password** в конфигурационном файле .env в секции PostgreSQL.
- **dtcs_dbport**. Порт для подключения к СУБД PostgreSQL. Подставляется значение, указанное для параметра **pg_port** в конфигурационном файле .env в секции PostgreSQL.
ПРИМЕЧАНИЕ. Все параметры подключения к базе данных должны совпадать с параметрами, указанными в [конфигурационном файле](#) Smart Service.
- **dtcs_csv_text_field**, **dtcs_csv_class_field**, **dtcs_csv_classes_field**, **dtcs_csv_external_id_field**. Названия полей в [CSV-файле](#) для обучения классификаторов через набор данных в формате CSV. Параметры являются внутренними, изменять не рекомендуется.
- **dtcs_repository**. Тип репозитория, используемого для хранения моделей классификации. Параметр является внутренним, изменять не рекомендуется.
- **dtcs_can_train_models**. Разрешить использовать этот сервис в качестве сервиса обучения классификаторов. Возможные значения: **True**, **False**. Значение по умолчанию **True**.
- **dtcs_queue_connection**. Строка подключения к RabbitMQ. Подставляется значение, указанное для параметра **rabbit_url** в конфигурационном файле .env в секции RabbitMQ.
- **dtcs_message_ttl_seconds**. Время хранения сообщений об обработке документа в RabbitMQ в секундах. Подставляется значение, указанное для [данного параметра](#) в конфигурационном файле .env.
- **dtcs_prefetch_count**. Количество одновременно обрабатываемых запросов на сервисе. Значение по умолчанию **0**. Если установлено такое значение, то оптимальное количество одновременно обрабатываемых запросов подбирается автоматически. Оно зависит от доступного ресурса системы, который рассчитывается как количество логических ядер процессора минус одно ядро для работы ОС. Формула для расчета параметра: $(\text{CPU}-1)/2$ с округлением вверх. Например, при CPU=8 доступный ресурс системы будет равен 7 (одно ядро отводится на работу ОС), а итоговое количество параллельно обрабатываемых запросов будет равно 4.
- **dtcs_allow_local_path**. Разрешить обучение классификаторов из локального каталога сервера. Возможные значения: **True**, **False**. Если установлено значение **False**, повышается безопасность работы. Значение по умолчанию **True**.
- **dtcs_max_text_extraction_requests**. Количество документов без текстового слоя, из которых при обучении классификатора может одновременно извлекаться текст. Рекомендуется указывать значение выше, чем общее количество обработчиков Text Extractor Service, но не слишком большое, так как возрастает нагрузка на CPU. Значение по умолчанию **50**.
- **dtcs_server_url**. Адрес сервиса Elastic APM Service. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dtcs_metrics_interval**. Интервал сбора метрик о состоянии памяти и CPU. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dtcs_span_frames_min_duration**. Сообщения, которые передаются в сервис Elastic APM Service. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dtcs_suffix**. Формат названия лог-файлов. Значение по умолчанию **%Y%m%d**.

Image Classifier Service

Сервис Image Classifier Service настраивается в одноименной секции конфигурационного файла .env.

Структура секции:

```
# ImageClassifierService
dics_replicas = {Количество запускаемых копий сервиса Image Classifier Service}
dics_external_port = {TCP-порт сервиса Image Classifier Service}
dics_debug = {Записывать в лог-файл отладочные сообщения}
dics_area = {Используемый раздел конфигурационного файла}
dics_queue_connection = ${rabbit_url}
dics_message_ttl_seconds = ${queues_message_ttl}
dics_prefetch_count = {Количество одновременно обрабатываемых сообщений}
dics_multithreaded_predict = {Использовать многопоточность}
dics_server_url = ${elastic_apm_server_urls}
dics_metrics_interval = ${elastic_apm_metric_interval}
dics_span_frames_min_duration = ${elastic_apm_span_frames_min_duration}
dics_suffix = {Формат названия лог-файлов}
```

- **dics_replicas.** Количество запускаемых копий сервиса Image Classifier Service. Возможные значения: **0** – сервис DICS не будет запущен при следующем запуске сервисов, **1** – сервис DICS будет запущен при следующем запуске сервисов. Значение по умолчанию **1**.
- **dics_external_port.** TCP-порт сервиса Image Classifier Service. Значение по умолчанию **9008**.
- **dics_debug.** Записывать в лог-файл отладочные сообщения. Возможные значения: **True**, **False**. Значение по умолчанию **True**.
- **dics_area.** Используемый в данный момент раздел конфигурационного файла. Используется для быстрого переключения между настройками окружения. Пример значения: **develop**.
- **dics_queue_connection.** Строка подключения к RabbitMQ. Подставляется значение, указанное для параметра **rabbit_url** в конфигурационном файле .env в секции RabbitMQ.
- **dics_message_ttl_seconds.** Время хранения сообщений об обработке документа в RabbitMQ в секундах. Подставляется значение, указанное для [данного параметра](#) в конфигурационном файле .env.
- **dics_prefetch_count.** Количество одновременно обрабатываемых запросов на сервисе. Значение по умолчанию **0**. Если установлено такое значение, то оптимальное количество одновременно обрабатываемых запросов подбирается автоматически. Оно зависит от доступного ресурса системы, который рассчитывается как количество логических ядер процессора минус одно ядро для работы ОС. Формула для расчета параметра: **(CPU-1)/2** с округлением вверх. Например, при CPU=8 доступный ресурс системы будет равен 7 (одно ядро отводится на работу ОС), а итоговое количество параллельно обрабатываемых запросов будет равно 4.
- **dics_multithreaded_predict.** Использовать несколько потоков при классификации изображения. Если сервис установлен на отдельном компьютере, установите значение **True**. Значение по умолчанию **False**.

- **dics_server_url.** Адрес сервиса Elastic APM Service. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dics_metrics_interval.** Интервал сбора метрик о состоянии памяти и CPU. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dics_span_frames_min_duration.** Если при обработке документа возникла ошибка и время обработки превышает значение параметра, в лог-файл записывается не только факт ошибки, но и подробная информация о ней. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dics_suffix.** Формат названия лог-файлов. Значение по умолчанию **%Y%m%d**.

Fact Extractor Service

Сервис Fact Extractor Service настраивается в одноименной секции конфигурационного файла .env.

Структура секции:

```
# FactExtractorService
dfes_replicas = {Количество запускаемых копий сервиса Fact Extractor Service}
dfes_external_port = {TCP-порт сервиса Fact Extractor Service}
dfes_debug = {Записывать в лог-файл отладочные сообщения}
dfes_area = {Используемый раздел конфигурационного файла}
dfes_queue_connection = ${rabbit_url}
dfes_message_ttl_seconds = ${queues_message_ttl}
dfes_prefetch_count = {Количество одновременно обрабатываемых сообщений}
dfes_server_url = ${elastic_apm_server_urls}
dfes_metrics_interval = ${elastic_apm_metric_interval}
dfes_span_frames_min_duration = ${elastic_apm_span_frames_min_duration}
dfes_suffix = {Формат названия лог-файлов}
```

- **dfes_replicas.** Количество запускаемых копий сервиса Fact Extractor Service. Возможные значения: **0** – сервис DFES не будет запущен при следующем запуске сервисов, **1** – сервис DFES будет запущен при следующем запуске сервисов. Значение по умолчанию **1**.
- **dfes_external_port.** TCP-порт сервиса Fact Extractor Service. Значение по умолчанию **9017**.
- **dfes_debug.** Записывать в лог-файл отладочные сообщения. Возможные значения: **True**, **False**. Значение по умолчанию **True**.
- **dfes_area.** Используемый в данный момент раздел конфигурационного файла. Используется для быстрого переключения между настройками окружения. Пример значения: **develop**.
- **dfes_queue_connection.** Строка подключения к RabbitMQ. Подставляется значение, указанное для параметра **rabbit_url** в конфигурационном файле .env в секции RabbitMQ.
- **dfes_message_ttl_seconds.** Время хранения сообщений об обработке документа в RabbitMQ в секундах. Подставляется значение, указанное для [данного параметра](#) в конфигурационном файле .env.

- **dfes_prefetch_count.** Количество одновременно обрабатываемых запросов на сервисе. Значение по умолчанию **0**. Если установлено такое значение, то оптимальное количество одновременно обрабатываемых запросов подбирается автоматически. Оно зависит от доступного ресурса системы, который рассчитывается как количество логических ядер процессора минус одно ядро для работы ОС. Формула для расчета параметра: **(CPU-1)/2** с округлением вверх. Например, при CPU=8 доступный ресурс системы будет равен 7 (одно ядро отводится на работу ОС), а итоговое количество параллельно обрабатываемых запросов будет равно 4.

ПРИМЕЧАНИЕ. Количество одновременно обрабатываемых запросов на экспорт или импорт прикладной разработки всегда равно 1, вне зависимости от установленного в параметре значения.

- **dfes_server_url.** Адрес сервиса Elastic APM Service. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dfes_metrics_interval.** Интервал сбора метрик о состоянии памяти и CPU. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dfes_span_frames_min_duration.** Если при обработке документа возникла ошибка и время обработки превышает значение параметра, в лог-файл записывается не только факт ошибки, но и подробная информация о ней. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dfes_suffix.** Формат названия лог-файлов. Значение по умолчанию **%Y%m%d**.

Fact Extractor Rules Service

Сервис Fact Extractor Rules Service настраивается в одноименной секции конфигурационного файла .env.

Структура секции:

```
# FactExtractorRulesService
dfesr_replicas = {Количество запускаемых копий сервиса Fact Extractor Rules Service}
dfesr_external_port = {TCP-порт сервиса Fact Extractor Rules Service}
dfesr_debug = {Записывать в лог-файл отладочные сообщения}
dfesr_area = {Используемый раздел конфигурационного файла}
dfesr_parser_file_name = {Имя файла синтаксического анализатора для извлечения фактов}
dfesr_parser_debug = {Записывать в лог-файл отладочные сообщения в процессе извлечения фактов и расчета вероятностей}
dfesr_parsers_count = {Количество экземпляров синтаксического анализатора в пуле}
dfesr_parser_timeout = {Таймаут в секундах, после которого обработка запроса прерывается}
dfesr_queue_connection = ${rabbit_url}://?heartbeat={Значение}
dfesr_message_ttl_seconds = ${queues_message_ttl}
dfesr_prefetch_count = {Количество одновременно обрабатываемых сообщений}
dfesr_server_url = ${elastic_apm_server_urls}
dfesr_metrics_interval = ${elastic_apm_metric_interval}
dfesr_span_frames_min_duration = ${elastic_apm_span_frames_min_duration}
dfesr_suffix = {Формат названия лог-файлов}
```

- **dfesr_replicas**. Количество запускаемых копий сервиса Fact Extractor Rules Service. Возможные значения: **0** – сервис DFES-R не будет запущен при следующем запуске сервисов, **1** – сервис DFES-R будет запущен при следующем запуске сервисов. Значение по умолчанию **1**.
- **dfesr_external_port**. TCP-порт сервиса Fact Extractor Rules Service. Значение по умолчанию **9018**.
- **dfesr_debug**. Записывать в лог-файл отладочные сообщения. Возможные значения: **True**, **False**. Значение по умолчанию **True**.
- **dfesr_area**. Используемый в данный момент раздел конфигурационного файла. Используется для быстрого переключения между настройками окружения. Пример значения: **develop**.
- **dfesr_parser_file_name**. Имя файла синтаксического анализатора, который применяется для извлечения фактов. Значение по умолчанию **parser.exe**. Параметр является внутренним, изменять не рекомендуется.
- **dfesr_parser_debug**. Записывать в лог-файл отладочные сообщения, возникающие в процессе извлечения фактов и расчета вероятностей. Возможные значения: **True**, **False**. Значение по умолчанию **False**.
- **dfesr_parsers_count**. Количество экземпляров синтаксического анализатора, которые хранятся в пуле. Используется для ускорения извлечения фактов. Рекомендуемое значение: на единицу больше, чем количество аппаратных потоков. Значение по умолчанию **4**.
- **dfesr_parser_timeout**. Время, после которого обработка запроса прерывается. В таком случае список извлеченных фактов возвращается пустой. Значение по умолчанию **30s**.
- **dfesr_queue_connection**. Строка подключения к RabbitMQ в формате amqp://{Имя пользователя}:{Пароль}@{Адрес}:{Порт}/{Виртуальный хост}?heartbeat={Значение}, где **heartbeat** – промежуток времени в секундах, за который в RabbitMQ дважды отправляется сообщение о состоянии сервиса. Если этого не происходит, то соединение между RabbitMQ и сервисом прерывается. Значение по умолчанию **300**.
- **dfesr_message_ttl_seconds**. Время хранения сообщений об обработке документа в RabbitMQ в секундах. Подставляется значение, указанное для [данного параметра](#) в конфигурационном файле .env.
- **dfesr_prefetch_count**. Количество одновременно обрабатываемых запросов на сервисе. Значение по умолчанию **0**. Если установлено такое значение, то оптимальное количество одновременно обрабатываемых запросов подбирается автоматически. Оно зависит от доступного ресурса системы, который рассчитывается как количество логических ядер процессора минус одно ядро для работы ОС. Формула для расчета параметра: **(CPU-1)/2** с округлением вверх. Например, при CPU=8 доступный ресурс системы будет равен 7 (одно ядро отводится на работу ОС), а итоговое количество параллельно обрабатываемых запросов будет равно 4.
 ПРИМЕЧАНИЕ. Количество одновременно обрабатываемых запросов на экспорт или импорт прикладной разработки всегда равно 1, вне зависимости от установленного в параметре значения.
- **dfesr_server_url**. Адрес сервиса Elastic APM Service. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.

- **dfesr_metrics_interval.** Интервал сбора метрик о состоянии памяти и CPU. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service..
- **dfesr_span_frames_min_duration.** Если при обработке документа возникла ошибка и время обработки превышает значение параметра, в лог-файл записывается не только факт ошибки, но и подробная информация о ней. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dfesr_suffix.** Формат названия лог-файла. Значение по умолчанию **%Y%m%d**.

Fact Extractor Learn Service

Сервис Fact Extractor Learn Service настраивается в одноименной секции конфигурационного файла .env.

Структура секции:

```
# FactExtractorLearnService
dfesl_replicas = {Количество запускаемых копий сервиса Fact Extractor Learn Service}
dfesl_max_replicas_per_node = {Ограничение количества запускаемых копий сервиса Fact Extractor Learn Service на одном сервере}
dfesl_external_port = {TCP-порт сервиса Fact Extractor Learn Service}
dfesl_debug = {Записывать в лог-файл отладочные сообщения}
dfesl_worker_cache_lifetime = {Время хранения результата обучения в кэше сервиса в секундах}
dfesl_can_train_models = {Разрешить использовать сервис как сервис обучения}
dfesl_dbname = ${pg_database}
dfesl_dbuser = ${pg_user}
dfesl_dbhost = ${pg_host}
dfesl_dbpass = ${pg_password}
dfesl_dbport = ${pg_port}
dfesl_queue_connection = ${rabbit_url}///?heartbeat={Значение}
dfesl_message_ttl_seconds = ${queues_message_ttl}
dfesl_prefetch_count = {Количество одновременно обрабатываемых сообщений}
dfesl_model_cache_lifetime = {Время хранения модели в пуле}
dfesl_subprocesses = {Количество подпроцессов}
dfesl_algorithm = {Алгоритм обучения}
dfesl_c1 = {Коэффициент L1-регуляризации}
dfesl_c2 = {Коэффициент L2-регуляризации}
dfesl_epsilon = {Точность вычисления показателя качества модели}
dfesl_delta = {Относительный показатель качества модели}
dfesl_slice_period = {Количество итераций обучения, через которое оценивается относительный показатель качества модели}
dfesl_max_iterations = {Максимальное число итераций обучения}
dfesl_all_possible_transitions = {Генерировать все сочетания классов}
dfesl_logging_period = {Интервал логирования процесса обучения в итерациях}
dfesl_test_size = {Размер выборки для расчета метрик}
dfesl_server_url = ${elastic_apm_server_urls}
dfesl_metrics_interval = ${elastic_apm_metric_interval}
dfesl_span_frames_min_duration = ${elastic_apm_span_frames_min_duration}
dfesl_suffix = {Формат названия лог-файлов}
```

- **dfesl_replicas**. Количество запускаемых копий сервиса Fact Extractor Learn Service. Возможные значения: **0** – сервис DFES-L не будет запущен при следующем запуске сервисов, **1** – сервис DFES-L будет запущен при следующем запуске сервисов. Значение по умолчанию **1**.
- **dfesl_external_port**. TCP-порт сервиса Fact Extractor Learn Service. Значение по умолчанию **9019**.
- **dfesl_debug**. Записывать в лог-файл отладочные сообщения. Возможные значения: **True**, **False**. Значение по умолчанию **True**.
- **dfesl_worker_cache_lifetime**. Время хранения результатов обучения модели в кэше сервиса Fact Extractor Learn Service. Указывается в секундах. Если при отправке сообщения о результатах сервису не удастся подключиться к RabbitMQ, то данные о последнем обучении сохраняются в кэше DFES-L в течение заданного периода. Это позволит не запускать обучение заново при восстановлении подключения к RabbitMQ. Значение по умолчанию **86400** (24 часа).
- **dfesl_can_train_models**. Разрешить использовать Fact Extractor Learn Service в качестве сервиса обучения. Возможные значения: **True**, **False**. Значение по умолчанию **True**.
- **dfesl_dbname**. Имя базы данных PostgreSQL. Подставляется значение, указанное для параметра **pg_database** в конфигурационном файле .env в секции PostgreSQL.
- **dfesl_dbuser**. Имя пользователя для подключения к СУБД PostgreSQL. Подставляется значение, указанное для параметра **pg_user** в конфигурационном файле .env в секции PostgreSQL.
- **dfesl_dbhost**. Адрес СУБД PostgreSQL. Подставляется значение, указанное для параметра **pg_host** в конфигурационном файле .env в секции PostgreSQL.
- **dfesl_dbpass**. Пароль для подключения к СУБД PostgreSQL. Подставляется значение, указанное для параметра **pg_password** в конфигурационном файле .env в секции PostgreSQL.
- **dfesl_dbport**. Порт для подключения к СУБД PostgreSQL. Подставляется значение, указанное для параметра **pg_port** в конфигурационном файле .env в секции PostgreSQL.
ПРИМЕЧАНИЕ. Параметры подключения к базе данных должны совпадать с параметрами, указанными в [конфигурационном файле](#) Smart Service.
- **dfesl_queue_connection**. Строка подключения к RabbitMQ в формате amqp://{Имя пользователя}:{Пароль}@{Адрес}:{Порт}/{Виртуальный хост}?heartbeat={Значение}, где **heartbeat** – промежуток времени в секундах, за который в RabbitMQ дважды отправляется сообщение о состоянии сервиса. Если этого не происходит, то соединение между RabbitMQ и сервисом прерывается. Значение по умолчанию **36000**.
- **dfesl_message_ttl_seconds**. Время хранения сообщений об обработке документа в RabbitMQ в секундах. Значение по умолчанию **86400**. Подставляется значение, указанное для [данного параметра](#) в конфигурационном файле .env.

- **dfesl_prefetch_count.** Количество одновременно обрабатываемых запросов на сервисе. Значение по умолчанию **0**. Если установлено такое значение, то оптимальное количество одновременно обрабатываемых запросов подбирается автоматически. Оно зависит от доступного ресурса системы, который рассчитывается как количество логических ядер процессора минус одно ядро для работы ОС. Формула для расчета параметра: $(\text{CPU}-1)/2$ с округлением вверх. Например, при CPU=8 доступный ресурс системы будет равен 7 (одно ядро отводится на работу ОС), а итоговое количество параллельно обрабатываемых запросов будет равно 4.

ПРИМЕЧАНИЕ. На некоторые типы запросов распространяются дополнительные условия, которые позволяют снизить нагрузку на систему. А именно:

- если значение параметра **0**, то количество одновременно обрабатываемых запросов на обучение или дообучение моделей всегда равно 1;
 - количество одновременно обрабатываемых запросов на экспорт или импорт прикладной разработки всегда равно 1, вне зависимости от установленного в параметре значения.
- **dfesl_model_cache_lifetime.** Время хранения модели в невыгружаемом пуле в минутах. Модели, к которым не обращаются в течение этого времени, выгружаются из пула. Укажите целое число больше нуля. Значение по умолчанию **60**.
 - **dfesl_subprocesses.** Количество подпроцессов для более быстрого обучения моделей извлечения фактов. Если параметр отсутствует, подпроцессы не выделяются. Значение по умолчанию **4**.
 - **dfesl_algorithm.** Алгоритм обучения модели. Поддерживается только алгоритм lbfgs.
 - **dfesl_c1, dfesl_c2.** Значение коэффициентов по умолчанию **0,1**.
 - **dfesl_epsilon.** Точность, с которой вычисляется параметр delta. Значение по умолчанию **0,00001**.
 - **dfesl_delta.** Допустимая разница в качестве обучения модели на разных итерациях. Если качество модели отличается не более чем на установленное значение, то считается, что качество не изменилось. Значение по умолчанию **0,04**.
 - **dfesl_slice_period.** Количество итераций обучения, через которое проверяется, что значение параметра **delta** не превышает установленного. Значение по умолчанию **10**.
 - **dfesl_max_iterations.** Максимальное количество итераций для обучения. Значение по умолчанию **250**.
 - **dfesl_all_possible_transitions.** Значение по умолчанию **True**.
 - **dfesl_logging_period.** Периодичность логирования процесса при обучении модели извлечения фактов. Задается в итерациях. Значение по умолчанию **10**. Если логирование процесса обучения не требуется, отключите его, указав значение **0**.
 - **dfesl_test_size.** Значение по умолчанию **0,25**.

ПРИМЕЧАНИЕ. Подробнее о параметрах обучения см. в документации к библиотеке Sklearn-crfsuite статью [API Reference](#), раздел CRF.

- **dfesl_server_url.** Адрес сервиса Elastic APM Service. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dfesl_metrics_interval.** Интервал сбора метрик о состоянии памяти и CPU. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.

- **dfesi_span_frames_min_duration.** Если при обработке документа возникла ошибка и время обработки превышает значение параметра, в лог-файл записывается не только факт ошибки, но и подробная информация о ней. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dfesi_suffix.** Формат названия лог-файла. Значение по умолчанию **%Y%m%d**.

Object Detection Service

Сервис Object Detection Service настраивается в одноименной секции конфигурационного файла .env.

Структура секции:

```
# ObjectDetectionService
dods_replicas = {Количество запускаемых копий сервиса Object Detection Service}
dods_external_port = {TCP-порт сервиса Fact Extractor Learn Service}
dods_debug = {Записывать в лог-файл отладочные сообщения}
dods_queue_connection = ${rabbit_url}
dods_message_ttl_seconds = ${queues_message_ttl}
dods_prefetch_count = {Количество одновременно обрабатываемых сообщений}
dods_server_url = ${elastic_apm_server_urls}
dods_metrics_interval = ${elastic_apm_metric_interval}
dods_span_frames_min_duration = ${elastic_apm_span_frames_min_duration}
dods_suffix = {Формат названия лог-файлов}
```

- **dods_replicas.** Количество запускаемых копий сервиса Object Detection Service. Возможные значения: **0** – сервис DODS не будет запущен при следующем запуске сервисов, **1** – сервис DODS будет запущен при следующем запуске сервисов. Значение по умолчанию **1**.
- **dods_external_port.** TCP-порт сервиса Object Detection Service. Значение по умолчанию 9047.
- **dods_debug.** Записывать в лог-файл отладочные сообщения. Возможные значения: **True**, **False**. Значение по умолчанию **True**.
- **dods_queue_connection.** Строка подключения к RabbitMQ. Подставляется значение, указанное для параметра **rabbit_url** в конфигурационном файле .env в секции RabbitMQ.
- **dods_message_ttl_seconds.** Время хранения сообщений об обработке документа в RabbitMQ в секундах. Подставляется значение, указанное для [данного параметра](#) в конфигурационном файле .env.
- **dods_prefetch_count.** Количество одновременно обрабатываемых запросов на сервисе. Значение по умолчанию **0**. Если установлено такое значение, то оптимальное количество одновременно обрабатываемых запросов подбирается автоматически. Оно зависит от доступного ресурса системы, который рассчитывается как количество логических ядер процессора минус одно ядро для работы ОС. Формула для расчета параметра: **(CPU-1)/2** с округлением вверх. Например, при CPU=8 доступный ресурс системы будет равен 7 (одно ядро отводится на работу ОС), а итоговое количество параллельно обрабатываемых запросов будет равно 4.
- **dods_server_url.** Адрес сервиса Elastic APM Service. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.

- **dods_metrics_interval.** Интервал сбора метрик о состоянии памяти и CPU. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dods_span_frames_min_duration.** Если при обработке документа возникла ошибка и время обработки превышает значение параметра, в лог-файл записывается не только факт ошибки, но и подробная информация о ней. Подставляется значение, указанное для данного параметра в конфигурационном файле .env в секции Elastic APM Service.
- **dods_suffix.** Формат названия лог-файлов. Значение по умолчанию **%Y%m%d**.

Storage Service

Сервис Storage Service настраивается в одноименной секции конфигурационного файла .env.

Структура секции:

```
# StorageService
storage_service_replicas = {Количество запускаемых копий сервиса хранения объектов}
storage_service_external_port = {TCP-порт сервиса хранения объектов}
storage_service_access_key = {Имя пользователя сервиса хранения объектов}
storage_service_secret_key = {Пароль пользователя сервиса хранения объектов}
```

- **storage_service_replicas.** Количество запускаемых копий сервиса хранения объектов. Возможные значения: **0** – сервис хранения объектов не будет запущен при следующем запуске сервисов, **1** – сервис хранения объектов будет запущен при следующем запуске сервисов. Значение по умолчанию **1**.
- **storage_service_external_port.** TCP-порт сервиса хранения объектов. Значение по умолчанию **9021**.
- **storage_service_access_key.** Имя пользователя для доступа к хранилищу.
- **storage_service_secret_key.** Пароль пользователя для доступа к хранилищу.

Elastic APM Service

Сервис Elastic APM Service настраивается в одноименной секции конфигурационного файла .env.

Структура секции:

```
# ElasticApm
elastic_apm_replicas = {Количество запускаемых копий сервиса Elastic APM Service}
elastic_apm_external_port = {TCP-порт сервиса Elastic APM Service}
elastic_apm_server_urls = {Адрес сервиса Elastic APM Service}
elastic_apm_log_level = {Минимальный уровень протоколирования сервиса Elastic APM Service}
elastic_apm_transaction_sample_rate = {Регулярность отправки трейсов в сервис Elastic APM Service}
elastic_apm_span_frames_min_duration = {Время обработки документа при возникновении ошибки, при котором в лог-файл записывается подробная информация о ней. В миллисекундах}
```

`elastic_apm_metric_interval` = {Интервал сбора метрик в секундах}
`elastic_apm_capture_body` = {Сообщения, которые передаются в сервис Elastic APM Service}
`elastic_apm_traces_path` = {Путь до каталога с файлами трассировок}
`elastic_apm_traces_file_name` = {Базовое имя файлов трассировки}
`elastic_apm_output_file_rotate_every_kb` = {Максимальный размер файла трассировки}
`elastic_apm_output_file_number_of_files` = {Максимальное количество файлов трассировки}

- **elastic_apm_replicas**. Количество запускаемых копий сервиса Elastic APM Service. Возможные значения: **0** – сервис Elastic APM Service не будет запущен при следующем запуске сервисов, **1** – сервис Elastic APM Service будет запущен при следующем запуске сервисов. Значение по умолчанию **1**.
- **elastic_apm_external_port**. TCP-порт сервиса Elastic APM Service. Значение по умолчанию **9082**.
- **elastic_apm_server_urls**. Адрес сервиса Elastic APM Service.
- **elastic_apm_log_level**. Минимальный уровень протоколирования Elastic APM Service.
- **elastic_apm_transaction_sample_rate**. Регулярность отправки трейсов в Elastic APM Service. Укажите значение **1.0**, если необходимо отправлять все трейсы, **0.5** – если каждый второй. Значение по умолчанию **1.0**.
- **elastic_apm_metric_interval**. Интервал сбора метрик о состоянии памяти и CPU. Значение по умолчанию **0s**, при котором метрики не собираются.
- **elastic_apm_capture_body**. Сообщения, которые передаются в сервис Elastic APM Service. Возможные значения:
 - **off** – сообщения не передаются;
 - **errors** – сообщения с ошибками;
 - **transactions** – сообщения без ошибок;
 - **all** – все сообщения.
- **elastic_apm_traces_path**. Путь до каталога с файлами трассировок. Значение по умолчанию **./etc/elastic-apm/traces**.
- **elastic_apm_file_name**. Базовое имя файлов трассировок. Например, при значении **apm-server** файлам присваиваются имена **apm-server**, **apm-server.1**, **apm-server.2** и т.д.
- **elastic_apm_output_file_rotate_every_kb**. Максимальный размер файла в килобайтах. Когда файл с трейсами достигает максимального размера, создается новый файл. Значение по умолчанию **10240**.
- **elastic_apm_output_file_number_of_files**. Максимальное количество файлов трассировок. Если количество файлов превышено, старые файлы перезаписываются. Значение по умолчанию **7**.

InfluxDB

Параметры базы данных InfluxDB, которая предназначена для хранения данных из Telegraf и их передачи в Grafana, настраиваются в одноименной секции конфигурационного файла `.env`.

Структура секции:

```
# InfluxDB
influxdb_replicas = {Количество запускаемых копий InfluxDB}
influxdb_user = {Имя пользователя InfluxDB}
influxdb_password = {Пароль пользователя InfluxDB}
```

- **influxdb_replicas**. Количество запускаемых копий базы данных InfluxDB. Возможные значения: **0** – база данных InfluxDB не будет запущена при следующем запуске сервисов, **1** – база данных InfluxDB будет запущена при следующем запуске сервисов. Значение по умолчанию **1**.
- **influxdb_user** и **influxdb_password**. Имя и пароль пользователя InfluxDB. Значение по умолчанию **admin**.

Grafana

Параметры информационной панели Grafana, которая предназначена для отображения метрик в реальном времени, настраиваются в одноименной секции конфигурационного файла `.env`.

Структура секции:

```
# Grafana
grafana_replicas = {Количество запускаемых копий Grafana}
grafana_external_port = {TCP-порт Grafana}
grafana_user = {Имя пользователя Grafana}
grafana_password = {Пароль пользователя Grafana}
```

- **grafana_replicas**. Количество запускаемых копий информационной панели Grafana. Возможные значения: **0** – Grafana не будет запущена при следующем запуске сервисов, **1** – Grafana будет запущена при следующем запуске сервисов. Значение по умолчанию **1**.
- **grafana_external_port**. TCP-порт Grafana. Значение по умолчанию **3000**.
- **grafana_user** и **grafana_password**. Имя и пароль пользователя Grafana. Значение по умолчанию **admin**.

Telegraf

Параметры серверного агента Telegraf, который собирает информацию о метриках и передает ее в базу данных InfluxDB, настраиваются в одноименной секции конфигурационного файла .env.

Структура секции:

```
# Telegraf
telegraf_replicas = {Количество запускаемых копий Telegraf}
```

- **telegraf_replicas**. Количество запускаемых копий Telegraf. Возможные значения: **0** – Telegraf не будет запущен при следующем запуске сервисов, **1** – Telegraf будет запущен при следующем запуске сервисов. Значение по умолчанию **1**.

Общие параметры

- **smart_services_version**. Номер текущей версии сервисов. Пример значения: **2022.7.1.0**.
- **queues_message_ttl**. Время хранения сообщений об обработке документа в RabbitMQ в секундах. Значение по умолчанию **86400**.

Чтобы изменить время хранения сообщений:

1. Запросите в службе поддержки скрипт удаления очередей RabbitMQ и расположите его в корневом каталоге сервисов рядом со скриптами остановки и удаления.
2. Измените значение параметра **queues_message_ttl** в конфигурационном файле .env.
3. Запустите скрипт, не останавливая сервисы Smart Services.

ContainersTimezone – параметры времени внутри контейнеров:

- **timezone**. Часовой пояс внутри контейнеров. Значение по умолчанию **Europe/Samara**.

Logs – параметры логирования сервисов:

- **log_path**. Путь до корневого каталога с лог-файлами. Значение по умолчанию **./logs/**
- **log_level**. Минимальный уровень протоколирования. Значение по умолчанию **Information**.
- **log_path_format**. Формат имени лог-файлов. Значение по умолчанию **{Date}.log**
- **dss_logs, dtes_logs, dis_logs, dps_logs, dfes_logs, dfesr_logs, dfesl_logs, dods_logs, dics_logs, dtcs_logs**. Подкаталоги для лог-файлов каждого сервиса Smart Services в корневом каталоге. Пример значения: **\${log_path}textextractor-service**.

ServicePlacement – параметры расположения сервисов на серверах. По умолчанию все сервисы запускаются на управляющем сервере, где значение **label** по умолчанию – **main**:

- **manager_node_label**. В параметре должно дублироваться значение **label** [управляющего сервера](#).
- **pg_node_label**. Содержит значение **label** сервера, на котором развернута СУБД PostgreSQL.

- **rabbit_node_label.** Содержит значение **label** сервера, на котором развернут брокер сообщений RabbitMQ.
- **storage_service_node_label, dtes_node_label, dfes_node_label, dfesr_node_label, dfesl_node_label, dods_node_label, dics_node_label, dtcs_node_label.** Содержит значение **label** сервера, на котором развернуты соответствующие сервисы Smart Services.

Portainer – параметры сервиса Portainer, который предоставляет графический веб-интерфейс для управления контейнерами после их запуска. Используется вместе с Portainer Agent:

- **portainer_external_port.** TCP-порт сервиса Portainer. Значение по умолчанию **8999**.

RabbitMQ – параметры запуска брокера сообщений RabbitMQ:

- **rabbit_replicas.** Количество запускаемых копий брокера сообщений RabbitMQ. Возможные значения: **0** – брокер сообщений не будет запущен при следующем запуске сервисов, **1** – брокер сообщений будет запущен при следующем запуске сервисов. Значение по умолчанию **1**.
- **rabbit_external_port.** TCP-порт сервиса RabbitMQ. Значение по умолчанию **5672**.
- **rabbit_user** и **rabbit_password.** Имя и пароль пользователя брокера сообщений.
- **rabbit_url.** Строка подключения к RabbitMQ в формате `amqp://{rabbit_user}:{rabbit_password}@{Виртуальный хост}:{Порт}`, где:
 - **Виртуальный хост** – виртуальный тенант RabbitMQ. Подробнее см. в документации RabbitMQ статью [Virtual Hosts](#). Пример значения: `amqp://guest:guest12345@localhost:5672/smart`.

PostgreSQL – параметры запуска СУБД PostgreSQL:

- **pg_replicas.** Количество запускаемых копий СУБД PostgreSQL. Возможные значения: **0** – СУБД PostgreSQL не будет запущена при следующем запуске сервисов, **1** – СУБД PostgreSQL будет запущена при следующем запуске сервисов. Значение по умолчанию **1**.
- **pg_host.** Адрес СУБД PostgreSQL. Значение по умолчанию **localhost**.
- **pg_user** и **pg_password.** Имя и пароль пользователя для подключения к СУБД PostgreSQL.
- **pg_database.** Имя базы данных PostgreSQL для Smart Service.
- **pg_database_dis.** Имя базы данных PostgreSQL для Identity Service.
- **pg_port.** Порт для подключения к СУБД PostgreSQL. Значение по умолчанию **5432**.
- **pg_connection.** Строка подключения к СУБД PostgreSQL в формате `User ID=${pg_user};Password=${pg_password};Host=${pg_host};Port=${pg_port};Database=${pg_database};Pooling={Использовать пул подключений к БД}`.

Устранение неисправностей

Сообщения об ошибках, предупреждениях и информационные сообщения, которые появляются во время работы, записываются в текстовые логи и трассировки. Это помогает администратору или разработчику проанализировать и своевременно устранить ошибки, возникшие в ходе работы.

Логирование

Лог-файлы сервисов Smart Services содержат полную информацию о работе сервисов. Уровень логирования и пути до лог-файлов задаются в конфигурационных файлах сервисов.

Трассировки запросов содержат записи о выполнении запросов сервисами Smart Services. В отличие от лог-файлов сервисов запись ведется в разрезе конкретного запроса. Например, трассировки могут содержать информацию о том, когда началась обработка запроса тем или иным сервисом, какие ошибки возникли в ходе обработки. Параметры записи задаются в конфигурационном файле сервиса.

СОВЕТ. Для анализа лог-файлов сервисов Smart Services и трассировок запросов рекомендуется использовать [панель управления Kibana](#). Для этого настройте их автоматическую загрузку в Elasticsearch:

- для лог-файлов – с помощью службы Filebeat;
- для трассировок измените [значение параметра elastic_apm_server_urls](#) на адрес удаленного сервера Elasticsearch в формате `http://{IP-адрес сервера}:{Порт}`.

Лог-файлы серверных компонентов содержат информацию о работе сервиса хранилищ MiniIO, брокера сообщений RabbitMQ и СУБД PostgreSQL. Ошибки серверных компонентов записываются в виде лог-файлов контейнера соответствующего серверного компонента. Чтобы просмотреть лог-файлы, выполните команду:

```
docker logs {имя контейнера}
```

Анализ ошибок

Иногда во время работы сервисов требуется перезапуск контейнеров. Например, после обучения большой модели извлечения фактов. Если после этого контейнеры не запустились или запустились частично, рекомендуется сначала перезагрузить сервер. Если контейнеры не запускаются и после перезагрузки сервера, перейдите к анализу ошибок с помощью логов.

Чтобы выяснить причину ошибки в работе сервисов Smart Services:

1. Проанализируйте [логи сервисов](#) и [трассировки запросов](#).

СОВЕТ. Для устранения ошибок в обработке документов рекомендуется использовать трассировки запросов, так как по логам сервисов сложнее отследить обработку отдельного запроса.

2. Проанализируйте лог-файлы сервиса хранилищ MiniIO, брокера сообщений RabbitMQ и СУБД PostgreSQL. Для анализа работы RabbitMQ также можно использовать плагин [RabbitMQ Management](#).
3. Если ошибка происходит при запуске сервиса или до начала логирования, информация о ней не будет включена в лог-файлы. Чтобы выявить такую ошибку воспользуйтесь одним из способов:

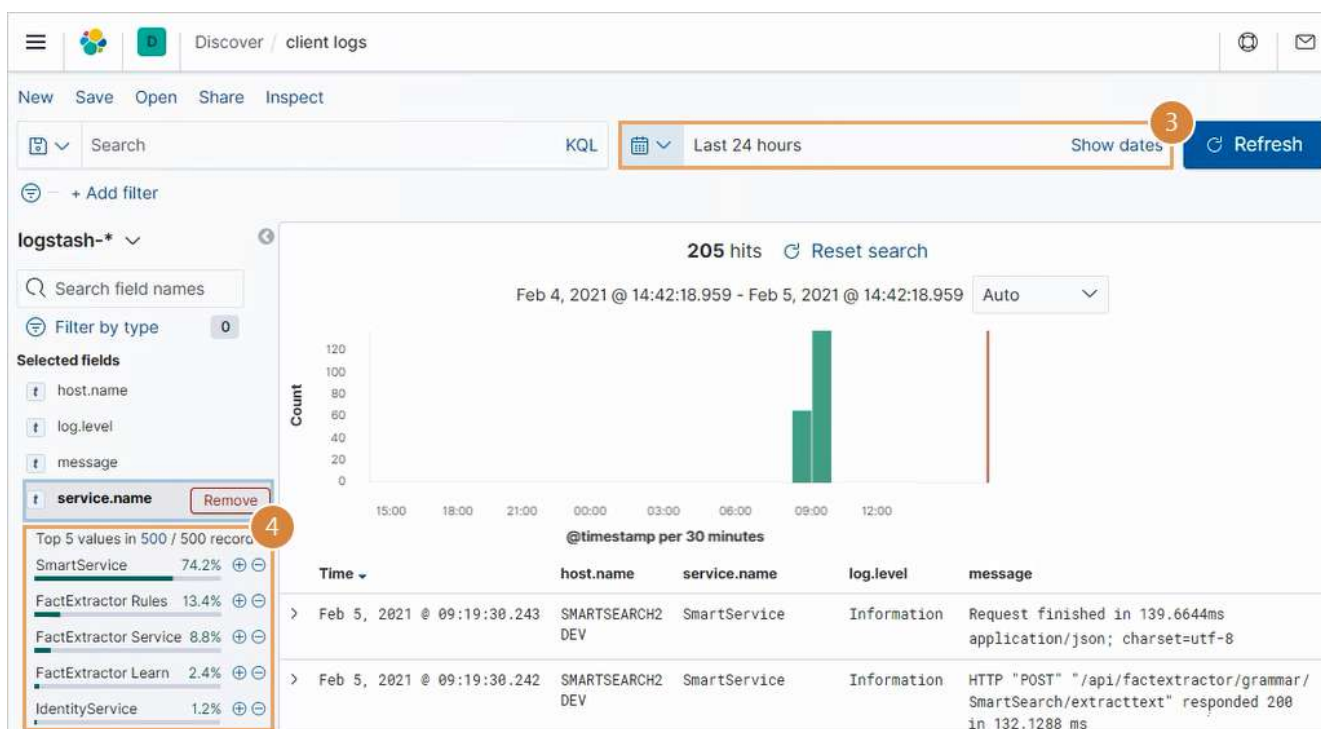
- изучите журнал событий Linux;
- изучите лог-файлы контейнера в файле `/var/lib/docker/containers/<container_id>/<container_id>-json.log`, где:
 - **<container_id>** – идентификационный номер контейнера. Чтобы узнать его, выполните команду:

```
docker inspect <имя контейнера> | grep Id
```

Анализ логов сервисов Smart Services

Для поиска информации в логах сервисов:

1. Запустите панель управления Kibana.
2. В главном меню в разделе **Kibana** перейдите в раздел **Discover** и выберите логи для анализа. Подробнее см. в документации Kibana статью [Discover](#).
3. Укажите период, за который нужно проанализировать логи:



Для выбора доступны часто используемые интервалы:

- **Today** – за сегодняшний день;
- **This week** – за текущую неделю;
- **Last <интервал>** – за указанное время, от последних 15 минут до последнего года.

Также в поле можно вручную указать необходимый диапазон дат.

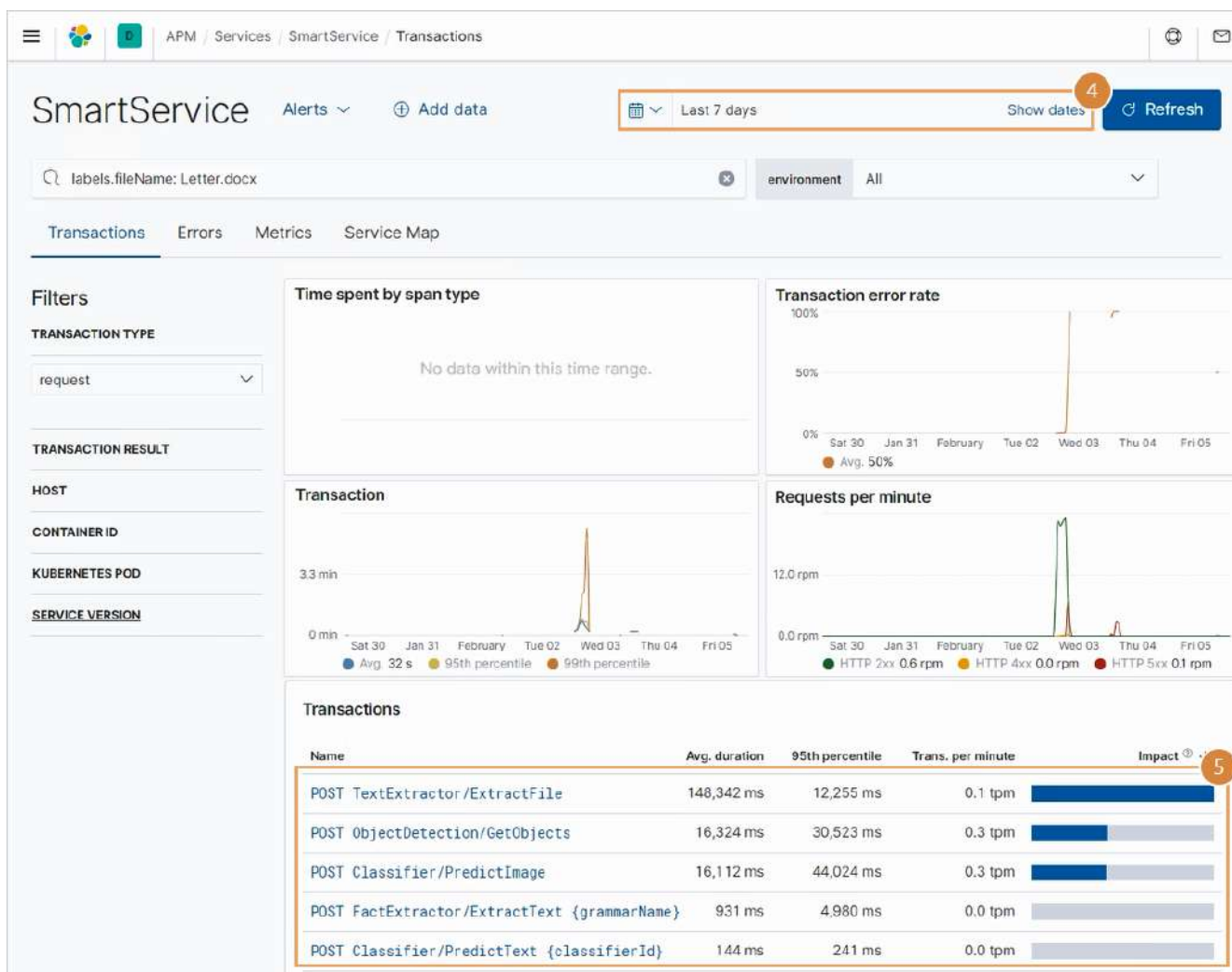
4. При необходимости отфильтруйте записи по сервису. Для этого в списке **Selected fields** выберите **service.name**. В открывшемся списке:
 - чтобы отфильтровать записи по одному сервису, нажмите ;
 - чтобы убрать записи по сервису из списка, нажмите .

Анализ трассировок запросов

Для поиска информации в трассировках запросов:

1. Запустите панель управления Kibana.
2. В главном меню в разделе **Observability** выберите пункт **APM**.
3. В списке сервисов выберите **Smart Service**. Это верхнеуровневый сервис, с которого начинается обработка запросов к сервисам Smart Services от внешних систем.

4. Укажите период, за который нужно проанализировать трассировки:

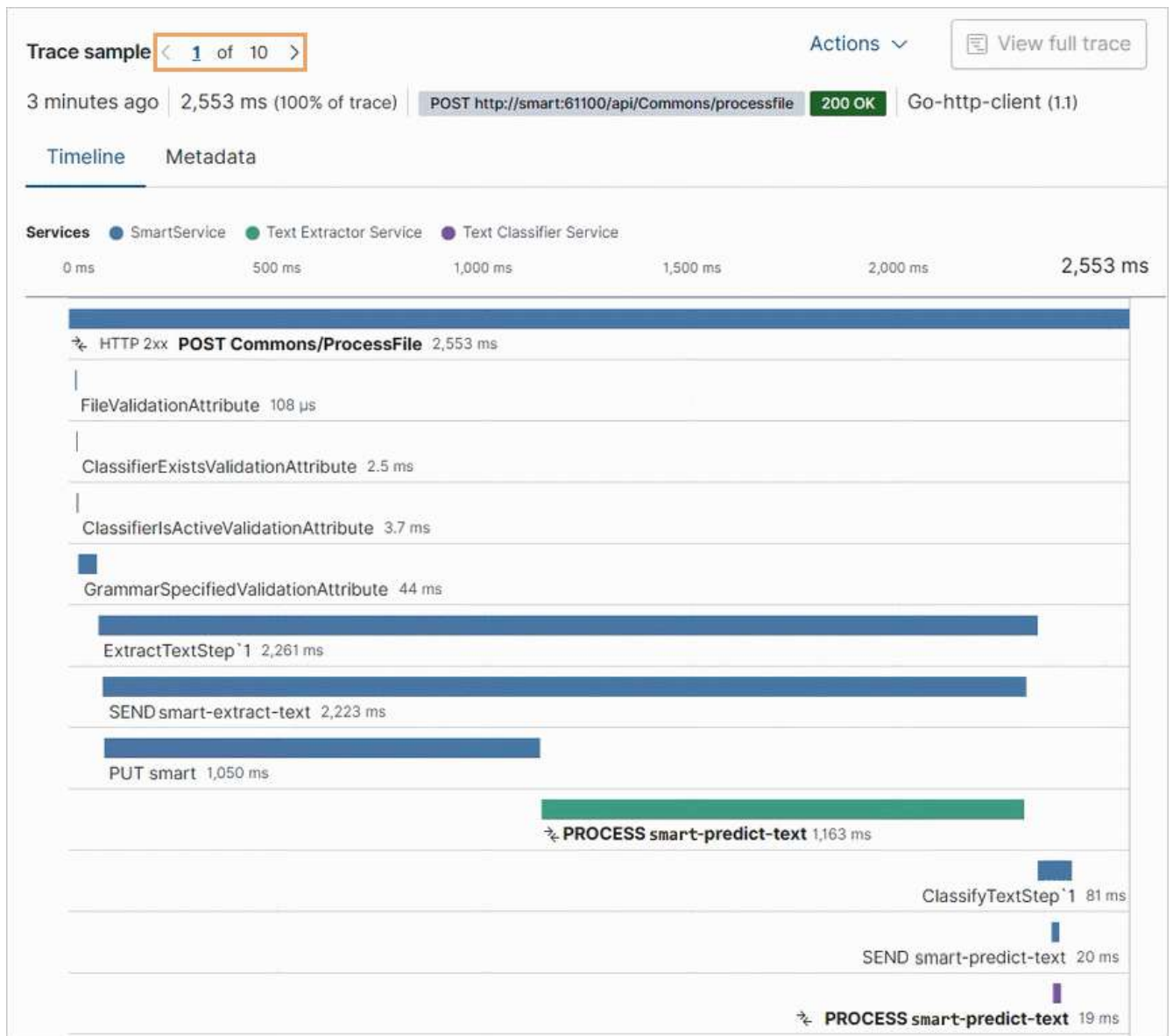


Для выбора доступны часто используемые интервалы:

- **Today** – за сегодняшний день;
- **This week** – за текущую неделю;
- **Last <интервал>** – за указанное время, от последних 15 минут до последнего года.

Также в поле можно вручную указать необходимый диапазон дат.

5. Отфильтруйте записи по типу запроса. Для этого в области Transactions нажмите на нужный тип. Например, **POST TextExtractor/ExtractFile** используется для извлечения текстового слоя из файла. Появится область **Trace sample** с информацией о запросах выбранного типа:



Область содержит информацию о том, какие сервисы обрабатывали запрос, распределение времени обработки между сервисами и т.д. Для переключения между запросами используйте кнопки < и >.

Если при выполнении запроса произошла ошибка, в записи появится предупреждение:



Чтобы посмотреть подробную информацию об ошибках, нажмите на соответствующую строку.

СОВЕТ. Чтобы найти запросы на обработку конкретного документа, дополнительно отфильтруйте записи по его названию. Для этого в строке поиска введите название в формате `labels.fileName: <Название документа>`, например **labels.fileName Letter.docx**.

6. При необходимости можно посмотреть записи из лог-файлов, относящиеся к выбранному запросу. Для этого в области **Trace sample** в выпадающем списке **Actions** выберите пункт **Trace logs**.

ПРИМЕЧАНИЕ. При необходимости в трассировки можно включить информацию о потреблении памяти сервисами. Для этого в [конфигурационных файлах](#) сервисов задайте интервал сбора метрик. Обратите внимание, что сбор метрик значительно увеличивает размер трассировок. По умолчанию значение параметра **0**, сбор метрик отключен.

Примеры анализа и устранения ошибок

В разделе описаны примеры ошибок в работе сервисов Smart Services и методы их устранения.

Пример 1. Тип документов распознается корректно, но факты не извлекаются

Решение

Вероятно, не запущен сервис извлечения фактов. В зависимости от класса документов, это может быть сервис извлечения фактов при помощи правил (Fact Extractor Rules Service) или сервис извлечения фактов при помощи обучаемых моделей (Fact Extractor Learn Service). Проверьте работу соответствующего сервиса, проанализировав [логи](#) или [трассировки запросов](#). Если сервис не запущен, в ответе на запрос извлечения фактов поле **facts** будет пустым. Лог сервиса будет содержать ошибку вида:

```
bad status code (503) during FactExtractor.post execution. 5 of 5 retries attempted.
```

Чтобы устранить ошибку, запустите сервис.

Пример 2. При запуске сервисов возникает ошибка подключения

В логи сервисов записывается сообщение об ошибке:

```
multiple exceptions: [Errno 10061] Connect call failed ('::1', 5672, 0, 0), [Errno 10061] connect call failed ('127.0.0.1', 5672)
```

Решение

Сервисы Smart Services не могут подключиться к очереди сообщений RabbitMQ. Проверьте, что брокер сообщений RabbitMQ запущен и учетные данные настроены корректно.

Если проблема не решена, проанализируйте лог-файлы RabbitMQ.

Часто ошибки RabbitMQ возникают из-за аварийного завершения его работы. В лог-файл записывается сообщение вида:

```
<дата> [info] <0.327.0> wal: recovering ["<имя_файла>.wal"]
```

```

<Data> [error] <0.327.0> CRASH REPORT Process <0.327.0> with 0 neighbours
exited with reason: no case clause matching {ok,<<0,0,0,0,0>>} in
ra_log_wal:open_existing/1 line 576 in gen_batch_server:init_it/6 line 156

<Data> [error] <0.326.0> Supervisor ra_log_wal_sup had child ra_log_wal
started with ra_log_wal:start_link({compute_checksums => true,dir =>
"c:/Users/DIRECT~1/AppData/Roaming/RabbitMQ/db/RABBIT~1/quor...",...},
[{max_batch_size,32768}]) at undefined exit with reason no case clause
matching {ok,<<0,0,0,0,0>>} in ra_log_wal:open_existing/1 line 576 in
context start_error

```

В этом случае удалите файл **<Имя_файла>.wal** и перезапустите RabbitMQ.

Пример 3. Некоторые документы при обработке сервисами Smart Services не классифицируются и заносятся в систему как простые

В логи сервиса Text Extractor Service записывается сообщение об ошибке вида:

```

visitor.Exception BackgroundImageRemoverVisitor: u != 0
at OpenCvSharp.NativeMethods.<...>

```

Решение

Запись логов, содержащая **u!=0**, означает нехватку оперативной памяти для корректной работы сервисов. Чтобы устранить ошибку, перезапустите Text Extractor Service и проверьте конфигурацию оборудования на соответствие системным требованиям. При необходимости увеличьте количество оперативной памяти на сервере.

Справочная информация

Ознакомьтесь со справочной информацией:

- [виды документов](#), которые обрабатывают сервисы Smart Services;
- [список фактов](#), которые умеют извлекать сервисы Smart Services;
- [ограничения](#) при извлечении фактов;
- [список языков](#), доступных для распознавания;
- [сторонние компоненты](#), которые входят в состав сервисов Smart Services.

Виды обрабатываемых документов

В базовой поставке Smart Services поддерживается обработка стандартных видов документов. Полнота извлечения *фактов* для них составляет 80% и выше.

Для корректной обработки документы следует оформлять по принятым формам:

Вид документа, включенный в базовую поставку	Рекомендация к оформлению документов
Акт выполненных работ	Общепринятая практика делового документооборота
Входящее письмо	ГОСТ Р 7.0.97-2016
Железнодорожный билет (электронный)	Рекомендуемая форма, установленная ОАО «РЖД»
Постановление судебных приставов	Приложение N 103 к приказу Федеральной службы судебных приставов N 318
Счет на оплату (русскоязычный)	Общепринятая практика делового документооборота
Счет-фактура	Приложение N 1 к постановлению Правительства РФ N 1137
Товарная накладная	Унифицированная форма № ТОРГ-12
Универсальный передаточный документ	Рекомендуемая форма

Сервисы Smart Services также обрабатывают дополнительные виды документов, для которых полнота извлечения фактов может быть ниже 80%:

- авиабилет;
- договор;
- дополнительное соглашение;
- корректировочный счет-фактура;
- свидетельство о постановке на налоговый учет;
- счет на оплату (англоязычный);
- товарно-транспортная накладная;
- транспортная железнодорожная накладная;
- транспортная накладная;
- универсальный корректировочный документ;
- экспедиторская расписка.

Список извлекаемых фактов

Вид документа	Извлекаемые факты
Входящее письмо	Вид документа
	Дата документа
	Номер документа
	Дата документа «В ответ на»
	Номер документа «В ответ на»
	Прочие даты документа
	Организационно-правовая форма (ОПФ) и Наименование контрагента
	Наименование головной организации
	ИНН и КПП контрагента
	ОГРН и ОГРНИП контрагента
	Р/с контрагента
	Наименование и БИК банка контрагента
	Номер телефона контрагента
	Адрес электронной почты контрагента
	Сайт организации контрагента
	Адресат
	Подписант письма
	Исполнитель
	Тип персоны
	Тема письма, начинающаяся с «О» или «Об» и расположенная сверху первой страницы письма
	Гриф доступа: «Конфиденциально», «Для служебного пользования», «Коммерческая тайна»
	Наименование прочей организации
	ИНН и КПП прочей организации
	ОГРН и ОГРНИП прочей организации
	Р/с прочей организации
	Наименование и БИК банка прочей организации
Договор (в том числе на английском языке)	Документы на русском языке:
	Вид документа
	Номер договора
	Дата договора
	Сумма
	ОПФ и Наименование контрагента
	ИНН и КПП контрагента
	Количество листов
	Валюта

Вид документа	Извлекаемые факты
	Документы на английском языке: Вид документа Номер договора Дата договора Наименование контрагента
Дополнительное соглашение	Номер документа Дата документа Номер ведущего документа Дата ведущего документа ОПФ и Наименование контрагента ИНН и КПП контрагента Подписант дополнительного соглашения
Счет-фактура	Вид документа Номер документа Дата документа Номер исправления Дата исправления Прочие даты документа ОПФ и Наименование контрагента ИНН и КПП контрагента Адрес контрагента Тип контрагента (грузоотправитель, грузополучатель, продавец, покупатель) Сумма без НДС, по документу Сумма НДС, по документу Сумма с НДС, по документу Валюта Наименование товара Код вида товара Единица измерения Код единицы измерения по ОКЕИ Количество товара (объем) Цена за единицу, по строке Сумма без НДС, по строке Сумма акциза, по строке Ставка НДС, по строке Сумма НДС, по строке Сумма с НДС, по строке Страна происхождения товара (код и краткое наименование) Регистрационный номер таможенной декларации

Вид документа	Извлекаемые факты
Корректировочный счет-фактура	Вид документа Номер документа Дата документа Номер исправления корректировки Дата исправления корректировки Номер корректируемого документа Дата корректируемого документа ОПФ и Наименование контрагента ИНН и КПП контрагента Адрес контрагента Тип контрагента (продавец, покупатель) Валюта документа
Универсальный передаточный документ	Вид документа Номер документа Дата документа Номер исправления Дата исправления Наименование документа-основания Дата документа-основания Номер документа-основания Статус (функция) УПД Прочие даты документа ОПФ и Наименование контрагента ИНН и КПП контрагента Адрес контрагента Тип контрагента (грузоотправитель, грузополучатель, продавец, покупатель) Сумма без НДС, по документу Сумма НДС, по документу Сумма с НДС, по документу Количество листов Валюта Код товара Наименование товара Код вида товара Количество товара (объем) Единица измерения Код единицы измерения по ОКЕИ Цена за единицу, по строке Сумма акциза, по строке

Вид документа	Извлекаемые факты
	Сумма без НДС, по строке Ставка НДС, по строке Сумма НДС, по строке Сумма с НДС, по строке Страна происхождения товара (код и краткое наименование) Регистрационный номер таможенной декларации
Универсальный корректировочный документ	Вид документа Номер документа Дата документа Номер исправления корректировки Дата исправления корректировки Номер корректируемого документа Дата корректируемого документа ОПФ и Наименование контрагента ИНН и КПП контрагента Адрес контрагента Тип контрагента (продавец, покупатель) Валюта документа Статус (функция) УКД
Товарная накладная (ТОРГ-12)	Вид документа Номер документа Дата документа Прочие даты документа ОПФ и Наименование контрагента ИНН и КПП контрагента Адрес контрагента Р/с контрагента Наименование и БИК банка контрагента Тип контрагента (грузоотправитель, грузополучатель, поставщик, плательщик) Сумма НДС, по документу Сумма с НДС, по документу Сумма без НДС, по документу Наименование документа-основания Дата документа-основания Номер документа-основания Номер транспортной накладной Дата транспортной накладной Валюта Наименование товара

Вид документа	Извлекаемые факты
	Код товара
	Единица измерения
	Код единицы измерения по ОКЕИ
	Вид упаковки
	Количество в одном месте
	Количество мест/штук
	Масса брутто, по строке
	Масса нетто, по строке
	Цена за единицу, по строке
	Сумма без НДС, по строке
	Ставка НДС, по строке
	Сумма НДС, по строке
	Сумма с НДС, по строке
Акт выполненных работ	Вид документа
	Номер документа
	Дата документа
	Прочие даты документа
	Наименование документа-основания
	Дата документа-основания
	Номер документа-основания
	Сумма без НДС, по документу
	Сумма НДС, по документу
	Сумма с НДС, по документу
	Валюта
	Целая часть суммы, выраженная числом
	Дробная часть суммы, выраженная числом
	Целая часть суммы, выраженная прописью
	Дробная часть суммы, выраженная прописью
	Валюта целой части
	Валюта дробной части
	ОПФ и Наименование контрагента
	ИНН и КПП контрагента
	ОГРН и ОГРНИП контрагента
	Р/с контрагента
	Наименование и БИК банка контрагента
	ФИО подписанта контрагента
	Тип контрагента (продавец, покупатель)
	Наименование работ/услуг
	Количество работ/услуг
	Единица измерения

Вид документа	Извлекаемые факты
Счет на оплату (русскоязычный)	Цена за единицу, по строке
	Сумма с НДС, по строке
	Сумма без НДС, по строке
Счет на оплату (англоязычный)	Вид документа
	Номер документа
	Дата документа
Счет на оплату (русскоязычный)	Наименование документа-основания
	Дата документа-основания
	Номер документа-основания
Счет на оплату (русскоязычный)	Номер заказа на поставку
	Дата получения
	Прочие даты документа
Счет на оплату (русскоязычный)	Сумма без НДС, по документу
	Сумма НДС, по документу
	Сумма с НДС, по документу
Счет на оплату (русскоязычный)	Валюта
	Целая часть суммы, выраженная числом
	Дробная часть суммы, выраженная числом
Счет на оплату (русскоязычный)	Целая часть суммы, выраженная прописью
	Дробная часть суммы, выраженная прописью
	Валюта целой части
Счет на оплату (русскоязычный)	Валюта дробной части
	ОПФ и Наименование контрагента
	ИНН и КПП контрагента
Счет на оплату (русскоязычный)	ОГРН и ОГРНИП контрагента
	Р/с контрагента
	Наименование и БИК банка контрагента
Счет на оплату (русскоязычный)	Тип контрагента (продавец, покупатель)
	Код товара
	Наименование товара
Счет на оплату (русскоязычный)	Количество товара
	Единица измерения
	Цена за единицу, по строке
Счет на оплату (русскоязычный)	Ставка НДС, по строке
	Сумма НДС, по строке
	Сумма с НДС, по строке
Счет на оплату (англоязычный)	Вид документа
	Номер документа
	Дата документа
Счет на оплату (англоязычный)	Прочие даты документа

Вид документа	Извлекаемые факты
	Наименование контрагента Тип контрагента (продавец, покупатель) Номер заказа Банковский счет Сумма документа Сумма НДС Наименование документа-основания Дата документа-основания Номер документа-основания Валюта
Транспортная накладная (ТРН)	Вид документа Номер документа Дата документа
Товарно-транспортная накладная (ТТН)	Вид документа Номер документа Дата документа ОПФ и Наименование контрагента ИНН и КПП контрагента Адрес контрагента Тип контрагента (грузоотправитель, грузополучатель, плательщик)
Транспортная железнодорожная накладная	Вид документа Номер документа Дата документа
Экспедиторская расписка	Вид документа Номер документа Дата документа
Паспорт гражданина РФ	Вид документа Серия и номер паспорта ФИО гражданина Пол гражданина Дата выдачи паспорта Дата рождения Место рождения Наименование организации, выдавшей паспорт Код подразделения организации, выдавшей паспорт
Свидетельство о постановке на налоговый учет	Вид документа ОПФ и Наименование организации/ФИО физического лица ИНН организации/физического лица

Вид документа	Извлекаемые факты
	КПП (для организаций) ОГРН (для организаций) Дата постановки на налоговый учет Серия и номер документа
Авиабилет (электронный)	Вид документа Маршрут авиабилета Дата отправления Дата прибытия Город отправления Город прибытия Стоимость билета (сумма и код валюты) ФИО персоны
Железнодорожный билет (электронный)	Вид документа Дата отправления Дата прибытия Маршрут Стоимость билета (сумма и код валюты) ФИО персоны
Постановление судебного пристава-исполнителя	Вид документа Номер и дата документа Номер и дата исполнительного производства Номер и дата исполнительного документа Наименование должника (ОПФ и Наименование организации/ ФИО физического лица) ИНН и КПП должника ОГРН должника СНИЛС должника Дата рождения должника Номер р/с должника для взыскания задолженности ФИО судебного пристава-исполнителя Должность судебного пристава-исполнителя Главное управление структурного подразделения судебного пристава-исполнителя Структурное подразделение судебного пристава-исполнителя Телефон структурного подразделения судебного пристава-исполнителя Сумма взыскания Валюта взыскания
По умолчанию (рекомендуется использовать для документов, которые не удалось классифицировать)	Дата документа

Вид документа	Извлекаемые факты
	ОПФ и Наименование организации/ФИО индивидуального предпринимателя
	ИНН и КПП организации/индивидуального предпринимателя
	ОГРН юридического лица
	ОГРНИП физического лица
	Сумма документа
	Валюта документа
	ФИО персоны
	Номер телефона
	Адрес электронной почты

Ограничения при извлечении

Ознакомьтесь с требованиями и ограничениями, которые нужно учесть при извлечении:

- [печатей и подписей](#);
- [фактов из скан-копий](#) документов.

Печатей и подписей

При извлечении из документов печатей и подписей необходимо учесть, что:

- обрабатываются только круглые печати размером от 30 до 60 миллиметров;
- обрабатываются только цветные печати. Подписи в документе могут быть любого цвета;
- нельзя определить, являются ли извлеченные печати и подписи разными или одинаковыми;
- нельзя определить, принадлежат ли печати и подписи организациям и лицам, указанным в документе.

Фактов из скан-копии документа

Чтобы *факты* корректно извлекались из скан-копий бумажных документов, убедитесь, что они соответствуют требованиям:

Требование	Возможные проблемы, если требование не соблюдено
Минимальное разрешение изображения – 300DPI	Не распознаны отдельные слова или символы или распознаны некорректно
Качество печати оригинального документа обеспечивает читабельность символов	
Текст документа имеет нулевой или положительный кернинг (межбуквенный интервал)	

Требование	Возможные проблемы, если требование не соблюдено
Текст документа имеет как минимум единичный интерлиньяж (междустрочный интервал)	
Текст документа не перекрыт графическими элементами: подписями, печатями, штампами, резолюциями и т.п.	
Скан-копия документа не содержит физических повреждений: загибов, пятен, отверстий от дыроколов, степлерных скоб, скрепок и т.п.	Не распознаны отдельные слова или символы или распознаны некорректно. При загибе угла страницы возможно некорректное определение формата таблиц документов, что приводит к некорректному извлечению фактов из таблиц
Таблицы имеют замкнутую границу. Границы таблиц и ячеек хорошо пропечатаны и не имеют пропусков	Таблица не извлечена в структурированном виде. Пропущены некоторые ячейки таблицы
Ячейки таблицы не содержат графических элементов: подписей, печатей, штампов, резолюций и т.п.	Содержание ячейки не распознано или распознано некорректно
Паспорт на странице расположен параллельно границам изображения	Не извлечены факты из паспорта или извлечены некорректно
Изображение документа имеет сбалансированную яркость и контрастность в части четкости и цветности изображения	Не распознаны отдельные слова или символы или распознаны некорректно Не извлечены факты из паспорта или извлечены некорректно

Список распознаваемых языков

Сервисы Smart Services поддерживают распознавание одноязычных и мультязычных документов.

По умолчанию включено распознавание документов на русском и английском языках. Чтобы добавить другой язык, обратитесь в службу поддержки.

Язык	Код языка	В стандартной поставке
Азербайджанский	aze	
Албанский	sqi	
Английский	eng	Да
Армянский	hye	
Белорусский	bel	Да
Болгарский	bul	
Венгерский	hun	
Грузинский	kat	
Датский	dan	
Иврит	heb	

Язык	Код языка	В стандартной поставке
Индонезийский	ind	
Испанский	spa	
Итальянский	ita	
Казахский	kaz	Да
Киргизский	kir	
Китайский	chi_sim	
Корейский	kor	
Латышский	lav	
Литовский	lit	
Монгольский	mon	
Немецкий	deu	
Норвежский	nor	
Польский	pol	
Португальский	por	
Румынский; молдавский	ron	
Русский	rus	Да
Сербский	srp	
Словацкий	slk	
Таджикский	tgk	
Тайский	tha	
Татарский	tat	
Турецкий	tur	
Узбекский	uzb	
Украинский	ukr	Да
Финский	fin	
Французский	fra	Да
Хорватский	hrv	
Чешский	ces	
Шведский	swe	
Эстонский	est	
Японский	jpn	

Сторонние компоненты

Компонент	Лицензия	Автор	Проект
aiohttp	The Apache License 2.0 Лицензионное соглашение	Nikolay Kim, Andrew Svetlov	Ссылка на проект
aio-pika	The Apache License 2.0 Лицензионное соглашение	Dmitry Orlov	Ссылка на проект
certifi	The Mozilla Public License Version 2.0 Лицензионное соглашение	Kenneth Reitz	Ссылка на проект
Common.Logging	The Apache License 2.0 Лицензионное соглашение	Mark Pollack, Erich Eichinger, Bruno Baia, Stephen Bohlen	Ссылка на проект
CS2K.NetCore	The BSD License Лицензионное соглашение	Anders Gustafsson, Cureos AB	Ссылка на проект
cython	The Apache License 2.0 Лицензионное соглашение	Stefan Behnel, Robert Bradshaw, Dag Seljebotn, Lisandro Dalcin	Ссылка на проект
dateparser	The BSD 3-Clause License Лицензионное соглашение	Waqas Shabir, Eugene Amirov, Artur Sadurski	Ссылка на проект
dynaconf	Лицензия: MIT License Лицензионное соглашение	Bruno Rocha	Ссылка на проект
EfficientNet PyTorch	The Apache License 2.0 Лицензионное соглашение	Luke Melas-Kyriazi	Ссылка на проект
elastic-apm	The BSD 3-Clause License Лицензионное соглашение	Elastic, Inc	Ссылка на проект
FileSignatures	Лицензия: MIT License Лицензионное соглашение	Neil Harvey, Rustam Sayfutdinov	Ссылка на проект
h5py	The BSD 3-Clause License Лицензионное соглашение	Andrew Collette	Ссылка на проект
joblib	The BSD 3-Clause License Лицензионное соглашение	Gael Varoquaux	Ссылка на проект
jsons	Лицензия: MIT License Лицензионное соглашение	Ramon Hagenaars	Ссылка на проект
Keras	The Apache License 2.0 Лицензионное соглашение	The Keras team	Ссылка на проект
Matplotlib	Лицензия: PSF LICENSE Лицензионное соглашение	The Matplotlib development team	Ссылка на проект
Microsoft .NET Core	Лицензия: MIT License Лицензионное соглашение	Microsoft Corporation	Ссылка на проект

Компонент	Лицензия	Автор	Проект
MinIO	The Apache License 2.0 Лицензионное соглашение	Minio, Inc.	Ссылка на проект
Newtonsoft.Json	Лицензия: MIT License Лицензионное соглашение	James Newton-King	Ссылка на проект
NLTK	The Apache License 2.0 Лицензионное соглашение	Steven Bird, Edward Loper, Ewan Klein	Ссылка на проект
Npgsql.EntityFrameworkCore.PostgreSQL	Лицензия: своя Лицензионное соглашение	The Npgsql Development Team	Ссылка на проект
NumPy	Лицензия: своя Лицензионное соглашение	The NumPy team	Ссылка на проект
OpenCV	The Apache License 2.0 Лицензионное соглашение	The OpenCV team	Ссылка на проект
opencv-python	Лицензия: MIT License Лицензионное соглашение	Olli-Pekka Heinisuo	Ссылка на проект
OpenCvSharp3	The Apache License 2.0 Лицензионное соглашение	shimat	Ссылка на проект
packaging	Both under the Apache License, Version 2 and The 2-Clause BSD License	Donald Stufft and individual contributors	Ссылка на проект
Pandas	The BSD 3-Clause License Лицензионное соглашение	The pandas team	Ссылка на проект
parsy	The MIT License Лицензионное соглашение	Jeanine Adkisson	Ссылка на проект
PdfSharpCore	Лицензия: MIT License Лицензионное соглашение	Empira Software GmbH, Cologne (Germany), David Dunscombe	Ссылка на проект
PDFtoImage	The MIT License Лицензионное соглашение	David Sungaila	Ссылка на проект
psutil	The BSD 3-Clause License Лицензионное соглашение	Giampaolo Rodola	Ссылка на проект
Psycopg	Лицензия: GNU Lesser General Public License Лицензионное соглашение	Daniele Varrazzo, The Psycopg Team	Ссылка на проект
PyJWT	The MIT License Лицензионное соглашение	Jeff Lindsay, Jose Padilla	Ссылка на проект
pymorphy2	The MIT License Лицензионное соглашение	Mikhail Korobov	Ссылка на проект
pyriwin32	Лицензия: своя Лицензионное соглашение	Brian R. Bondy, Glyph Lefkowitz, Jason R. Coombs, Lars Immisch, Mark Hammond	Ссылка на проект

Компонент	Лицензия	Автор	Проект
PyTorch	Лицензия: своя Лицензионное соглашение	Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan	Ссылка на проект
pyzipper	Лицензия: своя Лицензионное соглашение	Daniel Hillier, Sebastian Weigand	Ссылка на проект
RabbitMQ client library	The library is open-source , and is dual-licensed under Apache License v2 и Mozilla Public License v2.0	VMware, Inc. или ее дочерние компании	Ссылка на проект
Rtree	The MIT License Лицензионное соглашение	Sean Gilles, Howard Butler, and contributors	Ссылка на проект
scikit-image	Лицензия: своя Лицензионное соглашение	The scikit-image development team	Ссылка на проект
scikit-learn	The BSD 3-Clause License Лицензионное соглашение	The scikit-learn development team	Ссылка на проект
Serilog.Extensions.Logging	The Apache License 2.0 Лицензионное соглашение	Serilog Contributors	Ссылка на проект
servicemanager	The Apache License 2.0 Лицензионное соглашение	hmrc-web-operations	Ссылка на проект
sklearn-crfsuite	The MIT License Лицензионное соглашение	Mikhail Korobov	Ссылка на проект
sortedcontainers	The Apache License 2.0 Лицензионное соглашение	Grant Jenks	Ссылка на проект
stop-words	The BSD 3-Clause License Лицензионное соглашение	Alireza Savand, Francois	Ссылка на проект
structlog	Both under the Apache License, Version 2 and The MIT license	Hynek Schlawack	Ссылка на проект
TensorFlow	The Apache License 2.0 Лицензионное соглашение	Google Brain	Ссылка на проект
Tesseract	The Apache License 2.0 Лицензионное соглашение	Ray Smith, Zdenko Podobny	Ссылка на проект
Tomita-parser	The Mozilla Public License Version 2.0 Лицензионное соглашение	Яндекс	Ссылка на проект
torchvision	The BSD 3-Clause License Лицензионное соглашение	Torch Contributors	Ссылка на проект
urllib3	The MIT License Лицензионное соглашение	Andrey Petrov	Ссылка на проект
wincertstore	The Python Software Foundation License Version 2	Christian Heimes	Ссылка на проект

Компонент	Лицензия	Автор	Проект
	Лицензионное соглашение		