

12/3/24, 10:47 утра



Платформа, услуги и утилиты

Сгенерировано: 2024-12-03 10:47:49 GMT+0000

SAP Коммерция | 2205

Публичный

Оригинальное содержание:https://help.sap.com/docs/SAP_COMMERCE/d0224eca81e249cb821f2cdf45a82ace?locale=en-US&state=PRODUCTION&version=2205

Предупреждение

Этот документ был создан на основе SAP Help Portal и является неполной версией официальной документации по продукту SAP. Информация, включенная в пользовательскую документацию, может не отражать расположение тем на SAP Help Portal и может не иметь важных аспектов и/или корреляций с другими темами. По этой причине он не предназначен для продуктивного использования.

Для получения более подробной информации посетите сайт<https://help.sap.com/docs/ отказ от ответственности>.

Модуль поиска и навигации

Функциональность модуля поиска и навигации помогает вашим клиентам просматривать страницы ваших интернет-магазинов и просматривать результаты поиска на основе настроек фасета. Это может способствовать повышению конверсии, увеличению заказов и увеличению числа просмотров страниц людьми, которые используют функции поиска и навигации.

Функции	Архитектура	Выполнение
		
Шаблоны поисковых запросов Управление конфигурациями поиска	Расширение solrfacetsearch Расширение solrserver	Установка сервера Solr Конфигурация поиска Solr Facet Солр Безопасность Поддержка нескольких версий Solr

Возможности модуля поиска и навигации

Модуль поиска и навигации предоставляет ряд функций, связанных с созданием и управлением поисковыми конфигурациями. Например, вы можете создавать шаблоны поисковых запросов для своих поисковых конфигураций или управлять

[Шаблоны поисковых запросов](#)

Шаблоны поисковых запросов вводят многочисленные параметры настройки для получения улучшенных результатов поиска для клиентов. При создании шаблонов вы можете использовать различные атрибуты, такие как размер страницы, группировка или сортировка по различным каналам или даже частям витрины.

[Управление конфигурациями поиска](#)

Создавайте и управляйте конфигурациями поиска, индексации и сервера с помощью панели администрирования Backoffice.

Шаблоны поисковых запросов

Шаблоны поисковых запросов вводят многочисленные параметры настройки для получения улучшенных результатов поиска для клиентов. При создании шаблонов вы можете использовать различные атрибуты, такие как размер страницы, группировка или сортировка по различным каналам или даже частям витрины.

Вариант использования

До сих пор пользователь получал одни и те же результаты при поиске по ключевому слову или с помощью предложений, поскольку результаты регулировались настройками витрины. Пользователь хочет легко корректировать запрос в зависимости от того, чего он хочет добиться. Например, удалить поля, такие как изображения, из шаблона предложений, чтобы улучшить производительность или изменить сортировку, чтобы привлечь внимание к определенным продуктам.

ФУНКЦИИ

Создание шаблонов

Создавайте индивидуальные шаблоны, соответствующие вашим конфигурациям поиска и повышающие производительность.

Группировка вариантов

Используйте настройки шаблона для группировки вариантов определенного товара, чтобы сэкономить время ваших клиентов.

Выделение ключевых слов

Настройте выделение ключевых слов для терминов, чтобы сделать их более заметными на витрине магазина.

Создайте шаблон для предлагаемых терминов и настройте его, добавив поля сортировки и ограничения.

ЗАВИСИМОСТИ

Конкретных зависимостей для использования шаблонов поисковых запросов нет.

Шаблоны поисковых запросов в административной панели бэк-офиса

Узнайте, как управлять шаблонами поисковых запросов в Backoffice Administration Cockpit.

Backoffice Administration Cockpit поможет вам создать шаблоны и определить их настройки так, чтобы они соответствовали вашим потребностям. Перейдите по ссылкам ниже, чтобы подробно ознакомиться с функционалом.

Типы шаблонов поисковых запросов

Поддерживаются следующие два типа: ПО УМОЛЧАНИЮ и ПРЕДЛОЖЕНИЯ.

по умолчанию

Шаблон DEFAULT доступен в списке шаблонов с самого начала для работы с ним. В дальнейших разделах вы найдете информацию о том, что он содержит и как работает. Подробности см.[Возможности шаблона поискового запроса](#).

ПРЕДЛОЖЕНИЯ

Шаблон SUGGESTIONS поставляется с решением, однако не доступен в списке с самого начала, однако вы можете добавить его в любое время. Шаблон запроса SUGGESTIONS позволяет вам определять и настраивать поля, которые будут включены в результаты в списке предложений. Подробнее см.[ПРЕДЛОЖЕНИЯ Шаблон](#).

Витрина магазина

Всякий раз, когда упоминается витрина, это означает следующее:<https://apparel-uk.local:9002/yacceleratorstorefront/en>.

[Доступ к шаблонам поисковых запросов](#)

Расположение шаблонов поисковых запросов.

[Шаблон по умолчанию](#)

Шаблон DEFAULT — это шаблон по умолчанию, который идет вместе с конфигурацией поиска. Он изначально включает определенные свойства и настройки запроса, но его можно легко настроить.

[ПРЕДЛОЖЕНИЯ Шаблон](#)

Функция ПРЕДЛОЖЕНИЯ поддерживается изначально и позволяет вам настраивать параметры для предлагаемых элементов, которые появляются при вводе термина в поле поиска.

Доступ к шаблонам поисковых запросов

Расположение шаблонов поисковых запросов.

Контекст

Приведенные ниже шаги покажут вам, как перейти к шаблонам поисковых запросов в панели администрирования бэк-офиса.

Процедура

- В административной панели бэк-офиса перейдите к [Системный поиск и навигация Solr](#) [Фасетный поиск](#) [Конфигурация Индексированные типы](#).
Вы также можете использовать поле поиска для фильтрации всего дерева.

Появится список доступных (если таковые имеются) конфигураций поиска. Выберите ту, для которой вы создали шаблон поискового запроса, и щелкните ее, чтобы открыть редактор.

. Нажмите на **Индексированный тип** вкладку и перейдите к **Поиск конфигурации** раздел.

Вы можете увидеть шаблоны.

-Примечание

Шаблон DEFAULT присутствует всегда.

Дважды щелкните шаблон, если вы хотите его отредактировать.

apparel-ukProductType

INDEXED TYPE **INDEXED PROPERTIES** **FREE TEXT QUERY** **GROUPING** **ADMINISTRATION**

INDEXED TYPE

Identifier **Composed Type**

apparel-ukProductType Product [Product]

SEARCH CONFIGURATION

Search Query Templates

- DEFAULT
- SUGGESTIONS
- + Create new Search Query Template

Возможности шаблона поискового запроса

Откройте для себя функциональность шаблонов поисковых запросов на примере шаблона DEFAULT.

Шаблон DEFAULT доступен вам с самого начала. Он обеспечивает следующую функциональность:

- Основная информация о шаблоне запроса: основная информация о шаблоне и основные свойства, которые применяются ко всему шаблону.
- Свойства запроса: список свойств запроса, установленных для данного типа индекса. Для каждого свойства запроса можно дополнительно определить запрос свободного текста и настройки фасета.
- Free Text Query: Расширенный конструктор бесплатных текстовых запросов.
- Группировка: настройки, позволяющие группировать различные варианты одного и того же элемента.
- Сортировка: Настройки, позволяющие сортировать элементы.

Edit item DEFAULT



REFRESH SAVE

QUERY TEMPLATE QUERY PROPERTIES FREE TEXT QUERY GROUPING SORTING ADMINISTRATION

ESSENTIAL

Name	Indexed Type
DEFAULT	apparel-ukProductType

PROPERTIES

Show facets ?	Restrict fields in response ?
<input checked="" type="radio"/> True <input type="radio"/> False	<input type="radio"/> True <input checked="" type="radio"/> False
Enable Highlighting ?	Page Size
<input type="radio"/> True <input checked="" type="radio"/> False	0

Следующие шаги

Если вы хотите узнать больше о настройке шаблона DEFAULT, см.[Шаблон по умолчанию](#).

Следуйте инструкциям в[ПРЕДЛОЖЕНИЯ Шаблон](#) чтобы узнать больше о возможностях при создании и настройке собственного шаблона.

Шаблон по умолчанию

Шаблон DEFAULT — это шаблон по умолчанию, который идет вместе с конфигурацией поиска. Он изначально включает определенные свойства и настройки запроса, но его можно легко настроить.

Сценарий конфигурации

- На примере шаблона DEFAULT рассмотрим подробнее основные настройки:[Основные настройки](#)
- Определите свободный текстовый запрос для свойств запроса, чтобы улучшить процесс поиска:[Свойства запроса](#)
- Настройте параметры группировки, чтобы объединить варианты одного элемента и сделать просмотр результатов более быстрым и эффективным:[Настройка группировки вариантов](#).
- Узнайте, как выделить ключевые слова в результатах поиска, чтобы их было легче заметить:[Выделение ключевых слов](#)

Основные настройки

Основные настройки шаблона поискового запроса предоставляют вам следующую информацию:

- Имя:** имя вашего шаблона. Имя должно быть написано с заглавной буквы, иначе оно не будет соответствовать соглашению об именовании, и шаблон не будет работать.
- Индексированный тип:** индексированный тип, для которого создан шаблон.

Раздел «Свойства» содержит глобальные настройки шаблона.

- [Показать грани](#): Если установлено значение true, свойства запроса, выбранные в качестве фасетов, будут видны.
- [Ограничить поля в ответе](#): Если установлено значение true, будут отображаться только свойства запроса, заданные для включения в ответ. Подробности см. [Ограничить поля в ответе](#).
- [Включить подсветку](#): Если установлено значение true, свойства запроса с выбранной опцией подсветки будут выделены в результатах поиска. Подробнее см. [Выделение ключевых слов](#).

Свойства запроса

Свойства запроса являются свойствами индексированного типа. Они содержат собственные конфигурации поиска (например, настройки запроса свободного текста или настройки фасета) и также индексируются в процессе.

Edit item DEFAULT

QUERY TEMPLATE **QUERY PROPERTIES** FREE TEXT QUERY

QUERY PROPERTIES

Search Query Properties

itemtype : string	
code : string	
name : text	
description : text	
summary : text	
+ Create new Search Query Property	

После двойного щелчка по свойству вы увидите следующий набор вкладок:

- Основные настройки: позволяют решить, следует ли включать это свойство в ответ (ограничивая при этом количество полей) или использовать его для выделения.
- Свободный текстовый запрос: позволяет определять настройки, которые делают поиск более эффективным и точным:
 - **Бесплатный текстовый запрос**: облегчает поиск определенных терминов,
 - **Бесплатный текстовый нечеткий запрос**: облегчает поиск термина, даже если он написан с ошибкой,
 - **Бесплатный текстовый запрос с подстановочными знаками**: облегчает поиск термина, даже если вам известна только его первая, последняя или средняя часть,
 - **Запрос фразы бесплатного текста**: облегчает поиск всей фразы.

Подробную информацию об определении свободного текстового запроса см. [Определите детали свойства запроса](#).

- Настройки фасета: Здесь вы можете решить, должно ли свойство отображаться как фасет (если это возможно), какой тип фасета должен быть и какие поставщики должны к нему применяться. Подробнее см. [Настройки граний](#).

Определите детали свойства запроса

Если вы хотите сделать свой поиск эффективным и точным, и в то же время исключить возможные проблемы, возникающие из-за неправильно написанных и неполных терминов, вам следует правильно настроить параметры свободного текстового запроса.

Мастер поможет вам определить фразовый запрос, включить подстановочные знаки и настроить параметры нечеткости.

Добавить бесплатный текстовый запрос и фразовый запрос

Настройте свой свободный текст

Предпосылки

Для целей этого примера не определены настройки для каких-либо свойств. В результате, если вы зайдете в магазин и выполните поиск по определенному термину (например, belt), система покажет вам все доступные результаты, без каких-либо ограничений на то, что вы на самом деле искали.

Контекст

Чтобы показать, как работают запросы по свободному тексту и фразам, вам нужно определить настройки для свойства имени.

Процедура

- . Перейдите к шаблону DEFAULT. Если есть сомнения, следуйте инструкциям из [Доступ к шаблонам поисковых запросов](#) . Щелкните шаблон DEFAULT и перейдите к Свойства запроса . Выберите имя свойство и дважды щелкните его . В новом окне перейдите к Бесплатный текстовый запрос .

Это базовая настройка для каждого свойства. Если вы не включите бесплатный текстовый запрос для объекта недвижимости, он не будет включен в поиск.

- . Включить бесплатный текстовый запрос установив его в True . Не забудьте сохранить изменения.

Edit item name : text

REFRESH SAVE

SEARCH QUERY PROPERTY FREE TEXT QUERY FACET SETTINGS ADMINISTRATION

FREE TEXT QUERY

Free Text Query ? Free Text Query Min Term Length ?

True False 0

Free Text Query Boost ? 50

- . Если вы хотите, чтобы поисковый запрос имел определенную длину, вы можете задать ее в [Минимальная длина термина для бесплатного текстового запроса](#) . Это означает, что если вы введете 4, такой термин, как car, будет проигнорирован, поскольку он состоит всего из трех символов. Для целей этого примера оставим его равным 0.

- . Вы также можете повысить настройки для свободного текстового запроса, так что оценка увеличится на указанное значение. В нашем примере это 50.

Если вы вернетесь к магазину и введете в поиск слово «ремень», результаты будут ограничены товарами, в названии которых есть слово «ремень».

Включение запроса свободного текста приводит к ограниченным результатам, но вы наверняка хотите, чтобы они были ограничены еще больше. Попробуйте поискать belt green . Если вы попробуете это на витрине, результаты не изменятся. Чтобы иметь возможность искать фразы, вам нужно сначала включить запрос свободного текста.

. Перейдите в раздел «Свободный текстовый фразовый запрос» и установите значение «Истина».

FREE TEXT PHRASE QUERY

Free Text Phrase Query ?

True False

Free Text Phrase Query Slop ?

1

Free Text Phrase Query Boost ?

100

. Определите Query Slop, который представляет собой количество слов, разрешенных между терминами в запросе. 0 означает, что система будет искать точную фразу. Установите значение 1. Не забудьте сохранить изменения.

Вернитесь к витрине и найдите belt green. Термин, который вы ищете, должен быть первым. Вы можете заметить одно слово в середине фразы — именно для этого и нужен был запрос slop.

. Повысьте соответствие фразовому запросу по сравнению с соответствием свободному текстовому запросу, чтобы система сначала искала фразы.

Результаты

Вы можете искать отдельные термины и фразы, и результаты будут ограничены соответствием фразе.

(v) B2C Accelerator

belt green

SIGN IN / REGISTER

BRANDS STREETWEAR SNOW ACCESSORIES YOUTH

HOME / BELT GREEN

Shop by Stores

Find Stores

Shop by Price

- £0-£19.99 (6)
- £20-£49.99 (57)
- £50-£99.99 (10)
- £100-£199.99 (1)
- £400-£499.99 (5)

You searched for "belt green"

SORT BY: RELEVANCE

79 Products found

 Handshake Belt field green L £28.31	 Handshake Belt field green M £28.31	 Handshake Belt field green XL £28.31	 Granda Belt white L £40.46
---	---	--	--

Добавить подстановочный текстовый запрос и нечеткий запрос

Запрос фразы помогает вам искать целые фразы, но может случиться, что запись написана неправильно или неполна. Чтобы объяснить, как можно справиться с возможными проблемами, вы определите настройки для свойства запроса кода.

Процедура

. Дважды щелкните шаблон DEFAULT и перейдите к Свойства запроса.

. Выберите свойство запроса кода.

. Включите Free Text Query. Не забудьте сохранить изменения.

Попробуйте поискать определенный код, например 300604640. Предмет найден.

Активация Free Text Query приводит к тому, что элемент находится, но вам нужно знать точный код. Представьте себе ситуацию, когда вы помните только часть кода. Попробуйте ввести только часть вашего кода. Ничего не находится, или система возвращает случайные результаты. Чтобы избежать этого, вам нужно включить поддержку подстановочных знаков.

. Перейти к бесплатный текстовый запрос с подстановочными знаками раздел и установите его в

значение True.. Выберите тип запроса Wildcard.

У вас есть следующие варианты на выбор:

- **Прекс:** в начале будет применен подстановочный знак из нескольких символов.
- **Посткс:** в конце будет применен подстановочный знак из нескольких символов.
- **Прекс и посткс:** подстановочный знак из нескольких символов будет применен как в начале, так и в конце.

Выбирать Прекс. Сохраните изменения.

. Вы также можете установить минимальную длину термина подстановочных знаков, чтобы игнорировать подстановочные знаки с числом символов меньше указанного. Оставьте значение 0.

Результат должен быть следующим:

. Перейдите к витрине и введите 604640 .

Элемент найден. Попробуйте использовать разные варианты, чтобы проверить, работает ли поиск правильно.

Теперь вы можете искать элементы, даже если вы вводите только часть термина. Давайте рассмотрим случай, когда вы неправильно пишете запись, вводя 300603640 вместо 300604640. Опять ничего не найдено. Вот почему вам нужна поддержка нечеткости, чтобы система могла предложить вам правильный элемент.

. Перейдите в раздел «Свободный текстовый нечеткий запрос» и

включите его.. Установите уровень нечеткости на 1.

Введенное значение означает количество изменений между введенным вами текстом и совпадшим термином.

. Оставьте Минимальная длина нечеткого запроса свободного текста 0.

. Увеличьте запрос, введя 110 в Бесплатный текстовый нечеткий запрос Boost.

Результат должен быть следующим:

. Подойдите к витрине и введите код с одной неправильной цифрой, например 300603640

Результат будет содержать искомый вами элемент и подсказку о коде:

You searched for "300603640"

Did you mean: 300604640?

SORT BY:

RELEVANCE

11 Products found



Handshake Belt field green L

£28.31



Skull 80 SS youth white XL

£20.21



Результаты

Благодаря подстановочным знакам и нечетким запросам вы сможете легко найти нужный элемент, даже если введете часть термина или допустите в нем ошибку.

Настройки граней

Настройки фасета позволяют вам решить, должно ли определенное свойство запроса быть фасетом, а также добавить дополнительные настройки.

Вы можете определить настройки для каждого свойства запроса отдельно, используя [Настройки граней](#) раздел мастера. Если вы установите свойство запроса как фасет, оно будет отображаться на витрине.

Кроме того, вы можете использовать его в дальнейших конфигурациях, например, при настройке параметров поиска в адаптивном поиске.

Если свойство не может быть фасетом (например, из-за его типа), вы увидите предупреждающее сообщение.

Давайте посмотрим на настройки фасета для [категория](#) свойство.

Edit item category : string



REFRESH
SAVE

SEARCH QUERY PROPERTY
FREE TEXT QUERY
FACET SETTINGS
ADMINISTRATION

FACET SETTINGS

Facet

True False

Facet Value Display Name Provider

categoryFacetDisplayNameProvider

Facet Top Values Provider

defaultTopValuesProvider

Facet Type

Single select

Facet Value Sort Provider

Sort by displayed name

collectionName : text

allCategories : string

categoryPath : string

category : string

collection : string

brand : string

gender : string

price : double

style : string

swatchColors : string

Параметр	Описание
Фасет	Определяет, следует ли отображать свойство запроса как фасет.
Тип грани	<p>Тип фаски:</p> <ul style="list-style-type: none"> Рене: выбор одного значения грани Множественный выбор ИЛИ: выбор двух или более значений фасета, связанных условием ИЛИ Множественный выбор И: выбор двух или более значений фасета, связанных условием И
Поставщик отображаемого имени значения аспекта	Извлекает отображаемое имя для значения фасета.
Поставщик сортировки значений фасетов	Сортирует значения фасета по указанному свойству, такому как имя или количество продуктов. Сортировка применяется ко всему списку значений фасета и может отличаться при использовании поставщика верхних значений.
Facet — поставщик лучших ценностей	Предоставляет список значений фасета, которые имеют большее количество элементов, даже если фасеты имеют много значений. Оставшиеся значения свернуты.

Итоговая настройка фасета должна быть похожа на пример ниже:

Shop by Category

[Clothes](#) (499)

[T-Shirts](#) (350)

[Streetwear men](#) (268)

[T-Shirts men](#) (184)

[Snowwear women](#) (157)

[more categories...](#)

Настройка группировки вариантов

Группировка вариантов позволяет группировать похожие записи по определенному значению, например, по цвету.

Контекст

-Примечание

Вы можете реализовать конфигурацию группировки вариантов, используя [Модуль адаптивного поиска](#) также.

После настройки группировки вариантов, товар и все его варианты появятся на витрине. На следующем изображении вы можете увидеть результат группировки вариантов по цвету:



Her Logo SS Women heather peac...

£21.83



. Предварительный просмотр продукта.

. Варианты одного и того же продукта, сгруппированные по цветовому значению.

·Примечание

Если шаблоны включены, группировка должна быть установлена непосредственно в шаблоне. Если нет, группировка не работает должным образом. Если шаблоны не определены, применяется механизм отката. В этом случае конфигурация, определенная в [Индексированные типы поиска и навигации](#), используется. Вы можете определить его, нажав на индексированный тип и открыв [Группировкакладку](#).

Процедура

. Перейти к [Системный поиск и навигация](#) [Конфигурация поиска Solr Facet](#)  [Индексированные типы](#) .

. Выберите тип, для которого вы хотите определить группировку.

. В редакторе перейдите в [Конфигурация поиска индексированного типа](#) и дважды щелкните шаблон.

. Нажмите на [Группировкакладку](#) и настройте следующие элементы:

Edit item DEFAULT

The screenshot shows the 'Edit item DEFAULT' configuration interface. The 'GROUPING' tab is selected. There are four numbered callouts pointing to specific settings:

- Group results (True selected)
- Group property (baseProductCode)
- Group limit (999)
- Group facets (False selected)

Элемент	Имя	Описание	Пример
1.	Результаты групп	Указывает, будет ли система группировать и отображать несколько продуктов в одном результате поиска.	Истинный
2.	Имя поля группы	Имя поля, которое система использует для группировки результатов. Вы указываете атрибут, который используется для определения того, какие продукты следует группировать.	baseProductCode
3.	Групповой лимит	Определяет количество отображаемых вариантов.	100
4.	Групповые аспекты	Если включено, фасеты вычисляются для каждой группы. Если нет, они вычисляются для каждого документа.	ЛОЖЬ

. Чтобы определить значение, в соответствии с которым система позволяет клиентам выбирать варианты, в [местные.свойстваме](#), определить `commerceservices.variant.rollup.property.<baseSiteId>=<variant-attribute-qualifier>` свойство.

Например, если вы хотите сгруппировать варианты по цвету, введите свойство следующим образом:

`commerceservices.variant.rollup.property.apparel=style`.

Это свойство определяет критерии, которые используются для отображения вариантов под базовым изображением продукта. Это соответствует номеру 2 на предыдущем изображении. В то время как параметр поля группы определяет возвращаемые элементы, этот параметр определяет, какой параметр используется для отображения вариантов из возвращаемых результатов.

Выделение ключевых слов

Подсветка ключевых слов делает их легко заметными при поиске.

Чтобы включить подсветку ключевых слов, выполните следующие действия.

Процедура

- . Перейдите к шаблонам поисковых запросов в Backoffice Administration Cockpit. Если у вас возникли сомнения, следуйте инструкциям в [Доступ к шаблонам поисковых запросов](#).
- . Откройте шаблон.
- . Сначала нужно включить подсветку для шаблона. В разделе Шаблон запроса установите Включить подсветку в значение true.

Edit item DEFAULT

QUERY TEMPLATE **QUERY PROPERTIES** **FREE TEXT QUERY** **GROUPING** **SORTING** **ADMINISTRATION**

Name	Indexed Type
DEFAULT	apparel-ukProductType

PROPERTIES

Show facets	Restrict fields in response
<input checked="" type="radio"/> True <input type="radio"/> False	<input type="radio"/> True <input checked="" type="radio"/> False
Enable Highlighting	Page Size
<input checked="" type="radio"/> True <input type="radio"/> False	0

. Сохранять настройки.

. После включения глобальной подсветки выберите, какие свойства запроса следует использовать для подсветки. Перейти к Свойства запроса раздел.

Search Query Properties

itemtype : string	
code : string	
name : text	
description : text	
summary : text	
<input type="button" value="+ Create new Search Query Property"/>	

. Давайте выберем **имя** свойство, так что ключевые слова в имени элемента будут выделены. Дважды щелкните свойство, чтобы изменить его.

. В разделе «Свойства поискового запроса» установите **Использовать для выделения**на Истину.

Edit item name : text

itemtype : string
code : string
name : text
description : text
summary : text
manufacturerName : text
manufacturerAID : string
ean : string

SEARCH QUERY PROPERTY FREE TEXT QUERY FACET SETTINGS ADMINISTRATION

ESSENTIAL

Indexed Property	Priority
name	100
Include in Response <small>?</small>	<input checked="" type="radio"/> True <input type="radio"/> False
<input checked="" type="checkbox"/> Use for highlighting <small>?</small>	
<input checked="" type="radio"/> True <input type="radio"/> False	

. Сохранять настройки.

-Примечание

После закрытия текущего окна вы вернетесь к предыдущим окнам мастера. Не забудьте сохранить настройки по ходу дела, иначе изменения будут потеряны.

. Перейдите к витрине (например, <https://apparel-uk.local:9002/yacceleratorstorefront/en/>) и найдите элемент.

Ключевые слова различаются.

Beanie

Easy **Beanie** cyan Uni £16.16

The Brick **Beanie** husky Uni £14.54

Easy **Beanie** white Uni £16.16

Easy **Beanie** black Uni £16.16

£20-£49.99 (41)

Shop by Colour

BLACK (14)

Easy Beanie cyan Uni
£16.16

. Если вы перейдете в перспективу адаптивного поиска и просмотрите результаты поиска, вы также увидите выделенные ключевые слова.

Adaptive Search ▾

Search profile context: SimpleProfile - Simple Global ⓘ

Search in category: Global

Language English

Beanie| SEARCH

1 / 4 64 result(s) found

Default

Easy Beanie cyan Uni
£16.16

The Brick Beanie husky Uni
£14.54

ПРЕДЛОЖЕНИЯ Шаблон

Функция ПРЕДЛОЖЕНИЯ поддерживается изначально и позволяет вам настраивать параметры для предлагаемых элементов, которые появляются при вводе термина в поле поиска.

Контекст

Шаблон не предопределен. Вы определяете его в Backoffice Administration Cockpit. Используя шаблон, вы можете задать следующее:

- Запросите свойства, которые вы хотите увидеть, например, имя, код или описание элемента.
- Поиск фраз, подстановочные знаки и нечеткие запросы,
- Сортировка
- Выделение

Процедура

- . Создайте шаблон с помощью панели администрирования бэк-офиса.
- . Настройте шаблон, добавив свойства запроса и параметры произвольного текстового запроса.
- . Добавить сортировку.
- . Если вы хотите, чтобы результаты были предоставлены быстрее, вы можете ограничить поля в ответе.

Результаты

При поиске товаров в магазине шаблон возвращает в строке поиска предлагаемые товары, отсортированные по цене.

Создать шаблон поискового запроса

Вы создаете шаблон поискового запроса в административной панели Backoffice и задаете определенные параметры, которые позволят витрине распознавать его и предоставлять поисковые подсказки.

Процедура

- . В административной панели бэк-офиса перейдите к Система Поиск и навигация Конфигурация поиска Solr Facet Инdexированные типы.

Появится список всех конфигураций поиска.

- . Выберите конфигурацию поиска.

Откроется область редактора.

- . Выберите [Индексированный тип](#) вкладку.

- . В [Поиск конфигурации](#) раздел, нажмите [Создать новый шаблон поискового запроса](#).

Появится новое окно.

- . Введите имя **ПРЕДЛОЖЕНИЯ**.

-Примечание

Вы должны ввести имя, как указано выше. Если вы этого не сделаете, витрина будет использовать шаблон DEFAULT для поиска предложений. Для получения дополнительной информации см. раздел Naming Convention в: [Технические подробности шаблонов поисковых запросов](#).

- . Нажмите [Сделанный](#).

Шаблон создан. Теперь вы можете его настроить.

Настройте шаблон

Контекст

Для правильной работы шаблона необходимо добавить свойства запроса, которые вы хотите отобразить, и настроить параметры запроса свободного текста.

Результаты

Свойства запроса будут включены в поиск терминов. Продолжите настройку, определив параметры Free Text Query.

Добавить свойства запроса

Добавьте свойства запроса для вашего шаблона.

Процедура

- . Перейдите в Свойства запроса и нажмите [Создать новые свойства поискового запроса](#)

Create New Search Query Property

ESSENTIALS
Provide all mandatory fields

Indexed Property:

	...
--	-----

description

name +

code

itemtype

availableInStores

◀ ◀ 9 / 25 ▶ ▶

Нажмите [Сделанный](#) закончить.

. Таким же образом добавьте больше свойств, таких как: [код](#), [описание](#), [цена](#).

Вы можете увидеть их все в списке.

Запрос бесплатного текста Dene

Чтобы убедиться, что поиск выполняется правильно, вам необходимо задать настройки свободного текстового запроса.

Процедура

. В шаблоне перейдите к [Свойства запроса](#) раздел . Дважды

щелкните [имя](#) собственность . Перейти к [бесплатный текстовый](#)

[запрос](#) раздел.

. Включить [бесплатный текстовый запрос](#) установив его в значение True.

Включение свободного текстового запроса позволяет системе искать, находить и отображать термин.

. Включить [бесплатный текстовый запрос с подстановочными знаками](#) установив его на True, а Тип — на Prex и Postx.

Поскольку включен подстановочный запрос, термин будет найден, даже если вы введете только его часть.

. Сохраните конфигурацию. Повторите Шаг 5 и Шаг 6 еще раз для [код](#) [цена](#) [стоимость](#) [собственность](#). Помните, [Сохранять](#) Ваши изменения.

. Оставьте остальные настройки как есть. Если вы хотите узнать больше о настройках свободного текстового запроса, см. [Определите детали свойства запроса](#) . Подойдите к витрине и найдите [пояс](#). Предложения видны.

Сортировка

Сортировка позволяет вам расставить приоритеты в результатах, например, если вы хотите сделать некоторые элементы более заметными или упорядочить их.

Контекст

Чтобы отсортировать результаты по цене, выполните следующие действия.

Процедура

. Перейти к Сортировка вкладка.

На этой вкладке вы можете определить параметры сортировки для вашего шаблона ПРЕДЛОЖЕНИЯ.

Edit item SUGGESTIONS



REFRESH

SAVE

[QUERY TEMPLATE](#) [QUERY PROPERTIES](#) [FREE TEXT QUERY](#) [GROUPING](#) [SORTING](#) [ADMINISTRATION](#)

SORTING



Search Query Sorts

[+ Create new Search Query Sort](#)


. Нажмите на Создать новый поисковый запрос Сортировка кнопка.

В новом окне введите Цена Стоимость в поле и установите По возрастанию истине.

Create New Search Query Sort



ESSENTIALS

Provide all mandatory fields

Field Name:

Ascending:

 True

 False

CANCEL

DONE

Ваш список предложений теперь отсортирован по цене.

. Перейти к витрина и начать писать в поле поиска.

Вы можете увидеть список результатов, отсортированных по значению цены.

Результаты

Вы настроили шаблон ПРЕДЛОЖЕНИЯ, добавив свойства поискового запроса и определив параметры сортировки.

Ограничить поля в ответе

При настройке шаблона поискового запроса вы можете легко решить, какие свойства поискового запроса следует включить в результаты, отображаемые на витрине магазина.

Контекст

Для целей данного варианта использования предположим, что вы создали шаблон поискового запроса со следующими свойствами запроса:имя, цена Стоимость, коды img-515Wx515H чтобы убедиться, что изображения отображаются. Результаты, которые вы получите, написав camera в поле поиска, должны выглядеть примерно так, как показано на следующем рисунке:

Процедура

- . Перейдите к шаблону ПРЕДЛОЖЕНИЯ. Если вам нужна помощь, см.[Доступ к шаблонам поисковых запросов](#)
- . В Шаблон запроса вкладка редактора установлена Ограничить поля в ответе на true. При такой настройке только свойства с Включить в ответ проверено, будут включено в ответ.

The screenshot shows the SAP Fiori Query Template editor interface. At the top, there's a header with 'Edit item DEFAULT' and navigation tabs: 'Query Template' (selected), 'Query Properties', 'Free text query', 'Grouping', 'Sorting', and 'Administration'. Below the tabs, there are two main sections: 'Essential' and 'Properties'.

Essential Section:

- Name*: DEFAULT
- Indexed Type*: apparel-ukProductType ComposedTypeModel (87960992...)

Properties Section:

- Show facets*: True False
- Enable Highlighting*: True False
- Restrict fields in response*: True False (This field is highlighted with a yellow border.)
- Page Size: 0

Не забудьте нажать Сохранить, в противном случае ваши изменения будут потеряны.

- . Перейти к Свойства запроса вкладку и дважды щелкните свойство img-515Wx515H.

- . Набор Включить в ответ на false. Не забудьте сохранить изменения после редактирования.

Edit item img-515Wx515H : string

Refresh Save

Search Query Property Free Text Query Facet Settings Administration

Essential

Indexed Property* Priority*

img-515Wx515H 100

Include in Response* True False

Use for highlighting* True False

summary : text
itemtype : string
manufacturerAID : string
keywords : text
ean : string
img-515Wx515H : string
name : text

. Закройте окно свойств запроса, чтобы вернуться в главное окно. Сохраняешь изменения. . Перейдите к витрине и начните писать в поле поиска.

Изображения не включены в результаты.

Результаты

Вы можете легко сузить количество полей, отображаемых в ответе.

Технические подробности шаблонов поисковых запросов

Техническая информация о шаблонах поисковых запросов, такая как соглашение об именовании или использование резервного механизма.

В настоящее время вы можете выбрать один из двух предопределенных контекстов поиска для использования шаблонов поисковых запросов: DEFAULT и SUGGESTIONS. Вы можете найти тип перечисления для новых параметров контекста поискового запроса в `commerceservices-items.xml`:

```
<enumtype code="SearchQueryContext" generate="true" autocreate="true" dynamic="false">
    <description>Различные источники поискового запроса</description> <value
        код="DEFAULT" />
    <значение> код="ПРЕДЛОЖЕНИЯ" />
</enumtype>
```

Соглашение об именовании

Для определения того, какой шаблон запроса используется для какого контекста, используется определенное соглашение об именовании. Вот почему для контекста поиска и шаблона поискового запроса используется одно и то же имя. Например, чтобы использовать определенный шаблон запроса при поиске предложений, вы должны назвать его SUGGESTIONS, иначе он не будет работать. Если шаблон с определенным именем не найден, используется шаблон DEFAULT. Если шаблон DEFAULT отсутствует, используется резервный механизм.

Устаревший режим

Устаревший режим поиска по умолчанию установлен на значение false, и шаблоны поисковых запросов доступны для работы.

Поисковый запрос создается на основе конфигурации в `SolrSearchQueryTemplate` и `SolrSearchQueryProperty`. Если вы хотите работать в нетрадиционном режиме, но не хотите использовать шаблоны запросов, убедитесь, что для данного запроса не определены шаблоны индексированный Тип, чтобы мог работать механизм отката.

Механизм отката

Механизм отката был введен для сохранения совместимости для пользователей, работающих с предыдущими версиями. В таком случае конфигурация берется изSolrIndexedTypeиSolrIndexedPropertyтипы. Механизм отката используется, когда режим legacy выключен и шаблон не определен ни для одного контекста. Можно определить шаблон SUGGESTIONS и использовать его для предложений, но использовать механизм отката для контекста DEFAULT, то есть по отношению ко всем остальным поискам.

Дальнейшие шаги

Перейдите по ссылкам ниже, чтобы ознакомиться с моделью данных шаблонов поисковых запросов и просмотреть технические руководства:

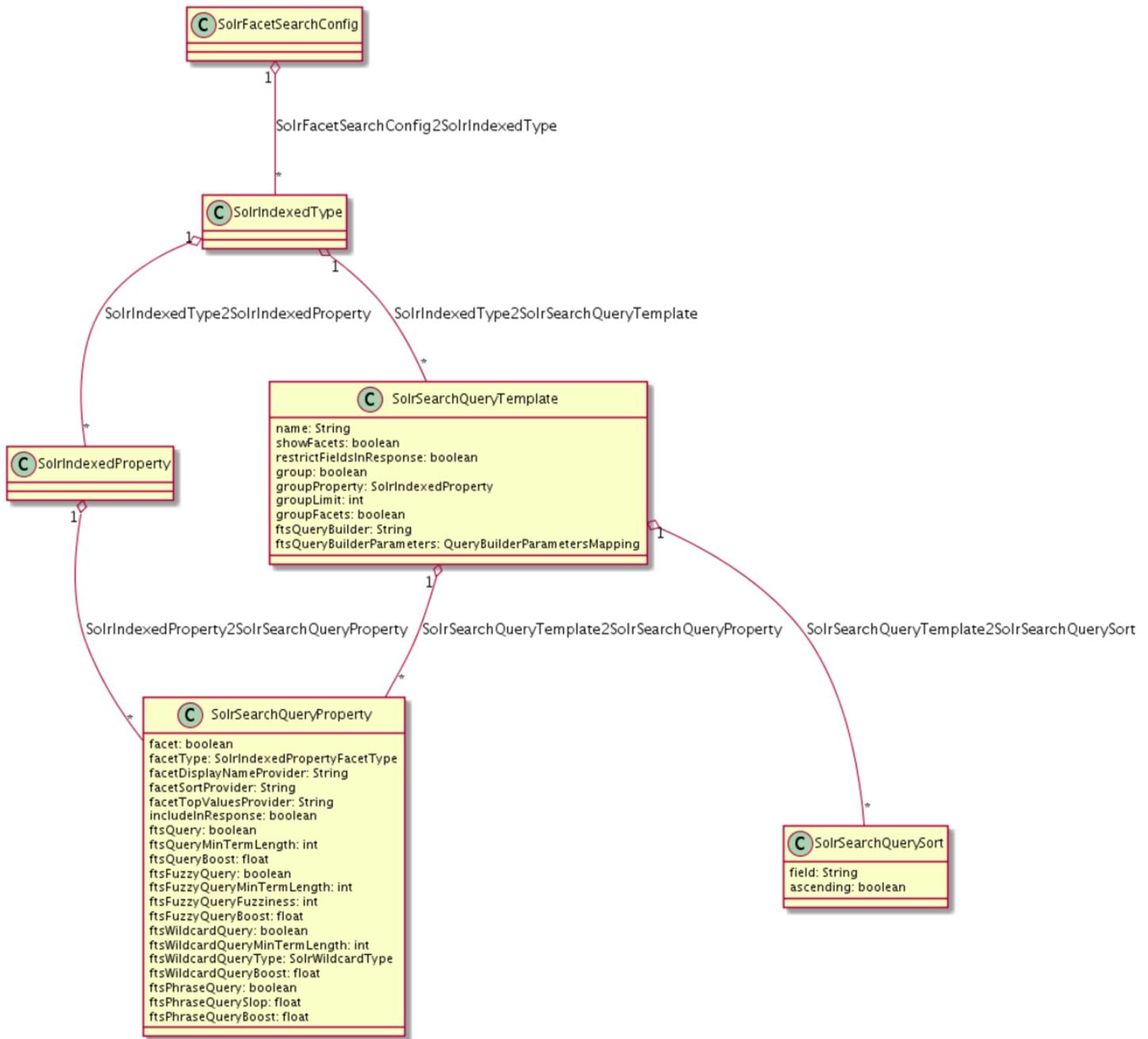
- [Модель данных шаблонов поисковых запросов](#)
- [Руководства для разработчиков шаблонов поисковых запросов](#)

Подробную информацию об управлении шаблонами поисковых запросов с помощью панели администрирования Backoffice см.[Шаблоны поисковых запросов в административной панели бэк-офиса](#).

Модель данных шаблонов поисковых запросов

Шаблоны поисковых запросов позволяют иметь более одной конфигурации для запросов свободного текстового поиска, которые можно использовать в различных контекстах, таких как предложения на витрине, основные результаты или панель администрирования бэк-офиса.

На следующем рисунке показана модель данных шаблона поискового запроса:



SolrSearchQueryTemplate

Атрибут	Имя	Описание
Имя	Имя	Имя, выбранное пользователем для шаблона.
показыватьФасеты	Показать грани	Включает или отключает фасеты в запросе или ответе.
restrictFieldsInResponse	Ограничить поля в ответе	Если отмечено, то только свойства свключитьВОтвет проверено, будет включено в ответ.
группа	Результаты группы	Если включено, результаты запроса будут сгруппированы.
groupProperty	Групповая собственность	Название свойства, по которому группируются результаты.
groupLimit	Групповой лимит	Количество результатов, возвращаемых для каждой группы.
группаFacets	Групповые аспекты	если включено, фасеты вычисляются для каждой группы, в противном случае — для каждого документа.
Размер страницы	Размер страницы	

Атрибут	Имя	Описание
ftsQueryBuilder	Бесплатный конструктор текстовых запросов	The идентификатор весенского компонента, который реализует FreeTextQueryBuilder интерфейс (например defaultFreeTextQueryBuilder, disMaxFreeTextQueryBuilder, multiFieldFreeTextQueryBuilder).
ftsQueryBuilderParameters	Параметры конструктора бесплатных текст-запросов	Параметры конструктора свободного текстового запроса.

SolrSearchQueryProperty

Атрибут	Имя	Описание
приоритет	Приоритет	
включитьВОтвет	Включить в ответ	Если выбрано, поле будет возвращено в ответе на запрос.
грань	Фасет	Проверьте, следует ли классифицировать недвижимость как фасетную
facetType	Тип грани	Тип грани (Уточнить, МножественныйВыборИ, МножественныйВыборИли)
facetDisplayNameProvider	Поставщик отображаемого имени значения аспекта	Поставщик отображаемых имен значений фасетов
facetSortProvider	Поставщик сортировки значений фасетов	Поставщик для сортировки значений фасетов
facetTopValuesProvider	Facet — поставщик лучших ценностей	Поставщик для выбора верхних значений граней
ftsQuery	Бесплатный текстовый запрос	Если включено, для этого свойства будет выполнен свободный текстовый запрос.
ftsQueryMinTermLength	Минимальная длина термина для бесплатного текстового запроса	Если длина термина меньше этого параметра, поисковый термин не будет учитываться.
ftsQueryBoost	Бесплатный текстовый запрос Boost	Мультипликативный коэффициент усиления для увеличения или уменьшения важности в запросе
ftsFuzzyQuery	Бесплатный текстовый нечеткий запрос	Если включено, для этого свойства будет выполнен нечеткий запрос с произвольным текстом.
ftsFuzzyQueryMinTermLength	Минимальная длина нечеткого запроса свободного текста	Если длина термина меньше этого параметра, поисковый термин не будет учитываться.
ftsFuzzyQueryНечеткость	Нечеткий запрос свободного текста Нечеткость	Требуемое сходство. Значение может быть 0, 1 или 2, будут сопоставлены только термины с более высоким сходством
ftsFuzzyQueryBoost	Бесплатный текстовый нечеткий запрос Boost	Мультипликативный коэффициент усиления для увеличения или уменьшения важности в запросе
ftsWildcardQuery	Бесплатный текстовый запрос с подстановочными знаками	Если включено, для этого свойства будет выполнен запрос с произвольным текстом и подстановочными знаками.
ftsWildcardQueryMinTermLength	Минимальная длина термина запроса с произвольным текстом	Если длина термина меньше этого параметра, поисковый термин не будет учитываться.
ftsWildcardQueryType	Тип запроса с произвольным текстом	Где будет применяться подстановочный знак из нескольких символов
ftsWildcardQueryBoost	Бесплатный текстовый запрос с подстановочными знаками Boost	Мультипликативный коэффициент усиления для увеличения или уменьшения важности в запросе
ftsPhraseQuery	Запрос фразы бесплатного текста	Если включено, для этого свойства будет выполнен запрос свободной текстовой фразы.
ftsPhraseQuerySlop	Бесплатный текстовый запрос фразы Slop	Количество других слов, разрешенных между словами в запросе, если ноль, то это точный поиск по фразе
ftsPhraseQueryBoost	Бесплатный текстовый фразовый запрос Boost	Мультипликативный коэффициент усиления для увеличения или уменьшения важности в запросе

SolrSearchQueryСортировка

Атрибут	Имя	Описание
поле	Имя поля	Поле, по которому выполняется сортировка. Например, добавление <code>ценаСтоимость</code> как имя приводит к тому, что список результатов сортируется по цене.
восходящий	По возрастанию	Порядок, в котором выполняется сортировка.

Руководства для разработчиков шаблонов поисковых запросов

Узнайте, как легко создавать, управлять и настраивать шаблоны поисковых запросов.

Внедрите шаблон поискового запроса по умолчанию

Узнайте, как настроить и использовать шаблон поискового запроса по умолчанию.

Процедура

- . Пример шаблона DEFAULT вы можете найти в разделе
[/apparelstore/resources/apparelstore/import/coredata/stores/apparel-uk/solr.impex](#).

Чтобы использовать его, обновите настройки следующим образом:

- `o7SolrIndexedTypekSolrSearchQueryTemplate`
 - `o7SolrIndexedPropertykSolrSearchQueryProperty`.

Подробную информацию о модели данных см. здесь: [Модель данных шаблонов поисковых запросов](#).

- . Для корректной работы шаблонов поисковых запросов необходимо отключить устаревший режим поиска.

Вы можете сделать это, изменив `/apparelstore/resources/apparelstore/import/coredata/stores/appareluk/solr.impex` следующим образом:

\$searchConfigName=apparel-ukPageSize

```
INSERT_UPDATE SolrSearchConfig;description[unique=true];legacyMode ;  
$searchConfigName:false
```

или с помощью Backoffice Administration Cockpit. Смотрите: Шаблоны поисковых запросов в административной панели бэк-офиса.

Шаблон DEFAULT теперь готов к использованию. Вы можете редактировать его с помощью Backoffice Administration Cockpit.

Реализуйте шаблон запроса предложений

Узнайте, как создать и управлять шаблоном запроса ПРЕДЛОЖЕНИЯ.

Шаблон запроса SUGGESTIONS можно настроить так же, как и шаблон DEFAULT.

Вы можете использовать приведенные ниже образцы данных для настройки шаблона SUGGESTIONS. Просто импортируйте файл `impexp` с помощью SAP Commerce Administration Console (Administration Console).

`$solrIndexedType=apparel-uk` Тип Продукта

Поисковый запрос Шаблон
вставить_обновление SolrSearchQueryTemplate;имя[уникальный=истина];индексированныйТип(идентификатор)[уникальный=истина];ftsQueryBuilder·ПРЕДЛОЖЕНИЯ:\$solrIndexedType:defaultFreeTextQueryBuilder

Свойства нефасетного поискового запроса

```
INSERT_UPDATE SolrSearchQueryProperty;indexedProperty(name, ;itemtype: solrIndexedType(идентификатор))[unique=true];поисковыйЗапрос
$solrIndexedType ;;;;;;;;:code:
$solrIndexedType :: :TRUE:90 :: :TRUE: :: :POSTFIX:45:2;
```

Если вы хотите узнать, как создать и настроить шаблон поискового запроса SUGGESTIONS с помощью панели администрирования Backoffice, см. [Шаблоны поисковых запросов в административной панели Бэк-офиса](#).

Реализуйте шаблон пользовательского поискового запроса

Контекст

Поскольку функция шаблона поискового запроса очень гибкая и настраиваемая, вы можете легко создать свой собственный шаблон. Например, вы собираетесь создать контекст поиска, который будет использоваться при просмотре категорий. Далее вы создадите шаблон поискового запроса CATEGORY с включенной функцией группировки продуктов. В результате вы увидите сгруппированные продукты на странице категорий и не сгруппированные на странице поиска (потому что для страницы поиска используется контекст DEFAULT).

Процедура

- . Создайте новый контекст, добавив элемент CATEGORY в SearchQueryContext тип перечисления в commerceServices-items.xml ле.

```
<enumtype code="SearchQueryContext" generate="false" autocreate="false" dynamic="false">
    <value code="КАТЕГОРИЯ" /> </
    enumtype>
```

- . Обновите систему с помощью `updatesystem` команды.
 - . Вам нужно позвонить `getProductSearchFacade().categorySearch` метод с вновь созданным контекстом.

В AbstractCategoryPageController.java класс заменить

```
searchPageData = getProductSearchFacade().categorySearch(categoryCode);
```

5

```
searchPageData = getProductSearchFacade().categorySearch(categoryCode, SearchQueryContext.CATEGORY);
```

Результат должен выглядеть так:

```
...
публичная пустота doSearch()
{
    показатьКатегорииТолько = ЛОЖЬ;
    если (searchQueryData.getValue() == null) {

        // Прямая ссылка на категорию без фильтрации
        searchPageData = getProductSearchFacade().categorySearch(categoryCode, SearchQueryContext.C if (categoryPage != null)
        {
            showCategoriesOnly = !categoryHasDefaultPage(categoryPage)
                && CollectionUtils.isNotEmpty(searchPageData.getSubCategories());
        }
    }
    еще
    {
        // У нас есть фильтрация поиска
        если (categoryPage == null || !categoryHasDefaultPage(categoryPage)) {

            // Загрузить страницу категории по умолчанию
        }
    }
}
```

```
    категорияСтраница = получитьСтраницуКатегорииПоУмолчанию();  
}  
  
окончательный SearchStateData searchState = new SearchStateData();  
searchState.setQuery(searchQueryData);  
  
окончательный PageableData pageableData = createPageableData(page, getSearchPageSize(), sortCode, s  
searchPageData = getProductSearchFacade().categorySearch(categoryCode, searchState, pageableData);  
}  
}  
...
```

. Создайте новый шаблон поискового запроса, который будет использоваться в контексте категории. Помните о соглашении об именовании — имя шаблона должно совпадать с именем контекста поиска.

Используйте SAP Commerce Administration Console (Консоль администрирования) для импорта следующего файла `imprex`:

```
;; всеПродвижения:$solrIndexedType ; ;МультиСелектор ;0 ;продвижениеFacetDisplayNameProvider
;; availableInStores:$solrIndexedType ;MultiSelectOr ;10000;apparelPointOfServiceFacetDisplayNameProvider
```

. Перейти к <https://apparel-uk.local:9002/yacceleratorstorefront/> для проверки группировки товаров при просмотре категорий.

Управление конфигурациями поиска

Создавайте и управляйте конфигурациями поиска, индексации и сервера с помощью панели администрирования Backoffice.

Вариант использования

Пользователь хочет определить конфигурации для поиска, индексации и сервера Solr. Кроме того, пользователь хотел бы определить ключевые слова, синонимы и стоп-слова для каждой конфигурации.

ФУНКЦИИ

Создание конфигураций поиска фасетов

Создавайте конфигурации фасетного поиска, включая конфигурации поиска и индексатора, а также такие данные, как ключевые слова, синонимы, стоп-слова и перенаправления ключевых слов.

Определение и настройка индексированных типов

Определите элементы, которые вы хотите индексировать, а также их свойства и запросы.

Создание диапазонов

Определите наборы диапазонов значений для заданных типов данных.

ЗАВИСИМОСТИ

Конкретных зависимостей для использования шаблонов поисковых запросов нет.

Создать конфигурацию поиска по фасетам

Используйте узел конфигурации поиска фасетов в дереве проводника, чтобы создать новую конфигурацию поиска фасетов и настроить ее в соответствии со своими потребностями.

Ознакомьтесь со следующими руководствами пользователя, чтобы успешно создавать и настраивать свои конфигурации.

. Сначала узнайте, как создать новую конфигурацию поиска по фасетам и познакомьтесь с расширенными параметрами конфигурации:[Добавить конфигурацию поиска](#).

. После того как вы создали новую конфигурацию или хотите настроить уже существующую, вы можете воспользоваться инструкциями, приведенными в следующих руководствах:

- Подробная информация о настройке резервного языка:[Настройка резервного языка для индексатора Solr](#).
- Подробности настройки перенаправлений ключевых слов:[Настройка перенаправлений ключевых слов](#).
- Подробности настройки синонимов:[Настроить синонимы](#). Подробности настройки стоп-слов:[Настроить стоп-слова](#).
- Подробная информация об определении поиска и настройке индексатора:[Конфигурация поиска и индексатора Dene](#).
- Подробности обновления индекса solr:[Обновление индекса Solr](#).

Добавить конфигурацию поиска

Узнайте, как создать новую конфигурацию поиска фасетов с помощью панели администрирования Backoffice.

Процедура

. Перейдите к Системный поиск и навигация поле поиска для ограничения результатов.

Конфигурация поиска Solr Facet

Конфигурации поиска фасетов

Вы также можете использовать

Список доступных конфигураций (если таковые имеются) отображается справа.

. Нажмите на кнопку в верхней части страницы.

Откроется новое окно.

. В разделе «Основы» заполните следующие поля

Create New Facet search configuration

ESSENTIALS Provide basic data **TYPES** Select indexed types **SESSION** Select attributes

Name: apparel-ukIndex

Solr server configuration: Default

Search configuration: Default

Indexer configuration: Default

CANCEL NEXT

- Имя: Уникальное имя для вашей конфигурации.
- Конфигурация сервера Solr: Конфигурация сервера Solr, которая будет использоваться. Доступные значения:[По умолчанию](#),[Локальное облако](#),[Локальный автономный](#).
- Конфигурация поиска: выбранная конфигурация поиска.[По умолчанию](#)Конфигурация доступна «из коробки».
- Конфигурация индексатора: Конфигурация индексатора, которая будет использоваться. Доступные значения:[По умолчанию](#),[Прямой](#),[Двухфазный](#).

-Примечание

Конфигурации по умолчанию подходят для большинства сред разработки. Для тестовой и производственной среды следует настроить конфигурации по умолчанию или создать новые.

Подробную информацию о создании конфигурации сервера Solr и конфигурации индексатора см. здесь:[Конфигурация поиска и индексатора Dene](#).

Нажмите [Следующий](#) чтобы продолжить.

. В разделе «Типы» выберите типы элементов, которые необходимо индексировать.

Create New Facet search configuration

X

ESSENTIALS

Provide basic data

TYPES

Select indexed types

SESSION

Select attributes

Indexed Properties: ?

apparel-ukProductType

...

BACK

CANCEL

NEXT

Нажмите Следующий чтобы продолжить.

. В разделе «Сессия» заполните следующие поля:

Create New Facet search configuration

X

ESSENTIALS

Provide basic data

TYPES

Select indexed types

SESSION

Select attributes

Catalog versions: ?

Apparel UK Content Catalog : Staged

...

Currencies: ?

Pound [GBP]

...

Languages: ?

English [en]

...

BACK

CANCEL

DONE

- Версии каталога
- Валюты
- Языки

. Нажмите Сделанный закончить

Ваша конфигурация теперь добавлена в список. Вы можете получить к ней доступ в любое время, чтобы внести изменения.

Расширенная настройка конфигурации поиска Solr Facet

Конфигурируемые предметы

Модель данных конфигурации поиска фасетов Solr включает в себя основной элемент `-SolrFacetSearchConfig` объект. Он представляет собой отдельную конфигурацию и остается связанным с другими элементами, которые при объединении позволяют определять различные аспекты конфигурации, такие как связанные языки, валюты или версии каталога.

Параметры настройки

Редактор конфигурации позволяет вам настроить конфигурацию для поиска фасетов Solr. Он содержит несколько вкладок с настраиваемыми элементами, подробно описанными в разделах ниже.

The screenshot shows the Solr Facet configuration interface. At the top, there's a table listing three indexes: apparel-deIndex, apparel-ukIndex, and electronicsIndex. The apparel-ukIndex row is highlighted in blue, indicating it is selected. Below the table, the configuration page for apparel-ukIndex is displayed. It includes tabs for PROPERTIES, INDEXED TYPES, KEYWORD REDIRECTS, SYNONYMS, STOPWORDS, CRON JOBS, and ADMINISTRATION. The PROPERTIES tab is active. Under PROPERTIES, there are fields for Name (apparel-ukIndex), Description (Apparel UK Solr Index), and Index name prefix (apparel-uk). Below this, there are sections for SEARCH AND INDEXER CONFIGURATION (with Solr server configuration, Indexer configuration, and Search configuration all set to Default) and SESSION ATTRIBUTES (with Catalog versions, Currencies, Languages, and Enabled language fallback mechanism). The Enabled language fallback mechanism has a True radio button selected.

Характеристики

Используя **Характеристики** на вкладке вы можете просмотреть свою конфигурацию, изменить ее и определить дополнительные параметры.

1. Свойства: основные свойства вашей конфигурации, такие как имя, описание и префикс имени индекса.

2. Конфигурация поиска и индексатора: В этом разделе вы можете определить и настроить следующие конфигурации:

Конфигурация сервера Solr

Редактор конфигурации сервера Solr позволяет вам настроить необходимые параметры конфигурации.

Конфигурация индексатора

Редактор конфигурации индексатора позволяет вам настраивать несколько параметров индексатора.

Поиск конфигурации

Конфигурация поиска позволяет вам определять параметры поиска, такие как количество результатов, отображаемых на странице, или сортировку.

Подробную информацию о том, как определить и настроить каждую из этих конфигураций, см. здесь:[Конфигурация поиска и индексатора Dene](#)

3. Атрибуты сеанса: Дополнительные настройки для вашей конфигурации, включая: версии каталога, языки или валюты. Здесь вы также можете включить механизм отката:[Управление диапазонами Solr](#).

Индексированные типы

Вы можете использовать вкладку Индексированные типы, чтобы добавлять или изменять типы элементов для индексации. Выберите предопределенный индексированный тип или создайте новый тип, щелкнув правой кнопкой мыши таблицу Типы элементов и выбрав соответствующее действие из контекстного меню. Элемент, соответствующий индексированному типу, имеет свой собственный узел дерева проводника в Backoffice Administration Cockpit. Он называется[Индексированные типы](#) и присутствует в каталоге Facet Search.

Перенаправления ключевых слов

Вкладка перенаправления ключевых слов позволяет пользователям определять слова, используемые для перенаправления их на определенные результаты поиска, такие как URL-адреса или элементы. Эти перенаправления ключевых слов определяются для конкретной конфигурации поиска по фасету и могут быть созданы для нескольких языков отдельно.

Подробную информацию об определении перенаправлений ключевых слов см. здесь:[Настройка перенаправлений ключевых слов](#)

Синонимы

Вкладка «Синонимы» позволяет пользователям создавать дополнительные слова для использования в качестве параметра поиска. Эти синонимы определены для конкретной конфигурации поиска по фасету и могут быть созданы для нескольких языков отдельно.

Подробную информацию об определении синонимов см.:[Настройте синонимы](#).

Стоп-слова

Для индексов с большим количеством общих слов, таких как the, a, эти слова делают индекс большим и замедляют фразовые запросы. Простым решением является фильтрация общих слов из полей, где они часто появляются. Вкладка Stopwords позволяет пользователям добавлять слова, которые будут использоваться в качестве параметров фильтрации. Эти стоп-слова определены для конкретной конфигурации поиска фасетов и могут быть созданы для нескольких языков отдельно.

Подробную информацию об определении стоп-слов см. здесь:[Настройте стоп-слова](#).

Задания Cron

Cronjobs, назначенные для определенной конфигурации поиска. Пример настройки cronjob см.:[Обновление индекса Solr с помощью Cronjob](#).

Конфигурация поиска и индексатора Dene

Узнайте, как использовать панель администрирования Backoffice для легкого определения и изменения конфигурации Solr.

Контекст

Backoffice Administration Cockpit дает вам возможность доступа и редактирования конфигурации сервера Solr и конфигурации Indexer. Каждый из этих элементов является отдельным типом платформы, поэтому у них есть свои собственные редакторы. У вас есть доступ к следующим конфигурациям:

- Конфигурация сервера Solr: позволяет вам определять аспекты, связанные с сервером, такие как конфигурация автономного сервера Solr, конфигурации сервера SolrCloud или конфигурации поискового клиента. Из коробки мы предоставляем конфигурации по умолчанию, локальное облако и локальные автономные конфигурации.
- Конфигурация индексатора: позволяет вам установить конфигурацию индексатора solr. Из коробки мы предоставляем конфигурации по умолчанию, прямые и двухфазные конфигурации.
- Конфигурация поиска: позволяет вам настроить общие параметры поиска, такие как размер страницы результатов или порядок сортировки по умолчанию. Конфигурация по умолчанию доступна из коробки.

-Примечание

Если свойства конфигураций заданы в панели администрирования Backoffice, то значение берется из свойства, в противном случае — из базы данных.

Процедура

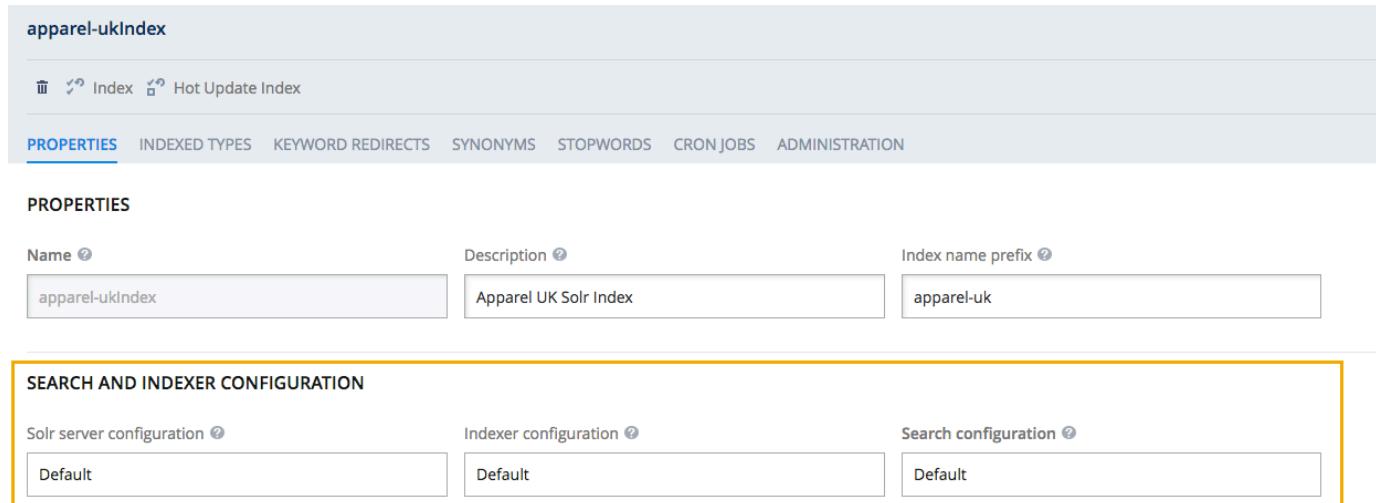
. Перейдите к [Системный поиск и навигация](#) ➔ Конфигурация поиска Solr Facet ➔ Конфигурации поиска фасетов .

Появится список доступных конфигураций (если такие имеются).

. Выберите существующую конфигурацию. Если вы хотите создать новую, следуйте инструкциям в: [Добавить конфигурацию поиска](#).

Откроется редактор, в котором можно определить дополнительные настройки.

. Под [Характеристики](#) найдете [Конфигурация поиска и индексатора](#) раздел.



apparel-ukIndex

Index Hot Update Index

PROPERTIES INDEXED TYPES KEYWORD REDIRECTS SYNONYMS STOPWORDS CRON JOBS ADMINISTRATION

PROPERTIES

Name	Description	Index name prefix
apparel-ukIndex	Apparel UK Solr Index	apparel-uk

SEARCH AND INDEXER CONFIGURATION

Solr server configuration	Indexer configuration	Search configuration
Default	Default	Default

Настройте сервер Solr

Процедура

. Используйте раскрывающееся меню для выбора [Создать новую конфигурацию сервера Solr](#) вariant.

SEARCH AND INDEXER CONFIGURATION

Solr server configuration ?

- Local Cloud
- Local Standalone
- apparel-deSolrServerConfig
- apparel-ukSolrServerConfig
- electronicsSolrServerConfig

+ Create new Solr server configuration

. Откроется окно конфигурации сервера Solr.

Create New Solr server configuration

SOLR SERVER CONFIGURATION

Identifier: ?

Mode: ?

Endpoint URLs: ?

CANCEL **DONE**

. Заполните поля, чтобы определить основные настройки вашей конфигурации.

Имя поля	Описание
Идентификатор	Уникальный идентификатор для конфигурации вашего сервера Solr.
Режим	<p>Вы можете выбрать режим работы из следующих значений:</p> <ul style="list-style-type: none"> <input type="radio"/> автономный <input type="radio"/> встроенный <input type="radio"/> облако

Имя поля	Описание
	<ul style="list-style-type: none"> o xml-экспорт
URL-адреса конечных точек	URL-адреса экземпляров Solr в этой конфигурации. Для обеспечения безопасной связи с Solr используйте протокол https. Подробнее о безопасной связи см. Безопасная связь с использованием SSL .

. Нажмите [DoneSaveSocketTimeoutException](#), закончить.

Вы должны найти свою конфигурацию в списке. Выберите ее, чтобы сделать активной. Не забудьте нажать, чтобы сохранить изменения.

. Чтобы добавить информацию о конфигурациях, щелкните конфигурацию, чтобы открыть редактор.

[Server configuration](#) [Client configuration](#) [Administration](#)

General configuration

Identifier* ⓘ

Mode* ⓘ

Endpoint URLs ⓘ

Solr query method

Standalone mode configuration

Use Master Node(s) exclusively for indexing*

True False

Cloud mode configuration

Number of shards

Replication factor

Automatically add replicas*

True False

Используйте вкладку «Сервер» для определения дополнительных настроек сервера Solr.

. Для метода запроса Solr, *Получатели* и *Почта* обеспечивают ту же функциональность с почти той же производительностью. Однако, *Почта* поддерживает больший размер заголовка.

. Для конфигурации Standalone Mode укажите, следует ли использовать ведущий узел(ы) исключительно для индексации. Если вы используете автономный Solr с режимом master-client, его можно установить в True.

. Для конфигурации облачного режима вы можете установить следующие значения:

- o Количество шардов: Когда коллекция слишком велика для одного узла, ее можно разбить и хранить в разделах, создав несколько шардов. Шард — это логическое разделение коллекции, содержащее подмножество документов из коллекции, так что каждый документ в коллекции содержится ровно в одном шарде. Вы можете оставить это значение равным 1, если только не требуется обрабатывать огромный объем данных.

-Примечание

Эта конфигурация действительна для всех индексированных типов, однако ее можно переопределить и определить для отдельного индексированного типа.

Подробности см. в: [Дополнительные настройки для индексированных типов](#).

- Фактор репликации: Фактор репликации определяет количество реплик, которые у вас будут для каждого индекса. Рекомендуется поддерживать это значение больше 1, чтобы избежать единой точки отказа.
 - Автоматически добавлять реплики: Используйте эту опцию, чтобы разрешить Solr автоматически создавать реплики при высокой нагрузке. Вы можете установить значение False, поскольку автоматически созданные реплики могут помешать выполнению полной индексной работы.
- . На вкладке конфигурации клиента вы можете добавлять и редактировать настройки для поискового клиента и клиента индексатора.

- Тайм-аут сокета: Тайм-аут происходит, когда поток данных прерывается. Если вы видите `SocketTimeoutException` в журнале приложений, вам может потребоваться обновить значение здесь. Однако, пожалуйста, имейте в виду, что исключение может быть следствием других исключений, и увеличение продолжительности тайм-аута может не помочь в таких случаях.

Доступные поля одинаковы для обоих клиентов и включают настройки тайм-аута и максимальное количество выполняемых подключений. Кроме того, эта вкладка позволяет вам настроить имя пользователя и пароль для клиента Solr и индексирующего клиента, чтобы обеспечить авторизованное и аутентифицированное подключение. Подробнее об авторизации см.[Поддержка аутентификации и авторизации для Solr](#).

Настройте индексатор

Процедура

- Используйте раскрывающееся меню для выбора **Создать новую конфигурацию индексатора** option. Выпадающее меню похоже на меню для конфигурации сервера Solr.
- Появится всплывающее окно.
- Заполните поля, чтобы завершить настройку.

Имя поля	Описание
Идентификатор	Уникальный идентификатор для конфигурации вашего индексатора.

Имя поля	Описание
Размер партии	Размер индексируемого пакета. Рекомендуется не превышать 1000. Ссылка 2446013  .
Путь экспорта	Определяет каталог, в который индексатор должен записывать индексируемые документы в режиме XML_EXPORT.
Количество темы	Количество потоков, используемых для индексации, обычно в два раза превышает количество имеющихся у вас ядер.
Режим индексатора	У вас есть следующие варианты на выбор: <ul style="list-style-type: none"> ○ ПРЯМО: Для полной индексной операции текущий индекс удаляется и заменяется новым. В течение определенного периода времени индекс не полностью доступен для запроса. ○ TWO_PHASE: Текущий индекс остается активным до тех пор, пока новый индекс не будет построен в новом ядре Solr. После этого новый индекс заменяет старый индекс и может быть запрошен пользователями без каких-либо периодов отключения. Внешним фронтенд-приложениям не нужно знать, какое ядро индексатора Solr активно в текущий момент.
Режим фиксации	У вас есть следующие варианты на выбор: <ul style="list-style-type: none"> ○ НИКОГДА: операция фиксации не будет выполнена явно ○ AFTER_INDEX: жесткая фиксация после индексации; это значение по умолчанию. Рекомендуется, когда размер пакета небольшой. ○ AFTER_BATCH: жесткая фиксация после каждой партии ○ СМЕШАННЫЙ: мягкая фиксация после каждой партии, жесткая фиксация после индексации
Режим оптимизации	У вас есть следующие варианты на выбор: <ul style="list-style-type: none"> ○ НИКОГДА: операция оптимизации не будет выполнена явно, это значение по умолчанию ○ AFTER_INDEX: оптимизировать после индексации <p>Рекомендуется установить значение NEVER. Этот параметр влияет на производительность индексации, особенно при большом объеме данных. Если вы выбираете AFTER_INDEX, пожалуйста, измените настройку SocketTimeOut на странице конфигурации сервера Solr.</p>
Распределенный Индексация	В распределенном механизме индексации элементы разбиваются на пакеты и индексируются параллельно. Чтобы узнать больше об индексации, см. Процесс индексации .
Игнорировать ошибки	Если игнорировать ошибки включен, это означает, что некоторые ошибки (связанные с данными) игнорируются во время индексации.
Макс. количество повторов	Максимальное количество повторных попыток в целом. Это значение можно установить на 3 или 4, чтобы обойти некоторые проблемы.
Максимальное количество повторных попыток	Максимальное количество повторных попыток на партию.

. Нажмите [Сделанный](#) и [закончить](#).

Вы должны найти свою конфигурацию в списке. Выберите ее, чтобы сделать активной. Не забудьте нажать [Сохранять](#), чтобы сохранить изменения.

Результаты

Вы успешно настроили параметры для сервера Solr и индексатора. Вы можете изменить их в любое время, перейдя на вкладку конфигурации Solr и выбрав конфигурацию, которую хотите изменить.

Настроить поиск

Процедура

. Используйте раскрывающееся меню для выбора [Создать новую конфигурацию поиска](#) option. Выпадающее меню выглядит аналогично представленным выше.

. Заполните поля.

Имя поля	Описание
Размер страницы результата	Определяет количество результатов на странице.
Порядок сортировки по умолчанию	Порядок сортировки по умолчанию.
Описание	Описание конфигурации вашего поиска

Настройка резервного языка для индексатора Solr

Инструкции по назначению резервного языка для конфигурации поиска фасетов Solr с помощью панели администрирования Backoffice.

Контекст

Механизм резервного языка отвечает за предоставление содержимого для локализованных атрибутов, если для текущих настроек локализации не указано значение. Продукты можно искать на основе их идентификатора, определенного специально для каждого языка. Однако в случае, если идентификатор не имеет значения для определенного языка, вы можете настроить резервный язык, который будет использоваться для поиска индексированного элемента.

Процедура

- . Перейдите к [Языки интернационализации](#)узел в дереве проводника.

Отобразится список уже настроенных языков (если таковые имеются).

- . Выберите язык, для которого вы хотите определить резервный язык.

Откроется редактор.

- . В редакторе перейдите к [Запасные языки](#)раздел. Используйте всплывающее окно или окно поиска ссылок, чтобы добавить резервный язык. Вы также можете создать новый, нажав [Создать новый язык](#).

Spanish [es]

ATTRIBUTES ADMINISTRATION

ESSENTIAL

ISO Code	Name
es	Spanish

COMMON

Active

True False

FALLBACK LANGUAGES

Fallback languages [?](#)

Latin-American Spanish [es_CO]	...

В списке появляется новый язык. Этот язык может быть использован конфигурацией поиска фасетов Solr в качестве резервного языка в случае, если определенный индексированный термин не существует для продукта на основном языке.

-Примечание

Механизм отката необходимо включать отдельно для каждой конфигурации поиска фасетов Solr.

. Перейдите к Системный поиск и навигация Solr Конфигурация поиска по фасетам Конфигурации поиска по фасетам

Появится список доступных конфигураций.

. Щелкните выбранную конфигурацию, чтобы открыть редактор.

. В редакторе перейдите к Характеристики вкладка. Внизу страницы установите Включен механизм резервного языка на Истину.

PROPERTIES INDEXED TYPES KEYWORD REDIRECTS SYNONYMS STOPWORDS CRON JOBS ADMINISTRATION

PROPERTIES

Name	Description	Index name prefix
apparel-ukIndex	Apparel UK Solr Index	apparel-uk

SEARCH AND INDEXER CONFIGURATION

Solr server configuration	Indexer configuration	Search configuration
Default	Default	Default

SESSION ATTRIBUTES

Catalog versions	Currencies	Languages	Enabled language fallback mechanism
Apparel Product Catalog : Online	Pound [GBP]	English [en]	<input checked="" type="radio"/> True <input type="radio"/> False
Apparel Product Catalog : Staged	
...	

Он включает резервные языковые механизмы для текущей конфигурации поиска фасетов Solr. Когда закончите, нажмите Сохранять чтобы сохранить изменения.

. После настройки резервных языков и включения механизма резервного языка для определенной конфигурации поиска фасетов вам необходимо обновить индекс, чтобы применить эти изменения. Подробности обновления конфигурации solr см.: [Обновление индекса Solr](#).

Настройка перенаправлений ключевых слов

Вы можете использовать функцию перенаправления ключевых слов, чтобы определить слова, которые будут перенаправлять ваших пользователей к определенным результатам поиска, таким как целевые URL-адреса или элементы.

Контекст

Чтобы создать новое перенаправление по ключевым словам, выполните следующие действия.

Процедура

. Перейдите к Системный поиск и навигация Solr Конфигурация поиска по фасетам Конфигурации поиска по фасетам

Список доступных конфигураций (если такие имеются) вы можете увидеть справа.

. Щелкните конфигурацию, в которую вы хотите добавить перенаправление по ключевому слову.

Откроется редактор.

. Перейти к Перенаправления ключевых слов вкладка.

Отображается список перенаправлений ключевых слов (если какие-либо из них уже добавлены).

apparel-ukIndex

Index Hot Update Index

PROPERTIES INDEXED TYPES **KEYWORD REDIRECTS** SYNONYMS STOPWORDS CRON JOBS ADMINISTRATION

GENERAL

Keyword redirects

EXACT basket	...
EXACT cart	...
EXACT accelerator	...
EXACT hymacc	...
EXACT help	...
...	

. Используйте более кнопку для поиска нового слова.

Открывается окно поиска. Используйте его для выбора ключевых слов, которые вы хотите добавить. Вы можете выбрать несколько за раз.

. Если вы хотите создать новое перенаправление ключевого слова, используйте раскрывающееся меню и нажмите кнопку [Создать новое перенаправление по ключевому слову](#) кнопка.

apparel-ukIndex

REGEX (Surfen | Surfing | Strand)

REGEX (Ski | Skifahren)

STARTS_WITH Lieferung

EXACT Hilfe

EXACT Warenkorb

+ Create new Keyword redirect

1 / 10

...

. Заполните поля в новом окне.



Create New Keyword redirect

ESSENTIALS

Provide all mandatory fields

Match type:

EXACT



Keyword:

basket

Ignore case:



Redirect:

/cart

Language:

English [en]

CANCEL

DONE

Имя поля	Описание
Тип соответствия	Устанавливает параметр соответствия, определяющий, как будет обрабатываться перенаправление ключевого слова. Доступные параметры: EXACT, STARTS_WITH, ENDS_WITH, CONTAINS, REGEX.
Ключевое слово	Ключевое слово, для которого вы создаете перенаправление.
Игнорировать регистр	Определяет, чувствительно ли ключевое слово к регистру.
Перенаправить	Элемент, на который перенаправляется ваше ключевое слово. Это может быть, например, продукт, веб-страница или каталог.
Язык	Язык вашего ключевого слова.

Нажмите [Сделанный](#) до конца. Ваше новое ключевое слово теперь доступно для использования.

. Если вы хотите удалить ключевое слово из списка, просто наведите на него курсор, пока не появится [Появляется](#) кнопка. Нажмите кнопку.

Ключевое слово удалено из списка.

. Если вы хотите удалить ключевое слово навсегда, войдите в режим редактора, щелкнув по записи ключевого слова.

Появляется редактор.

. Используйте значок корзины в левом верхнем углу редактора, чтобы удалить перенаправление ключевого слова.

Ключевые слова были удалены.

Настроить синонимы

Синонимы в фасетном поиске Solr позволяют определить дополнительное слово, которое можно использовать в качестве параметра поиска.

Контекст

Использование синонимов не означает, что вы получите те же результаты при поиске определенных элементов. Если у вас есть синоним, скажем, от А до В, вы, скорее всего, получите разные результаты при поиске А и при поиске В. Это связано с тем, что синонимы нельзя использовать для некоторых типов запросов (например, подстановочных и нечетких запросов).

Solr предоставляет вам список синонимов по умолчанию, а также вы можете определить свои собственные синонимы в панели администрирования бэк-офиса.

-Примечание

Синонимы по умолчанию также используются в функциях автозаполнения и проверки орфографии.

Процедура

- . Перейдите к [Системный поиск и навигация Конфигурация поиска Solr Facet](#) Конфигурации поиска фасетов в дереве проводника.

Список доступных конфигураций (если таковые имеются) отображается справа.

- . Выберите конфигурацию и щелкните по ней.

Появляется редактор.

- . Выберите [Синонимы](#) вкладку.

hats -> caps
shoes,sneaker,sneakers,trainer -> shoe
shades,glasses -> sunglasses
...

- . На этом этапе вы можете выбрать уже созданные синонимы или добавить новые.

. Для того, чтобы выбрать уже существующий синоним, откройте [Поиск ссылок](#)коно. Вы также можете начать вводить имя синонима, чтобы найти и добавить его.

- . Выберите из списка синонимы, которые вы хотите добавить в свою конфигурацию, и нажмите [Выбирать](#).

Выбранные синонимы отображаются в списке.

- . Чтобы добавить новый синоним, нажмите кнопку [Создать новую конфигурацию синонимов](#)кнопка.

apparel-ukIndex

Index Hot Update Index

PROPERTIES INDEXED TYPES KEYWORD REDIRECTS **SYNONYMS** STOPWORDS CRON JOBS ADMINISTRATION

SYNONYMS

EXPORT TO SOLR

Synonyms configuration 

hats -> caps	...
shoes,sneaker,sneakers,trainer -> shoe	...
shades,glasses -> sunglasses	...
Videokamera -> Camcorder	...
af -> Autofokus	...
deals -> special offers	...
small,mini -> compact	...
video camera -> camcorder	...
+ Create new Synonyms configuration	▼

. В новом окне заполните поля.

Create New Synonyms configuration 

ESSENTIALS
Provide all mandatory fields

From:
Hats

To:
Caps

Language:
English [en]

CANCEL **DONE**

Имя поля	Описание
От	Слово, которое вы используете при поиске определенного предмета, например, «шляпа».

Имя поля	Описание
K	Синоним искомого термина, например «сар».
Язык	<p>Язык синонима.</p> <p>-Примечание</p> <p>Мастер позволяет выбрать один язык за раз, поэтому для создания синонимов на нескольких языках запустите мастер еще раз.</p>

. Нажмите [Сделанный](#) и закончить.

Синоним виден в списке.

. Запомнить щелчок [Сохранять](#), в противном случае вы потеряете внесенные изменения. .

Экспортируйте синонимы на сервер solr с помощью [Экспорт в Solr](#) кнопка.

. Если экспорт был выполнен успешно, в верхней части страницы появится подтверждение. . Повторно

проиндексируйте элементы, используя тип операции индексации FULL.

. Если вы хотите удалить синоним из списка, наведите на него курсор и используйте [Х](#) кнопку, чтобы удалить его.

Синоним больше не присутствует в списке. Нажмите [Сохранять](#) чтобы сохранить изменения.

The screenshot shows the SAP interface for managing synonyms. At the top, there is a navigation bar with tabs: PROPERTIES, INDEXED TYPES, KEYWORD REDIRECTS, **SYNONYMS**, STOPWORDS, CRON JOBS, and ADMINISTRATION. Below the tabs, the word "SYNONYMS" is displayed in bold. A large blue button labeled "EXPORT TO SOLR" is visible. The main area contains a table titled "Synonyms configuration". The table lists four entries:

- hats -> caps
- shoes,sneaker,sneakers,trainer -> shoe
- shades,glasses -> sunglasses
- Hats -> Caps

To the right of each entry is a small red "X" icon. A yellow arrow points from the text "Если вы хотите удалить синоним навсегда, щелкните по записи." to the "X" icon in the first row. At the bottom of the table, there is a "..." button.

. Если вы хотите удалить синоним навсегда, щелкните по записи.

а. Появляется редактор. Используйте мусорное ведро значок в левом верхнем углу, чтобы удалить синоним.

Edit item hats -> caps

SYNONYM ADMINISTRATION

SYNONYM

From hats **To** caps

Language English [en]

6. Вас просят подтвердить свое решение. Нажмите [Да](#) для подтверждения.

. Не забудьте снова экспортить список на сервер Solr, чтобы сообщить ему об изменениях.

Настроить СТОП-СЛОВА

Поисковый сервер Solr поставляется со списком стоп-слов по умолчанию. Однако вы можете использовать Backoffice Administration Cockpit для создания дополнительного списка общих слов, таких как a, the и подобные, которые не должны учитываться механизмами индексации.

Процедура

. Перейдите к Системный поиск и навигация Конфигурация поиска Solr Facet Конфигурации поиска фасетов в дереве проводника.

Список доступных конфигураций (если таковые имеются) отображается справа.

. Выберите конфигурацию и щелкните по ней.

Появляется редактор.

. Выберите Стоп-слова вкладка.

apparel-ukIndex

Index Hot Update Index

PROPERTIES INDEXED TYPES KEYWORD REDIRECTS SYNONYMS **STOPWORDS** CRON JOBS ADMINISTRATION

STOPWORDS

EXPORT TO SOLR

Stopwords

a	...
the	
	...

. На этом этапе вы можете выбрать уже созданные стоп-слова или добавить новые.

. Чтобы выбрать уже существующее стоп-слово, используйте значок увеличительного стекла, чтобы открыть окно поиска ссылок. Вы также можете начать вводить имя стоп-слова, чтобы найти его и добавить.

. Чтобы добавить новое стоп-слово, нажмите кнопку **Создать новое стоп-слово** на кнопке.

Index Hot Update Index

PROPERTIES INDEXED TYPES KEYWORD REDIRECTS SYNONYMS **STOPWORDS**

STOPWORDS

EXPORT TO SOLR

Stopwords

a	...
the	
	...
the	
a	
+ Create new Stopword	

. В новом окне заполните поля.

Create New Stopword



ESSENTIALS

Provide all mandatory fields

Stopword:

Language:

CANCEL

DONE

Имя поля	Описание
Стоп-слово	Стоп-слово по вашему выбору.
Язык	<p>Язык стоп-слова.</p> <p>-Примечание</p> <p>Мастер позволяет выбрать один язык за раз, поэтому для создания стоп-слов на нескольких языках запустите мастер еще раз.</p>

. Нажмите [Сделанный](#) закончить.

Стоп-слово отображается в списке.

. Запомнить щелчок [Сохранять](#), в противном случае вы потеряете внесенные изменения..

Экспортируйте стоп-слова на сервер solr с помощью [Экспорт в Solr](#) кнопка.

. Если экспорт был выполнен успешно, в верхней части страницы появится подтверждение.. Если вы хотите удалить стоп-слово навсегда, щелкните запись.

а. Появляется редактор. Используйте [мусорное ведро](#) значок в левом верхнем углу, чтобы удалить стоп-слово.

Edit item die



REFRESH

SAVE

[STOPWORD](#) ADMINISTRATION

STOPWORD



Stopword

Language

6. Вам будет предложено подтвердить свое решение. Нажмите [Да](#) для подтверждения.

. Если вы хотите удалить стоп-слово из списка, наведите на него курсор и используйте кнопку, чтобы удалить его.

STOPWORDS

EXPORT TO SOLR

Stopwords

a	
the	
die	✖
...	

Стоп-слово больше не присутствует в списке. Нажмите Сохранять чтобы сохранить изменения.

. Не забудьте снова экспортить список на сервер Solr, чтобы сообщить ему об изменениях.

Обновление индекса Solr

Инструкции по обновлению индекса Solr с помощью административной панели бэк-офиса.

Существует несколько способов обновления индекса Solr, и вы можете использовать визуальный интерфейс Backoffice Administration Cockpit для выполнения всех необходимых действий. Вы можете создать, обновить или удалить индекс Solr.

Мастер операций индексатора

Вы можете создать, обновить и удалить индекс Solr. Шаги, представленные ниже, покажут вам, как обновить индекс Solr с помощью мастера операций индексатора.

Мастер горячего обновления индексатора

Основная цель мастера обновления индексатора Hot Update Wizard — позволить вам выполнять быстрые, специальные обновления индекса Solr. Он полезен, если возникает необходимость изменить атрибуты только одного или нескольких типов индексированных элементов. Вы также можете удалить определенные индексированные элементы. Раздел проведет вас через этот процесс.

Обновление индекса Solr с помощью Cronjob

Вы можете настроить cron job, отвечающий за обновление индексов Solr. Выполните следующие шаги, чтобы выполнить эту задачу.

Мастер операций индексатора

Вы можете создать, обновить и удалить индекс Solr. Шаги, представленные ниже, покажут вам, как обновить индекс Solr с помощью мастера операций индексатора.

Процедура

. Перейдите к Системный поиск и навигация Конфигурация поиска Solr Facet ➤ Конфигурации поиска фасетов ➤ в дереве проводника.

Список доступных конфигураций (если такие имеются) отображается справа.

. Выберите конфигурацию и щелкните по ней.

Появляется редактор.

. Нажмите на Индексную кнопку.

The screenshot shows the SAP Solr configuration interface. At the top, there are three entries: 'apparel-ukIndex Apparel UK Solr Index', 'appareLUKConfig', and 'electronicsIndex Electronics Solr Index'. Below this, a specific index configuration for 'apparel-ukIndex' is displayed. The 'Index' tab is selected and highlighted with a yellow box. Other tabs include 'Hot Update Index', 'PROPERTIES', 'INDEXED TYPES', 'KEYWORD REDIRECTS', 'SYNONYMS', 'STOPWORDS', 'CRON JOBS', and 'ADMINISTRATION'. Under the 'PROPERTIES' tab, fields are shown for 'Name' (apparel-ukIndex), 'Description' (Apparel UK Solr Index), and 'Index name prefix' (apparel-uk).

. В окне мастера выберите значение операции.

The screenshot shows the 'Create New SolrIndexerOperationWizard' dialog box. The title bar says 'Create New SolrIndexerOperationWizard' with a close button. The main area is titled 'SOLR INDEXER' and sub-titled 'Provide indexer configuration'. Below this, a section labeled 'Operation values' contains a dropdown menu with the value 'full' selected. At the bottom left is a 'START' button, and at the bottom right is a 'CANCEL' button.

У вас есть следующие варианты на выбор:

- FULL: Пересоздает индекс, то есть удаляет существующий индекс и пересоздает новый по заданным свойствам. Во время полного обновления индекс может быть недоступен в полном масштабе, пока новый индекс не будет полностью создан.
- ОБНОВЛЕНИЕ: Обновляет определенные существующие индексированные элементы. Обновляются только различия и затрагиваются только обновленные элементы. Другие индексированные элементы доступны для просмотра.
- DELETE: используется редко. Удаляет определенные индексированные элементы из индекса, в то время как другие элементы не затрагиваются, и индекс остается доступным. Примером использования может быть случай, когда определенный элемент становится недоступным, и вы уверены, что он не будет доступен в будущем, вы можете настроить операцию Delete, чтобы удалить этот элемент из индекса Solr.

. Нажмите [Начинать](#) чтобы начать процесс индексации.

. После завершения индексации отобразится окно подтверждения. Нажмите [Сделанный](#) чтобы закрыть мастер.

Мастер горячего обновления индексатора

Основная цель мастера обновления индексатора Hot Update Wizard — позволить вам выполнять быстрые, специальные обновления индекса Solr. Он полезен, если возникает необходимость изменить атрибуты только одного или нескольких типов индексированных элементов. Вы также можете удалить определенные индексированные элементы. Раздел проведет вас через этот процесс.

Процедура

. Перейдите к Системный поиск и навигация Solr Конфигурация поиска по фасетам Конфигурации поиска по фасетам в дереве проводника.

Список доступных конфигураций (если таковые имеются) отображается справа.

. Выберите конфигурацию и щелкните по ней.

Появляется редактор.

. Нажмите на Индекс горячих обновлений кнопку, чтобы открыть мастер.

The screenshot shows the SAP Solr Configuration interface. At the top, there are two tabs: 'apparel-ukIndex' and 'Apparel UK Solr Index'. Below them is a list of configurations: 'appareLUKConfig' and 'electronicsIndex Electronics Solr Index'. Under 'apparel-ukIndex', there are several tabs: 'PROPERTIES' (which is selected), 'INDEXED TYPES', 'KEYWORD REDIRECTS', 'SYNONYMS', 'STOPWORDS', 'CRON JOBS', and 'ADMINISTRATION'. The 'PROPERTIES' tab contains fields for 'Name' (apparel-ukIndex), 'Description' (Apparel UK Solr Index), and 'Index name prefix' (apparel-uk). A yellow box highlights the 'Hot Update Index' tab under 'apparel-ukIndex'.

. В разделе «Тип и операция» укажите следующие значения.

The screenshot shows the 'Create New SolrIndexerHotUpdateWizard' dialog. The title bar says 'Create New SolrIndexerHotUpdateWizard' with a close button. Below it, there are two tabs: 'TYPE AND OPERATION' (selected) and 'ITEMS'. The 'TYPE AND OPERATION' tab has the sub-instruction 'Set indexed type and indexer operation'. It contains two dropdown menus: 'Item type' (set to 'Product') and 'Operation values' (set to 'update'). At the bottom are 'BACK' and 'CANCEL' buttons.

Имя поля	Описание
Тип товара	Тип индексируемого элемента. Вы можете выбрать из типов, перечисленных на вкладке «Индексируемые типы». Подробнее об определении индексированных типов: Создание и настройка индексированных типов .
Эксплуатационные значения	Значение операции индексатора. У вас есть два значения на выбор: обновить или удалить. В зависимости от выбранной операции элементы будут добавлены или удалены из индекса.

. В разделе Элементы выберите элементы для индексации. Вы можете выбрать из доступных элементов в списке или создать новый элемент.

. Если вы хотите создать новый элемент, нажмите кнопку **Создать новый (название элемента)** кнопка.

-Примечание

Название кнопки и поля мастера различаются в зависимости от типа элемента.

Обновление индекса Solr с помощью CronJob

Вы можете настроить cron job, отвечающий за обновление индексов Solr. Выполните следующие шаги, чтобы выполнить эту задачу.

Контекст

Обновление всего документа для больших и сложных каталогов с целью включения всего нескольких изменений может занять много времени и неэффективно. Частичное обновление — это способ обновить только подмножество атрибутов документа Solr, одновременно сохраняя значение других.

Новая задача для частичного обновления может быть создана через Backoffice Administration Cockpit. Как и в случае с любым другим заданием cron, вы можете указать один или несколько временных интервалов для запуска этой задачи или запустить ее немедленно.

Процедура

- . Перейдите к **Фоновые процессы системы** ➔ **CronJobs** в дереве проводника.

Список определенных cronjobs отображается справа.

- . Нажмите на  кнопку в верхней части страницы.

Отобразится список доступных заданий cron.

- . Выбирайте **Cronjob для solr Indexer (внешний)**.
- . Используйте всплывающее окно мастера для настройки параметров вашего cronjob.

Create New Cronjob for solr indexer



PROVIDE ALL MANDATORY FIELDS

Language:

English [en]

Code:

Logs Operator: 

AND

Process on server node group: 

Log to file:

True False

Allow system recovery after changes:

True False

Настройте параметры. Наиболее важными для вас являются следующие:

- Код: уникальный код для вашего cronjob, который позволяет легко его отличить. Пример: частичный Update-electronicsIndex-cronJob
- Индексируемый тип: тип для индексации. Пример: Продукт.
- Операция индексатора: способ индексации. Для выполнения частичного обновления необходимо выбрать частичное обновление.
- Запрос: запрос, используемый для индексации. Пример: выберите {PK} из Продукта.
- Конфигурация Solr: конфигурация поиска фасетов, к которой применяется cronjob.
- Определение задания: определение, содержащее настройки для конкретного задания. Выберите solrExtIndexerJob.

. Нажмите Сделано и закончить.

Ваша задача cronjob теперь присутствует в списке.

Щелкните запись cronjob, чтобы увидеть ее подробности.

Как видите, настройки cronjob сгруппированы по вкладкам.

Code	Job definition	Current status	Last result	Timetable
update-apparel-delIndex-cronjob	solrIndexerJob	FINISHED	SUCCESS	Interval gap: 1 minute

solrIndexerJob : full-apparel-delIndex-cronJob - FINISHED - SUCCESS

ESSENTIAL

Code	Current status	Job definition	Last result
full-apparel-delIndex-cronJob	FINISHED	solrIndexerJob	SUCCESS

Каждая вкладка содержит раздел Essential, где вы можете найти общие настройки для вашего cronjob. Ниже этого раздела вы можете найти информацию о спецификации для конкретной вкладки:

- Журнал: Информация о журналах и этапах работы. Подробнее о журналах:
- Задача: Специфическая информация индексатора и настройки уведомлений.
- Запустить как: параметры, которые позволяют запускать задачу от имени определенного пользователя или с определенными атрибутами сеанса, указывать выделенный узел сервера для запуска задачи (для оптимизации балансировки нагрузки) и выбирать приоритет задачи.
- Расписание: настройки, позволяющие указать один или несколько временных интервалов для запуска этой задачи, или вы можете запустить задачу немедленно.
- Восстановление системы: Настройки, позволяющие указать, должна ли система восстанавливаться после изменений.
- Администрирование: Расширенные параметры конфигурации.

Запуск CronJob

Контекст

У вас есть два варианта запуска cronjob: вы можете запустить cronjob немедленно или запланировать его на определенный временной интервал.

Процедура

. Чтобы запланировать cronjob, перейдите на страницу [Расписание](#) вкладка.

solrIndexerJob : full-apparel-delIndex-cronJob - FINISHED - SUCCESS

ESSENTIAL

Code	Current status	Job definition	Last result
full-apparel-delIndex-cronJob	FINISHED	solrIndexerJob	SUCCESS

Timetable	Last start time	Enabled	Last end time
Daily at 03:05:00	Jan 27, 2017 9:04:36 AM	<input checked="" type="radio"/> True <input type="radio"/> False	Jan 27, 2017 9:05:23 AM

SCHEDULE

You can specify one or more time slots to run this task in, or you can run the task immediately.

Trigger

Daily at 03:05:00 - Sat Jan 28 03:0...

. Чтобы запланировать cronjob, перейдите на страницу [Расписание](#) вкладка.

solrIndexerJob : full-apparel-delIndex-cronJob - FINISHED - SUCCESS

LOG **TASK** **RUN AS** **TIME SCHEDULE** **SYSTEM RECOVERY** **ADMINISTRATION**

Timetable	Last start time	Enabled
Daily at 03:05:00	Jan 27, 2017 9:04:36 AM	<input checked="" type="radio"/> True <input type="radio"/> False

SCHEDULE

You can specify one or more time slots to run this task in, or you can run the task immediately.

Trigger

Daily at 03:05:00 - Sat Jan 28 03:0...

+ Create new Trigger

. Нажмите на [Создать новый триггер](#) кнопку.

. Появится окно мастера.

. Заполните поля, чтобы предоставить обязательную информацию о расписании. Имя cronjob уже заполнено для вас.

Create New Trigger



ALL REQUIRED FIELDS

Cronjob:

solrIndexerJob : full-apparel-delIndex-cronJob - FINISHED - SUCCESS

Active:



True



False

Next Activation time:

Jan 27, 2017 2:45:33 PM



Cron expression:

CANCEL

DONE

Имя поля	Описание
CronJob	Имя cronjob. Поле заполняется автоматически.
Активный	Флаг, указывающий, активирован ли триггер.
Время следующей активации	Время срабатывания триггера.
Выражение Cron	Выражение Cron, управляющее планированием времени.

. Нажмите [Сделанный](#) для завершения планирования.

Вы успешно настроили основные параметры для триггера cronjob. Чтобы добавить более конкретные параметры, выберите триггер из списка и дважды щелкните его, чтобы открыть редактор.

Создание и настройка индексированных типов

Узнайте, как создать и настроить индексированный тип, определив его свойства и запросы.

Каждый предмет в Solr имеет определенный тип. Существует определенное количество типов, идущих csolrfacetsearchрасширение. Однако вы также можете добавить новый тип элемента Solr. Добавляя новые типы, вы можете определить свой собственный набор элементов, которые может искать служба SolrFacetSearch.

Панель администрирования бэк-офиса предоставляет вам мастер, который поможет вам легко создать и настроить каждый элемент.

-Кончик

Вам не нужно заполнять всю информацию и поля при создании нового типа элемента. Вы можете закончить в любое время, нажав [Сделанный](#) и вернитесь в главный редактор, чтобы добавить недостающую информацию, отредактировав свои элементы.

Добавить индексированные типы Solr

Процедура

. Перейдите к Системный поиск и навигация доступные Конфигурация поиска Solr Facet Индексированные типы Справа вы можете увидеть список типы элементов (если таковые определены).

. Нажмите на кнопку в верхней части редактора.

Появится всплывающее окно.

. В Типраздел, вам нужно выбрать типы для индексации. Укажите уникальный идентификатор и выберите составной тип для вашего элемента.

INDEXED TYPE	INDEXED PROPERTIES	FREE TEXT QUERY	GROUPING	SOLR SORTS	ADMINISTRATION
Resolution, 80	string	true	false	false	false
feature-size	text	false	false	false	false
feature-videoCap	text	false	false	false	false
feature-aperture	text	false	false	false	false
feature-picMode	text	false	true	false	false
feature-memCards	text	false	false	false	false
feature-display	text	false	false	false	false

-Примечание

Вы можете завершить создание нового типа в любой момент, нажав кнопку Сделанный. Вы сможете вернуться к редактору позже, чтобы внести изменения.

Чтобы определить свойства на этом этапе, нажмите Следующий.

Определение свойств для индексированных типов

Вы можете кодифицировать индексированные типы, используя свойства.

Контекст

Вы уже создали индексированный тип и хотите его дополнительно определить, добавив дополнительные настройки.

Процедура

. Перейдите к [Системный поиск и навигация Solr](#) [Фасетный поиск](#) [Конфигурация Индексированные типы](#). Выберите конфигурацию, для которой вы создали индексированный тип.

. Выберите [Индексированные свойства](#) вкладку и щелкните [Создать новое свойство](#).

INDEXED TYPE	INDEXED PROPERTIES	FREE TEXT QUERY	GROUPING	SOLR SORTS	ADMINISTRATION
	Resolution, 80	string	true	false	false
	feature-size	text	false	false	false
	feature-videoCap	text	false	false	false
	feature-aperture	text	false	false	false
	feature-picMode	text	false	true	false
	feature-memCards	text	false	false	false
	feature-display	text	false	false	false

. В основной раздел, определите следующие настройки:

Имя поля	Описание
Имя	Уникальное название объекта недвижимости.
Тип	Определяет пользовательский тип свойства.
Фасет	Определяет, следует ли классифицировать свойство как фасет.
Многозначный	Определяет, может ли свойство содержать несколько значений.
Валюта	Определяет, может ли недвижимость хранить валюту.
Локализованный	Определяет, можно ли локализовать свойство.
Использовать для проверки орфографии	Определяет, следует ли проверять орфографию свойства.
Использовать для автозаполнения	Определяет, должно ли свойство подлежать автоматическому заполнению.
Включить в ответ	Определяет, возвращается ли поле в ответе на запрос.
Использовать для выделения	Определяет, используется ли поле для выделения поисковых терминов в результатах поиска.

. Нажмите [Следующий](#), чтобы продолжить.

. В [Другой](#) разделе укажите любые дополнительные значения для вашего объекта недвижимости.

. Нажмите [Сделанный](#).

Результаты

В список добавлен новый объект недвижимости.

Запросы определения для индексированного типа

Контекст

Раздел «Конфигурация индексатора» на вкладке «Индексированные типы» предоставляет мастер для создания запросов индексатора.

Процедура

- . Перейдите к [Системный поиск и навигация Solr](#) [Фасетный поиск](#) [Конфигурация Индексированные типы](#). Выберите конфигурацию, для которой вы хотите создать поисковый запрос, и щелкните по ней, чтобы открыть редактор.

The screenshot shows the SAP Solr Configuration interface. At the top, there is a search bar with a yellow 'SEARCH' button. Below the search bar, there are tabs for 'Identifier' and 'Composed Type'. Under 'Identifier', there is a list item 'apparegi-ukrproducttype' with the value 'Product [Product]'. Under 'Composed Type', there is a list item 'electronicsProductType' with the value 'Product [Product]'. A blue header bar indicates the selected item is 'electronicsProductType ComposedTypeModel (8796097904722@15)'. Below this, there are buttons for 'REFRESH' and 'SAVE'. A navigation bar at the bottom includes 'INDEXED TYPE', 'INDEXED PROPERTIES', 'FREE TEXT QUERY', 'GROUPING', 'SOLR SORTS', and 'ADMINISTRATION'. The main content area is divided into two sections: 'SEARCH CONFIGURATION' and 'INDEXER CONFIGURATION'. The 'SEARCH CONFIGURATION' section contains a 'Search Query Templates' section with a 'Create new Search Query Template' button. The 'INDEXER CONFIGURATION' section contains an 'Indexer queries' section with a table. The table has two columns: 'Type' and 'Query value'. There is one row for 'full' type with a complex SQL query. The entire 'INDEXER CONFIGURATION' section is highlighted with a yellow border.

Type	Query value
full	<pre>SELECT {PK} FROM {Product} WHERE ({varianttype} IS NULL OR {varianttype} NOT IN ({{ SELECT {PK} FROM {varianttype} WHERE {code} = 'Electronics' }}) AND {code} NOT IN({{ SELECT {code} FROM {GenericVariantProduct} }}) SELECT DISTINCT tbl.pk, tbl.code FROM ({{{ SELECT DISTINCT {p:PK} AS pk, {p:code} AS code, {p:varianttype} AS varianttype FROM {Product AS p LEFT JOIN CustomerReview AS cr ON {cr:product}={p:PK} } WHERE {p:modifiedtime} >= ?lastIndexTime OR {cr:modifiedtime} >= ?lastIndexTime }}} UNION</pre>

- . Нажмите на [Создать новый запрос индексатора](#) в нижней части раздела, чтобы создать новый запрос индексатора.

Появится новое окно.

- . В разделе «Основные сведения» укажите значения обязательных полей.

Имя поля	Описание
Идентификатор	Уникальное имя запроса.
Тип	Тип вашего запроса. У вас есть следующие значения на выбор: <ul style="list-style-type: none"> ○ FULL: пересоздает индекс ○ ОБНОВЛЕНИЕ: обновление некоторых документов в индексе ○ PARTIAL_UPDATE: позволяет выбрать поля для обновления ○ УДАЛИТЬ: удаляет документы из индекса
Значение запроса	Гибкий поисковый запрос.
Вставить текущую дату	Проверяет, если текущая дата параметр должен быть введен в запрос.
Ввести текущее время	Проверяет, если текущее время параметр должен быть введен в запрос.
Вставить последнее время индекса	Проверяет, если lastIndexTime параметр должен быть введен в запрос.

. Перейдите в раздел Другие. Выберите пользователей, если запросы выполняются с привилегиями пользователя. Нажмите Сделанный и закончить.

. Теперь вы можете определить дополнительные настройки для индексированного типа, используя основной редактор.

Дополнительные настройки для индексированных типов

Шаблоны поисковых запросов

Шаблоны поисковых запросов вводят многочисленные параметры настройки для получения улучшенных результатов поиска для клиентов. При создании шаблонов вы можете использовать различные атрибуты, такие как размер страницы, группировка или сортировка по различным каналам или даже частям витрины. Подробную информацию о создании и управлении шаблонами поисковых запросов см. [Шаблоны поисковых запросов](#).

Другие

Для каждого из индексированных типов вы также можете определить дополнительные настройки — вы найдете их в Индексированные типы вкладка в разделе Другие. Вы можете определить следующие пункты:

- Поставщик удостоверений: Поставщик удостоверений для типа элемента
- Модель загрузчика
- Поставщик значений полей модели: устанавливает поставщик значений полей, который зависит от этого типа, а не от конкретных свойств.
- Поставщик значений свойств по умолчанию
- Преобразователь результатов: преобразователь результатов (идентификатор компонента Spring), необходимый для преобразования данных результатов Solr в простой объект данных, который можно использовать для отображения данных индексированного типа (например, продукта) непосредственно на внешнем интерфейсе.

Расширенная конфигурация

В разделе Advanced Configuration вы можете определить дополнительные параметры для индексированного типа, например, конфигурацию шарда. Количество шардов и фактор репликации можно настроить для каждого facetsearchconfig, в этом случае все индексированные типы, связанные с ним, будут использовать ту же конфигурацию. Подробнее о настройке facetsearchconfig пункт см.: [Конфигурация поиска и индексатора Dene](#).

Однако можно переопределить конфигурацию для отдельного индексированного типа с помощью раздела «Расширенная конфигурация».

ADVANCED CONFIGURATION

Additional parameters of Indexed type

solr.collection.numShards ▶ 2	
solr.collection.replicationFactor ▶ 2	
+ Add new item	

-Примечание

Если параметры не установлены, ни во время конфигурирования `facetsearchconfig` элемента для отдельного индексированного типа будут использоваться значения по умолчанию:

- `solr.collection.numShards = 1`
- `solr.collection.replicationFactor = 1`

Управление диапазонами Solr

Инструкции по управлению наборами диапазонов Solr с помощью административной панели бэк-офиса.

Фасеты ранжирования для сервера Solr можно создать для любого поля даты или числового поля, которое поддерживает запросы диапазона. С помощью диапазонов вы можете составить ряд запросов диапазона для цен, дат и т. д. Создание фасетов на основе диапазона позволяет вам точнее нацеливать результаты поиска.

Поисковый сервер Solr поддерживает числовые и датовые диапазоны, которые помогают вам получать результаты на основе определенных условий. Некоторые свойства могут быть проиндексированы в соответствии с диапазоном, к которому они принадлежат. Например, для подхода поиска по фасетам удобно иметь ценовые диапазоны в качестве фасетов.

Типы данных

Вы можете создать наборы диапазонов значений для заданного типа данных. Это родительские элементы, которые группируют диапазоны значений одного типа. Разрешены следующие типы: string, int, double, oat и date. Определив наборы диапазонов, вы обычно хотите заполнить их фактическими диапазонами значений. Это простые элементы, состоящие из имени, значения from и to.

-Примечание

При создании диапазона помните, что верхние и нижние пределы (от и до) проверяются в соответствии с типом набора диапазонов. Следовательно, для числовых типов принимаются только числовые значения. Аналогично, значения даты не принимаются, если они не следуют `ГГГГ-ММ-ДД\ЧЧ:ММ` формат.

[Создать диапазоны](#)

[Изменить диапазоны](#)

[Удалить диапазоны](#)

Создать диапазоны

Процедура

. Перейдите к  [Система](#) ➔ [Поиск и навигация](#) ➔ [Конфигурация поиска Solr Facet](#) ➔ [Диапазоны](#).

Справа отображается список уже созданных наборов диапазонов (если такие имеются).

. Нажмите на  кнопку для запуска мастера.

. В [Основы](#) раздел, укажите необходимую информацию.

Create New Set of ranges



ESSENTIALS

Provide all mandatory fields

Name:

electronicsPriceRangeEUR

Type:

double

CANCEL

DONE

Имя поля	Описание
Имя	Уникальное имя для вашего набора диапазонов.
Тип	Тип набора, например: текстовый, двойной, овсяный, строковый.

. Нажмите [Сделанный](#) закончить.

Определенный набор диапазонов появится в списке.

. Щелкните запись, чтобы войти в режим

редактора.. Появится редактор.

. В разделе Essential выберите конфигурацию поиска фасетов, к которой применяется ваш набор.

Конфигурация добавлена.

Name	Type
ElectronicsPriceRangeEUR	double
MegaPixelRange	double

ElectronicsPriceRangeEUR:double

RANGES ADMINISTRATION

ESSENTIAL

Name	Qualifier	Facet search configuration
ElectronicsPriceRangeEUR	EUR	electronicsIndex
		...

RANGE VALUES

Type	Ranges						
double	<table border="1"> <thead> <tr> <th>Name</th> <th>From</th> <th>To</th> </tr> </thead> <tbody> <tr> <td>EURO-EU...</td> <td>0</td> <td>50</td> </tr> </tbody> </table>	Name	From	To	EURO-EU...	0	50
Name	From	To					
EURO-EU...	0	50					
Create new Value range							

. В разделе «Значения диапазона» нажмите кнопку [Создать новый диапазон значений](#) и выберите вариант.

Укажите требуемые значения.

Create New Value range X

RANGES
Provide range values

Name:

From:

To:

[CANCEL](#) DONE

Имя поля	Описание
Имя	Уникальное имя для вашего диапазона значений
От	Минимальное значение диапазона.

Имя поля	Описание
K	Максимальное значение для диапазона.

. Нажмите [Сделанный](#) и закончить.

В список добавлен новый диапазон.

ESSENTIAL

Name

ElectronicsPriceRangeEUR

Qualifier

EUR

Facet search configuration

electronicsIndex

...

RANGE VALUES

Type

double

Ranges

Name	From	To
EURO-EUR...	0	50
EURO-EUR...	0	49.99

[+ Create new Value range](#)

. Теперь вы можете добавить еще несколько диапазонов в набор.

Изменить диапазоны

Процедура

. Перейдите к [Системный поиск и навигация](#) ➤ [Конфигурация поиска Solr Facet](#) ➤ [Диапазоны](#).

Справа отображается список уже созданных наборов диапазонов (если таковые имеются).

. Выберите набор, который вы хотите изменить, и щелкните по нему.

Появляется редактор.

ElectronicsPriceRangeEUR:double

RANGES ADMINISTRATION

ESSENTIAL

Name	Qualifier	Facet search configuration
ElectronicsPriceRangeEUR	EUR	electronicsIndex

RANGE VALUES

Type	Ranges									
double	<table border="1"> <thead> <tr> <th>Name</th> <th>From</th> <th>To</th> </tr> </thead> <tbody> <tr> <td>EURO-EUR...</td> <td>0</td> <td>50</td> </tr> <tr> <td>EURO-EUR...</td> <td>0</td> <td>49.99</td> </tr> </tbody> </table>	Name	From	To	EURO-EUR...	0	50	EURO-EUR...	0	49.99
Name	From	To								
EURO-EUR...	0	50								
EURO-EUR...	0	49.99								
<div style="border: 1px solid #ccc; padding: 5px; display: inline-block;"> Remove Edit Details </div>										

Теперь вы можете редактировать настройки своего набора, добавлять, изменять и удалять диапазоны внутри набора.

. Нажмите [Сохранить](#), чтобы сохранить изменения.

Удалить диапазоны

Процедура

. Перейдите к [Системный поиск и навигация](#) ➔ Конфигурация поиска Solr Facet ➔ [Диапазоны](#).

Справа отображается список уже созданных наборов диапазонов (если таковые имеются).

. Выберите набор, который вы хотите изменить, и щелкните по нему.

Появляется редактор.

. Используйте значок корзины, чтобы удалить набор диапазонов.

ElectronicsPriceRangeEUR:double

RANGES ADMINISTRATION

ESSENTIAL

Name	Qualifier	Facet search configuration
ElectronicsPriceRangeEUR	EUR	electronicsIndex

RANGE VALUES

Type	Ranges									
double	<table border="1"> <thead> <tr> <th>Name</th> <th>From</th> <th>To</th> </tr> </thead> <tbody> <tr> <td>EURO-EUR...</td> <td>0</td> <td>50</td> </tr> <tr> <td>EURO-EUR...</td> <td>0</td> <td>49.99</td> </tr> </tbody> </table>	Name	From	To	EURO-EUR...	0	50	EURO-EUR...	0	49.99
Name	From	To								
EURO-EUR...	0	50								
EURO-EUR...	0	49.99								
<div style="border: 1px solid #ccc; padding: 5px; display: inline-block;"> X Remove </div>										

Включить сбор статистики

The `solrfacetsearch` расширение поддерживает сбор статистических данных по запросам, отправленным на сервер Solr.

Примечание

Чтобы включить эту функцию, вы должны работать в устаревшем режиме, а Solr должен быть установлен локально. Если вы соответствуете этим предварительным условиям, вам необходимо установить соответствующие свойства в конфигурации Solr. Вы можете настроить `solrfacetsearch` для сбора важной информации с целью ее экспорта на сервер Solr для дальнейшего анализа. Используйте файл конфигурации для установки необходимых параметров и сбора информации, такой как временная метка или номер результата.

Базовая конфигурация

Процедура

- . Включить сбор статистики.

```
solrStats.enableCollectingStatistics=true
```

. По умолчанию Solr собирает статистические данные в файлы, расположенные в следующей папке:

```
solrStats.filesLocation=${HYBRIS_LOG_DIR}/solrstats
```

Файлы, содержащие статистические данные, имеют префикс, как в примере, приведенном ниже.

```
solrStats.filePrefix=stats.log
```

- . Файлы обновляются каждый час, это настраивается следующим свойством

```
solrStats.dateFormat=rrrr-MM-дд.ЧЧ
```

Собранные данные

Если вы включите сбор статистических данных, то после выполнения каждого запроса на сервере Solr будет сохраняться следующая информация:

- Дата
- Конфигурация индекса
- Язык
- Запрос
- Количество результатов

Используйте Cronjob для сбора статистики

Контекст

Вам следует создать cron job для сохранения агрегированных значений в базе данных. Затем сохраняется следующая информация: дата, индекс, конфигурация, язык, запрос, среднее количество результатов для запроса и количество.

Процедура

- . Перейдите к [Фоновые процессы системы](#) 

Список определенных cronjobs отображается справа.

. Нажмите на  кнопка.

Появится список доступных типов заданий cron.

. Выберите SolrQueryСтатистикаКоллекторCronJob cronjob..

Используйте окно мастера, чтобы указать требуемые значения.

Наиболее важные ценности, которые вам следует предоставить:

- Код: код (имя) для вашего cronjob, который позволяет его легче отличить. Это может быть, например: Solr Statistics Collection for
- Определение задания: тип задания, которое должно быть выполнено. Для этого конкретного задания выберите SolrQueryСтатистикаСборщикРабота.

Оставшиеся поля можно заполнить в любой момент в режиме редактора.

Create New SolrQueryStatisticsCollectorCronJob

PROVIDE ALL MANDATORY FIELDS

Retry execution, if the job can not be executed yet:

True False

Once-only executable:

True False

Process on server node group: [?](#)

Code:

Process on server node: [?](#)

Owner:

...

Allow system recovery after changes:

True False

. Нажмите Сделанный/закончить.

Ваш новый cronjob теперь виден в списке. Чтобы запустить его, выполните действия, описанные в: [Запуск Cronjob](#).

Запрос статистики

Для получения статистики вы можете использовать следующий запрос:

ВЫБЕРИТЕ {PK} ИЗ {SolrQueryAggregatedStats}

SolrQueryAggregatedStats содержит следующие атрибуты:

- время
- indexConfig

- ЯЗЫК
- запрос
- считать
- avgNumberOfResults

Архитектура модуля поиска и навигации

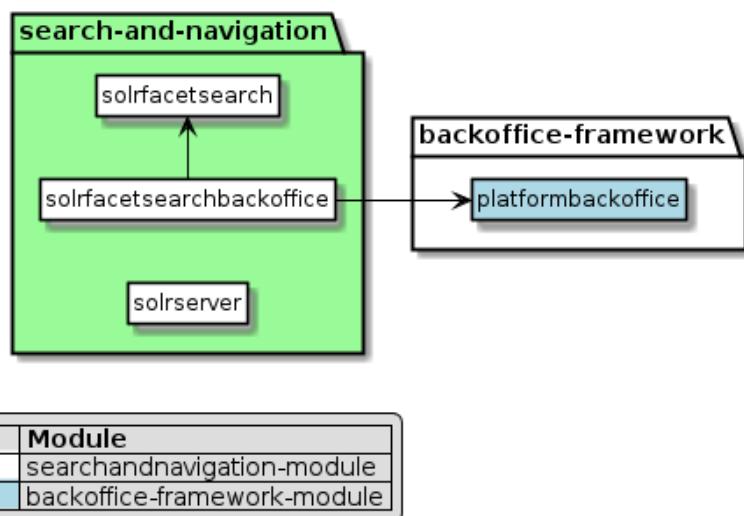
Модуль поиска и навигации представляет собой набор расширений, обеспечивающих функциональность фасетного поиска на основе сервера Solr. Фасетный поиск позволяет вам очищать или перемещаться по коллекции информации, используя ряд дискретных атрибутов.

Зависимости

Следующий UML отображает зависимости между всеми расширениями, обеспечивающими функциональность:

Нажмите на расширение, чтобы просмотреть подробную информацию о нем.

Это изображение интерактивное. Наведите курсор на каждую область для описания. Щелкните выделенные области для получения дополнительной информации.



Обратите внимание, что карты изображений в файлах PDF не являются интерактивными.

Рецепты

Полный список рецептов SAP Commerce, которые могут включать этот модуль, см.[Рецепты установщика](#).

Полный список рецептов пакетов расширения интеграции SAP Commerce Cloud, которые могут включать этот модуль, см.[Справочник рецептов установщика](#).

Расширения

Модуль поиска и навигации состоит из следующих расширений:

[Расширение backofficesolrsearch](#)

Расширение backofficesolrsearch — это расширение, которое добавляет Apache Solr Search вбэк-офис. С помощью поиска Apache Solr, бэк-офис обеспечивает более быструю, эффективную и оптимизированную функцию поиска с фасетной навигацией и индексацией практически в реальном времени.

[Расширение solrfacetsearch](#)

The solrfacetsearch расширение обеспечивает функциональность фасетного поиска и навигации на основе сервера Apache Solr. Оно позволяет вам выполнять фасетный (также называемый размерным) поиск по элементам SAP Commerce, таким как продукты и содержимое WCMS.

[Расширение solrserver](#)

The solrserver расширение включает в себя автономный сервер Solr, который может автоматически настраиваться, запускаться и останавливаться вместе с платформой.

[Расширение solrfacetsearchbackoffice](#)

The `solrfacetsearchbackoffice` расширение предоставляет интерфейс для создания и управления конфигурациями поиска с помощью панели администрирования Backoffice.

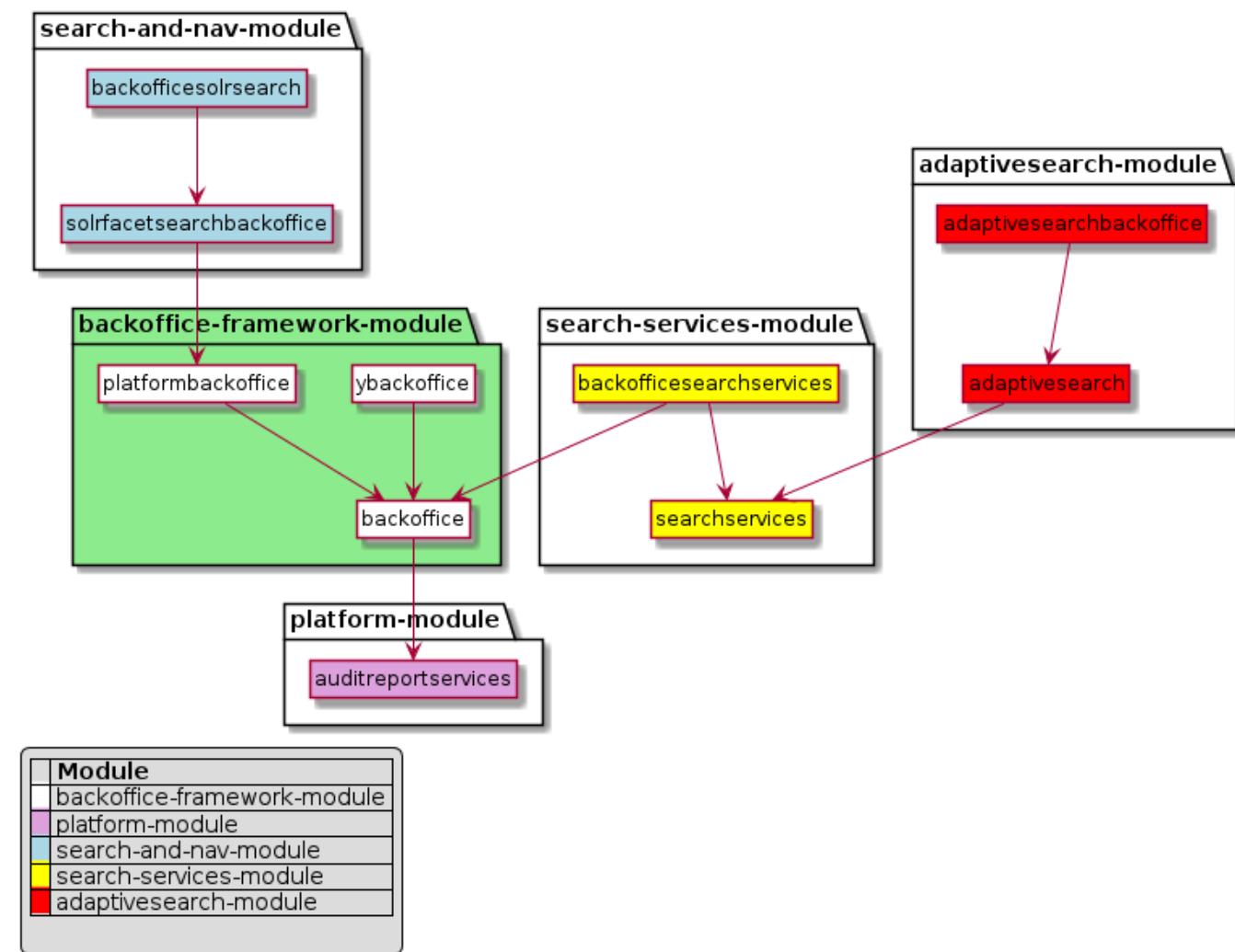
Расширение backofficesolrsearch

Расширение `backofficesolrsearch` — это расширение, которое добавляет Apache Solr Search в бэк-офис. С помощью поиска Apache Solr, бэк-офис обеспечивает более быструю, эффективную и оптимизированную функцию поиска с фасетной навигацией и индексацией практически в реальном времени.

О расширении

Имя	Каталог	Связанный модуль
бэкофисsolrsearch	hybris/bin/modules/search-andnavigation/backofficesolrsearch	Модуль поиска и навигации

Зависимости



Зависимости

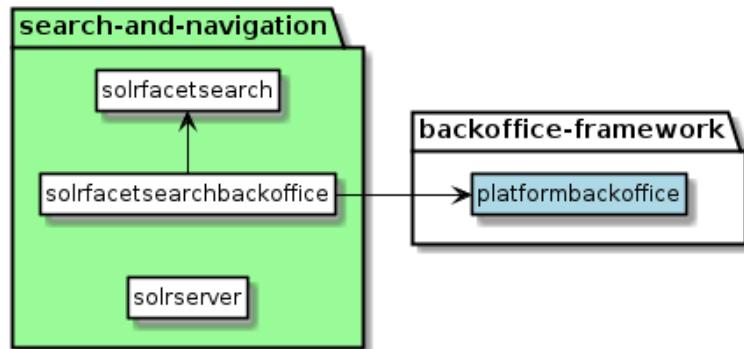
Расширение solrfacetsearch

The `solrfacetsearch` расширение обеспечивает функциональность фасетного поиска и навигации на основе сервера Apache Solr. Оно позволяет вам выполнять фасетный (также называемый размерным) поиск по элементам SAP Commerce, таким как продукты и содержимое WCMS.

-Осторожность

Индексировать только данные, связанные с продуктом. Хранение и индексирование любых конфиденциальных данных в Solr может представлять угрозу безопасности.

ЗАВИСИМОСТИ



Module
searchandnavigation-module
backoffice-framework-module

Расширение solrserver

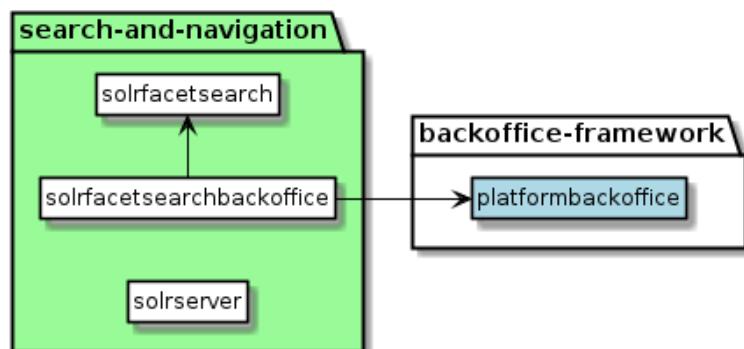
The **solrserver** extension includes an autonomous Solr server, which can be automatically configured, started, and stopped together with the platform.

This extension is not required by default, but it is very convenient to include it in CI or development environments. It is possible to create multiple Solr instances. A Solr instance represents a combination of configuration, data, and log files. Only one instance per machine can be started/stopped.

-Примечание

Не используйте **solrserver** расширение на производственных системах. SAP рекомендует использовать автономный сервер Solr на отдельной машине. Вы можете скопировать **solrserver** из расширения и запустить его как автономный сервер.

ЗАВИСИМОСТИ

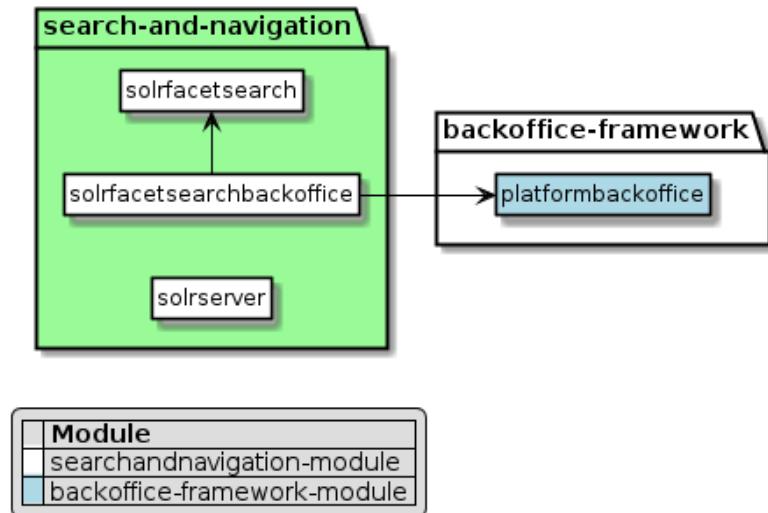


Module
searchandnavigation-module
backoffice-framework-module

Расширение solrfacetsearchbackoffice

The `solrfacetsearchbackoffice` расширение предоставляет интерфейс для создания и управления конфигурациями поиска с помощью панели администрирования Backoffice.

Зависимости



Реализация модуля поиска и навигации

Установите и настройте сервер Solr, чтобы использовать функции индексации и поиска. Вы также можете определить конфигурации для нескольких версий Solr и узнать, как сделать свой сервер Solr безопасным.

Установка сервера Solr

Для корректной работы расширения его необходимо подключить к работающему серверу Solr. Установка и настройка зависят от того, в каком режиме вы хотите запустить сервер Solr.

Конфигурация поиска Solr Facet

Настройку поиска по фасетам можно выполнить через графический интерфейс или панель администрирования бэк-офиса. [Солр](#)

Безопасность

Функции безопасности Solr включают поддержку шифрования связи с Solr (а также между узлами Solr) с использованием SSL и поддержку аутентификации и авторизации, предоставляемых фреймворками безопасности Solr. Функции безопасности включены по умолчанию, когда вы используете `solrserver` расширение. В другом случае вы можете использовать предоставленный пример конфигурации, который вы можете легко включить.

Поддержка нескольких версий Solr

SAP Commerce поддерживает более одной версии сервера Solr. [Общая](#)

функциональность

Часть функционала, используемого в модуле поиска и навигации, является общим для поиска и индексации. Например, оба они используют API слушателей.

Функциональность индексации

Существует ряд различных тем, касающихся функциональности индексирования. Каждая тема документирована в отдельном руководстве..

Функциональность поиска

Чтобы в полной мере использовать функциональность поиска, ознакомьтесь со стратегиями, используемыми в процессе поиска, и узнайте, как настраивать поиск с помощью API поиска, поставщиков сортировки и пользовательского сопоставления.

Установка сервера Solr

Для корректной работы расширения его необходимо подключить к работающему серверу Solr. Установка и настройка зависят от того, в каком режиме вы хотите запустить сервер Solr.

-Примечание

Убедитесь, что вы включили `solrfacetsearch` расширение для SAP Commerce путем редактирования `localextensions.xml` в `/hybris/config/` каталог.

Совместимость с сервером Apache Solr

В зависимости от текущей версии релиза может быть задействовано несколько различных версий сервера Apache Solr. Матрица совместимости вместе с информацией о поддержке нескольких версий Solr доступна в [Поддержка нескольких версий Solr](#).

Воспользуйтесь ссылками ниже, чтобы узнать больше о функциях, предоставляемых модулем поиска и навигации.

Соображения безопасности

При установке Apache Solr следует учитывать некоторые соображения безопасности. Подробнее о функциях безопасности см. [Солр Безопасность](#).

Быстрая установка автономного сервера Solr

Быстрая установка сервера Solr в автономном режиме.

Контекст

Следуйте инструкциям по быстрой установке и настройке сервера Solr на локальном компьютере.

Процедура

. Включить solrserver расширение в вашем localextensions.xml файле.

Расширение теперь готово к использованию. Для получения дополнительной информации о настройке расширений см. [Установка на основе указанных расширений](#).

. Создайте экземпляр сервера Solr. Вы можете использовать экземпляр по умолчанию, включенный в solrserver расширение.

Конфигурация по умолчанию следующая:

```
solrserver.instances.default.autostart=true
solrserver.instances.default.mode=standalone
solrserver.instances.default.hostname=localhost
solrserver.instances.default.port=8983
solrserver.instances.default.memory=512m
```

Если вам нужно, вы можете переопределить некоторые свойства в вашем местные свойства. Для получения дополнительной информации см.: [Расширение solrserver](#).

. Запустите сервер SAP Commerce.

Поскольку автозапуск включен для экземпляра сервера Solr по умолчанию, сервер Solr будет запускаться и останавливаться вместе с платформой. Для получения дополнительной информации см.: [Установка SAP Commerce вручную](#) и [Установка SAP Commerce с использованием рецептов установщика](#).

. Настройте SAP Commerce для использования сервера Solr.

Наиболее важными атрибутами, которые следует учитывать при обновлении конфигурации сервера Solr, являются:

- Режим: установите автономный.
- URL-адреса конечных точек: должны содержать один главный URL-адрес со следующим значением: `http://localhost:8983/solr`.

Подробную информацию о том, как легко настроить параметры сервера Solr с помощью панели администрирования Backoffice, см. здесь: [Создать конфигурацию поиска по фасетам](#).

Быстрая установка сервера Cloud Solr

Установите и настройте сервер Solr в облачном режиме.

Контекст

Следуйте инструкциям по быстрой установке и настройке сервера Solr на локальном компьютере.

Процедура

. Включить `solrserver` расширение в вашем `localextensions.xml` ле.

Расширение теперь готово к использованию. Для получения дополнительной информации о настройке расширений см.[Установка на основе указанных расширений](#).

. Создайте экземпляр сервера Solr. Вы можете использовать экземпляр облака, включенный в `solrserver` расширение. Однако, вам придется отключить автозапуск в свойствах проекта для экземпляра по умолчанию и включите его для экземпляра облака, как показано в блоке кода ниже.

```
solrserver.instances.default.autostart=false
solrserver.instances.cloud.autostart=true
```

Если вы хотите, вы можете переопределить следующую конфигурацию по умолчанию, включенную в свойствах, чтобы сделать ее подходящим для ваших нужд:

```
# отключает автозапуск для экземпляра Solr по умолчанию
solrserver.instances.default.autostart=false

solrserver.instances.cloud.autostart=true
solrserver.instances.cloud.mode=cloud
solrserver.instances.cloud.hostname=localhost
solrserver.instances.cloud.port=8983
solrserver.instances.cloud.memory=512m
solrserver.instances.cloud.zk.host=
solrserver.instances.cloud.zk.upconfig=true
```

Для получения дополнительной информации см.:[Расширение solrserver](#).

. Запустите сервер SAP Commerce.

Поскольку для экземпляра сервера облака Solr включен автозапуск, сервер Solr будет запускаться и останавливаться вместе с платформой. Для получения дополнительной информации см.:[Установка SAP Commerce вручную](#) и[Установка SAP Commerce с использованием рецептов установщика](#).

. Настройте SAP Commerce для использования сервера Solr.

Наиболее важными атрибутами, которые следует учитывать при обновлении конфигурации сервера Solr, являются:

- Режим: установите облачный.
- URL-адреса конечных точек: должны содержать один URL-адрес со значением `localhost:9983`, это URL-адрес сервера ZooKeeper.

Подробную информацию о том, как легко настроить параметры сервера Solr с помощью панели администрирования Backoffice, см. здесь:[Создать конфигурацию поиска по фасетам](#).

Конфигурация поиска Solr Facet

Настройку поиска по фасетам можно выполнить через графический интерфейс или панель администрирования бэк-офиса.

Для того, чтобы плавно индексировать информацию, серверу Solr нужна допустимая конфигурация. Общая конфигурация не только определяет конфигурацию сервера Solr, но и создает набор директив, определяющих конкретные элементы платформы и степень, в которой они должны быть индексированы. Конфигурация поиска фасетов Solr является постоянной с точки зрения элементов в базе данных.

Технические аспекты этой конфигурации рассматриваются в дальнейших разделах этого документа. Хотя этот раздел содержит технические подробности, вы можете выполнить некоторые действия по конфигурации самостоятельно, используя пользовательский интерфейс Backoffice Administration Cockpit.

Файлы конфигурации сервера Solr

Как описано в руководстве по установке, существуют различные способы настройки сервера Solr. Конфигурация по умолчанию должна работать в большинстве случаев, однако вы можете захотеть немного изменить ее, чтобы настроить под свои нужды.

Конг-сеты

Возможно иметь различную конфигурацию для каждого индексированного типа. Это достигается с помощью именованных конфигурационных наборов, которые представляют собой общие каталоги конфигурации, хранящиеся в настраиваемом configsets базовый каталог.

-КОНЧИК

Информацию о настройке Solr путем создания файлов конфигурации см.[Создание и добавление пользовательских файлов Solr](#).

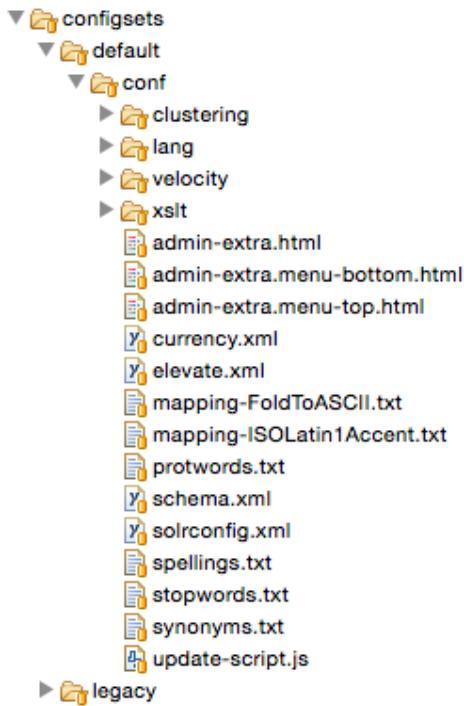
Чтобы создать конфигурационные наборы, просто добавьте новый каталог в configsets базовый каталог. Congset идентифицируется по имени этого каталога. Рекомендуется использовать существующий cong set в качестве шаблона. Структура представлена справа.

Наиболее важные файлы:

- schema.xml: определяет структуру вашего индекса, включая поля и типы полей;
- solrconfig.xml: определяет большинство параметров для настройки самого Solr.

-Примечание

Изменить невозможно configsets уже существующего индекса на сервере Solr. В таком случае измените атрибут congset в индексированном типе, удалите соответствующий индекс с сервера Solr (конфигурация и данные) и выполните повторную индексацию.



Синонимы и стоп-слова

Синонимы и стоп-слова можно определить с помощью Backoffice Administration Cockpit. После создания их необходимо отправить на сервер Solr. Это делается автоматически в процессе индексации или вручную в Backoffice Administration Cockpit с помощью Экспортировать стоп-слова/синонимы действие, однако использование этого действия не рекомендуется.

Время запроса против времени индексации

Solr различает синонимы/стоп-слова времени запроса и синонимы/стоп-слова времени индекса, и анализаторы могут быть определены для обоих. Обновления времени выполнения не запускают переиндексацию данных. Если синонимы/стоп-слова обновляются, а время индекса отличается от времени запроса, это может привести к ошибке несоответствия. В таком случае может потребоваться переиндексация.

Для получения дополнительной информации см.:[Процесс индексации](#).

Предложения

Solr может предлагать варианты для искомых слов. Вы можете использовать один из следующих компонентов для активации этой функции:

- SpellCheckComponent
- SuggestComponent

По умолчанию сервер Solr использует SpellCheckerComponent. Вы можете изменить эту конфигурацию в solrconfig.xml. Чтобы переключиться на компонент SuggestComponent, в файле config.xml закомментируйте следующие блоки кода:

```

<requestHandler имя="/suggest" класс="solr.SearchHandler"> <lst
имя="defaults">
    <стр name="spellcheck">истина</стр>
    <стр name="spellcheck.dictionary">по умолчанию</стр>
    <стр name="spellcheck.onlyMorePopular">истина</стр>
    <стр name="spellcheck.count">5</стр>
    <стр имя="spellcheck.collate">верно</стр>
</lst>
<arr name="components">
    <стр>предложить</стр>
</arr>
</requestHandler>

<searchComponent name="проверка орфографии"      class="solr.SpellCheckComponent">
    <стр name="queryAnalyzerFieldType">text_spell</стр>
    <lst name="spellchecker">
        <стр name="name">default</стр>
        <стр name="classname">solr.DirectSolrSpellChecker</стр>
        <стр name="field">проверка орфографии</стр>
    </lst>
    <lst name="проверка орфографии">
        <стр имя="имя">ru</стр>
        <стр name="имя_класса">solr.DirectSolrSpellChecker</стр>
        <стр name="поле">spellcheck_en</стр>
    </lst>
    <lst name="проверка орфографии">
        <стр имя="имя">cs</стр>
        <стр name="имя_класса">solr.DirectSolrSpellChecker</стр>
        <стр name="поле">spellcheck_cs</стр>
    </lst>
    <lst name="проверка орфографии">
        <стр имя="имя">de</стр>
        <стр name="имя_класса">solr.DirectSolrSpellChecker</стр>
        <стр name="поле">spellcheck_de</стр>
    </lst>
    <lst name="проверка орфографии">
        <стр имя="имя">ec</стр>
        <стр name="имя_класса">solr.DirectSolrSpellChecker</стр>
        <стр name="поле">spellcheck_es</стр>
    </lst>
    <lst name="проверка орфографии">
        <стр name="name">es_CO</стр>
        <стр name="classname">solr.DirectSolrSpellChecker</стр>
        <стр name="field">spellcheck_es_co</стр>
    </lst>
    <lst name="проверка орфографии">
        <стр имя="имя">fr</стр>
        <стр name="имя_класса">solr.DirectSolrSpellChecker</стр>
        <стр name="поле">spellcheck_fr</стр>
    </lst>
    <lst name="проверка орфографии">
        <стр имя="имя">привет</стр>
        <стр name="имя_класса">solr.DirectSolrSpellChecker</стр>
        <стр name="поле">spellcheck_hi</стр>
    </lst>
    <lst name="проверка орфографии">
        <стр имя="имя">xy</стр>
        <стр name="имя_класса">solr.DirectSolrSpellChecker</стр>
        <стр name="поле">spellcheck_hu</стр>
    </lst>
    <lst name="проверка орфографии">
        <стр имя="имя">идентификатор</стр>
        <стр name="имя_класса">solr.DirectSolrSpellChecker</стр>
        <стр name="поле">идентификатор_проверки_орфографии</стр>
    </lst>
    <lst name="проверка орфографии">
        <стр имя="имя">это</стр>
        <стр name="имя_класса">solr.DirectSolrSpellChecker</стр>
        <стр name="поле">spellcheck_it</стр>
    </lst>
    <lst name="проверка орфографии">
        <стр имя="имя">я</стр>
        <стр name="имя_класса">solr.DirectSolrSpellChecker</стр>
        <стр name="поле">spellcheck_ja</стр>
    </lst>
    <lst name="проверка орфографии">
        <стр имя="имя">ко</стр>
        <стр name="имя_класса">solr.DirectSolrSpellChecker</стр>
        <стр name="поле">spellcheck_ko</стр>
    </lst>

```

```

</lst>
<lst name="проверка орфографии">
    <ctrp имя="имя">pl</str>
    <ctrp name="имя_класса">solr.DirectSolrSpellChecker</str>
    <ctrp name="поле">spellcheck_pl</str>
</lst>
<lst name="проверка орфографии">
    <ctrp имя="имя">пт</str>
    <ctrp name="имя_класса">solr.DirectSolrSpellChecker</str>
    <ctrp name="поле">spellcheck_pt</str>
</lst>
<lst name="проверка орфографии">
    <ctrp имя="имя">ru</str>
    <ctrp name="имя_класса">solr.DirectSolrSpellChecker</str>
    <ctrp name="поле">spellcheck_ru</str>
</lst>
<lst name="проверка орфографии">
    <ctrp имя="имя">ж</str>
    <ctrp name="имя_класса">solr.DirectSolrSpellChecker</str>
    <ctrp name="поле">spellcheck_zh</str>
</lst>
<lst name="проверка орфографии">
    <ctrp name="name">zh_TW</str>
    <ctrp name="classname">solr.DirectSolrSpellChecker</str>
    <ctrp name="field">spellcheck_zh_tw</str>
</lst>
</searchComponent>

```

Раскомментируйте следующие блоки кода:

```

<!--
<имя обработчика запроса="/suggest"    класс="solr.SearchHandler">
    <lst name="defaults">
        <ctrp name="suggest">истина</str>
        <ctrp name="suggest.dictionary">по умолчанию</str>
        <ctrp name="suggest.count">5</str>
    </lst>
    <arr name="components">
        <str>предложить</str>
    </arr>
</requestHandler>
-->

<!--
<searchComponent name="suggest"      класс="solr.SuggestComponent">
    <lst name="suggester">
        <ctrp name="name">default</str>
        <ctrp name="dictionaryImpl">DocumentDictionaryFactory</str>
        <ctrp name="lookupImpl">FreeTextLookupFactory</str>
        <ctrp name="suggestAnalyzerFieldType">text_spell</str>
        <ctrp name="suggestFreeTextAnalyzerFieldType">text_spell</str>
        <ctrp name="field">autosuggest</str>
        <ctrp name="buildOnCommit">истина</str>
        <ctrp name="buildOnOptimize">истина</str>
        <ctrp name="exactMatchFirst">истина</str>
        <ctrp name="separator"><![CDATA[ ]]></str>
    </lst>
    <lst name="suggester">
        <ctrp имя="имя">ru</str>
        <ctrp name="dictionaryImpl">DocumentDictionaryFactory</str>
        <ctrp name="lookupImpl">FreeTextLookupFactory</str>
        <ctrp name="suggestAnalyzerFieldType">text_spell_en</str>
        <ctrp name="suggestFreeTextAnalyzerFieldType">text_spell_en</str>
        <ctrp name="field">autosuggest_en</str>
        <ctrp name="buildOnCommit">истина</str>
        <ctrp name="buildOnOptimize">истина</str>
        <ctrp name="exactMatchFirst">истина</str>
        <ctrp name="separator"><![CDATA[ ]]></str>
    </lst>
    <lst name="suggester">
        <ctrp имя="имя">cs</str>
        <ctrp name="dictionaryImpl">DocumentDictionaryFactory</str>
        <ctrp name="lookupImpl">FreeTextLookupFactory</str>
        <ctrp name="suggestAnalyzerFieldType">text_spell_cs</str>
        <ctrp name="suggestFreeTextAnalyzerFieldType">text_spell_cs</str>
        <ctrp name="field">autosuggest_cs</str>
        <ctrp name="buildOnCommit">истина</str>
        <ctrp name="buildOnOptimize">истина</str>
        <ctrp name="exactMatchFirst">истина</str>
        <ctrp name="separator"><![CDATA[ ]]></str>
    </lst>
    <lst name="suggester">

```

```

<стр имя="имя">де</стр>
<стр name="dictionaryImpl">DocumentDictionaryFactory</стр>
<стр name="lookupImpl">FreeTextLookupFactory</стр>
<стр name="suggestAnalyzerFieldType">text_spell_de</стр>
<стр name="suggestFreeTextAnalyzerFieldType">text_spell_de</стр>
<стр name="field">autosuggest_de</стр>
<стр name="buildOnCommit">истина</стр>
<стр name="buildOnOptimize">истина</стр>
<стр name="exactMatchFirst">истина</стр>
<стр name="separator"><![CDATA[ ]]></стр>

</lst>
<lst name="suggester">
    <стр имя="имя">ec</стр>
    <стр name="dictionaryImpl">DocumentDictionaryFactory</стр>
    <стр name="lookupImpl">FreeTextLookupFactory</стр>
    <стр name="suggestAnalyzerFieldType">text_spell_es</стр>
    <стр name="suggestFreeTextAnalyzerFieldType">text_spell_es</стр>
    <стр name="field">autosuggest_es</стр>
    <стр name="buildOnCommit">истина</стр>
    <стр name="buildOnOptimize">истина</стр>
    <стр name="exactMatchFirst">истина</стр>
    <стр name="separator"><![CDATA[ ]]></стр>
</lst>
<lst name="suggester">
    <стр name="name">es_CO</стр> name="dictionaryImpl">DocumentDictionaryFactory</стр>
    <стр name="lookupImpl">FreeTextLookupFactory</стр>
    <стр name="suggestAnalyzerFieldType">text_spell_es_co</стр>
    <стр name="suggestFreeTextAnalyzerFieldType">text_spell_es_co</стр>
    <стр name="field">autosuggest_es_co</стр>
    <стр name="buildOnCommit">истина</стр>
    <стр name="buildOnOptimize">истина</стр>
    <стр name="exactMatchFirst">истина</стр>
    <стр name="separator"><![CDATA[ ]]></стр>
</lst>
<lst name="suggester">
    <стр имя="имя">fr</стр>
    <стр name="dictionaryImpl">DocumentDictionaryFactory</стр>
    <стр name="lookupImpl">FreeTextLookupFactory</стр>
    <стр name="suggestAnalyzerFieldType">text_spell_fr</стр>
    <стр name="suggestFreeTextAnalyzerFieldType">text_spell_fr</стр>
    <стр name="field">autosuggest_fr</стр>
    <стр name="buildOnCommit">истина</стр>
    <стр name="buildOnOptimize">истина</стр>
    <стр name="exactMatchFirst">истина</стр>
    <стр name="separator"><![CDATA[ ]]></стр>
</lst>
<lst name="suggester">
    <стр имя="имя">привет</стр>
    <стр name="dictionaryImpl">DocumentDictionaryFactory</стр>
    <стр name="lookupImpl">FreeTextLookupFactory</стр>
    <стр name="suggestAnalyzerFieldType">text_spell_hi</стр>
    <стр name="suggestFreeTextAnalyzerFieldType">text_spell_hi</стр>
    <стр name="field">autosuggest_hi</стр>
    <стр name="buildOnCommit">истина</стр>
    <стр name="buildOnOptimize">истина</стр>
    <стр name="exactMatchFirst">истина</стр>
    <стр name="separator"><![CDATA[ ]]></стр>
</lst>
<lst name="suggester">
    <стр имя="имя">xy</стр>
    <стр name="dictionaryImpl">DocumentDictionaryFactory</стр>
    <стр name="lookupImpl">FreeTextLookupFactory</стр>
    <стр name="suggestAnalyzerFieldType">text_spell_hu</стр>
    <стр name="suggestFreeTextAnalyzerFieldType">text_spell_hu</стр>
    <стр name="field">autosuggest_hu</стр>
    <стр name="buildOnCommit">истина</стр>
    <стр name="buildOnOptimize">истина</стр>
    <стр name="exactMatchFirst">истина</стр>
    <стр name="separator"><![CDATA[ ]]></стр>
</lst>
<lst name="suggester">
    <стр имя="имя">идентификатор</стр>
    <стр name="dictionaryImpl">DocumentDictionaryFactory</стр>
    <стр name="lookupImpl">FreeTextLookupFactory</стр>
    <стр name="suggestAnalyzerFieldType">text_spell_id</стр>
    <стр name="suggestFreeTextAnalyzerFieldType">text_spell_id</стр>
    <стр name="field">autosuggest_id</стр>
    <стр name="buildOnCommit">истина</стр>
    <стр name="buildOnOptimize">истина</стр>
    <стр name="exactMatchFirst">истина</стр>
    <стр name="separator"><![CDATA[ ]]></стр>
</lst>
<lst name="suggester">
    <стр name="name">это</стр>

```

```

<ctrp name="dictionaryImpl">DocumentDictionaryFactory</str>
<ctrp name="lookupImpl">FreeTextLookupFactory</str>
<ctrp name="suggestAnalyzerFieldType">text_spell_it</str>
<ctrp name="suggestFreeTextAnalyzerFieldType">text_spell_it</str>
<ctrp name="field">autosuggest_it</str>
<ctrp name="buildOnCommit">истина</str>
<ctrp name="buildOnOptimize">истина</str>
<ctrp name="exactMatchFirst">истина</str>
<ctrp name="separator"><![CDATA[ ]]></str>

</lst>
<lst name="suggester">
    <ctrp имя="имя">я</str>
    <ctrp name="dictionaryImpl">DocumentDictionaryFactory</str>
    <ctrp name="lookupImpl">FreeTextLookupFactory</str>
    <ctrp name="suggestAnalyzerFieldType">text_spell_ja</str>
    <ctrp name="suggestFreeTextAnalyzerFieldType">text_spell_ja</str>
    <ctrp name="field">autosuggest_ja</str>
    <ctrp name="buildOnCommit">истина</str>
    <ctrp name="buildOnOptimize">истина</str>
    <ctrp name="exactMatchFirst">истина</str>
    <ctrp name="separator"><![CDATA[ ]]></str>
</lst>
<lst name="suggester">
    <ctrp имя="имя">ко</str>
    <ctrp name="dictionaryImpl">DocumentDictionaryFactory</str>
    <ctrp name="lookupImpl">FreeTextLookupFactory</str>
    <ctrp name="suggestAnalyzerFieldType">text_spell_ko</str>
    <ctrp name="suggestFreeTextAnalyzerFieldType">text_spell_ko</str>
    <ctrp name="field">autosuggest_ko</str>
    <ctrp name="buildOnCommit">истина</str>
    <ctrp name="buildOnOptimize">истина</str>
    <ctrp name="exactMatchFirst">истина</str>
    <ctrp name="separator"><![CDATA[ ]]></str>
</lst>
<lst name="suggester">
    <ctrp имя="имя">pl</str>
    <ctrp name="dictionaryImpl">DocumentDictionaryFactory</str>
    <ctrp name="lookupImpl">FreeTextLookupFactory</str>
    <ctrp name="suggestAnalyzerFieldType">text_spell_pl</str>
    <ctrp name="suggestFreeTextAnalyzerFieldType">text_spell_pl</str>
    <ctrp name="field">autosuggest_pl</str>
    <ctrp name="buildOnCommit">истина</str>
    <ctrp name="buildOnOptimize">истина</str>
    <ctrp name="exactMatchFirst">истина</str>
    <ctrp name="separator"><![CDATA[ ]]></str>
</lst>
<lst name="suggester">
    <ctrp имя="имя">pt</str>
    <ctrp name="dictionaryImpl">DocumentDictionaryFactory</str>
    <ctrp name="lookupImpl">FreeTextLookupFactory</str>
    <ctrp name="suggestAnalyzerFieldType">text_spell_pt</str>
    <ctrp name="suggestFreeTextAnalyzerFieldType">text_spell_pt</str>
    <ctrp name="field">autosuggest_pt</str>
    <ctrp name="buildOnCommit">истина</str>
    <ctrp name="buildOnOptimize">истина</str>
    <ctrp name="exactMatchFirst">истина</str>
    <ctrp name="separator"><![CDATA[ ]]></str>
</lst>
<lst name="suggester">
    <ctrp имя="имя">ru</str>
    <ctrp name="dictionaryImpl">DocumentDictionaryFactory</str>
    <ctrp name="lookupImpl">FreeTextLookupFactory</str>
    <ctrp name="suggestAnalyzerFieldType">text_spell_ru</str>
    <ctrp name="suggestFreeTextAnalyzerFieldType">text_spell_ru</str>
    <ctrp name="field">autosuggest_ru</str>
    <ctrp name="buildOnCommit">истина</str>
    <ctrp name="buildOnOptimize">истина</str>
    <ctrp name="exactMatchFirst">истина</str>
    <ctrp name="separator"><![CDATA[ ]]></str>
</lst>
<lst name="suggester">
    <ctrp имя="имя">ж</str>
    <ctrp name="dictionaryImpl">DocumentDictionaryFactory</str>
    <ctrp name="lookupImpl">FreeTextLookupFactory</str>
    <ctrp name="suggestAnalyzerFieldType">text_spell_zh</str>
    <ctrp name="suggestFreeTextAnalyzerFieldType">text_spell_zh</str>
    <ctrp name="field">autosuggest_zh</str>
    <ctrp name="buildOnCommit">истина</str>
    <ctrp name="buildOnOptimize">истина</str>
    <ctrp name="exactMatchFirst">истина</str>
    <ctrp name="separator"><![CDATA[ ]]></str>
</lst>
<lst name="suggester">
    <ctrp name="name">zh_TW</str>
    <ctrp name="dictionaryImpl">DocumentDictionaryFactory</str>

```

```

<ctrp name="lookupImpl">FreeTextLookupFactory</str>
<ctrp name="suggestAnalyzerFieldType">text_spell_zh_tw</str>
<ctrp name="suggestFreeTextAnalyzerFieldType">text_spell_zh_tw</str>
<ctrp name="field">autosuggest_zh_tw</str>
<ctrp name="buildOnCommit">истина</str>
<ctrp name="buildOnOptimize">истина</str>
<ctrp name="exactMatchFirst">истина</str>
<ctrp name="separator"><![CDATA[ ]]></str>
</lst>
</searchComponent>
-- >

```

После перезапуска сервера Solr новый метод подсказок активируется.

Перенаправления ключевых слов

Для получения дополнительной информации см.: [Настройка перенаправлений ключевых слов](#)

Все типы совпадений могут быть чувствительны к регистру.

Перенаправления зависят от конфигурации поиска и языка. По умолчанию мы поддерживаем пять типов соответствий запроса:

SAP Commerce предоставляет механизм перенаправления ключевых слов. Эта функция позволяет пользователям определять определенные слова, которые при сопоставлении во время поиска работают как триггеры, заставляя клиента перенаправляться на указанные результаты поиска.

- ТОЧНЫЙ:Запрос должен соответствовать заданной фразе.
- НАЧИНАЕТСЯ_С:Запрос должен начинаться с указанной фразы.
- ЗАВЕРШАЕТСЯ_С:Запрос должен заканчиваться указанной фразой.
- СОДЕРЖИТ:Запрос должен содержать указанную фразу.
- РЕГУЛЯРНОЕ ВЫРАЖЕНИЕ:Запрос должен соответствовать заданному выражению.

Сопутствующая информация

[Схема установки сервера Solr xml](#)
[Solr Cong xml](#) ↗

Установка сервера Solr для автономного режима

Узнайте, как установить и настроить автономный сервер Solr таким образом, чтобы его можно было использовать в SAP Commerce.

Быстрая установка и настройка

Контекст

Следуйте инструкциям по быстрой установке и настройке сервера Solr на локальном компьютере.

Процедура

- . Включите solrserver расширение в вашем localextensions.xml файле.

Расширение теперь готово к использованию. Для получения дополнительной информации о настройке расширений см. [Установка на основе указанных расширений](#).

- . Создайте экземпляр сервера Solr. Вы можете использовать экземпляр по умолчанию, включенный в solrserver расширение.

Конфигурация по умолчанию следующая:

```

solrserver.instances.default.autostart=true
solrserver.instances.default.mode=standalone
solrserver.instances.default.hostname=localhost

```

```
solrserver.instances.default.port=8983
solrserver.instances.default.memory=512m
```

Если вам нужно, вы можете переопределить некоторые свойства в вашем местные свойства. Для получения дополнительной информации см.: [Расширение solrserver](#).

. Запустите сервер SAP Commerce.

Поскольку автозапуск включен для экземпляра сервера Solr по умолчанию, сервер Solr запускается и останавливается вместе с платформой. Для получения дополнительной информации см.: [Установка SAP Commerce вручную](#) и [Установка SAP Commerce с использованием рецептов установщика](#).

. Настройте SAP Commerce для использования сервера Solr.

Наиболее важными атрибутами, которые следует учитывать при обновлении конфигурации сервера Solr, являются:

- Режим: установите автономный.
- URL-адреса конечных точек: должны содержать один главный URL-адрес со следующим значением: `http://localhost:8983/solr`.

Подробную информацию о том, как легко настроить параметры сервера Solr с помощью панели администрирования Backoffice, см. здесь: [Создать конфигурацию поиска по фасетам](#).

Установка сервера Solr

Использование расширения solrserver

Включив `solrserver` расширение в вашем `localextensions.xml`, вы получаете экземпляр сервера Solr, готовый к использованию. Конфигурация по умолчанию гарантирует, что автономный сервер Solr настраивается, запускается и останавливается вместе с платформой. Это удобно в средах CI или разработки. Обратите внимание, что свойство `mode` должно быть установлено на `standalone`.

Следующий код показывает пример конфигурации:

```
# отключает автозапуск для экземпляра Solr по умолчанию
solrserver.instances.default.autostart=false

solrserver.instances.standalone.autostart=true
solrserver.instances.standalone.mode=автономный
solrserver.instances.standalone.hostname=локальный
solrserver.instances.standalone.port=8983
solrserver.instances.standalone.memory=512m
```

Использование внешнего автономного сервера

Каждая версия SAP Commerce может поддерживать более одной версии Solr для обеспечения совместимости. Каждая версия включает в себя каталог сервера, содержащий сервер и примененные настройки, а также каталог настроек, содержащий модификации, которые можно применить к официальному релизу Solr. Для шагов установки см. раздел Установка в [Поддержка нескольких версий Solr](#).

Для запуска и остановки сервера Solr используйте следующие команды:

-Примечание

В системах на базе Unix скрипт `bin/solr` может быть неисполнимым по умолчанию. Чтобы сделать его исполняемым, выполните следующую команду в корневом каталоге сервера Solr:

```
:#chmod +x bin/solr
```

- В системах Linux:

```
# ./bin/solr старт -р 8983
Ожидание до 30 секунд, чтобы увидеть Solr, работающий на порту 8983 [/]
Запущен сервер Solr на порту 8983 (pid=23092). Удачного поиска!
```

. /bin/solr стоп -р 8983

Отправка команды остановки Solr, работающему на порту 8983... ожидание 5 секунд, чтобы процесс Jetty 23092 остановился

- В системах Windows:

bin\solr.cmd старт -р 8983

Ожидание до 30, чтобы увидеть Solr, работающий на порту 8983. Запустил сервер Solr на порту 8983. Удачного поиска!

bin\solr.cmd стоп -р 8983

Остановка процесса Solr 6256, работающего на порту 8983.

Ожидание 0 секунд. Нажмите клавишу для продолжения...

Конфигурация кластера

В этом разделе описывается, как настроить кластер серверов Solr с использованием репликации. Это особенно полезно, когда:

- У вас высокая нагрузка на систему из-за поисковых запросов, и вы хотите распределить нагрузку между несколькими серверами.
- Индексация потребляет много ресурсов, поэтому вам необходимо разделить индексацию и поиск.

В конфигурации кластера есть ведущий сервер, на котором происходит индексация, и один или несколько подчиненных серверов. Все индексы на ведущем сервере реплицируются на каждый из подчиненных серверов, так что и ведущий, и подчиненные серверы имеют одинаковые индексы.

-Примечание

Полная операция индексации не работает, если подчиненный сервер не запущен. Однако простой сервера не является проблемой во время запросов или обновлений индекса. В таких ситуациях он только уменьшает общую емкость кластеризованной установки Solr. Если ваш подчиненный экземпляр Solr отключен во время обновлений, ваш главный экземпляр Solr индексируется и обслуживает входящие запросы.

Вы можете включить функцию репликации индекса в экземпляре Solr из коробки. Запросы на репликацию обрабатываются ReplicationHandler, который может быть настроен в solrconfig.xml. По умолчанию соответствующий раздел в solrconfig.xml закомментирован:

```
<имя обработчика запроса="/replication" class="solr.ReplicationHandler" >
  <!--
    Чтобы включить простую репликацию master/slave, раскомментируйте один из
    разделов ниже, в зависимости от того, должен ли этот экземпляр solr быть «master»
    или «slave». Если этот экземпляр является «slave», вам также нужно будет заполнить
    masterUrl, чтобы указать на реальную машину.

  -->
  <lst name="мастер">
    <стр name="replicateAfter">коммит</стр> name="replicateAfter">запуск</стр>
    <стр name="confFiles">schema.xml,stopwords.txt,synonyms.txt</стр>
    <стр
  </lst>
  <!--
    <lst name="раб">
      <стр name="masterUrl">http://your-master-hostname:port/solr/${solr.core.name}/replication</стр>
      <стр name="pollInterval">00:00:60</стр>
    </lst>
  -->
</requestHandler>
```

Ниже приведены примеры конфигурации с использованием solrserver расширение:

- местные.свойства:

```
solrserver.instances.master.autostart=true
solrserver.instances.master.mode=standalone
solrserver.instances.master.hostname=localhost
solrserver.instances.master.port=8983
solrserver.instances.master.memory=512m
```

```
solrserver.instances.slave.autostart=true
solrserver.instances.slave.mode=standalone
solrserver.instances.slave.hostname=localhost
solrserver.instances.slave.port=8984
solrserver.instances.slave.memory=512m
```

- config/solr/instances/master/configsets/default/conf/solrconfig.xmlле:

```
<имя обработчика запроса="/replication" class="solr.ReplicationHandler" >
<lst name="мастер">
<стр name="replicateAfter">коммит</стр> name="replicateAfter">запуск</стр>
<стр name="confFiles">schema.xml,stopwords.txt,synonyms.txt</стр>
<стр
</lst>
</requestHandler>
```

- config/solr/instances/slave/configsets/default/conf/solrconfig.xmlле:

```
<имя обработчика запроса="/replication" class="solr.ReplicationHandler" >
<lst name="раб">
<стр name="masterUrl">http://your-master-hostname:port/solr/${solr.core.name}/replication</стр>
<стр name="pollInterval">00:00:30</стр>
</lst>
</requestHandler>
```

Конфигурация SAP Commerce в административной панели бэк-офиса

Обновите конфигурацию сервера Solr с помощью панели администрирования Backoffice.

Наиболее важные характеристики, которые следует учитывать:

- Режим: установите автономный.
- URL-адреса конечных точек: для URL-адресов серверов, входящих в кластер, один из них должен быть установлен как главный, например, <http://master:8983/solr>, <http://slave:8983/solr>.

Подробную информацию о том, как легко настроить параметры сервера Solr с помощью Backoffice Administration Cockpit, см.[Настройте сервер Solr](#).

Расширенная конфигурация сервера Cloud Solr

Узнайте, как установить и настроить облачный сервер Solr таким образом, чтобы его можно было использовать в SAP Commerce.

Использование расширения solrserver

Включив `solrserver` в вашем `localextensions.xml` вы получите экземпляр сервера Solr, готовый к использованию. Для быстрого руководства по установке см.[Быстрая установка сервера Cloud Solr](#).

Использование внешнего автономного сервера

Каждая версия SAP Commerce имеет возможность поддерживать более одной версии Solr для обеспечения совместимости. Каждая версия включает в себя каталог сервера, содержащий сервер и примененные настройки, а также каталог настроек, содержащий модификации, которые могут быть применены к официальному релизу Solr. Для шагов установки см. раздел Установка в [Поддержка нескольких версий Solr](#).

Для запуска и остановки сервера Solr используйте следующие команды:

-Примечание

В системах на базе unix скрипт bin/solr может быть неисполнимым по умолчанию. Чтобы сделать его исполняемым, выполните следующую команду в корневом каталоге сервера Solr:

```
#chmod +x bin/solr
```

- В системах Linux:

```
# ./bin/solr старт -р 8983
```

Ожидание до 30 секунд, чтобы увидеть Solr, работающий на порту 8983 [/]

Запущен сервер Solr на порту 8983 (pid=23092). Удачного поиска!

```
# ./bin/solr стоп -р 8983
```

Отправка команды остановки Solr, работающему на порту 8983... ожидание 5 секунд, чтобы процесс Jetty 23092 остановился

- В системах Windows:

```
# bin\solr.cmd старт -р 8983
```

Ожидание до 30, чтобы увидеть Solr, работающий на порту

8983. Запустил сервер Solr на порту 8983. Удачного поиска!

```
# bin\solr.cmd стоп -р 8983
```

Остановка процесса Solr 6256, работающего на порту 8983.

Ожидание 0 секунд. Нажмите клавишу для продолжения...

Аутентификация и авторизация Zookeeper

По умолчанию аутентификация и авторизация не применяются, когда сервер пытается присоединиться к квorumу. Вот почему рекомендуется включить их.

Инструкции о том, как это сделать, см. [Плагины авторизации и аутентификации в режиме Solr Cloud](#)



Конфигурация кластера

В этом разделе описывается, как настроить кластер серверов Solr с помощью SolrCloud. Это особенно полезно, когда вам нужны расширенные возможности отказоустойчивости и высокой доступности.

SolrCloud — это гибкий распределенный поиск и индексация без ведущего узла для выделения узлов, сегментов и реплик. Вместо этого Solr использует ZooKeeper для управления этими расположениями в зависимости от файлов конфигурации и схем. Запросы и обновления можно отправлять на любой сервер. Solr использует информацию в базе данных ZooKeeper, чтобы выяснить, какие серверы должны обрабатывать запрос.

Более подробную информацию об облаке см. здесь: <https://cwiki.apache.org/confluence/display/solr/SolrCloud>



Пример конфигурации (с использованием solrserver расширение):

- местные.свойства:

```
solrserver.instances.cloud1.autostart=false
solrserver.instances.cloud1.mode=cloud
solrserver.instances.cloud1.port=8983
solrserver.instances.cloud1.memory=512m
solrserver.instances.cloud1.zk.host=
solrserver.instances.cloud1.zk.upconfig=true
solrserver.instances.default.autostart=false
solrserver.instances.cloud1.hostname=<hostIPAddress> / локальный хост
```

```
solrserver.instances.cloud2.autostart=false
solrserver.instances.cloud2.mode=cloud
solrserver.instances.cloud2.port=8984
solrserver.instances.cloud2.memory=512m
solrserver.instances.cloud2.zk.host=localhost:9983
```

```
solrserver.instances.cloud2.zk.upconfig=false
solrserver.instances.cloud2.hostname=<IP-адрес_хоста> / локальный хост
```

-Примечание

Обратите внимание, что автозапуск серверов отключен, поскольку порядок запуска серверов Solr не может быть гарантирован. В этом примере сервер Solr для экземпляра cloud1 должен быть запущен до сервера для экземпляра cloud2. Это связано с тем, что сервер Solr для экземпляра cloud1 также запускает встроенный сервер ZooKeeper.

- Запуск серверов:

```
# муравей startSolrServer - Имя_экземпляра=cloud1
# муравей startSolrServer - Имя_экземпляра=cloud2
```

Конфигурация в административной панели бэк-офиса

Обновите конфигурацию сервера Solr с помощью панели администрирования Backoffice.

Наиболее важные характеристики, которые следует учитывать:

- Режим: установите его на облачно.
- URL-адреса конечных точек: должны содержать один или несколько URL-адресов серверов ZooKeeper. URL-адреса должны соответствовать шаблону {имя_хоста}: {порт}: localhost:9983

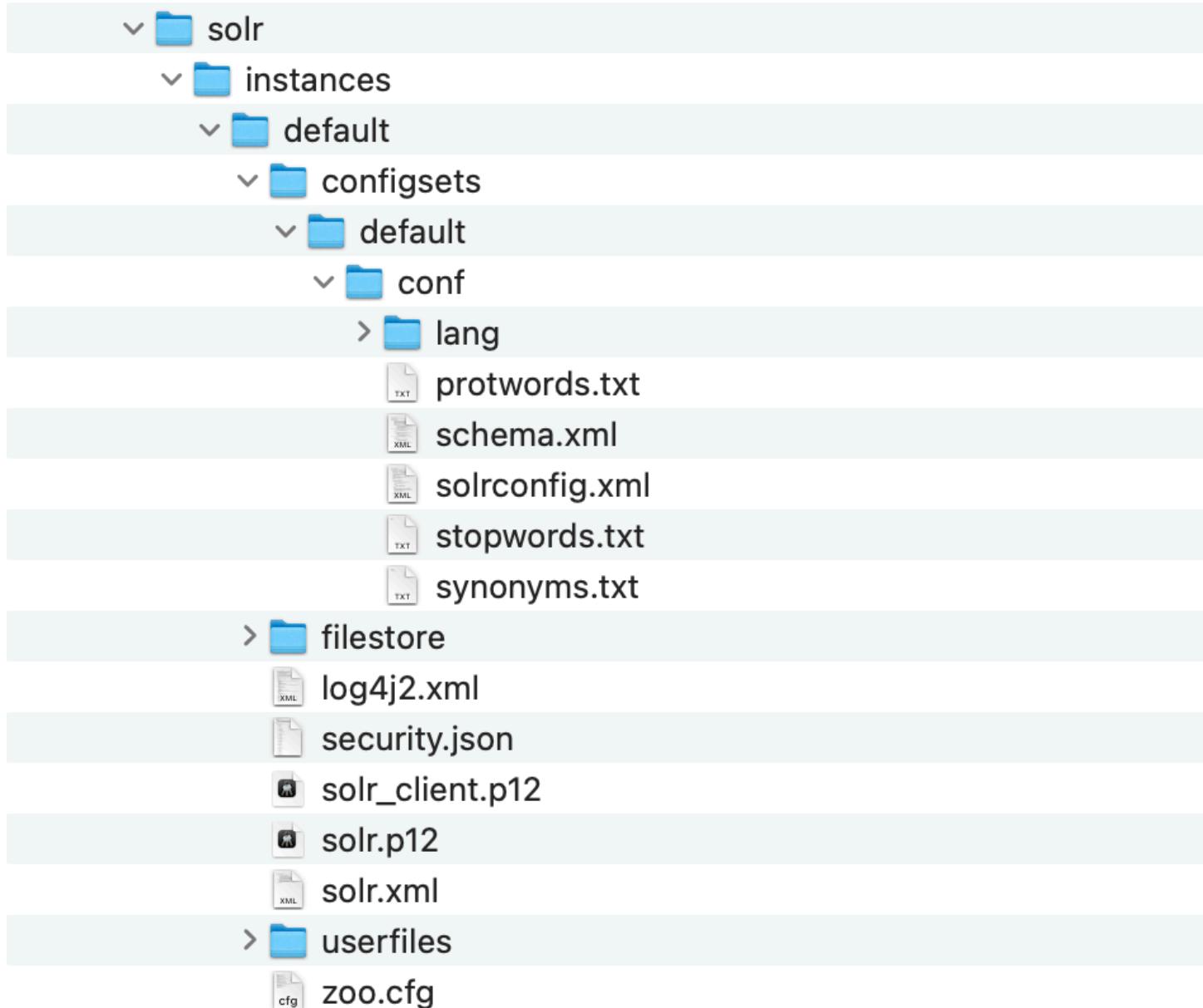
Подробную информацию о том, как легко настроить параметры сервера Solr с помощью панели администрирования Backoffice, см. здесь:[Создать конфигурацию поиска по фасетам](#).

Настройка и управление экземплярами Solr

Для правильной работы необходимо настроить сервер Solr с помощью набора свойств.

Файлы конфигурации сервера Solr

Каждый экземпляр имеет свой собственный набор файлов конфигурации, который размещается в каталоге конфигурации экземпляра. Эти файлы создаются автоматически при создании экземпляра. По умолчанию он имеет структуру, похожую на следующую:



Общие настройки экземпляра Solr

Определите общие настройки для ваших экземпляров с помощью свойства проекта `Свойства экземпляра Solr`.

Управление экземплярами Solr

Экземплярами и серверами Solr можно управлять:

- Автоматически: серверы Solr для экземпляров с автозапуском=истиной запускаются при инициализации глобального контекста Spring и останавливаются при закрытии контекста. Подробнее см.[Автоматический запуск сервера Solr](#).
- Вручную: Вы можете использовать задачи Ant для управления своими экземплярами. Подробности см.[Задачи Ant сервера Solr](#).

-Примечание

Если при запуске сервера Solr соответствующий экземпляр Solr не существует, он создается автоматически.

Свойства экземпляра Solr

Функциональность, предоставляемая расширением, позволяет управлять более чем одним экземпляром. Каждый экземпляр имеет имя и может быть настроен с использованием свойств `вместные.свойства`.

Вам необходимо настроить keyStore и trustStore Свойства для режимов сервера и клиента. Ознакомьтесь со следующей таблицей, чтобы узнать больше о свойствах экземпляра:

Имя свойства	Значение по умолчанию	Описание
solrserver.instances.<имя экземпляра>.autostart	ЛОЖЬ	Если установлено значение true, данный экземпляр Solr запускается/останавливается автоматически вместе с платформой.
solrserver.instances.<имя экземпляра>.mode	автономный	Режим, в котором должен быть запущен сервер Solr: он может быть автономным или облачным.
solrserver.instances.<имя экземпляра>.zk.host		Запустите Solr с определенной строкой подключения ZooKeeper. Если эта опция не указана, Solr запустит встроенный экземпляр ZooKeeper. Это также используется для загрузки конфигурации. Пример: сервер1:2181, сервер2:2181.
solrserver.instances.<имя экземпляра>.zk.upcong	истинный	Если установлено значение true, конфигурация будет загружена на сервер ZooKeeper после запуска сервера Solr.
solrserver.instances.<имя экземпляра>.zk.prop.<имя свойства>		Устанавливает общекластерные свойства в ZooKeeper.
solrserver.instances.<имя экземпляра>.hostname	локальный хост	Запускает сервер Solr с определенным именем хоста.
solrserver.instances.<имя экземпляра>.port	8983	TCP-порт, используемый сервером Solr (каждый экземпляр должен использовать свой порт).
solrserver.instances.<имя экземпляра>.cong.dir	<HYBRIS_CONFIG_DIR>/solr/instances/<имя экземпляра>	Каталог, содержащий конфигурацию экземпляра Solr.
solrserver.instances.<имя экземпляра>.data.dir	<HYBRIS_DATA_DIR>/solr/instances/<имя экземпляра>	Каталог, содержащий данные экземпляра Solr.
solrserver.instances.<имя экземпляра>.log.dir	<HYBRIS_LOG_DIR>/solr/instances/<имя экземпляра>	Каталог, содержащий журналы экземпляров Solr.
solrserver.instances.<имя экземпляра>.memory	512 м	Объем памяти (кучи), который будет использоваться процессом сервера Solr.
solrserver.instances.<имя экземпляра>.javaoptions		Дополнительные параметры командной строки, которые будут использоваться процессом сервера Solr.
solrserver.instances.<имя экземпляра>.authtype	базовый	Тип аутентификации, поддерживается только базовый. Пустое значение отключает аутентификацию.
solrserver.instances.<имя экземпляра>.user		Пользователь, используемый для аутентификации.
solrserver.instances.<имя экземпляра>.пароль		Пароль, используемый для аутентификации.
solrserver.instances.<имя экземпляра>.ssl.enabled	истинный	Если установлено значение true, SSL включен.
Для конфигурации режима сервера		
solrserver.instances.<имя экземпляра>.ssl.keyStoreType	ПКС12	Тип хранилища ключей.
solrserver.instances.<имя экземпляра>.ssl.keyStore	Файл solr.p12 в каталоге конфигурации экземпляра	Файл keyStore.

Имя свойства	Значение по умолчанию	Описание
solrserver.instances.<имя экземпляра>.ssl.keyStorePassword		Пароль хранилища ключей.
solrserver.instances.<имя экземпляра>.ssl.trustStoreType	ПКС12	Тип trustStore.
solrserver.instances.<имя экземпляра>.ssl.trustStore	Файл solr.p12 в каталоге конфигурации экземпляра	Файл trustStore.
solrserver.instances.<имя экземпляра>.ssl.trustStorePassword		Пароль trustStore.
Для конфигурации клиентского режима		
solrserver.instances.<имя экземпляра>.ssl.client.keyStoreType	ПКС12	Тип хранилища ключей клиента.
solrserver.instances.<имя экземпляра>.ssl.client.keyStore	Файл solr.p12 в каталоге конфигурации экземпляра	Файл keyStore.
solrserver.instances.<имя экземпляра>.ssl.client.keyStorePassword		Пароль хранилища ключей клиента.
solrserver.instances.<имя экземпляра>.ssl.client.trustStoreType	ПКС12	Тип trustStore клиента.
solrserver.instances.<имя экземпляра>.ssl.client.trustStore	Файл solr.p12 в каталоге конфигурации экземпляра	Клиентский файл trustStore.
solrserver.instances.<имя экземпляра>.ssl.client.trustStorePassword		Пароль trustStore клиента.
solrserver.instances.<имя экземпляра>.ssl.needClientAuth	ЛОЖЬ	Если установлено значение true, клиентам необходимо пройти аутентификацию.
solrserver.instances.<имя экземпляра>.ssl.wantClientAuth	ЛОЖЬ	Если установлено значение true, разрешить клиентам аутентификацию (но не обязательно).

Автоматический запуск сервера Solr

Процедура

- . Проверьте, запущен ли соответствующий Solr. Если нет, перейдите к шагу 6.
- . Проверьте, является ли запущенный сервер Solr правильным. Если нет, выдайте ошибку.
- . Если принудительно Перезапустить свойство установлено в false перейдите к шагу 6.
- 6. . Остановите сервер Solr.
- . Проверьте, правильно ли остановлен сервер Solr. Если нет, выдайте ошибку..
- Запустите сервер Solr.
- . Проверьте, правильно ли запущен сервер Solr. Если нет, выдайте ошибку.

Некоторые части алгоритма можно настроить с помощью свойств `вместные.свойства` ie. Доступны следующие свойства:

Имя свойства	Значение по умолчанию	Описание
solrserver.forceПерезапуск	истинный	Если при запуске платформы уже запущен соответствующий сервер Solr, он будет перезапущен.
solrserver.failOnError	истинный	Если во время запуска любого сервера Solr возникнет ошибка, она рассматривается как серьезная ошибка, и вся платформа не будет работать.

Задачи Ant сервера Solr

The solrserver расширение предоставляет некоторые дополнительные задачи Ant, которые можно использовать для управления экземплярами и серверами Solr.

Используя параметр, можно переопределить некоторые свойства экземпляра. Имя экземпляра по умолчанию — default. Доступны следующие задачи ant:

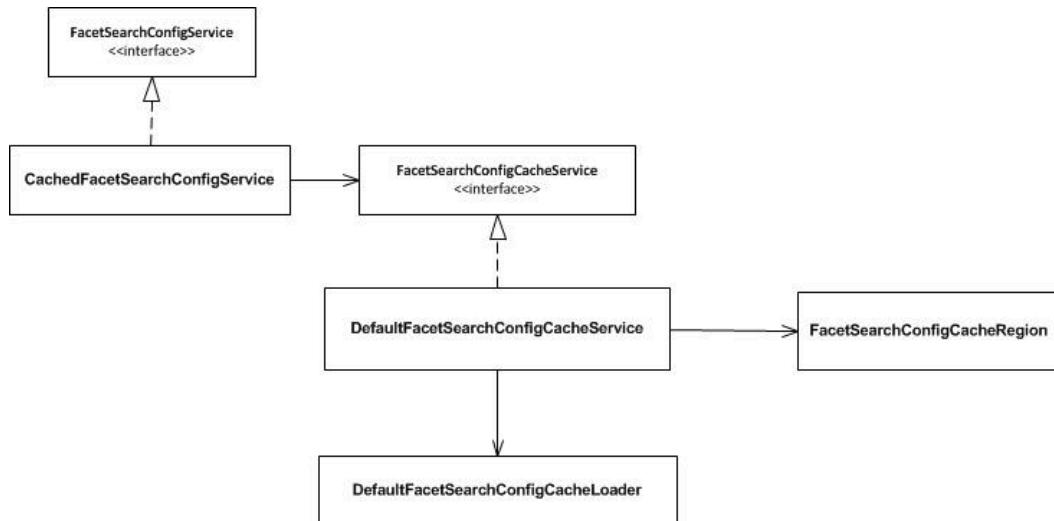
Название задачи	Описание	Параметры
createSolrInstance	Создает экземпляр Solr.	разрешенные параметры: • - Имя_экземпляра
удалитьSolrInstance	Удаляет экземпляр Solr. Это означает удаление всех файлов конфигурации, данных и журналов.	разрешенные параметры: • - Имя_экземпляра
загрузитьSolrConfig	Загружает конфигурацию в экземпляр ZooKeeper.	разрешенные параметры: • - Имя_экземпляра
startSolrServer	Запускает сервер Solr.	разрешенные параметры: • - Имя_экземпляра
startSolrServers	Запускает все серверы Solr, которые имеют автозапуск установлено значение true.	
stopSolrServer	Останавливает сервер Solr.	разрешенные параметры: • - Имя_экземпляра
stopSolrServers	Останавливает все серверы Solr, которые имеют автозапуск установлено значение true.	
настроитьSolrServer	Применяет настройки SAP Commerce к официальному выпуску Solr.	разрешенные параметры: • - DsolrServerPath

Кэширование конфигурации Solr

В этом документе описывается кэш конфигурации Solr.

Кэш-конструкция поиска граней была создана для улучшения производительности функциональности поиска solr. Кэш-конструкция используется cachedFacetSearchCongService, который определен в solrfacetsearch-spring.xml. По умолчанию ему назначен псевдоним facetSearchCongService.

На рисунке представлен обзор структуры класса, используемой для реализации кэша конфигурации поиска фасетов.



Область кэширования для объекта FacetSearchCong

Кэш для объектов FacetSearchCong был реализован как один из кэшей региона платформы hybris. Его можно отслеживать в SAP Commerce Administration Console (Administration Console).

The screenshot shows the SAP Commerce Administration Console interface. In the top navigation bar, the 'Console' tab is selected. Below it, the 'Cache' section is open. On the left, there's a tree view with nodes like 'typesystemCacheRegion', 'entityCacheRegion', 'queryCacheRegion', 'mediaCacheRegion', 'scriptsCacheRegion', and 'defaultFacetSearchConfigCacheRegion'. The 'defaultFacetSearchConfigCacheRegion' node is expanded. To its right, there's a summary table with metrics: Max Size (500), Max Reached Size (3), Hit/Miss Ratio (92%), and a detailed table showing 1 entry for '_FACET_SEARCH_CONFIG_'. The table has columns: Type, Hits, Misses, Ratio, Invalidations, Evictions, Fetches, and InstanceCount. The entry shows 38 Hits, 3 Misses, 93% Ratio, 0 Invalidations, 0 Evictions, 3 Fetches, and 3 InstanceCount. Below the table, it says 'Showing 1 to 1 of 1 entries'. At the bottom left, there's a 'Clear cache' button.

Определение этого региона можно найти в global-solrfacetsearch-spring.xmlле, как показано в следующем примере:

```
<!-- Кэш конфигурации поиска фасетов Solr -->
<имя псевдонима="defaultFacetSearchConfigCacheRegion" псевдоним="facetSearchConfigCacheRegion" /> <имя
компоненты="defaultFacetSearchConfigCacheRegion" class="de.hybris.platform.solrfacetsearch.config.cache.impl.
    <конструктор-аргумент имя="имя" значение="defaultFacetSearchConfigCacheRegion"/> имя="maxEntries"
    <конструктор-аргумент значение="${facetsearchconfig.cache.maxEntries}"/> имя="statsEnabled" значение="$
    <конструктор-аргумент {regioncache.stats.enabled}">
    <имя свойства="handledTypes">
        <массив>
            <value>_FACET_SEARCH_CONFIG_</value>
        </массив>
    </имя>
</свойство>
</боб>
<bean id="facetSearchConfigCacheRegistrar" class="de.hybris.platform.regioncache.region.CacheRegis
```

Смотрите также:[Региональный кэш](#).

Аннулирование кэша

Данные FacetSearchCong создаются на основе многих объектов модели, и их необходимо аннулировать каждый раз, когда изменяется один из этих объектов. Механизм аннулирования был реализован на основе предположения, что поисковый конг не меняется очень часто. Весь кэш очищается всякий раз, когда изменяется любой объект модели типа, относящегося к данным FacetSearchCong. InvalidationListener определен в DefaultFacetSearchCongCacheService.

Для получения дополнительной информации см.:[Указание развертывания для типов платформы hybris](#).

В файле создается набор кодов типов, для которых кэш должен быть признан недействительным.solrfacetsearch-spring.xmlле. Значения кода типа соответствуют коду типа атрибут из элемента <deployment> в определении типа (...elements.xml).

Следующий фрагмент кода взят изsolrfacetsearch-spring.xmlле:

```
<bean id="defaultFacetSearchConfigInvalidationTypeSet"
    class="de.hybris.platform.solrfacetsearch.config.cache.FacetSearchConfigInvalidationTypeSet"> <конструктор-аргумент>
    <набор>
        <value>2200</value>      <!--SolrFacetSearchConfig <-->
        <value>2204</value>      <!--SolrValueRangeSet --> <!--
        <value>2205</value>      <!--SolrValueRange -->
        <value>2206</value>      <!--SolrSearchConfig      -->
        <value>2207</value>      <!--SolrIndexedType      -->
        <value>2208</value>      <!--SolrIndexedProperty <!-->
        <value>2209</value>      <!--SolrIndexerQuery --> <!--
        <value>2210</value>      <!--SolrIndexerQueryParameter <!-->
        <value>2211</value>      <!--SolrIndexConfig      -->
        <value>2212</value>      <!--SolrServerConfig     -->
        <value>2213</value>      <!--SolrEndpointUrl     -->
```

```

<value>2218</value>      <!--SolrSynonymConfig --> <!--
<value>2221</value>      SolrAbstractKeywordRedirect <!--
<value>2222</value>      SolrStopWord -->
<value>2223</value>      <!--SolrFacetSearchKeywordRedirect -->
<значение>32</значение> <!--Язык -->
<значение>33</значение> <!--Валюта -->
<value>601</value>       <!--ВерсияКаталога -->
<значение>610</значение> <!--ClassAttributeAssignment --> <!--
<value>2013</value>      BaseStore -->

</набор>
</конструктор-арг>
</боб>

```

Набор кодов типов можно расширить, добавив еще один компонент типа `de.hybris.platform.solrfacetsearch.config.cache.FacetSearchConfigInvalidationTypeSet`.

Пример конфигурации:

```

<bean id="someTypeSet" class="de.hybris.platform.solrfacetsearch.config.cache.FacetSearchConfigInvalidationTypeSe
<конструктор-аргумент>
<набор>
<value>9251</value>
<value>9256</value>
</набор>
</конструктор-арг>
</боб>

```

Добавление валют в поисковую систему Solr Facet

Добавление новой валюты в витрину магазина и назначение ее продуктам позволяет отображать продукты в этой валюте. Прежде чем клиенты смогут фильтровать результаты поиска по валюте, необходимо добавить валюту в поиск фасета Solr.

Предпосылки

- Вы добавили валюту в нужную витрину.
- Вы назначили цены на свои продукты в валюте.

Контекст

Прежде чем клиенты смогут фильтровать свои продукты, необходимо добавить новую валюту в конфигурацию фасета Solr и назначить диапазон для валюты.

Процедура

- В Backoffice назначьте валюту фасету Solr config. . Перейти к [Поиск Помощь Диапазоны](#). . Создать новый 
- [SolrValueRangeSet](#) для новой валюты. . Введите имя, выберите [двойной тип диапазона](#) и щелкните [Сделанный](#). Набор диапазонов создан и открыт в редакторе.
- На [Диапазоны](#) вкладка, под [Существенный](#), введите квалификатор для заданного диапазона. .
- Под [Диапазоны](#), добавьте новый диапазон к типу.
- Убедитесь, что максимальная стоимость диапазона как минимум равна максимальной цене вашей продукции.
- На [Администрирование](#) вкладка, под [Несвязанный](#), выберите несвязанное свойство [цена](#).
- Выполните полную индексацию по правильному индексу Solr.

Результаты

Созданная вами валюта и диапазон отображаются на витрине.

Солр Безопасность

Функции безопасности Solr включают поддержку шифрования связи с Solr (а также между узлами Solr) с использованием SSL и поддержку аутентификации и авторизации, предоставляемых фреймворками безопасности Solr. Функции безопасности включены по умолчанию, когда вы используете solrserverрасширение. В другом случае вы можете использовать предоставленный пример конфигурации, который вы можете легко включить.

При установке Apache Solr следует учитывать следующие соображения безопасности:

- Хотя Apache Solr предоставляет некоторые функции безопасности, не рекомендуется выставлять его напоказ внешнему миру, например, Интернету. Серверы Solr следует размещать в демилитаризованной зоне (DMZ) за экраном.
- Всегда используйте безопасный клиентский компьютер и веб-браузер при использовании интерфейса администрирования Apache Solr. Помните о риске [CSRF](#) атаки - всегда завершайте сеанс браузера и выходите из системы после завершения своей задачи, а также избегайте открытия потенциально поврежденных веб-сайтов и писем электронной почты во время аутентификации с помощью консоли администрирования.
- Следуйте стандартным процедурам защиты веб-приложений, например:
 - Серверы Solr не следует запускать с правами root или администратора.
 - Права доступа к файлам конфигурации и файлам данных должны быть ограничены.
- Если вы индексируете пользовательский контент, такой как обзоры или комментарии, то имейте в виду, что атаки межсайтового скрипtingа (XSS) представляют риск, если результаты поиска отображаются. Чтобы избежать этого риска, контент необходимо правильно экранировать, чтобы предотвратить его неправильную интерпретацию.

Безопасная связь с использованием SSL

Для обеспечения безопасной связи с использованием SSL вам необходимо:

- действительный сертификат SSL.
- установить системные свойства, связанные с SSL.

Мы предоставляем образец `solr.jksle`, содержащий сертификаты, которые могут быть использованы для разработки.

-Примечание

Образцы сертификатов SSL предоставляются для сред разработки и позволяют использовать только localhost. Для производственных сред следует генерировать новые сертификаты. Система должна доверять новым сертификатам, поэтому не забудьте добавить их в хранилище доверия JVM по умолчанию или настроить дополнительное хранилище доверия таким же образом, как платформа делает для образцов сертификатов. Для примера см. `расширенные.свойства` в платформа/ресурсах:

```
# Дополнительное хранилище доверия. Если настроено хранилище доверия (в формате JKS), оно добавляется как резервное хранилище доверия
# JVM. Его цель — предоставить доверенный самоподписанный сертификат CA для разработчиков/тестировщиков convenie
additional(javax.net.ssl.trustStore=${platformhome}/resources/devcerts/ydevelopers.jks
additional(javax.net.ssl.trustStorePassword=ваsh_пароль
```

Подробную информацию о различных вариантах включения SSL см. [Включение SSL](#).

Конфигурация сервера Solr

Использование расширения `solrserver`

Чтобы включить SSL при использовании `solrserver`расширение, добавьте следующие свойства в `местные.свойства`:

<code>solrserver.instances.<экземпляр></code>	<code>имя>.ssl.enabled=true</code>
<code>solrserver.instances.<экземпляр></code>	<code>имя>.ssl.keyStorePassword=ваsh_пароль</code>
<code>solrserver.instances.<экземпляр></code>	<code>имя>.ssl.trustStorePassword=ваsh_пароль</code>

Конфигурация по умолчанию предполагает, что существует файл `keyStore/trustStore` с именем `solr.jksb` каталоге конфигурации экземпляра.

При использовании облачного режима (SolrCloud) добавьте также следующее свойство:

`solrserver.instances.<имя экземпляра>.zk.prop.urlScheme=https`

-Примечание

Некоторые общекластерные свойства (такие какurlScheme) необходимо настроить до запуска любого узла Solr. Сервер Solr будет перезапущен, чтобы убедиться, что конфигурация применена правильно.

Чтобы полностью отключить SSL, установите следующие свойства:

<code>solrserver.экземпляры.<экземпляр</code>	<code>имя>.ssl.enabled=false</code>
<code>solrserver.экземпляры.<экземпляр</code>	<code>имя>.ssl.keyStorePassword=</code>
<code>solrserver.экземпляры.<экземпляр</code>	<code>имя>.ssl.trustStorePassword=</code>
<code>solrserver.экземпляры.<экземпляр</code>	<code>имя>.zk.prop.urlScheme=http</code>

Использование внешнего автономного сервера

SSL не включен по умолчанию при использовании внешнего автономного сервера. Чтобы включить его, следуйте инструкциям в[Включение SSL](#) Для . Чтобы было проще вас мы предоставляем примерную конфигурацию, включающую:

- асервер/solr/solr.jksфайл, содержащий образцы сертификатов,
- abin/solr.in.cmdsbin/solr.in.shфайлы, определяющие пример конфигурации (они оба прокомментированы по умолчанию).

Конфигурация сервера Solr в административной панели бэк-офиса

Единственное необходимое изменение — это изменение URL-адреса сервера Solr с http на https. Например, вместо`http://localhost:8983/solr`это должно быть `https://localhost:8983/solr`. Используйте панель администрирования Backoffice для доступа к конфигурации сервера Solr и установки нового значения:[Настройте сервер Solr](#).

Поддержка аутентификации и авторизации для Solr

Вся конфигурация аутентификации и авторизации хранится вбезопасность.jsonле. Образецбезопасность.jsonпредоставляется файл, содержащий следующих пользователей:

-Примечание

Для каждого имени пользователя, указанного в таблице, используйте пароль, который вы определили для этой учетной записи.

Пользователь	Описание
солрадмин	пользователь-администратор, может выполнять любые операции
solrserver	для связи между серверными узлами
solrclient	может выполнять поисковые запросы
solrindexingclient	может выполнять поисковые запросы и индексацию

Подробную информацию о различных опциях и плагинах см.[Плагины аутентификации и авторизации](#).



-Примечание

Для производственных сред необходимо сгенерировать новые пароли, например, пароль можно изменить с помощью API Solr. Подробности см.[Добавление пользователя или редактирование пароля](#).



Конфигурация сервера Solr

Использование расширения solrserver

Чтобы включить аутентификацию и авторизацию при использованииsolrserverрасширение, добавьте следующие свойства вместные.свойствае:

solrserver.instances.<экземпляр	имя>.authType=basic
solrserver.instances.<экземпляр	имя>.user=solrserver
solrserver.instances.<экземпляр	имя>.пароль=ваш_пароль

Theбезопасность.jsonle находится в каталоге конфигурации экземпляра. Полный список поддерживаемых свойств см.[Расширение solrserver](#).

Чтобы полностью отключить аутентификацию и авторизацию, удалитебезопасность.jsonle и установите следующие свойства:

solrserver.instances.<экземпляр	имя>.authType=
solrserver.instances.<экземпляр	имя>.пользователь=
solrserver.instances.<экземпляр	имя>.пароль=

Использование внешнего автономного сервера

Аутентификация и авторизация не включены по умолчанию при использовании внешнего автономного сервера. Чтобы включить их, следуйте инструкциям в справочнике Solr в[Плагины аутентификации и авторизации](#)¹.

Пример конфигурации представлен всервер/solr/security.json примере. Чтобы включить его, переименуйте файл вбезопасность.json.

Конфигурация сервера Solr в административной панели бэк-офиса

Единственное необходимое изменение — настроить пользователя и пароль для клиентов Solr в конфигурации сервера Solr. Инструкции см.[Настройте сервер Solr](#).

Поддержка нескольких версий Solr

SAP Commerce поддерживает более одной версии сервера Solr.

Apache Solr находится в стадии активной разработки, что приводит к частым выпускам функций в текущей основной версии. Предыдущая основная версия все еще получает некоторые исправления безопасности и ошибок для выпусков функций в качестве версии с долгосрочной поддержкой (LTS). Более старые версии считаются версиями с окончанием основного обслуживания (EOMM) и больше не обновляются. В результате вы можете использовать более одной версии сервера Solr для каждой версии SAP Commerce.

Совместимость версий Solr

В зависимости от текущей версии релиза могут быть задействованы несколько различных версий сервера Apache Solr.

В следующей таблице представлен краткий обзор совместимости:

Модуль поиска и навигации SAP Commerce	Версия сервера Solr	Комментарии
6.7	7.2.x2	Поставляется с 7.7.x из патча 6.7.0.15
1808	7.4.x2	Поставляется с 7.7.x из патча 1808.10
1811	7.5.x2	Поставляется с 7.7.x из патча 1811.6
1905	7.7.x2 и выше	
2005	8.4.x2 и выше	
2011	8.6.3 и выше	
2105	8.9.0 и выше	
2205	8.11.1 и выше	

¹)Используйте сервер Solr из предыдущей версии SAP Commerce и установите атрибут устаревшего режима в конфигурации индексатора на значение true.

²)x обозначает версию с исправленными ошибками, обеспечивающими обратную совместимость.

Структура каталога

Структура каталога была изменена для поддержки нескольких версий. Имя каталога состоит только из основной и дополнительной части версии, поскольку настройки и сервер могут иметь разные версии патчей. Вы можете найти подробности о включенных версиях в `meta.properties` (примеры приведены ниже).

-Примечание

В релиз включен только один сервер Solr, и его версия может со временем меняться, даже для релизов обслуживания. Предоставленный сервер Solr может быть полезен для CI или сред разработчиков, но для производственных систем используйте внешний автономный сервер.

Пример `solr/resources/7.4/server/meta.properties`:

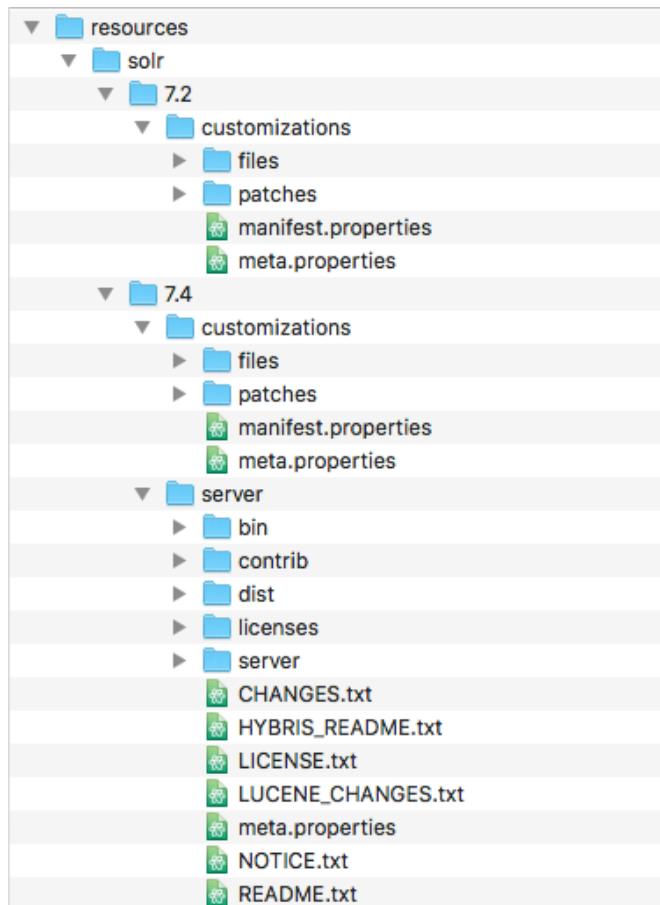
`версия=7.4.0`

`настройкиVersion=7.4.3`

Пример `solr/resources/7.4/customizations/meta.properties`:

`версия=7.4.3`

Структура каталога следующая:



Для каждой версии существуют следующие каталоги:

- сервер: содержит сервер Solr с примененными настройками.
- customizations: содержит модификации, которые будут применены к официальному релизу Solr. Эти модификации делают сервер Solr более легким и совместимым с платформой SAP Commerce. Список изменений см. в `HYBRIS_README.txt`.

Установка внешнего сервера Solr

Настройте внешний автономный сервер Solr для локальной разработки в SAP Commerce.

Процедура

- . Загрузите официальный релиз Solr с сайта <https://solr.apache.org/downloads.html>.
- . Извлеките содержимое архива в каталог.
- . Примените настройки SAP Commerce с помощью следующей команды:

```
ant configureSolrServer -DsolrServerPath=/путь/к/solr
```

Общая функциональность

Часть функционала, используемого в модуле поиска и навигации, является общим для поиска и индексации. Например, оба они используют API слушателей.

API слушателей

Вы можете использовать прослушиватели для перехвата и настройки некоторых частей процессов индексации и поиска.

Слушатели следуют правилам приоритета. Перед методами слушатели с более высоким приоритетом вызываются раньше слушателей с более низким приоритетом. После методов они вызываются в обратном порядке. Применяются следующие приоритеты, от высшего к низшему: глобальные слушатели, слушатели, настроенные в FacetSearchCong, и, наконец, слушатели, настроенные для IndexedType.

-Кончик

The `solrQueryDebuggingListener` регистрирует информацию о проанализированном запросе. Это может быть полезно для решения проблем, возникающих на этапе разработки.

Создание пользовательского прослушивателя

Чтобы создать слушателя, вам необходимо реализовать один из поддерживаемых типов слушателей. Следующий пример кода показывает, как создать слушателя:

Пример пользовательского прослушивателя

```
открытый класс EnableLocalizationFallbackListener реализует IndexerListener, IndexerBatchListener {
    частный I18NService i18nService;

    @Необходимый
    public void setI18nService(final I18NService i18nService) {
        это.i18nService = i18nService;
    }

    публичный I18NService getI18nService() {
        вернуть i18nService;
    }

    @Переопределить
    public void beforeIndex(конечный контекст IndexerContext) выдает исключение IndexerException {
        enableLocalizationFallback(context.getFacetSearchConfig(), context.getIndexedType());
    }

    @Переопределить
    public void afterIndex(контекст final IndexerContext) выдает исключение IndexerException {
        // НЕТ
    }

    @Переопределить
    public void afterIndexError(конечный контекст IndexerContext) выдает IndexerException {
        // НЕТ
    }
}
```

```

}

@Переопределить
public void beforeBatch(final IndexerBatchContext batchContext) выдает IndexerException {
    enableLocalizationFallback(batchContext.getFacetSearchConfig(), batchContext.getIndexedType());
}

@Переопределить
public void afterBatch(final IndexerBatchContext batchContext) выдает IndexerException {
    // НЕТ
}

@Переопределить
public void afterBatchError(final IndexerBatchContext batchContext) выдает IndexerException {
    // НЕТ
}

protected void enableLocalizationFallback(final FacetSearchConfig facetSearchConfig, final IndexedType indexe {
    i18nService.setLocalizationFallbackEnabled(true);
}
}

```

В следующем примере показана конфигурация Spring:

```
<bean id="enableLocalizationFallbackListener" class="com.example.EnableLocalizationFallbackListener">
    <имя_свойства="i18nService" ref="i18nService" /> </bean>
```

Регистрация слушателя

Регистрация слушателя по всему миру

Вы можете зарегистрировать слушателя глобально, добавив конфигурацию Spring. Приоритет по умолчанию равен 100, если он не указан как часть объявления компонента Spring. Приоритеты выше 1000 зарезервированы для использования системой. Ниже приведен пример поздравления Spring:

```
<bean id="enableLocalizationFallbackListenerDefinition" parent="solrListenerDefinition">
    <имя_свойства="приоритет" значение="150" />
    <property name="listener" ref="enableLocalizationFallbackListener" /> </bean>
```

Конфигурация поиска по фасету

Добавьте идентификаторы Spring bean-компонентов слушателей в атрибут listeners в конфигурации поиска по фасетам. Слушатели, которые появляются первыми в списке, имеют наивысший приоритет.

Индексированный тип

Добавьте идентификаторы Spring bean-компонентов слушателей к атрибуту listeners в IndexedType. Слушатели, которые появляются первыми в списке, имеют наивысший приоритет.

Функциональность индексации

Существует ряд различных тем, касающихся функциональности индексирования. Каждая тема документирована в отдельном руководстве.

Процесс индексации

Узнайте больше о типах индексирования, чтобы успешно начать процесс индексации и устранить возможные ошибки. [API индексации](#)

В этом документе содержится информация об API верхнего уровня для функциональности индексации и о том, как выполнить некоторые простые настройки. [API поставщика ценностей](#)

Узнайте, как можно написать поставщик значений, используя доступный API.

Идентификаторы документов

Каждый индексированный документ должен иметь уникальный идентификатор. В большинстве случаев подойдет идентификатор по умолчанию, однако вы также можете определить свой собственный.

Процесс индексации

Узнайте больше о типах индексирования, чтобы успешно начать процесс индексации и устранить возможные ошибки.

Поскольку индексация может занять много времени, элементы разбиваются на партии и индексируются параллельно.

Есть два способа заставить это работать:

- Использование индексатора worker (для каждого worker назначается поток)
- Использование распределенной обработки и механизма задач, что является предлагаемым решением для индексации в кластерной среде.

Изменяя значение `распределеннаяИндексация` атрибут конфигурации Solr, можно переключиться с одного на другой способ индексации. В случае распределенной индексации можно настроить группу узлов кластерного процесса, который будет выполняться (`nodeGroup` атрибут конфигурации Solr).

Индексация операций

Вместо индексации всех доступных элементов, в некоторых случаях вам может понадобиться индексировать только новые доступные элементы или обновить те, которые изменились с момента последней операции индексации. Вы можете выбрать среди следующих операций индексации:

- FULL: пересоздает индекс, все элементы будут проиндексированы на основе запроса FULL индекса;
- ОБНОВЛЕНИЕ: обновляет некоторые документы в индексе. Элементы для индексации/обновления обычно выбираются запросом ОБНОВЛЕНИЕ;
- PARTIAL_UPDATE: аналогично операции UPDATE, но позволяет выбирать поля для обновления (для более быстрых обновлений);
- DELETE: удаляет документы из индекса. Элементы для удаления обычно выбираются запросом DELETE.

-Примечание

Рекомендуется регулярно использовать операцию полной индексации, чтобы быть уверенным, что данные, которые уже были удалены из базы данных, также удалены из solr.

Режим индексации для ПОЛНОЙ операции

При выполнении операции FULL поддерживаются следующие режимы индексации:

- ПРЯМОЕ: индексация происходит непосредственно в живом индексе. Новые и обновленные документы отмечаются, а старые удаляются после успешной операции;
- TWO_PHASE: индексация происходит по временному индексу. После успешной операции временный индекс заменяет текущий живой.

Ограничения операции PARTIAL_UPDATE

PARTIAL_UPDATE потенциально намного быстрее обычного UPDATE, однако он также имеет следующие ограничения:

Инструкции по выполнению частичных обновлений см. здесь: [Обновление индекса Solr с помощью Cronjob](#).

- требуется сохранено=истина по всем атрибутам на SolrcHEMA.xml (например, в настоящее время полный текст_язык поля имеют сохранено=ложь по умолчанию и это значение следует изменить на true);
- не следует использовать в атрибутах, которые используются для проверки орфографии/предложений;
- Постоянное обновление индекса может иметь побочные эффекты для производительности поиска.

Запуск процесса индексации

Использование IndexerService

Служба IndexerService предоставляет методы, которые позволяют выполнять все операции с индексами, упомянутые ранее. Важно отметить, что:

- Если метод позволяет передавать индексируемые элементы в качестве параметра, это означает, что запрос индекса по умолчанию для операции использоваться не будет;
- Единственными методами, которые получают индексированные свойства в качестве параметра, являются методы, используемые для операции PARTIAL_UPDATE.

Инструкции по обновлению индекса см. здесь:[Обновление индекса Solr](#).

Использование задания Cron

Индексирование также может быть запущено CronJob. Доступны следующие типы Cron Job:

- SolrIndexerCronJob (в сочетании csolrIndexerРаботаРабота):
 - поддерживает операции индекса FULL, UPDATE и DELETE; не
 - поддерживает операцию индекса PARTIAL_UPDATE;
 - использует запросы индексатора по умолчанию;
 - пользователем сеанса будет тот, который настроен в запросе индексатора;
 - использует данные из вторичного источника данных, доступного только для чтения, если вы настроили его таким образом в Backoffice
- SolrExtIndexerCronJob (в сочетании csolrExtIndexerJobРабота):
 - поддерживает операции индекса UPDATE, PARTIAL_UPDATE и DELETE;
 - не поддерживает операцию FULL index; не
 - обновляет время последнего индекса;
 - пользователь сеанса будет настроен в задании Cron; имеет
 - дополнительные атрибуты:
 - индексированныйТип:индексированный тип для использования, например, «Product» или «Product_indexName»;
 - индексированныеСвойства:свойства для индексирования, используются только в том случае, если операция индексирования — PARTIAL_UPDATE;
 - запрос:гибкий поисковый запрос, который выбирает элементы для индексации;
 - queryParameterProvider:id bean-компонент spring, который позволяет создавать параметры для передачи в запрос. Он должен реализовывать один из следующих типов:Параметрпровидер, ContextAwareParameterProviderили CronJobAwareParameterProvider, все изde.hybris.platform.solrfacetsearch.providerупаковка.

Обработка приостановки и возобновления

Механизм индексации поддерживает приостановку и возобновление работы платформы. Когда индексация запущена, а платформа по какой-то причине приостановлена, процесс индексации будет изящно приостановлен, а затем возобновлен после запуска платформы. Пакеты, для которых индексация началась до приостановки, будут полностью проиндексированы до приостановки работы платформы.

Эту функцию поддерживают:

- индексация по умолчанию
- распределенная индексация

Выбор любого типа индексации приводит к одноковому поведению функций приостановки и возобновления.

Более подробную информацию о приостановке и возобновлении см.[Приостановка и возобновление работы SAP Commerce](#).

Обработка ошибок

Что касается обработки ошибок в процессе индексации, начиная с версии hybris 5.5.0, применяется следующее:

- если обнаружено исключение, процесс индексации отменяется;
- в этом случае он отменяет воркеры, которые не начали выполнение, а также пытается отменить воркеры, работающие в данный момент;
- Только успешно полностью проиндексированные элементы будут видны при поиске.

-Примечание

УстановивигнорироватьОшибкиа в значение true в конфигурации индексатора, некоторые ошибки игнорируются в процессе индексации.

API индексации

В этом документе содержится информация об API верхнего уровня для функциональности индексации и о том, как выполнить некоторые простые настройки.

Служба индексации

Запустить процесс индексации легко: вам нужно всего лишь вызвать один из методов службы индексатора, как показано в примере ниже.

```
открытый интерфейс IndexerService {

    /**
     * Выполняет полную индексную операцию, которая воссоздает индекс. Все типы, связанные с фасетным поиском
     * конфигурация учитывается. Элементы для индексации выбираются на основе полного запроса индекса.
     *
     * @param facetSearchConfig
     *         - конфигурация для экземпляра индексатора
     *
     * @throws IndexerException
     *         исключение возникает, когда в конфигурации отсутствует полный запрос индекса или во время индексации возникает
     *         непредвиденная ошибка.
     */
    void performFullIndex(FacetSearchConfig facetSearchConfig) выдает исключение IndexerException;

    /**
     * То же, что и {@link #performFullIndex(FacetSearchConfig)}, но позволяет передавать подсказки индексатора в качестве параметра.
     *
     * @param facetSearchConfig
     *         - конфигурация для экземпляра индексатора
     * @param indexerHints
     *         - подсказки индексатора
     *
     * @throws IndexerException
     *         исключение возникает, когда в конфигурации отсутствует полный запрос индекса или во время индексации возникает
     *         непредвиденная ошибка.
     */
    void performFullIndex(FacetSearchConfig facetSearchConfig, Map<String, String> indexerHints) бросает IndexerEx

    /**
     * Обновление некоторых элементов индекса. Все типы, связанные с конфигурацией поиска фасетов, рассматриваются
     * Элементы для обновления выбираются на основе запроса на обновление индекса.
     *
     * @param facetSearchConfig
     *         - конфигурация для экземпляра индексатора
     *
     * @throws IndexerException
     *         исключение возникает, если запрос на обновление индекса отсутствует в конфигурации или возникает непредвиденная ошибка во время
     *         индексации.
     */
    void updateIndex(FacetSearchConfig facetSearchConfig) выдает исключение IndexerException;

    /**
     * То же, что и {@link #updateIndex(FacetSearchConfig)}, но позволяет передавать подсказки индексатора в качестве параметра.
     *
     * @param facetSearchConfig
     *         - конфигурация для экземпляра индексатора
     * @param indexerHints
     *         - подсказки индексатора
     *
     * @throws IndexerException
     *         исключение возникает, если запрос на обновление индекса отсутствует в конфигурации или возникает непредвиденная ошибка во время
     *         индексации.
     */
    void updateIndex(FacetSearchConfig facetSearchConfig, Map<String, String> indexerHints) выдает IndexerExcepti

    /**
     * Обновляет некоторые элементы индекса для определенного типа. Элементы для обновления выбираются на основе обновления
     * запроса.
     *
     * @param facetSearchConfig
     *         - конфигурация для экземпляра индексатора
     * @param indexedType
     *         - выбранный тип
     *
     * @throws IndexerException
     *         исключение возникает, если запрос на обновление индекса отсутствует в конфигурации или возникает непредвиденная ошибка во время
     *         индексации.
     */
    void updateTypeIndex(FacetSearchConfig facetSearchConfig, IndexedType indexedType) выдает исключение IndexerException;
```

```

/**
 * То же, что и {@link #updateTypeIndex(FacetSearchConfig, IndexedType)}, но позволяет передавать подсказки индексатора как
 * параметр.
 *
 * @param facetSearchConfig
 *         - конфигурация для экземпляра индексатора
 * @param indexedType
 *         - выбранный тип
 * @param indexerHints
 *         - подсказки индексатора
 *
 * @throws IndexerException
 *         исключение возникает, если запрос на обновление индекса отсутствует в конфигурации или возникает непредвиденная ошибка во время
 *         индексации.
 */
void updateTypeIndex(FacetSearchConfig facetSearchConfig, IndexedType indexedType, Map<String, String> indexe
    выдает исключение IndexerException;

/**
 * Обновляет некоторые элементы индекса для определенного типа.
 *
 * @param facetSearchConfig
 *         - конфигурация для экземпляра индексатора
 * @param indexedType
 *         - выбранный тип
 * @param pks
 *         - пакеты товаров для обновления
 *
 * @throws IndexerException
 *         исключение возникает, когда во время индексации возникает непредвиденная ошибка.
 */
void updateTypeIndex(final FacetSearchConfig facetSearchConfig, final IndexedType indexedType, List<PK> pks)
    выдает исключение IndexerException;

/**
 * То же, что и {@link #updateTypeIndex(FacetSearchConfig, IndexedType, List)}, но позволяет передавать подсказку индексатора
 * параметр.
 *
 * @param facetSearchConfig
 *         - конфигурация для экземпляра индексатора
 * @param indexedType
 *         - выбранный тип
 * @param pks
 *         - пакеты товаров для обновления
 * @param indexerHints
 *         - подсказки индексатора
 *
 * @throws IndexerException
 *         исключение возникает, когда во время индексации возникает непредвиденная ошибка.
 */
void updateTypeIndex(final FacetSearchConfig facetSearchConfig, final IndexedType indexedType, List<PK> pks,
    Map<String, String> indexerHints) выдает исключение IndexerException;

/**
 * Обновляет некоторые свойства некоторых элементов индекса для определенного типа.
 *
 * @param facetSearchConfig
 *         - конфигурация для экземпляра индексатора
 * @param indexedType
 *         - выбранный тип
 * @param indexedProperties
 *         - свойства для обновления
 * @param pks
 *         - пакеты товаров для обновления
 *
 * @throws IndexerException
 *         исключение возникает, когда во время индексации возникает непредвиденная ошибка.
 */
void updatePartialTypeIndex(final FacetSearchConfig facetSearchConfig, final IndexedType indexedType,
    final Collection<IndexedProperty> indexedProperties, List<PK> pks) выдает IndexerException;

/**
 * То же, что и {@link #updatePartialTypeIndex(FacetSearchConfig, IndexedType, Collection, List)}, но позволяет
 * подсказки индексатора в качестве параметра.
 *
 * @param facetSearchConfig
 *         - конфигурация для экземпляра индексатора
 * @param indexedType
 *         - выбранный тип
 * @param indexedProperties
 *         - свойства для обновления
 * @param pks
 *         - пакеты товаров для обновления
 * @param indexerHints
 *         - подсказки индексатора
 */

```

```

/*
 * @броски IndexerException
 *      исключение возникает, когда во время индексации возникает непредвиденная ошибка.
 */
void updatePartialTypeIndex(final FacetSearchConfig facetSearchConfig, final IndexedType indexedType,
                           окончательная коллекция <IndexedProperty> indexedProperties, список <PK> pks, карта <String, строка> indexerHints) выдает
                           исключение IndexerException;

/**
 * Удаляет некоторые элементы из индекса. Все типы, связанные с конфигурацией поиска фасетов, учитываются
 * Элементы для удаления выбираются на основе запроса на удаление индекса.
 *
 * @param facetSearchConfig
 *      - конфигурация для экземпляра индексатора
 *
 * @throws IndexerException
 *      исключение возникает, если запрос на удаление индекса отсутствует в конфигурации или возникает непредвиденная ошибка во время
 *      индексации.
 */
void deleteFromIndex(FacetSearchConfig facetSearchConfig) выдает исключение IndexerException;

/**
 * То же, что и {@link #deleteFromIndex(FacetSearchConfig)}, но позволяет передавать подсказки индексатора в качестве параметра.
 *
 * @param facetSearchConfig
 *      - конфигурация для экземпляра индексатора
 * @param indexerПодсказки
 *      - подсказки индексатора
 *
 * @throws IndexerException
 *      исключение возникает, если запрос на удаление индекса отсутствует в конфигурации или возникает непредвиденная ошибка во время
 *      индексации.
 */
void deleteFromIndex(FacetSearchConfig facetSearchConfig, Map<String, String> indexerHints) выдает IndexerExc

/**
 * Удаляет некоторые элементы из индекса для определенного типа. Элементы для удаления выбираются на основе удаления
 * запрос.
 *
 * @param facetSearchConfig
 *      - конфигурация для экземпляра индексатора
 * @param индексированныйТип
 *      - выбранный тип
 *
 * @throws IndexerException
 *      исключение возникает, если запрос на удаление индекса отсутствует в конфигурации или возникает непредвиденная ошибка во время
 *      индексации.
 */
void deleteTypeIndex(FacetSearchConfig facetSearchConfig, IndexedType indexedType) выдает исключение IndexerException;

/**
 * То же, что и {@link #deleteTypeIndex(FacetSearchConfig, IndexedType)}, но позволяет передавать подсказки индексатора как
 * параметр.
 *
 * @param facetSearchConfig
 *      - конфигурация для экземпляра индексатора
 * @param индексированныйТип
 *      - выбранный тип
 * @param indexerПодсказки
 *      - подсказки индексатора
 *
 * @throws IndexerException
 *      исключение возникает, если запрос на удаление индекса отсутствует в конфигурации или возникает непредвиденная ошибка во время
 *      индексации.
 */
void deleteTypeIndex(FacetSearchConfig facetSearchConfig, IndexedType indexedType, Map<String, String> indexe
выдает исключение IndexerException;

/**
 * Удаляет некоторые элементы из индекса для определенного типа.
 *
 * @param facetSearchConfig
 *      - конфигурация для экземпляра индексатора
 * @param индексированныйТип
 *      - выбранный тип
 * @param ПКС
 *      - пакеты предметов, подлежащих удалению
 *
 * @throws IndexerException
 *      исключение возникает, когда во время индексации возникает непредвиденная ошибка.
 */
void deleteTypeIndex(FacetSearchConfig facetSearchConfig, IndexedType indexedType, List<PK> pks) бросает Индекс

/**
 * То же, что и {@link #deleteTypeIndex(FacetSearchConfig, IndexedType, List)}, но позволяет передавать подсказку индексатора

```

12/3/24, 10:47 утра

```
* параметр.  
*  
* @param facetSearchConfig  
*      - конфигурация для экземпляра индексатора  
* @param indexedType  
*      - выбранный тип  
* @param pks  
*      - пакеты предметов, подлежащих удалению  
* @param indexerHints  
*      - подсказки индексатора  
*  
* @throws IndexerException  
*      исключение возникает, когда во время индексации возникает непредвиденная ошибка.  
*/  
void deleteTypeIndex(FacetSearchConfig facetSearchConfig, IndexedType indexedType, List<PK> pks,  
Map<String, String> indexerHints) выдает исключение IndexerException;  
  
...  
}
```

Некоторые из наиболее распространенных вариантов использования:

- Выполнение полной индексной операции:

```
indexerService.performFullIndex(facetSearchConfig);
```

- Обновление некоторых документов в индексе:

```
indexerService.updateTypeIndex(facetSearchConfig, индексированныйТип, ПКС);
```

- Удаление некоторых документов из индекса:

```
indexerService.deleteTypeIndex(facetSearchConfig, индексированныйТип, ПКС);
```

Контекст и слушатели

Контекст

Некоторые объекты контекста могут быть доступны в процессе индексации. Вы можете получить к ним доступ, используя соответствующую фабрику или, в некоторых случаях, как параметр метода (например, слушатели).

Доступны следующие типы контекста:

- IndexerQueryContext (вместе с IndexerQueryContextFactory): представляет контекст, действительный в течение всего времени запроса индексатора;
- IndexerContext (вместе с IndexerContextFactory): представляет контекст, действительный на протяжении всей операции индексирования индексатора, однако он недействителен внутри пакета индексатора (пакет индексатора может выполняться в другом потоке или даже на другой машине);
- IndexerBatchContext (вместе с IndexerBatchContextFactory): представляет контекст, действительный на протяжении пакета индексатора. Каждый пакет выполняется в отдельном потоке, и этот контекст действителен только для соответствующего потока.

Слушатели

В этом разделе описываются типы прослушивателей, которые можно реализовать для перехвата процесса индексации.

ИндексаторЗапросЛистенер

Вы используете IndexerQueryListener для перехвата выполнения запросов индексатора. Следующий фрагмент кода показывает интерфейс IndexerQueryListener:

```
/**  
 * Интерфейс для получения уведомлений о выполнении запросов индексатора.  
 */  
открытый интерфейс IndexerQueryListener {  
  
    /**  
     * Обрабатывает уведомление о том, что запрос индексатора вот-вот начнет выполняться.  
    */
```

12/3/24, 10:47 утра

```
/*
 * @param queryContext
 *      - {@сылка IndexerQueryContext}
 *
 * @throws IndexerException
 *      если произошла ошибка
 */
void beforeQuery(IndexerQueryContext queryContext) выдает исключение IndexerException;

/**
 * Обрабатывает уведомление о том, что запрос индексатора только что завершен.
 *
 * @param queryContext
 *      - {@link IndexerQueryContext}
 *
 * @throws IndexerException
 *      если произошла ошибка
 */
void afterQuery(IndexerQueryContext queryContext) выдает исключение IndexerException;

/**
 * Обрабатывает уведомление о сбое запроса индексатора (это также может быть связано с сбоем прослушивателей).
 *
 * @param queryContext
 *      - {@link IndexerQueryContext}
 *
 * @throws IndexerException
 *      если произошла ошибка
 */
void afterQueryError(IndexerQueryContext queryContext) выдает исключение IndexerException;
}
```

ИндексаторСлушатель

Вы используете IndexerListener для перехвата выполнения операции индексатора. Следующий фрагмент кода показывает интерфейс IndexerListener:

```
/**
 * Интерфейс для получения уведомлений об экземплярах {@link IndexerContext}.
 */
открытый интерфейс IndexerListener {

    /**
     * Обрабатывает уведомление о том, что индексация для определенного {@link IndexerContext} скоро начнется.
     *
     * @param контекст
     *      - {@link IndexerContext}
     *
     * @throws IndexerException
     *      если произошла ошибка
     */
    void beforeIndex(контекст IndexerContext) выдает исключение IndexerException;

    /**
     * Обрабатывает уведомление о том, что индексация для определенного {@link IndexerContext} только что завершена.
     *
     * @param контекст
     *      - {@link IndexerContext}
     *
     * @throws IndexerException
     *      если произошла ошибка
     */
    void afterIndex(контекст IndexerContext) выдает исключение IndexerException;

    /**
     * Обрабатывает уведомление о том, что индексация для определенного {@link IndexerContext} не удалась.
     *
     * @param контекст
     *      - {@link IndexerContext}
     *
     * @throws IndexerException
     *      если произошла ошибка
     */
    void afterIndexError(контекст IndexerContext) выдает исключение IndexerException;
}
```

ИндексаторПакетныйСлушатель

Вы можете использовать IndexerBatchListener для перехвата выполнения операции индексатора для определенного пакета. Следующий фрагмент кода показывает интерфейс IndexerBatchListener:

```
/**
 * Интерфейс для получения уведомлений об экземплярах {@link IndexerBatchContext}.
 */
открытый интерфейс IndexerBatchListener {

    /**
     * Обрабатывает уведомление о том, что обработка для определенного {@link IndexerBatchContext} скоро начнется.
     *
     * @param пакетныйКонтекст
     *      - {@link IndexerBatchContext}
     *
     * @throws IndexerException
     *      если произошла ошибка
     */
    void beforeBatch(IndexerBatchContext batchContext) выдает исключение IndexerException;

    /**
     * Обрабатывает уведомление о том, что обработка для определенного {@link IndexerBatchContext} только что была завершена
     *
     * @param пакетныйКонтекст
     *      - {@link IndexerBatchContext}
     *
     * @throws IndexerException
     *      если произошла ошибка
     */
    void afterBatch(IndexerBatchContext batchContext) выдает исключение IndexerException;

    /**
     * Обрабатывает уведомление о том, что обработка определенного {@link IndexerBatchContext} завершилась неудачей.
     *
     * @param пакетныйКонтекст
     *      - {@link IndexerBatchContext}
     *
     * @throws IndexerException
     *      если произошла ошибка
     */
    void afterBatchError(IndexerBatchContext batchContext) выдает исключение IndexerException;
}
```

Советы по индексированию

Подсказки индексатора предназначены для предоставления дополнительной информации процессу индексирования. Подсказки могут быть доступны в любое время; они являются частью контекста индексатора и контекста пакета индексатора.

Поддерживаются следующие подсказки:

Намекать	Описание	Возможные значения	C
commitMode	Переопределяет режим фиксации, определенный в конфигурации индексатора.	СМЕШАННЫЙ, НИКОГДА, ПОСЛЕ_ИНДЕКСА, ПОСЛЕ_ПАРТИИ	5.6
optimizeMode	Переопределяет режим оптимизации, определенный в конфигурации индексатора.	НИКОГДА, ПОСЛЕ_ИНДЕКСА, AFTER_FULL_INDEX	5.6

API поставщика ценности

Узнайте, как можно написать поставщик значений, используя доступный API.

API имеет следующие преимущества:

- Он группирует индексированные свойства, которые используют одного и того же поставщика значений;
- Он предоставляет больше контекстной информации, например, можно получить доступ ко всем элементам, которые индексируются для определенного пакета.

Создание индивидуального поставщика

Создайте поставщика пользовательских значений, используя **КвалификаторПоставщики** или реализация **ValueResolver** интерфейс.

Расширение **AbstractValueResolver** и использование **КвалификаторПоставщик**

-Примечание

Следующая процедура показывает рекомендуемый способ создания поставщика значений.

Он следует за наблюдаемыми шаблонами и избегает дублирования кода в ваших поставщиках значений. Вы можете реализовать следующие методы:

- **addFieldValues**: Обязательный метод. Он отвечает за разрешение индексируемых значений. Данные и данные квалификатора доступны как часть параметра `resolverContext`.
- **Данные загрузки**: Необязательный метод. По умолчанию он возвращает нулевой. Он загружает данные, действительные в контексте модели.
- **loadQualifierData**: Необязательный метод. По умолчанию он возвращает нулевой. Он загружает данные, которые действительны в контексте модели и квалификации. Например, квалифер может быть определенным языком или валютой.

Следующий пример кода показывает **AbstractValueResolver** сорт:

```
/**
 * Абстрактный класс для преобразователей значений, которые хотят использовать {@link QualifierProvider}
 *
 * @param <T>
 *         тип модели <MDATA>
 * @param
 *         тип данных, действительный в контексте модели <QDATA>
 * @param
 *         тип данных, действительный в контексте модели и квалификатора
 */
открытый абстрактный класс AbstractValueResolver<T расширяет ItemModel, MDATA, QDATA> реализует ValueResolver<T> {
    ...
}

/**
 * Загружает данные, действительные в контексте модели.
 *
 * @param пакетныйКонтекст
 *         - текущий контекст пакета индексатора
 * @param индексированныеСвойства
 *         - индексированные свойства, которые используют один и тот же распознаватель значений
 * @param модель
 *         - значения должны быть разрешены для данного экземпляра модели
 *
 * @throws FieldValueProviderException
 *         если произошла ошибка
 */
защищенный MDATA loadData(final IndexerBatchContext batchContext, final Collection<IndexedProperty> indexedProperties)
    окончательная модель T) выдает FieldValueProviderException
{
    вернуть ноль;
}

/**
 * Загружает данные, действительные в контексте модели и квалификатора.
 *
 * @param пакетныйКонтекст
 *         - текущий контекст пакета индексатора
 * @param индексированныеСвойства
 *         - индексированные свойства, которые используют один и тот же распознаватель значений
 * @param модель
 *         - значения должны быть разрешены для данного экземпляра модели
 *
 * @throws FieldValueProviderException
 */

```

```

*           если произошла ошибка
* /
защищенный QDATA loadQualifierData(final IndexerBatchContext batchContext, final Collection<IndexedProperty> i
    окончательная модель T, окончательный квалификатор Qualifier) выдает FieldValueProviderException
{
    вернуть ноль;
}

защищенный абстрактный void addFieldValues(final InputDocument document, final IndexerBatchContext batchContext,
    final IndexedProperty indexedProperty, final T model, ValueResolverContext<MDATA, QDATA> resolverCont выдает
    FieldValueProviderException;
...
}

```

Следующий пример кода показывает КвалификаторПоставщикинтерфейс:

```

/***
 * Этот интерфейс обеспечивает поддержку различных типов квалификаторов.
 */
открытый интерфейс QualifierProvider {

    /**
     * Возвращает все поддерживающие этим поставщиком типы.
     *
     * @return поддерживающие типы
     */
    Установить<Класс<?>> getSupportedTypes();

    /**
     * Проверяет, можно ли применять/использовать квалификаторы с индексированным свойством, переданным в качестве параметра.
     *
     * @param индексированноеСвойство
     *           - индексированное свойство
     *
     * @return {@code true}, если можно использовать квалификаторы, {@code false} в противном случае
     */
    boolean canApply(IndexedProperty indexedProperty);

    /**
     * Возвращает все возможные квалификаторы для заданной конфигурации индекса и индексированного типа.
     *
     * @param facetSearchConfig
     *           - конфигурация поиска фасетов
     * @param индексированныйТип
     *           - индексированный тип
     *
     * @return доступные квалификаторы
     */
    Коллекция<Квалификатор> getAvailableQualifiers(FacetSearchConfig facetSearchConfig, IndexedType indexedType);

    /**
     * Применяет квалификатор, переданный как параметр. Обычно это заключается в установке некоторых атрибутов на сеансе
     * путем вызова службы для установки текущего языка сеанса, валюты и т. д.
     *
     * @param квалификатор
     *           - {@link Qualifier}, который будет применен
     */
    пустота applyQualifier(Квалификатор      квалификатор);

    /**
     * Возвращает текущий квалификатор. Обычно это заключается в получении некоторых атрибутов из сеанса и создания

```

```

* соответствующий квалификатор.
*
* @return текущий квалификатор
*/
Квалификатор getCurrentQualifier();
}

```

Ниже показан пример кода Java для `ProductUrlsValueResolver`:

```

открытый класс ProductUrlsValueResolver расширяет AbstractValueResolver<ProductModel, Object, Object> {

частный UrlResolver<ProductModel> urlResolver;

public UrlResolver<ProductModel> getUrlResolver() {

    возврат urlResolver;
}

@Необходимый
public void setUrlResolver(final UrlResolver<ProductModel> urlResolver) {

    этот.urlResolver = urlResolver;
}

@Переопределить
защищенный void addFieldValues(final InputDocument document, final IndexerBatchContext batchContext,
    окончательный IndexedProperty indexedProperty, окончательный ProductModel продукт,
    окончательный ValueResolverContext<Object, Object> resolverContext) выдает FieldValueProviderException
{
    окончательная строка productUrl = urlResolver.resolve(product); если (!
        StringUtils.isBlank(productUrl))
    {
        document.addField(indexedProperty, productUrl, resolverContext.getFieldQualifier());
    }
}
}

```

Ниже представлена конфигурация Spring:

```

<bean id="productUrlsValueResolver" class="com.example.ProductUrlsValueResolver" parent="abstractValueResolver">
    <property name="qualifierProvider" ref="languageQualifierProvider" /> <property
    name="urlResolver" ref="productModelUrlResolver" /> </bean>

```

Реализация **ValueResolver** Интерфейс

-Примечание

Это один из возможных способов создания поставщика ценности, однако не рекомендуемый.

Следующий фрагмент кода показывает `ValueResolver` интерфейс:

```

/***
* <p>
* Реализации этого интерфейса отвечают за разрешение индексируемых значений. Этот интерфейс работает
* на индексированный уровень собственности.
* </p>
*
* @see TypeValueResolver
*/
открытый интерфейс ValueResolver<T extends ItemModel> {

    /**

```

```

* Разрешает индексируемые значения. Индексированные свойства, которые используют один и тот же распознаватель значений, группируются и
* метод вызывается один раз для каждой из этих групп.
*
* @param документ
*      - документ, который будет проиндексирован, все решенные значения должны быть добавлены в качестве полей в этот документ
* @param пакетныйКонтекст
*      - текущий контекст пакета индексатора
* @param индексированныеСвойства
*      - индексированные свойства, которые используют один и тот же распознаватель значений
* @param модель
*      - значения должны быть разрешены для данного экземпляра модели
*
* @throws FieldValueProviderException
*      если произошла ошибка
*/
void resolve(final InputDocument document, final IndexerBatchContext batchContext,
            final Collection<IndexedProperty> indexedProperties, модель T) выдает FieldValueProviderException;
}

```

В следующем примере показан код Java:

```

открытый класс ProductUrlsValueResolver реализует ValueResolver<ProductModel> {

частный CommonI18NService commonI18NService;
частный UrlResolver<ProductModel> urlResolver;

public CommonI18NService getCommonI18NService() {
    вернуть commonI18NService;
}

@Необходимый
public void setCommonI18NService(final CommonI18NService commonI18NService) {
    этот.commonI18NService = commonI18NService;
}

public UrlResolver<ProductModel> getUrlResolver() {
    возврат urlResolver;
}

@Необходимый
public void setUrlResolver(final UrlResolver<ProductModel> urlResolver) {
    этот.urlResolver = urlResolver;
}

@Переопределить
public void resolve(final InputDocument document, final IndexerBatchContext batchContext,
                   final Collection<IndexedProperty> indexedProperties, final ProductModel model) выдает FieldValueProv
{
    // индекс нелокализованных свойств
    для (final IndexedProperty indexedProperty : indexedProperties) {
        если (!indexedProperty.isLocalized()) {
            addFieldValue(документ, инdexированное свойство, модель, null);
        }
    }

    // индекс локализованных свойств
    окончательная Языковая Модель текущийЯзык = commonI18NService.getCurrentLanguage(); попробуйте

    {
        окончательный FacetSearchConfig facetSearchConfig = пакетныйконтекст.getFacetSearchConfig();
        окончательный IndexConfig indexConfig = facetSearchConfig.getIndexConfig();
        окончательная коллекция<LanguageModel> languages = indexConfig.getLanguages();

        для (конечный язык LanguageModel : языки) {
            commonI18NService.setCurrentLanguage(язык);
        }
    }
}

окончательно

```

12/3/24, 10:47 утра

```
{ commonI18NService.setCurrentLanguage(текущийЯзык);
}

protected void addFieldValue(final InputDocument document, final IndexedProperty indexedProperty, final ProductModel продукт)
{
    окончательный язык LanguageModel) выдает FieldValueProviderException
{
    окончательная строка urlString = urlResolver.resolve(product); если (!
    StringUtils.isBlank(urlString))
    {
        document.addField(indexedProperty, urlString, language == null ? null : language.getIsoCode());
    }
}
```

В следующем примере показана конфигурация Spring:

```
<bean id="productUrlsValueResolver" class="com.example.ProductUrlsValueResolver">
    <property name="commonI18NService" ref="commonI18NService" /> <property
    name="urlResolver" ref="productModelUrlResolver" /> </bean>
```

Использование параметров

Еще одним улучшением является возможность иметь несколько параметров (с использованием `IndexedProperty`). Следующий пример показывает, как можно воспользоваться этим преимуществом.

```
открытый класс ProductUrlsValueResolver расширяет AbstractValueResolver<ProductModel, Object, Object> {

    public static final String OPTIONAL_PARAM = "необязательный"; public static final
    boolean OPTIONAL_PARAM_DEFAULT_VALUE = true;

    частный UrlResolver<ProductModel> urlResolver;

    public UrlResolver<ProductModel> getUrlResolver() {

        возврат urlResolver;
    }

    @Необходимый
    public void setUrlResolver(final UrlResolver<ProductModel> urlResolver) {

        этот.urlResolver = urlResolver;
    }

    @Переопределить
    защищенный void addFieldValues(final InputDocument document, final IndexerBatchContext batchContext,
        окончательный IndexedProperty indexedProperty, окончательный ProductModel продукт,
        окончательный ValueResolverContext<Object, Object> resolverContext) выдает FieldValueProviderException
    {
        логическое значение hasValue = false;

        окончательная строка urlString = urlResolver.resolve(product); если (!
        StringUtils.isBlank(urlString))
        {
            имеетЗначение = истина;
            document.addField(indexedProperty, urlString, resolverContext.getFieldQualifier());
        }

        если (!имеетЗначение)
        {
            окончательный логический параметр isOptional = ValueProviderParameterUtils.getBoolean(indexedProperty, OPTIONAL_PARAM,
                OPTIONAL_PARAM_DEFAULT_VALUE);
            если (!isOptional)
            {
                throw new FieldValueProviderException("Значение не определено для индексированного свойства" + indexedProperty);
            }
        }
    }
}
```

Следующий код представляет собой конфигурацию Spring:

```
<bean id="productUrlsValueResolver" class="com.example.ProductUrlsValueResolver" parent="abstractValueResolver">
    <property name="qualifierProvider" ref="languageQualifierProvider" /> <property
    name="urlResolver" ref="productModelUrlResolver" /> </bean>
```

Доступные решатели и поставщики

Модель Атрибуты Значение Решатель

The Model Attribute Value Resolver получает значения из атрибутов в модели. По умолчанию, если атрибут параметра не указан, он пытается получить атрибут с тем же именем, что и настроенный для индексированного свойства.

Параметр	Значение по умолчанию	Описание
необязательный	истинный	Если false, указывает, что разрешенные значения не должны быть нулевыми и не пустой строкой (для каждого квалифера). Если эти условия не выполняются, выдается исключение типа FieldValueProviderException.
атрибут		Если указано, это имя атрибута.
расколоть	ЛОЖЬ	Если true, разбивает любое разрешенное значение на совпадения с регулярным выражением (только если значение имеет тип String).
splitRegex	\c+	Если split имеет значение true, то это регулярное выражение, которое следует использовать.
формат		Идентификатор форматирующего компонента, который будет использоваться для форматирования объекта значения атрибута перед применением разделения.
оценитьВыражение	ЛОЖЬ	Если true, то предполагается, что имя атрибута является языком выражений Spring, который необходимо оценить.

Доступные поставщики квалификационных услуг

Доступны следующие поставщики квалификаций:

- NoOpQualifierProvider: Поставщик фиктивных квалификаторов для случаев, когда квалификаторы не требуются.
- LanguageQualifierProvider: Более качественный поставщик языков.
- CurrencyQualifierProvider: Более качественный поставщик валют.

Идентификаторы документов

Каждый индексированный документ должен иметь уникальный идентификатор. В большинстве случаев подойдет идентификатор по умолчанию, однако вы также можете определить свой собственный.

Создание поставщика удостоверений

Чтобы создать своего поставщика удостоверений, вам необходимо сначала реализовать Поставщик удостоверений интерфейс, который обеспечивает уникальное строковое представление индексированного идентификатора типа, который генерируется с учетом ограничений типа модели:

```
открытый интерфейс IdentityProvider<T> {

    /**
     * Предоставляет уникальное строковое представление индексированного идентификатора типа, который генерируется относительно мод
     * ограничения.
     *
     * @param indexConfig
     * @param модель
     * @возвращаться идентификатор
     */
    String getIdentifier (IndexConfig indexConfig, модель T);
}
```

TheProductIdentityProviderКласс, который определяет уникальный идентификатор поставщика, является примером реализации интерфейса:

```
открытый класс ProductIdentityProvider реализует IdentityProvider<ProductModel> {

    @Переопределить
    public String getIdentifier (окончательный IndexConfig indexConfig, окончательный продукт ProductModel) {

        окончательный CatalogVersionModel catalogVersion = product.getCatalogVersion();
        окончательный String code = product.getCode();
        вернуть catalogVersion.getCatalog().getId() + "/" + catalogVersion.getVersion() + "/" + код;
    }
}
```

Вам необходимо не забыть создать bean-компонент в контексте Spring для вашего поставщика удостоверений:

```
<bean id="productIdentityProvider" class="com.example.ProductIdentityProvider"/>
```

Регистрация поставщика удостоверений

Пользовательский поставщик удостоверений должен быть настроен в индексированном типе.

Группировка в общей среде

Использование распределенной группировки результатов в шардированной среде SolrCloud имеет некоторые оговорки. Чтобы это работало правильно, все документы в каждой группе должны быть размещены на одном шарде, и этого можно достичь путем индексации с использованием составной Idмаршрутизатор. Чтобы использовать маршрутизатор compositeId, уникальный идентификатор документа должен быть предварен значением атрибута (по которому мы хотим выполнить группировку), за которым следует символ "!" и уникальный идентификатор документа.

ShardAwareProductIdentityProvider, которая является реализацией по умолчанию, используемой для сегментированной среды, позволяет группировать на основе продукта, создавая составной Idc префиксом, который является базовым идентификатором продукта.

Сопутствующая информация

[Распределенная группировка результатов.](#)

[Предостережения. Шарды и индексирование данных в](#)

[SolrCloud. Введение в индексирование Solr](#)

Функциональность поиска

Чтобы в полной мере использовать функциональность поиска, ознакомьтесь со стратегиями, используемыми в процессе поиска, и узнайте, как настраивать поиск с помощью API поиска, поставщиков сортировки и пользовательского сопоставления.

[Процесс поиска](#)

Общая информация о процессе поиска и различных стратегиях поиска.

[API поиска](#)

Узнайте больше об API верхнего уровня для функции поиска и выполните некоторые простые настройки. [Поставщики услуг](#)

[индивидуальной сортировки](#)

В этом документе содержится информация о настройке поставщиков сортировки для поиска фасетов Solr.

[Прямой доступ к запросу Solr](#)

В этом документе содержится информация о прямом доступе к запросам solr.

[Результаты поиска пользовательских карт](#)

В этом разделе описывается, как предоставлять пользовательские объекты передачи данных и преобразователи результатов.

Процесс поиска

Общая информация о процессе поиска и различных стратегиях поиска.

Различные стратегии поиска были реализованы для того, чтобы иметь более расширенную функциональность поиска, но в то же время сохранить совместимость с предыдущими версиями. В зависимости от ваших потребностей вы можете выбрать использование одной или другой. Более подробную информацию о них вы найдете в следующих разделах.

Создание более тяжелых поисковых запросов с большим количеством фасетов и полей поиска может вызвать проблемы с ресурсами Solr. Вы должны:

- Определяйте новый аспект и поле поиска только тогда, когда это необходимо.
- Регулярно проверяйте свои настройки и удаляйте те аспекты, которые больше не нужны.
- Запустите тесты производительности, чтобы проверить, может ли ваша текущая инфраструктура выдерживать ожидаемую пользовательскую нагрузку (особенно после любых новых изменений).

·Примечание

Поисковые запросы и результаты поиска не подвержены ограничениям FlexibleSearch. Для получения дополнительной информации см.[Ограничения](#).

LegacyFacetSearchStrategy

Эта стратегия используется, когда в конфигурации поиска включен режим legacy. Она существует для целей совместимости, и при ее использовании функциональность поиска ведет себя так же, как и в предыдущих версиях.

Главное ограничение заключается в том, что многие условия и параметры запроса берутся из конфигурации, а не из объекта поискового запроса. Это затрудняет наличие запросов, которые ведут себя по-разному в зависимости от контекста. Например, сложно указать аспекты, которые должны быть включены в ответ динамическим образом.

DefaultFacetSearchStrategy

Эта стратегия будет использоваться, когда режим legacy отключен в конфигурации поиска. Некоторые из новых функций доступны только при использовании этой стратегии.

Главное преимущество заключается в том, что почти каждое условие и параметр запроса можно задать непосредственно в объекте поискового запроса. Это делает его более подходящим для случаев, когда вам нужны запросы, зависящие от контекста. Благодаря своей простоте он также упрощает создание новой функциональности поверх него. Следующая функциональность доступна только при использовании этой стратегии:

- оператор запроса по умолчанию (для конкретного запроса);
- фильтровать запросы;
- свободные текстовые поисковые запросы;
- грани (не значения граней);
- поля, которые необходимо включить в ответ;
- усиливает.

·Примечание

The SolrQueryPostProcessor и SolrResultPostProcessor Интерфейсы не поддерживаются этой стратегией.

FacetSearchListener Интерфейс может быть использован для аналогичной функциональности.

Бесплатный текстовый поиск

Одной из новых функций этой стратегии является поддержка свободных текстовых поисковых запросов непосредственно в объекте поискового запроса. Поскольку существуют различные способы выполнения свободного текстового запроса, мы создали FreeTextQueryBuilder Интерфейс. Реализации этого интерфейса отвечают за преобразование представления в поисковом запросе в то, что Solr/Lucene может понять.

Доступны следующие бесплатные конструкторы текстовых запросов:

- MultiFieldFreeTextQueryBuilder: строит запрос таким образом, что итоговая оценка будет равна сумме оценок всех подзапросов;
- DisMaxFreeTextQueryBuilder: похоже на MultiFieldFreeTextQueryBuilder, но группирует некоторые подзапросы. Оценка для группы будет максимальной оценкой подзапросов, которые принадлежат этой группе (а не суммой). Этот конструктор запросов поддерживает следующие параметры:

Ключ	Значение по умолчанию	Возможные значения	Описание
groupByQueryType	истинный	правда, ложь	Изменяет способ группировки дизъюнктивных макс-запросов. Если установлено значение true, запросы также группируются по типу (где типы: свободный текстовый запрос, свободный текстовый нечеткий запрос, свободный текстовый подстановочный запрос).
галстук	0.0	0,0-1,0	<p>Позволяет вам настроить, насколько сильно на окончательную оценку запроса повлияют оценки полей с более низкими оценками по сравнению с полем с более высокими оценками:</p> <ul style="list-style-type: none"> ○ значение «0,0» делает запрос чистым «запросом на дизъюнкцию максимума»; ○ значение «1,0» делает запрос чистым «запросом дизъюнктивной суммы», где не имеет значения, какова максимальная оценка подзапроса. <p>Уравнение связи: (баллы совпадающего пункта с наивысшим баллом) + ((параметр связи) * (баллы любых других совпадающих пунктов))</p>

[Фильтровать запрос](#)

Функция фильтрации запросов является частью модуля поиска и навигации и используется для упрощения фильтрации содержимого поискового запроса.

Целью было заменить использование скрытых граней для фильтрации результатов. Следовательно, SolrSearchFilterQueryData список был представлен как часть SolrSearchQueryData, содержащий список необходимых данных для фильтрации.

```
<!-- ФильтрЗапросОператор -->
<enum class="de.hybris.platform.commerceservices.search.solrfacetsearch.data.FilterQueryOperator">
    <значение>И</значение>
    <значение>ИЛИ</значение>
</enum>

<bean class="de.hybris.platform.commerceservices.search.solrfacetsearch.data.SolrSearchFilterQueryData">
    <свойство имя="ключ" тип="Строка"/>
    <свойство имя="operator" тип="de.hybris.platform.commerceservices.search.solrfacetsearch.data.FilterQueryData" имя="values"/>
    <свойство тип="java.util.Set<String>"/>
</боб>
```

The Популятор, который заполняет данные для Популятор, корректирован для заполнения индексированных свойств как запросов фильтра, в случае если они не относятся к типу Facet с оператором OR; в противном случае они будут обрабатываться как обычные фасеты.

Популятор также собирает списки SolrSearchQueryData и добавляет их как запросы фильтра с AND в качестве оператора по умолчанию, на случай, если оператор не существует.

Примером решения запроса фильтра является реализация в коммерция услуга расширение и используется, например, в модуле комплектации для поиска продуктов, доступных для данного компонента.

Группировка

Группировка результатов помещает документы с общим значением поля в группы и возвращает верхние документы для каждой группы. Есть две реализации, которые вы можете использовать:

Метод	Приоритет	Дополнительная информация
Свернуть и развернуть	По умолчанию	https://solr.apache.org/guide/8_9/collapse-and-expand-results.html
Группа	Необязательный	https://solr.apache.org/guide/8_9/resultgrouping.html

Collapse and expand — это метод по умолчанию, который имеет лучшую производительность. Однако при использовании реализации **collapse and expand** выделение работает только для документов верхнего уровня. Чтобы использовать предыдущую реализацию группировки, задайте следующее свойство в конфигурации проекта:

```
solrfacetsearch.search.grouping.method=группа
```

Метод по умолчанию установлен следующим образом:

```
solrfacetsearch.search.grouping.method=свернуть-развернуть
```

API поиска

Узнайте больше об API верхнего уровня для функции поиска и выполните некоторые простые настройки.

Для выполнения поиска вам необходимо вызвать один из методов, предоставляемых **FacetSearchServiceИнтерфейс**. Создание и заполнение поискового запроса, а также обработка результатов могут показаться более сложными, поэтому мы привели несколько примеров использования методов интерфейса.

Создание поискового запроса

Начните с создания объекта **Search Query**. Для этого вы можете использовать **ПоискЗапросконструктор** для создания пустого поискового запроса:

```
SearchQuery searchQuery = new SearchQuery(facetSearchConfig, indexedType);
```

Вы также можете использовать **FacetSearchQueryМетоды** интерфейса:

- создать **ПоисковыйЗапросМетод** создания пустого поискового запроса:

```
SearchQuery searchQuery = facetSearchService.createSearchQuery(facetSearchConfig, indexedType);
```

- создать **ПопулярныйПоисковыйЗапросметод** создания заполненного поискового запроса. Подробности о группировке, фасетах и полях берутся из конфигурации поиска.

```
SearchQuery searchQuery = facetSearchService.createPopulatedSearchQuery(facetSearchConfig, indexedType);
```

- создать **FreeTextSearchQueryметод** создания заполненного поискового запроса. Подробности о группировке, фасетах, полях и свободном текстовом поиске взяты из конфигурации поиска.

```
SearchQuery searchQuery = facetSearchService.createFreeTextSearchQuery(facetSearchConfig, indexedType, user
```

- создать **ПоисковыйЗапросИзШаблонаметод** создания заполненного поискового запроса из шаблона. Подробности о группировке, фасетах и полях берутся из шаблона.

```
SearchQuery searchQuery = createSearchQueryFromTemplate(facetSearchConfig, indexedType, queryTemplateName)
```

- создать **ПоисковыйЗапросИзШаблонаметод** создания заполненного поискового запроса из шаблона. Подробная информация о группировке, фасетах, полях и свободном текстовом поиске взята из шаблона.

```
SearchQuery searchQuery = createFreeTextSearchQueryFromTemplate(facetSearchConfig, indexedType, queryTempla
```

Заполнение поискового запроса

В предыдущем разделе мы привели несколько примеров того, как создать объект поискового запроса. В некоторых случаях запрос не только создавался, но и заполнялся некоторыми параметрами запроса. Этого может быть недостаточно, и вы можете захотеть добавить некоторые дополнительные параметры запроса. Вот некоторые из наиболее распространенных вариантов использования:

- Добавление условий поиска:

```
searchQuery.addQuery(поле, значениеполя);
```

- Добавление условий поиска, не влияющих на подсчет очков:

```
searchQuery.addFilterQuery(поле, значение поля);
```

- Добавление граней и значений граней:

```
searchQuery.addFacet(поле);
searchQuery.addFacetValue(поле, fieldValue);
```

- Сортировка результатов:

```
searchQuery.addSort(поле);
```

-Примечание

Некоторые параметры запроса в объекте поискового запроса будут использоваться только вместе с новой стратегией поиска. При использовании устаревшей стратегии поиска эта информация будет взята непосредственно из конфигурации, а не из объекта поискового запроса. Затронутые параметры:

- Оператор запроса по умолчанию
- Фильтровать запросы
- Запросы на бесплатный текстовый поиск
- Границы (не значения граней)
- Поля, которые необходимо включить в
- ответ Boosts

Выполнение поискового запроса

После того, как объект поискового запроса создан и заполнен, нам нужно выполнить его, чтобы получить результат. Для этого нам просто нужно вызвать один из методов поиска изFacetSearchServiceинтерфейса.

- Пример 1:

```
РезультатПоиска searchResult = facetSearchService.search(searchQuery);
```

- Пример 2, то же, что и раньше, но с подсказками по поиску:

```
SearchResult searchResult = facetSearchService.search(searchQuery, searchHints);
```

Использование результата поиска

После выполнения запроса мы получаем объект результата поиска. Некоторые из наиболее распространенных вариантов использования:

- Получить количество результатов, соответствующих условиям поиска. Пример:

```
long numberOfResults = searchResult.getNumberOfResults();
```

- Получить результирующие документы. Это возвращает только часть всех результатов поиска в зависимости от размера страницы и смещения в поисковом запросе. Пример:

```
Список<Документ> документов = searchResult.getDocuments();
```

- Получите грани. Пример:

```
Список<Facet> фасетов = searchResult.getFacets();
```

Контекст и слушатели

Контекст

Некоторые объекты контекста доступны во время процесса поиска. Вы можете получить к ним доступ, используя соответствующую фабрику, или, в некоторых случаях, как параметр метода (например, слушатели).

Доступны следующие типы контекста:

- FacetSearchContext (вместе с FacetSearchContextFactory): этот интерфейс представляет контекст, действительный на протяжении всего поиска.

Слушатели

В этом разделе вы можете найти список типов прослушивателей, которые вы можете реализовать для перехвата процесса поиска.

FacetSearchListener

Используйте `FacetSearchListener` интерфейс, если вы хотите перехватить выполнение поисковых запросов.

```
/***
 * Интерфейс для получения уведомлений о выполнении фасетного поиска.
 */
открытый интерфейс FacetSearchListener {

    /**
     * Обрабатывает уведомление о том, что служба поиска фасетов собирается начать выполнение.
     *
     * @param facetSearchContext
     *         - {@link FacetSearchContext}
     *
     * @throws FacetSearchException
     *         если произошла ошибка
     */
    void beforeSearch(FacetSearchContext facetSearchContext) выдает исключение FacetSearchException;

    /**
     * Обрабатывает уведомление о том, что служба поиска по фасетам только что завершила работу.
     *
     * @param facetSearchContext
     *         - {@link FacetSearchContext}
     *
     * @throws FacetSearchException
     *         если произошла ошибка
     */
    void afterSearch(FacetSearchContext facetSearchContext) выдает исключение FacetSearchException;

    /**
     * Обрабатывает уведомление о сбое службы поиска фасетов (это также может быть связано со сбоем прослушивателей).
     *
     * @param facetSearchContext
     *         - {@link FacetSearchContext}
     *
     * @throws FacetSearchException
     *         если произошла ошибка
     */
    void afterSearchError(FacetSearchContext facetSearchContext) выдает FacetSearchException; }
```

Поисковые подсказки

Поисковые подсказки — это способ передачи дополнительной информации в процесс поиска. Эти подсказки можно получить в любое время, они являются частью контекста поиска. Пока не поддерживается ни одна конкретная подсказка.

Поставщики услуг индивидуальной сортировки

В этом документе содержится информация о настройке поставщиков сортировки для поиска фасетов Solr.

Вы можете выбрать один из поставщиков сортировки, которые идут в комплекте csolrfacetsearchрасширение. Вы также можете определить свой собственный поставщик сортировки, как описано в этом документе.

Поставщики сортировки по умолчанию

The solrfacetsearchрасширение поставляется с несколькими поставщиками сортировки. См. код ниже для примера.

```
<bean id="facetNameSortProviderAscending"          class="de.hybris.platform.solrfacetsearch.config.impl.DefaultFacetSortP
      <свойство имя="компаратор"      ref="facetDisplayNameComparator"/>
      <свойство имя="по убыванию"    значение="false"/>
  </боб>

<bean id="facetCountSortProviderDescending"        class="de.hybris.platform.solrfacetsearch.config.impl.DefaultFace
      <свойство имя="компаратор"      ref="facetCountComparator"/>
      <свойство имя="по убыванию"    значение="истина"/>
  </боб>

...
```

Различные поставщики сортировки всегда используют один и тот же базовый класс:

`de.hybris.platform.solrfacetsearch.config.impl.DefaultFacetSortProvider`. Единственное отклонение — это разница в реализации Comparator и nисходящем ag. Подробнее о создании настраиваемых поставщиков сортировки см. [Создание индивидуальных поставщиков сортировки](#).

Создание индивидуальных поставщиков сортировки

Контекст

Узнайте, как создать собственного поставщика сортировки.

Процедура

. Реализовать компаратор для `de.hybris.platform.solrfacetsearch.search.FacetValue` класс. Это было бы
Компаратор<ЗначениеФасета> и его можно использовать для сортировки значений фасета. Например, предположим, что вам нужно сделать сортировку по имени значения фасета.

открытый класс `CustomFacetNameComparator` реализует `Comparator<FacetValue>` {

```
@Переопределить
public int compare(final FacetValue значение1, final FacetValue значение2) {

    если (значение1 == null || значение2 == null) {

        возврат 0;
    }
    вернуть значение1.getName().compareTo(value2.getName());
}
```

. Зарегистрируйте свой компаратор в контексте Spring и определите поставщик сортировки, который его использует.

```
<!-- КОМПАРАТОР -->
<bean id="customFacetNameComparator" class="de.hybris.platform.solrfacetsearch.search.impl.comparat

<!-- ПОСТАВЩИК СОРТИРОВКИ -->
<bean id="customFacetNameSortProviderAscending"          класс="de.hybris.platform.solrfacetsearch.config.impl.DefaultFacetSortProvider
      <свойство имя="компаратор"      ref="customFacetNameComparator"/>
      <свойство имя="по убыванию"    значение="false"/>
  </боб>
```

. Укажите локализованные имена для вашего компонента в файлах свойств локализации в расширение\ресурсы\локализацияdirectory.
Соглашение для ключа свойства — `solrfacetsearch.sortprovider.yourbeannamealllowercase`

```
# Сортировать имена компонентов поставщика
solrfacetsearch.sortprovider.facetnamesortproviderascending=Сортировать по отображаемому имени
solrfacetsearch.sortprovider.facetcountsortpointerdescending=Сортировать по количеству значений фасета
solrfacetsearch.sortprovider.sizeattributesortprovider=Сортировать по размеру
solrfacetsearch.sortprovider.numericfacetsortproviderdesc=Сортировать по числовому значению по убыванию
solrfacetsearch.sortprovider.numericfacetsortproviderasc=Сортировать по числовому значению по возрастанию

solrfacetsearch.sortprovider.customfacetnamesortproviderasc=Сортировать по имени по возрастанию
```

. Скомпилируйте базу кода и перезапустите сервер Hybris.

Прямой доступ к запросу Solr

В этом документе содержится информация о прямом доступе к запросам solr.

-Примечание

Эта функциональность считается устаревшей и работает только при включенном устаревшем режиме в конфигурации поиска. Аналогичная функциональность может быть достигнута с помощью интерфейса FacetSearchListener. Для получения дополнительной информации см.[Процесс поиска](#) и[API поиска](#) документация.

Обзор

Постпроцессоры расширяют классы преобразователей по умолчанию с помощью дополнительной логики, которая зависит от конкретной конфигурации и реализации.

Постпроцессоры поисковых запросов

Реализация службы de.hybris.platform.solrfacetsearch.search.impl.DefaultFacetSearchService по умолчанию использует специальный преобразователь запросов.

```
открытый интерфейс SolrQueryConverter {
    /**
     * Преобразует экземпляр {@link SearchQuery} в действительный {@link SolrQuery}.
     */
    SolrQuery convertSolrQuery(SearchQuery searchQuery) выдает FacetSearchException; }
}
```

Основная цель API — обеспечить перевод поискового запроса в запрос сервера Solr. По умолчанию convertSolrQuery метод вызывается в DefaultFacetSearchService для
re.hybris.platform.solrfacetsearch.search.impl.DefaultFacetSearchService.translateSearchQuery(SearchQuery).

Результатирующее содержимое запроса поступает из Hybris в интеграцию Solr, которая реализована в solrfacetsearchрасширение. Однако, можно использовать некоторые другие функции сервера Solr, которые пока не поддерживаются напрямую solrfacetsearchрасширение.

Вот почему реализация по умолчанию
de.hybris.platform.solrfacetsearch.search.SolrQueryConverter.convertQuery(SearchQuery) методу предоставлены настраиваемые обработчики постобработки запросов, которые поддерживают настройку процесса преобразования по умолчанию.

Существует настраиваемый список доступных постпроцессоров запросов. Все изменения, поступающие от этих постпроцессоров, применяются в том же порядке, в котором они указаны в файле конфигурации. Это делается в конце процесса перевода.

Конвертер запросов объявлен как spring bean и используется службой DefaultFacetSearchService. Поэтому все хуки постпроцессора должны быть объявлены на уровне конвертера.

Пользовательский конвертер запросов и конфигурация

Чтобы объявить хук, вам необходимо реализовать de.hybris.platform.solrfacetsearch.search.SolrQueryPostProcessor Интерфейс. Пример кода ниже представляет собой пример такой реализации.

```
открытый класс SimpleSolrQueryPostProcessor реализует SolrQueryPostProcessor {

    @Переопределить
    Процесс SolrQuery (окончательный запрос SolrQuery, окончательный SearchQuery solrSearchQuery) {

        query.setSortField("id", ORDER.asc);
        query.setStart(Integer.valueOf(0));
        query.setRows(Integer.valueOf(10));
    }
}
```

Весенняя конфигурация

DefaultSolrQueryConverter содержит список, ссылающийся на все определенные постпроцессоры запросов. Постпроцессоры могут быть определены в форме экземпляра списка утилит. Перевод по умолчанию, определенный в конвертере, выполняется в начале процесса. Постпроцессоры применяются к запросу один за другим в конце процесса преобразования.

```
<псевдоним имя="defaultSolrQueryConverter" псевдоним="solrQueryConverter"/>
<bean id="defaultSolrQueryConverter" class="de.hybris.platform.solrfacetsearch.search.impl.DefaultSolrQueryConverter">
    <свойство name="queryPostProcessors" ref="solrQueryPostProcessors"/>
    <свойство name="solrFieldNameProvider" ref="solrFieldNameProvider"/> value="${facet.limit.default}"></property>
    <свойство имя="defaultLimit" property> value="${solr.indexedProperty.forbidden.char}"></property>
    <свойство имя="запрещенныйСимвол" />
</боб>

<alias name="defaultSolrQueryPostProcessors" alias="solrQueryPostProcessors"/> <util:list
id="defaultSolrQueryPostProcessors">
    <реф bean="simpleSolrQueryPostProcessor" />
</util:list>

<bean id="simpleSolrQueryPostProcessor" class="my.package.SimpleSolrQueryPostProcessor" />
```

Результат поиска Постпроцессоров

Аналогично постпроцессорам запросов, которые работают с объектом запроса Solr, существуют также постпроцессоры результатов, которые могут изменять объект результата поиска, возвращаемый поиском по индексу Solr.

Постпроцессоры результатов поиска работают наде.хybris.platform.solrfacetsearch.search.impl.SolrSearchResultуровень. Если вы хотите зарегистрировать постпроцессор, вам необходимо предоставить правильную конфигурацию пружины и создать собственную реализацию.

Весенняя конфигурация

Все постпроцессоры по умолчанию зарегистрированы в defaultFacetSearchService.

```
<alias name="defaultFacetSearchService" псевдоним="facetSearchService"/>
<псевдоним имя="defaultFacetSearchService" псевдоним="solrFacetSearchService"/> идентификатор="defaultFacetSearchService"
<боб класс="de.hybris.platform.solrfacetsearch.search.impl.DefaultFacetSearchService">
    <имя свойства="userService" ref="userService" /> <имя
    свойства="i18nService" ref="i18nService" /> <имя
    свойства="i18nService" ref="solrService" /> имя="catalogService"
    <свойство ref="catalogService" /> имя="fieldNameProvider"
    <свойство /> ref="solrFieldNameProvider"/>
    <свойство имя="solrQueryConverter" /> ref="solrQueryConverter"/>
    <свойство имя="resultPostProcessors" /> ref="solrSearchResultPostProcessors"/>
</боб>

<alias name="defaultSolrSearchResultPostProcessors" alias="solrSearchResultPostProcessors"/> <util:list
id="defaultSolrSearchResultPostProcessors">
    <реф bean="idleSolrResultPostProcessor"/>
</util:list>

<bean id="idleSolrResultPostProcessor" class="my.package.IdleSolrResultPostProcessor" />
```

Постпроцессоры результатов поиска должны реализовываться de.хybris.platform.solrfacetsearch.search.SolrResultPostProcessor абстрактный класс. Пример кода содержит пример.

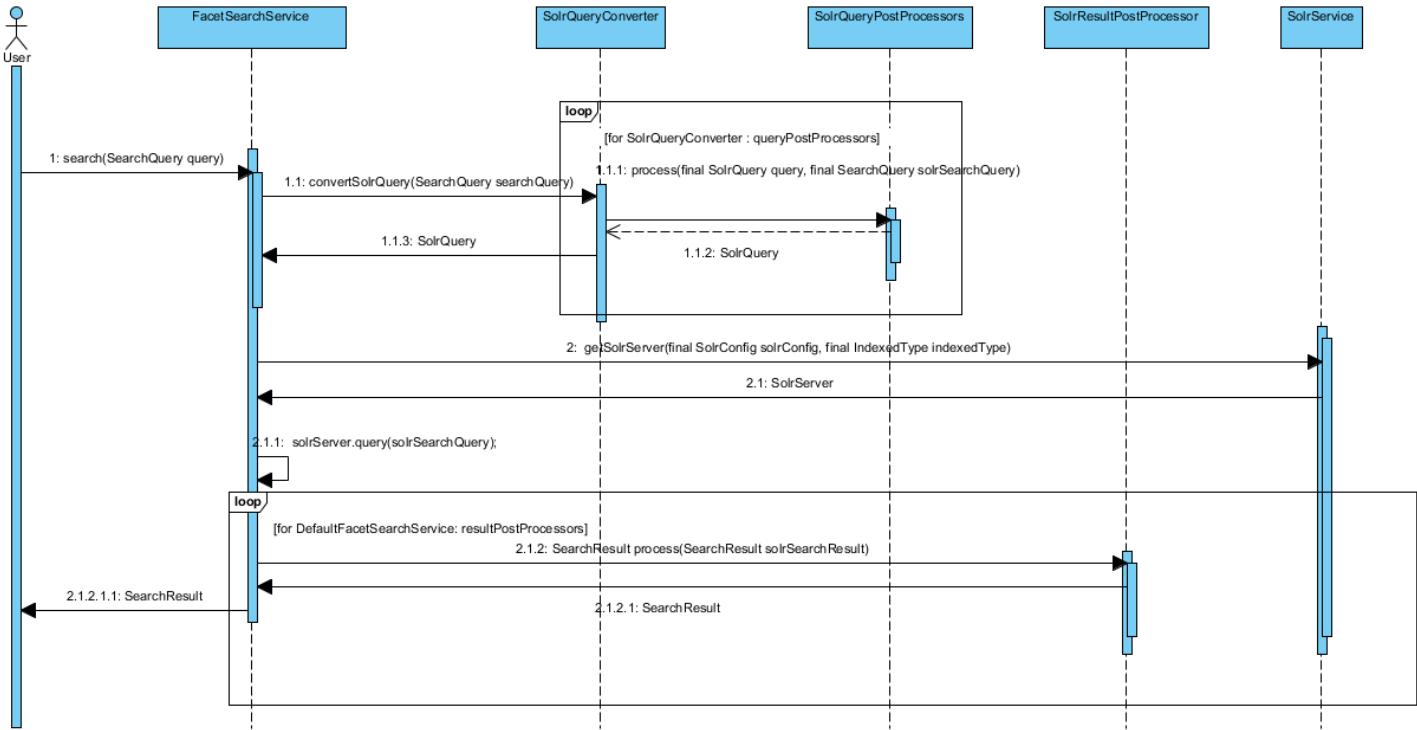
```
открытый класс IdleSolrResultPostProcessor реализует SolrResultPostProcessor {
```

```

@Переопределить
публичный процесс SearchResult(SearchResult solrSearchResult) {
    //сделать что-нибудь с результатом поиска: скрыть некоторые DTO результата, отсортировать значения фасетов, манипулировать с помощью breadcrumb
}
}

```

Обзор постобработки запросов и результатов поиска Solr



Дальнейшая настройка конвертера запросов

Для обеспечения гибкости можно задать несколько опций в реализации преобразователя запросов Solr. Таблица содержит пояснения к этим опциям.

Имя настраиваемого атрибута	Описание	Возможные значения	Значение по умолчанию
полевой оператор	Логический оператор для полей фасета в запросе Solr. Конвертер использует его при построении запроса Solr.	И/ИЛИ	И
фасетСортировать	Возможность сортировки значений фасетов в фасетах.	КОЛИЧЕСТВО/ИНДЕКС	ИНДЕКС
defaultFacetLimit	Ограничение по умолчанию для значений фасета для одного фасета.	ИНТ	свойство проекта = facet.limit.default
ЗапрещенныйСимвол	Символ является разделителем в именах полей Solr (например, имя_en_string)	чар	свойство проекта = solr.индексированное.свойство.запрещено.чар

Стандартная настройка конвертера запросов Solr.

```

<псевдоним имя="defaultSolrQueryConverter" псевдоним="solrQueryConverter"/>
<bean id="defaultSolrQueryConverter" class="de.hybris.platform.solrfacetsearch.search.impl.DefaultSolrQueryConverter">
    <свойство name="queryPostProcessors" ref="solrQueryPostProcessors"/>
    <свойство name="solrFieldNameProvider" ref="solrFieldNameProvider"/> value="${facet.limit.default}"></свойство>
    <свойство имя="defaultLimit" property> value="${solr.indexedproperty.forbidden.char}"</property>
    <свойство имя="запрещенныйСимвол"
    </боб>

```

Следующий пример кода демонстрирует пример настройки в Spring.

```
<псевдоним имя="customSolrQueryConverter" псевдоним="solrQueryConverter"/>
<bean id="customSolrQueryConverter" родитель="defaultSolrQueryConverter">
    <свойство имя="fieldOperator" значение="OR"/>
    <свойство имя="facetSort" значение="COUNT"/>
</боб>
```

Сопутствующая информация

[Spring Framework в SAP Commerce](#)

Результаты поиска пользовательских карт

В этом разделе описывается, как предоставлять пользовательские объекты передачи данных и преобразователи результатов.

-Примечание

Эта функциональность считается устаревшей. Если вам нужна похожая функциональность, напишите пользовательский конвертер, который преобразует Document в ваш Data Transfer Object. Подробнее о том, как получить список документов из результатов поиска, см.[API поиска](#).

Вы можете преобразовать результат поиска Solr в удобный объект передачи данных. Кроме того, вы также можете настроить процесс преобразования, реализовав собственные конвертеры и классы DTO. Это поможет вам обрабатывать информацию, полученную с сервера Solr, не совершая дополнительных обращений к базе данных.

Результат Конвертер

Преобразование документов Solr повышает производительность solrfacetsearchрасширение, позволяя ему использовать результат Solr для отображения индексированного контента в веб-фронтенде. Цель состоит в том, чтобы преобразовать SOLR org.apache.solr.common.SolrDocument, который представляет собой API-представление результата Solr в виде легкого, настраиваемого объекта DTO.

-Примечание

The solrfacetsearch Расширение поставляется с простой реализацией для преобразования продукта. В следующих частях документа оно будет служить примером:

- Конвертер:de.hybris.platform.solrfacetsearch.search.product.DefaultSolrProductConverter, который преобразует результирующий документ Solr в объект DTO.
- DTO:de.hybris.platform.solrfacetsearch.search.product.SolrProductData

Чтобы воспользоваться преимуществами преобразованного объекта DTO, вам понадобятся два элемента:

- DTO: Это должен быть ваш собственный класс DTO. Объект DTO должен быть простым объектом POJO, который содержит все свойства индексированного контента, который вы хотите представить.
- Конвертер: Это должна быть ваша собственная реализация преобразователя результатов Solr. Конвертер должен реализовывать de.hybris.platform.servicelayer.dto.converter.Converter<ИСТОЧНИК, ЦЕЛЬ>интерфейс, представляющий собой универсальный API преобразования на уровне сервисов платформы.

Реализовав все необходимые классы, вам нужно связать индексированный тип в вашей конфигурации Solr с вашим конвертером. Сделав это, вы можете вызвать новый специальный метод из нашего поискового API de.hybris.platform.solrfacetsearch.search.SearchResult:<T> Список<T> getResultData(). Этот метод возвращает результаты поиска с поддержкой DTO.

Использование результатов поиска с поддержкой DTO

Контекст

В следующих разделах описываются шаги, необходимые для использования результатов поиска с поддержкой DTO.

Процедура

. Создайте класс POJO, содержащий поля для свойств, которые вы хотите прочитать из результатов поиска Solr. Например:
de.hybris.platform.solrfacetsearch.search.product.SolrProductData

SolrProductData

```
открытый класс SolrProductData {

    частный Нить код;
    частный Нить имя;
    частный Нить описание;
    частный Нить Версия каталога;
    частный Нить каталог;
    частный Длинный рк;
    частный Коллекция<Строка> категории;
    частный Двойной цена;
    частный Нить ean;

    /**
     * @возвращаться the код
     */
    публичная строка получитьКод()
    {
        возвращаться КОД;
    }

    /**
     * @парам код
     *          код для установки
     */
    public void setCode(конечный код строки) {

        этот.код = код;
    }

    /**
     * @возвращаться the имя
     */
    публичная строка получитьИмя()
    {
        возвращаться ИМЯ;
    }

    /**
     * @парам имя
     *          имя для установки
     */
    public void setName(конечное имя строки) {

        это.имя = имя;
    }

    /**
     * @возвращаться the описание
     */
    публичная строка получитьОписание()
    {
        возвращаться ОПИСАНИЕ;
    }

    /**
     * @парам описание
     *          описание для установки
     */
    public void setDescription(финальное строковое описание) {

        это.описание = описание;
    }

    /**
     * @возвращаться the каталогВерсия
     */
    публичная строка получитьCatalogVersion()
    {
        возвращаться Версия каталога;
    }

    /**
     * @парам каталогВерсия
     *          каталогВерсия для установки
     */
    public void setCatalogVersion(final String catalogVersion) {

        this.catalogVersion = catalogVersion;
    }
}
```

```

    }

    /**
     * @возвращаться the каталог
     */
    публичная строка получитьКаталог()
    {
        возвращаться каталог;
    }

    /**
     * @парам каталог
     *      каталог для установки
     */
    public void setCatalog(конечный строковый каталог) {

        этот.каталог = каталог;
    }

    /**
     * @возвращаться the ПК
     */
    публичный длинный получитьPk()
    {
        возвращаться ПК;
    }

    /**
     * @парам ПК
     *      ПК для установки
     */
    public void setPk(final Long pk) {

        этот.ПК = pk;
    }

    /**
     * @возвращаться the категории
     */
    публичная коллекция<String> { получитьКатегории()

        категории возврата;
    }

    /**
     * @парам категории
     *      категории для установки
     */
    public void setCategories(Collection<String> категории) {

        this.categories = категории;
    }

    /**
     * @возвращаться the цена
     */
    публичный двойной получитьЦену()
    {
        возвращаться цена;
    }

    /**
     * @парам цена
     *      цену, которую нужно установить
     */
    public void setPrice(double цена) {

        эта.цена = цена;
    }

    /**
     * @возвращаться the ean
     */
    публичная строка получитьEan()
    {
        возвращаться ean;
    }

    /**
     * @парам ean
     *      ean установить
     */
    public void setEan(String ean) {

        этот.ean = ean;
    }

```

}

}

The `solrfacetsearch` расширение имеет абстрактную реализацию для преобразователей результатов Solr: `de.hybris.platform.solrfacetsearch.search.AbstractSolrConverter<T>`. Он реализует Конвертеринтерфейс (`de.hybris.platform.servicelayer.dto.converter.Converter<ИСТОЧНИК, ЦЕЛЬ>`) и предоставляет защищенные методы утилит для получения индексированных значений свойств из документов Solr. Для перевода индексированных свойств Solr он использует `de.hybris.platform.solrfacetsearch.provider.ИмяПоляПоставщикинтерфейс`.

открытый абстрактный класс `AbstractSolrConverter<T>` реализует `Converter<SolrResult, T>` {

```
частный ПоставщикИмениПоля fieldNameProvider;
/***
 * Возвращает пустой экземпляр шаблона для цели конверсии
 */
защищенный абстрактный T createDataObject();
...
```

. РасширитьАннотация `SolrConverter` класс для вашего класса DTO для косвенной реализации `de.hybris.platform.servicelayer.dto.converter.Converter<ИСТОЧНИК, ЦЕЛЬ>` интерфейс для него. Например, `de.hybris.platform.solrfacetsearch.search.product.DefaultSolrProductConverter`

открытый класс `DefaultSolrProductConverter` расширяет `AbstractSolrConverter<SolrProductData>` {

```
частная статическая окончательная строка КАТАЛОГ = "каталог";
частная статическая окончательная строка ПК = "pk";
частная статическая окончательная строка ОПИСАНИЕ = "description";
частная статическая окончательная строка ИМЯ = "name";
частная статическая окончательная строка CATALOG_VERSION = "catalogVersion";
частная статическая окончательная строка CATEGORIES = "category"; частная
статическая окончательная строка PRICE = "priceValue";
частная статическая конечная строка CODE = "code";
частная статическая конечная строка EAN = "ean";
```

`@Переопределить`

public SolrProductData convert(final SolrResult solrResult, final SolrProductData target) бросает Со {

```
target.setCatalog(this.<String> getValue(solrResult, CATALOG)); target.setPk(this.<Long>
getValue(solrResult, PK)); target.setName(this.<String> getValue(solrResult, NAME));
target.setDescription(this.<String> getValue(solrResult, DESCRIPTION));
target.setCatalogVersion(this.<String> getValue(solrResult,
target.setCategories((Collection<String>                                         ВЕРСИЯ_КАТАЛОГА));
                                                 это.получитьЗначение(solrResult, КАТЕГОРИИ));
                                                 получитьЗначение(solrResult, ЦЕНА));
target.setPrice(this.<Double>                                         получитьЗначение(solrResult, КОД));
target.setCode(this.<String>                                         получитьЗначение(solrResult, ЕАН));
target.setEan(this.<String>                                         получитьЗначение(solrResult, ЕАН));

возврат цели;
}
```

`@Переопределить`

```
защищенный SolrProductData     создатьDataObject()
{
    возвращаться новый SolrProductData();
}
```

. Зарегистрируйте конвертер в Spring Context.

```
<псевдоним имя="defaultSolrProductConverter" псевдоним="mySolrProductConverter"/>
<bean id="defaultSolrProductConverter" class="de.hybris.platform.solrfacetsearch.search.product.Def
```

. Если вы используете Backoffice Administration Cockpit для настройки индекса Solr, вы также найдете соответствующее поле редактора для конвертера документов Solr. Наведите указатель мыши на метку Результат конвертера для отображения справочной информации с кратким описанием поля ввода текста.

Multithreaded Synchronization

- Retention Rule
- Search and Navigation
 - Facet Search Configurations
 - Indexed Types
 - Ranges
 - Backoffice
 - Search Profiles
 - Search Profile Activation Sets

SAVED QUERIES

No queries

apparel-ukProductType ComposedTypeModel (8796097970258@21)

INDEXED TYPE INDEXED PROPERTIES FREE TEXT QUERY GROUPING SOLR SORTS ADMINISTRATION

SEARCH CONFIGURATION

INDEXER CONFIGURATION

OTHERS

Identity provider Model loader Model fields values provider

Result converter

Result converter (spring bean id) needed to convert Solr result data into a simple data object that could be used to display indexed type (i.e. product) data directly on the front end.