

Р. А. Кацора

РАЗРАБОТКА И РЕАЛИЗАЦИЯ ПОИСКОВОЙ СИСТЕМЫ ДЛЯ ЭЛЕКТРОННОЙ БИБЛИОТЕКИ

Реализован проект клиент-серверного приложения для электронной библиотеки на основе платформы полнотекстового поиска Apache Solr. Приложение построено на базе технологий Spring Framework, Spring Boot и Angular Framework. Web-интерфейс приложения был построен с использованием библиотеки компонентов Angular Material. Проект представляет из себя web-сервер, развернутый на локальной машине, под управлением контейнера сервлетов Tomcat, а также индекс библиотеки книг, который управляется платформой полнотекстового поиска Apache Solr.

Одна из наиболее важных проблем в современном информационном мире – это хранение электронных документов. Создание систем для обработки, анализа, хранения, контроля и управления документами призвано решить эту проблему как для рядовых пользователей, так и для корпоративных клиентов. Предприятиям, обладающим большими базами документов, важно учитывать наличие у себя тех или иных документов, а также организовывать оперативный доступ к ним.

С этой целью для более эффективной работы предприятия создана поисковая система на основе платформы полнотекстового поиска Apache Solr. Также разработано web-приложение, использующее функционал данной поисковой системы. Данное web-приложение в удобной и понятной форме принимает запросы от пользователя и возвращает результаты поиска. Результаты выдаются частями для уменьшения нагрузки на канал передачи данных и сервер приложения. Предусмотрена возможность отфильтровать результаты поиска, используя средства поисковой системы. Кроме того, существует возможность загрузки файлов книг для индексации поисковой платформой. Было решено ряд задач:

- разработка непосредственно самой поисковой системы;
- создание web-приложения.

Технологии Spring, Spring Boot и Angular Framework были выбраны для наибольшей устойчивости, а также контроля ресурсов, выделяемых сервером.

Spring Framework выбран из-за наличия IoC-контейнера (Inversion of Control), что позволяет создать модульное и конфигурируемое приложение. Кроме того, данный фреймворк имеет модуль Spring Boot, который упрощает процесс разработки web-приложения.

Серверная часть приложения представляет собой Restful web-сервис, построенный с использованием Spring Boot. Клиентская часть приложения представляет собой web-приложение, разработанное с использованием Angular Framework.

За сборку приложения отвечает Apache Maven. Он же отвечает за управление зависимостями данного проекта. Для того, чтобы выполнить сборку, необходимо воспользоваться командой mvn clean install. Также, в ходе сборки могут быть выполнены тесты исходного кода приложения. В результате в каталоге target будет создан файл с расширением .war, который можно установить на web-сервер. Кроме того, данный war-файл будет содержать все зависимости приложения в каталоге WEB-INF\lib.

В качестве web-сервера был выбран Apache Tomcat.

Для того, чтобы установить приложение на web-сервер, необходимо выполнить сборку приложения, а затем полученный файл с расширением .war поместить в каталог webapps web-сервера.

Для запуска web-сервера можно воспользоваться консолью и из корневого каталога выполнить команду `bin\catalina start` или воспользоваться `bin\startup.bat` для Windows и `bin\startup.sh` для Linux.

Данное web-приложение использует фреймворк Spring для построения каркаса приложения.

Класс `DocumentServiceApplication` является центральным элементом приложения и используется для запуска Spring Boot-приложения.

Класс `ServletInitializer` предоставляет конфигурацию Spring Boot-приложения.

Класс `WebConfig` содержит описание классов-бинов приложения и конфигурацию области `DispatcherServlet`.

Класс `SwaggerConfig` содержит конфигурацию Swagger фреймворка, используемого для визуального представления Rest API приложения.

Хранением данных, а также их обработкой и индексированием занимается Apache Solr. Данные сохраняются в коллекции (`collection/core`). В данном случае создана коллекция с именем `booklibrary`.

Информация о книгах представлена записями – полями, каждое из которых содержит следующую информацию:

- `id` – уникальный идентификатор;
- `identifier`– ISBN;
- `title` – название;
- `authors` – список авторов;
- `description` – описание;
- `content` – содержимое.

Конфигурация коллекции находится в файле `solrconfig.xml`. В данном файле можно изменить типы полей книги, добавить новые, определить необходимость их индексации и многое другое.

Бизнес логика приложения представлена интерфейсами `SearchService` который возвращает список документов, соответствующих поисковому запросу, ограниченному списком авторов, с указанием номера начальной строки выборки Apache Solr и максимального количества записей в результате) и `UploadService` и их реализацией.

Интерфейс `UploadService` выполняет добавление документа в индекс Apache Solr формата `Epub`.

За сборку проекта и управление зависимостями отвечает `Node.js`. Файл конфигурации содержит список зависимостей, а также список команд для развертывания и отладки проекта с помощью командной строки в секции «`scripts`».

Для того, чтобы выполнить сборку, необходимо воспользоваться командой `npm run build` без аргументов для сборки в «`development`» режиме или с аргументом `--prod` для сборки в режиме «`production`». В результате каталог `dist` будет содержать скомпилированные файлы проекта, готовые к развертыванию на web-сервер.

Среда `Node.js` предоставляет встроенный web-сервер для быстрого развертывания проекта. Для этого можно воспользоваться командой `npm run start`. В результате произойдет сборка и развертывание проекта. Хост, порт и путь к приложению могут быть изменены, например, при установке на удаленный web-сервер. Кроме того, web-сервер будет отслеживать изменения исходных файлов проекта и, при необходимости, выполнять повторную сборку измененных файлов и повторное развертывание приложения.

Данное приложение содержит 4 компонента: `document`; `search`; `upload`; `page-not-found`.

Компонент `document.component.ts` представляет собой контейнер, в котором будет отображаться содержимое запрошенного документа. Компонент `search` представляет собой форму поиска, секцию фильтров, а также список результатов поиска. Компонент `upload`. представляет собой форму для загрузки файла на сервер для последующего индексирования платформой полнотекстового поиска Apache Solr. Компонент `page-not-found` представляет собой содержимое страницы с ошибкой в случае запроса на недопустимый адрес.

После открытия поисковой системы в браузере пользователь оказывается на главной странице, где отображается строка для ввода поискового запроса.

При вводе поискового запроса пользователю автоматически будут предложены наиболее релевантные варианты названий книг в окне подсказки ниже поисковой строки. Количество предложений конфигурируемо и выбрано в размере пяти.

Нажав кнопку «Search» или клавишу «Enter» на клавиатуре, будет осуществлен поиск по индексу Apache Solr.

Результаты отобразятся пользователю ниже строки поиска в виде контейнеров. Каждый контейнер содержит информацию о конкретной книге – название, список авторов и выдержка из текста книги, в которой фигурирует текст поискового запроса или аннотация, если книга была добавлена в выборку по совпадению автора или названия.

После того, как будут отображены результаты поиска, пользователь может дополнительно отфильтровать результаты – указать одного из авторов или год публикации книги с помощью фасетного поиска.

Список авторов найденных книг отображается слева от результатов поиска. Также для каждого автора указывается количество книг, удовлетворяющих запросу. После выбора автора останутся только книги, одним из авторов которых является данный автор. Выбранный автор переместится в начало списка и значок с количеством книг изменится на крестик, который позволяет удалить данного автора из фильтра.

После выбора необходимой книги будет осуществлен переход на страницу с полным текстом данной книги.

Также реализован постраничный просмотр результатов поиска. Переключение между страницами осуществляется с помощью специальной области внизу главной страницы. Можно воспользоваться как и кнопками для перехода на одну страницу вперед или назад, так и кнопками для перехода к началу или в конец списка результатов поиска.

В случае если поиск не дал результатов, приложение выведет соответствующее уведомление пользователю.

Пользователь имеет возможность загружать документы на сервер. При необходимости загружаемые документы можно дополнить информацией – заголовком, списком авторов, датой публикации и аннотацией.

Литература

1 Кацора Р. А. Разработка клиент-серверного приложения для электронной библиотеки документов / Р. А. Кацора // Новые математические методы и компьютерные технологии в проектировании, производстве и научных исследованиях: материалы XXI Республиканской научной конференции студентов и аспирантов, Гомель, 19–21 марта 2018 г. – Гомель, 2018. – С. 142–143.

2 Кацора Р. А. Разработка клиент-серверного приложения для электронной библиотеки документов / Р. А. Кацора // Вычислительные методы, модели и образовательные технологии: сборник материалов Международной научно-практической конференции, Брест, 19–20 октября 2017 г. – Брест, 2017. – С. 234.

УДК 004.42

А. Р. Качкан

РЕАЛИЗАЦИЯ СЕРВЕРНОЙ ЧАСТИ ПРИЛОЖЕНИЯ «ПОДБОР РЕШЕНИЙ, СООТВЕТСТВУЮЩИХ ПРЕДНАЗНАЧЕНИЮ ЧЕЛОВЕКА»

Статья посвящена описанию серверной части приложения для профориентации. Разработанная версия клиент-серверного web-приложения обеспечивает регистрацию в приложении; добавление новых тестов в базу данных; редактирование уже