

ПРОЕКТИРОВАНИЕ REST API ДЛЯ РЕАЛИЗАЦИИ ВЕБ-СЕРВИСА ИНТЕРНЕТ-МАГАЗИНА

Молтановский Роман Дмитриевич

студент Уральского Федерального Университета имени первого Президента России Б.Н. Ельцина, РФ, г. Екатеринбург

Тимошенко Сергей Иванович

научный руководитель,

Аннотация. Целью статьи является проектирование REST API с подбором оптимальных инструментов и фреймворков для разработки и продакшена веб-сервиса магазина бытовой техники.

Основным методом является подбор технологий и проектирование. Результатом исследования является выбор оптимальных инструментов и фреймворков и готовая схема REST API веб-сервиса интернет-магазина.

Ключевые слова: Spring Boot, Vue.js, REST API, PostgreSQL, HTTP, интернет-магазин.

Введение

Для разработки серверной части программы будет использоваться язык Java, фреймворк Spring Boot, Hibernate. Клиентская часть программы будет создаваться с использованием языка JavaScript и фреймворка Vue.js и рассчитана на функциональный и интуитивно понятный интерфейс, взаимодействие с сервером осуществляется с помощью веб-браузера.

Обмен данными между клиентом и сервером будет осуществляться через протокол HTTP, используя JavaScript библиотеку Axios на стороне клиента и java класс RestController на стороне сервера.

1. Слои целевой REST API

Spring Boot — это java фреймворк, целью которого является упрощение создания приложений на основе Spring. Он позволяет наиболее простым способом создать web-приложение, требуя от разработчиков минимум усилий по его настройке и написанию кода. **Spring Boot** обеспечивает по умолчанию весь необходимый набор классов и библиотек для построения бекенда REST API.

Построение REST API будет проходить по схеме: Слой сущностей данных -> Сервисный слой обработки данных сервером -> Слой REST Контроллеров Spring -> Vue.js HTTP Client -> Сервисный слой обработки данных клиентом.

1) Слой сущностей данных

Все данные веб-сервиса представлены в виде java-классов, в базе данных хранятся как

сущности, отмеченные аннотацией @Entity. За формирование таблиц и преобразование java-кода в SQL отвечает фреймворк Hibernate, за очереди запросов к базе данных отвечает SpringData Jpa - механизм для взаимодействия с сущностями базы данных, обертка над JPA, реализуется через java-интерфейс JpaRepository.

2) Сервисный слой обработки данных сервером

Сервисный слой обработки данных служит для реализации бизнес-логики веб-сервиса, содержит в себе несколько подсистем, такие как:

Система генерации товаров, Система обновление БД, Система администрирования контента. На этом уровне данные проходят через большое количество различных обработок и процедур, в результате которых данные сохраняются в БД и отправляются на клиент.

3) Слой REST Контроллеров Spring

На контроллерном слое, сервер веб-сервиса принимает и обрабатывает HTTP запросы от клиента, принимает параметры и данные запросов с клиента, затем передает входящие данные на сервисный слой различным обработчикам данных, в зависимости от типа HTTP метода и URL адреса запроса.

Выбранный метод обработчика с помощью методов репозитория JpaRepository запрашивает нужные данные из БД, преобразует их в обрабатываемые классы, реализует бизнес-логику, сериализует исходящие данные в JSON и возвращает данные обратно в контроллер, который в свою очередь по протоколу HTTP возвращает данные на клиент, в качестве ответа от сервера.

4) Vue.js HTTP Client

В качестве HTTP клиента на клиентской стороне веб-сервиса выступает JavaScript библиотека Axios. Axios это клиентская библиотека, которая по умолчанию использует промисы (promise) - процесс обработки, который должен быть запущен в случае успеха и неуспеха окончания обработки функции, передает сформированные данные в HTTP запросе на сервер и принимает от него ответ.

5) Слой обработки данных клиентом

Сервисный слой на стороне клиента напоминает по структуре слой на стороне сервера, методы клиентского сервисного слоя готовят и обрабатывают данные как для отправки на сервер, так и для рендера на страницах приложения.

2. Дополнительные сервисы

Для работы в продакшн-среде веб-сервиса магазина бытовой техники необходим сервер-хостинг и сервер для хранения медиа-данных, поэтому серверная-часть веб-сервиса будет включать в себя функционал по загрузке медиа-контента по протоколу HTTPS, а также обработку веб-мониторинга данных о состоянии сервиса в продакшн-среде.

3. Итоговая схема REST API

В результате проектирования структуры REST API, конечный результат можно представить в виде диаграммы на рисунке 1.

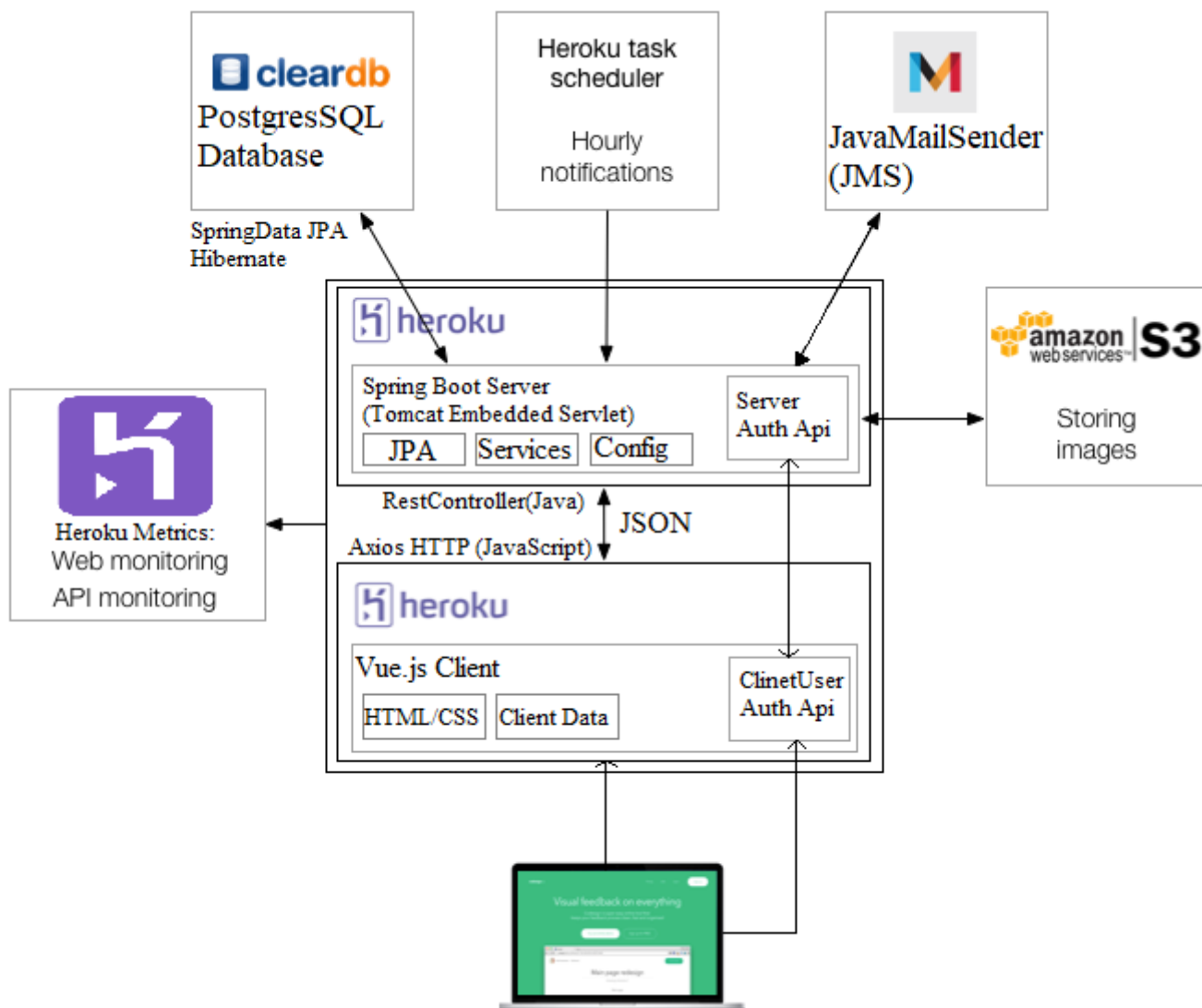


Рисунок 1. Итоговая схема REST API

Список литературы:

1. Официальный сайт проекта Spring: сайт. URL: <http://spring.io/> (дата обращения 15.10.2020). - Текст : электронный
2. Официальный сайт проекта Vue.js: сайт. URL: <https://vuejs.org/> (дата обращения 10.10.2020). - Текст : электронный
3. Rest-Assured : сайт. URL: <https://rest-assured.io/> (дата обращения 10.09.2020). - Текст : электронный
4. Spring Getting Started Guides: сайт. URL: <https://spring.io/guides/> (дата обращения 10.09.2020). - Текст : электронный