Distributed methods and technologies based on XML - Final Project: Music Library Management

Ionaşcu Alexandru, Dumitrescu Călin

February 2020

1 Introduction

For this project, we chose to make a music app. The database with songs, artists, publish date, genres is an XML document. The user can add new data, delete or modify from the browser interface [Fig.1]. Apart from the input dialog the interface also contains buttons for the different functions and an overview of the data inside the database: List of songs, List of authors, List of genres.

2 Technologies

XML stands for an extensible markup language. It is used to lay a set of rules for text formatting and data presentation. XML is extensible because it is a framework, anyone can add to and modify to suit their needs. Markup means it is used to define text. XML is just a way of making data both human and machine-readable.

To manipulate XML documents, XPath is used. XPath is a language for locating nodes in a document tree. In other words, XPath is used to navigate an XML document. XPath is an important part of XSLT which will be described briefly, hereafter. Selecting nodes can be done in multiple ways. For instance using axes, the relationship between them (parent/ancestors/siblings), or by node name, attributes, predicates or even wildcards.

XSLT stands for extensible stylesheet language transformations. As the name implies, it is a language for transforming XML documents. It is necessary because XML is a generic markup language as opposed to HTML for example. Browsers won't know how to display the information and computer programs won't be able to consume XML documents unless the documents have the specific structure for that individual program. This app makes use of the Apache Camel integration engine to facilitate interactions between services and technologies and spring boot to create REST web services.

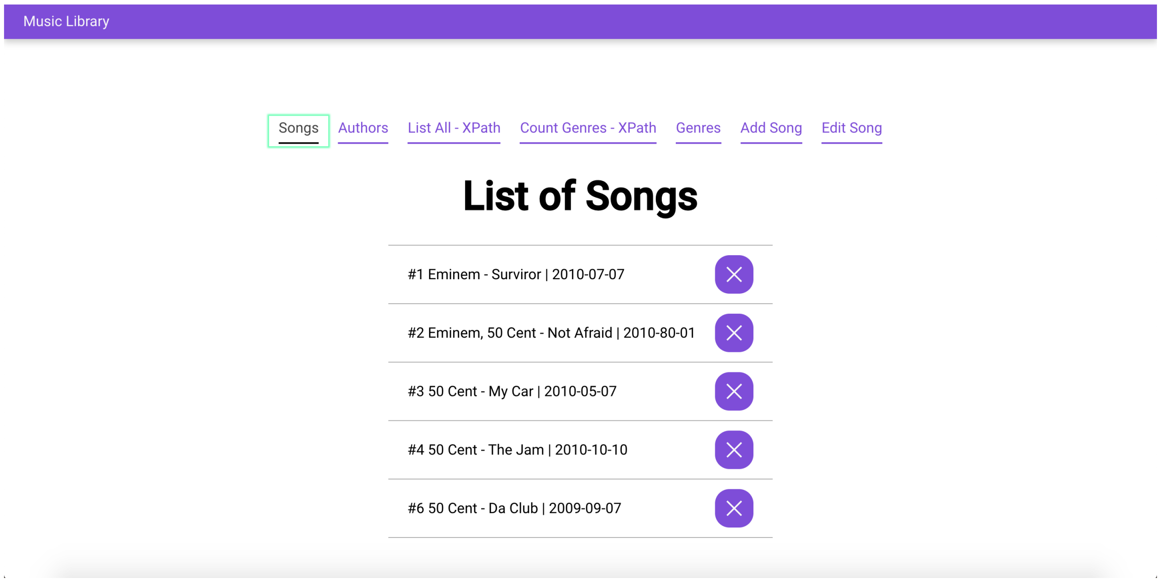
3 Implementation

Camel is also known as a routing and mediation engine as it effectively routes data between endpoints. Concept: Endpoint receives the message → Route transports message → EIP (Enterprise Integration Pattern) processes the message → Route → Endpoint sends the message to consuming program.

```
@Component
public class UpdateSongProcessor implements Processor {

        @Autowired
        private MusicLibraryService service;


        @Override
        public void process(Exchange exchange) throws Exception {
                service.updateSong(exchange.getIn().getBody(Song.class));
        }
}
```

REST uses HTTP requests to GET, PUT, POST and DELETE data resources. Example of GET (retrieve resource representation/information only):

```
rest()
    .get("/genres")
    .produces(MediaType.APPLICATION_XML_VALUE)
    .route()
        .setBody(() -> service.listCategories())
        .endRest();

rest()
    .put("/song")
    .consumes(MediaType.APPLICATION_JSON_VALUE)
    .type(Song.class)
    .outType(Song.class)
        .route()
        .process(updateSongProcessor)
        .endRest();
```



List of Songs

#1 Eminem - Surviror | 2010-07-07

#2 Eminem, 50 Cent - Not Afraid | 2010-80-01

#3 50 Cent - My Car | 2010-05-07

#4 50 Cent - The Jam | 2010-10-10

#6 50 Cent - Da Club | 2009-09-07

Figure 1: React-based GUI