Automating XML generation from SQL

**Author**:Rabi

**Date**:14th Aug,2023



XML generation from dynamically generate SQL from UI.

It would save time for QA team where XML preparation required from Database manually

1. **Challenges of Manual XML Preparation**

* Manually preparing XML from the database can be a **time-consuming** and **error-prone** process.
* Let's delve into the challenges that the QA team encounters when handling this task.
* **Challenge 1: Potential Errors:**
  + **Human Error**- When QA team members manually extract data from the database to create XML files, there's a higher risk of **human errors** creeping in.



* + **Incorrect data extraction**, **incorrect tag names**- and missing or misplaced elements are common issues that can arise due to manual entry.



Example:

Imagine a scenario where the QA team needs to extract **customer order data** for generating an XML report. A small mistake, such as misplacing a closing tag or omitting a key attribute, could result in an **XML file that's not valid or accurate**.

* **Challenge 2: Time Consumption:**
  + **Time Consuming**- Preparing XML manually can consume a significant **amount of time**, especially when dealing with **large datasets**.



* + **XML Schema following**- QA team members need to manually write queries, retrieve data, format it into XML structure, and ensure it adheres to the required schema.

Example:

Consider a case where the QA team needs to generate XML reports for multiple products across various categories. Manually creating XML for each product, complete with **appropriate tags** and attributes, can be time-intensive and **delay the testing process.**

* **Challenge 3: Inconsistencies:**
  + **Inconsistent** XML formatting can occur due to variations in how different QA team members approach the task.
  + **Lack of standardized processes** can lead to varying XML structures and naming conventions.

Example:

Suppose the QA team is working on XML reports for different regions, and each member follows a slightly different structure. This inconsistency can create confusion, impact data analysis, and hinder collaboration.

1. **Solution: Automated XML Generation**

Automating XML generation using Spring Boot, JPA, and the H2 Database addresses these challenges by providing a streamlined, consistent, and efficient approach.

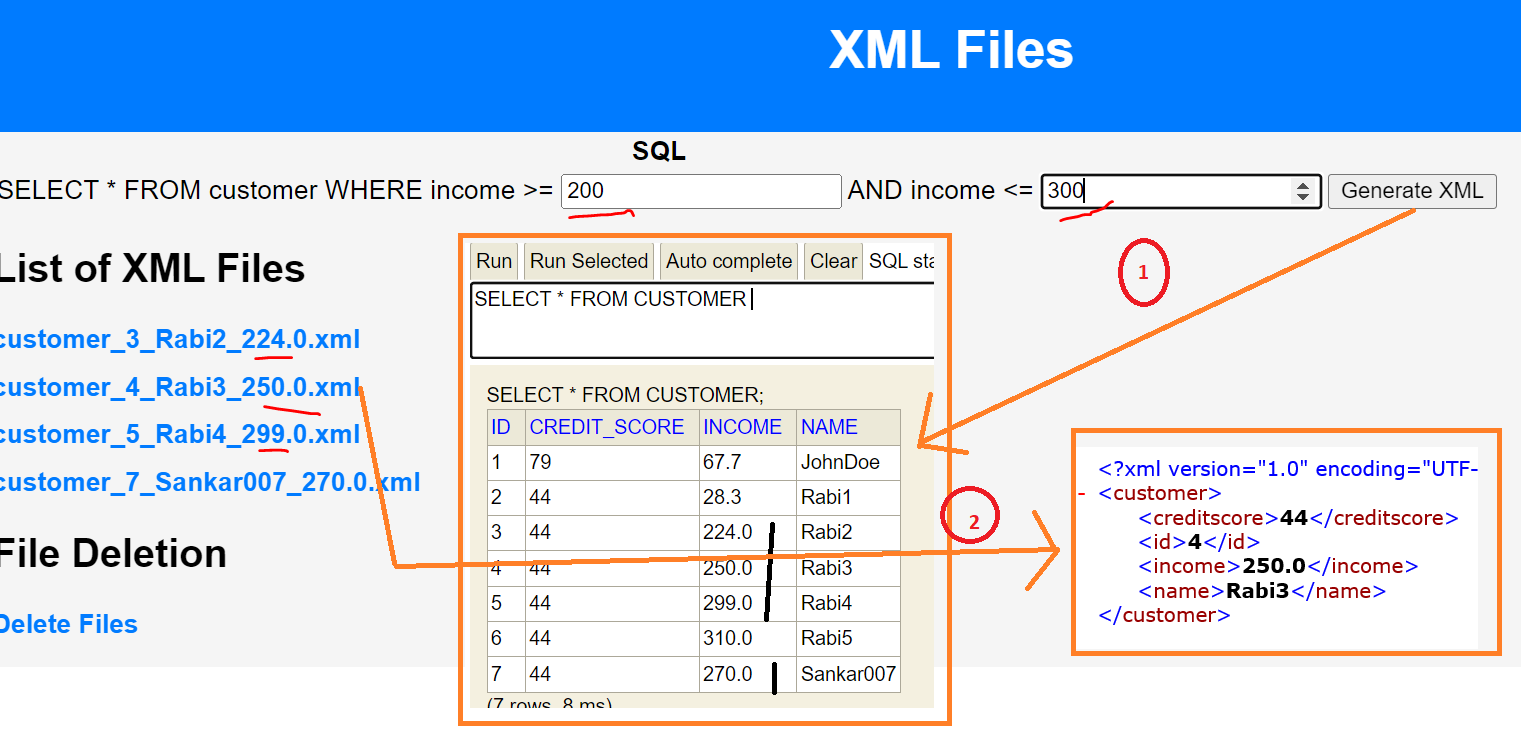


* Benefits of Automated XML Generation

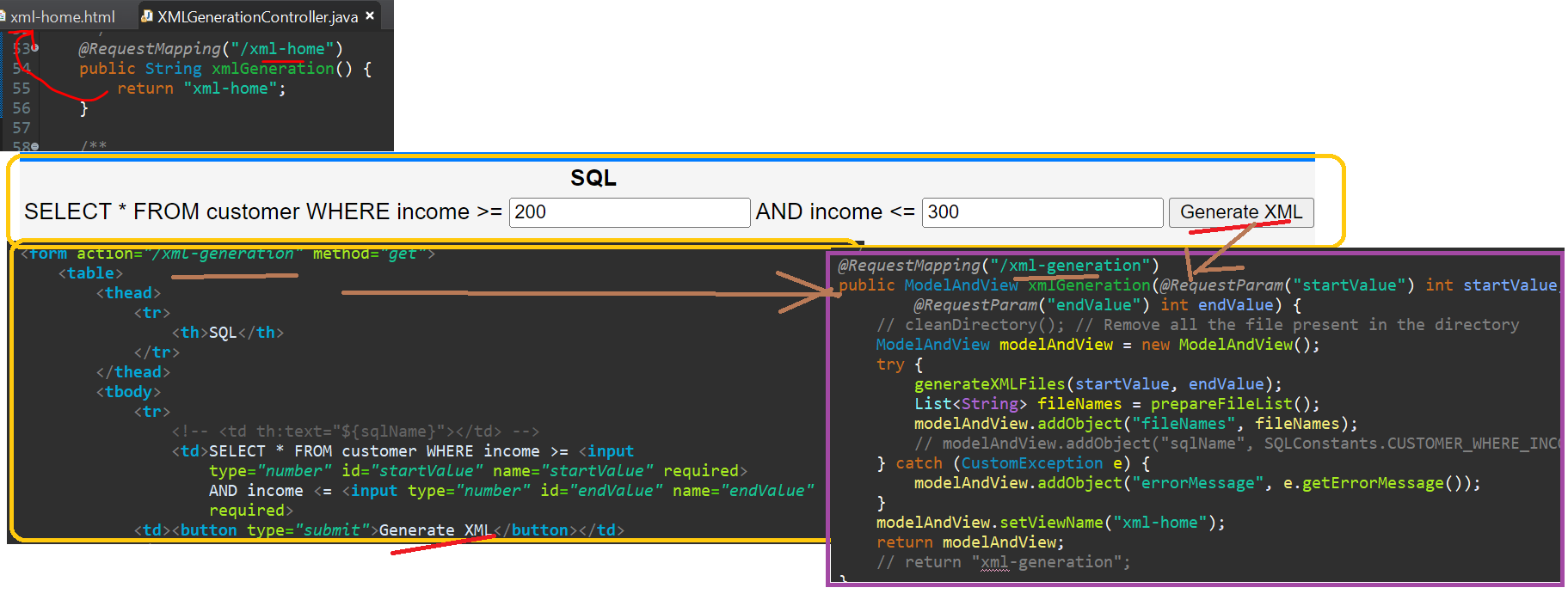
List the benefits of using the automated approach:

* + - Time-saving for both development and QA teams
    - Reduced chances of human error
    - Consistency in XML output
    - Improved efficiency and productivity

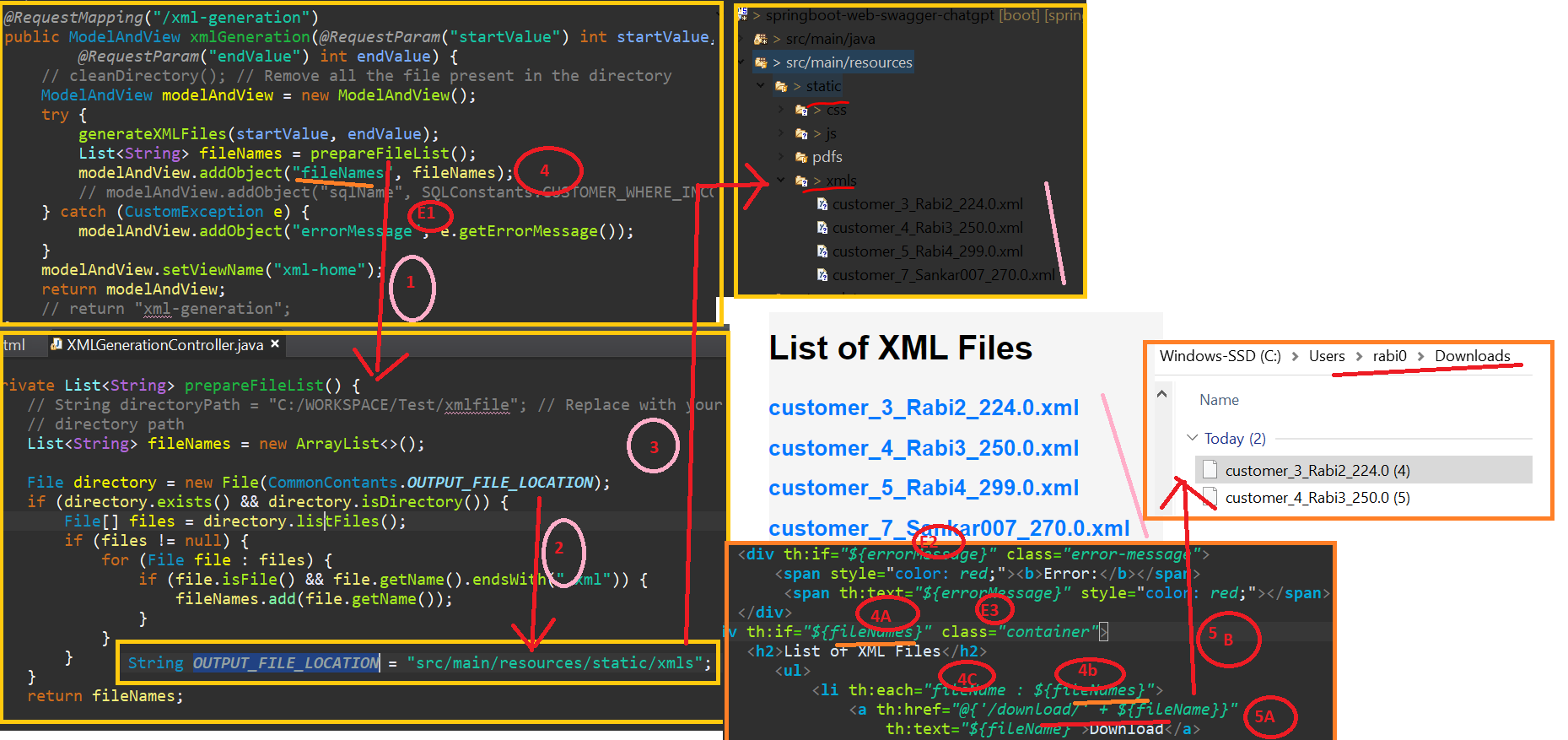
1. XML generation workflow



1. Landing Page and submit :-



1. Show list of files and Download the XML files and Download
2. Xml file generation – generateXMLFiles()
3. Show the list of #1 – prepareFileList()



6. Error Scenario -No Data

