

National Informatics Centre, Agartala.



SUMMER TRAINING – 2023

Task: Parsing of HL7 message into readable format and generation of laboratory report (PDF format).

SUBMITTED BY:

AKASH CHOWDHURY.

GROUP: 1A

Date of Submission: 02/07/2023

GUIDED BY:

Nilkamal Dey Purkayastha.

Abhijit Bhattacharjee.

Atanu Majumder.

CONTENTS

Ex.No	Particular	Page No	
1	Problem Statement	3	
2	Task Description	4-7	
3	Input Segment	8	
4	Output Segment	9-10	
5	Conclusion	11	

Problem Statement

The problem is to parse HL7 messages into a readable format and generate laboratory reports in PDF. HL7 is a standard protocol used for exchanging healthcare data. The challenge is to extract relevant information from complex HL7 messages, including patient demographics, test results, diagnoses, and treatments. The goal is to develop a solution that parses HL7 messages into a human-readable format and generates well-formatted PDF reports. The solution should be robust, efficient, and handle a large volume of messages. It should also accommodate different HL7 versions and adhere to privacy regulations for patient data security. Overall, the objective is to improve clinical workflows by simplifying the parsing and report generation processes.

Task Description

Task Description: Parsing HL7 Message into Readable Format and Generating Laboratory Report (PDF Format).

Objective:

The objective of this task is to develop a functionality that parses an HL7 message and converts it into a readable format. Additionally, the task involves generating a laboratory report in PDF format based on the parsed HL7 message.

Code Overview:

The provided code is implemented in Java using the Spring Framework. It consists of a controller class named **OruMessageController** that handles HTTP POST requests for generating the PDF report. The controller method **generatePdf** takes the HL7 message as input and returns a **ResponseEntity** containing the generated PDF file.

Technology Stack:

The solution is implemented using the following technologies:

Java: The core programming language used for writing the code.

Springboot Framework: The Springboot framework is utilized for building the web application and handling HTTP requests.

HAPI (HL7 API): The HAPI library version 2 is employed for parsing HL7 messages and extracting relevant information.

iText: The iText library is used for generating the PDF report and applying formatting to the document.

The combination of Java, Spring Framework, HAPI, and iText provides a robust and efficient solution for parsing HL7 messages, transforming them into a readable format, and generating high-quality PDF reports.

Parsing the HL7 Message:

To parse the HL7 message, the code utilizes the HAPI (HL7 API) library, which provides a set of tools for working with HL7 messages. The HAPI context is created using the **DefaultHapiContext** class, and the parser is obtained from the context. The ORU (Observation Result) message is then extracted from the input string using the parser.

In case of any parsing errors, the code catches the **HL7Exception** and returns an appropriate error response. Otherwise, it proceeds with further processing of the message.

The Terser utility from HAPI is used to extract specific segments and fields from the parsed HL7 message. It allows accessing and manipulating individual fields using a path-like syntax. In the code, Terser is used to extract information such as the sending application, patient identification details, OBR (Observation Request) segment information, and OBS (Observation) segment details.

Generating the PDF Report:

For generating the PDF report, the code utilizes the **iText** library, which is a popular Java library for creating and manipulating PDF documents. The code creates a new Document instance with A4 page size and opens it for writing. It also sets up the necessary fonts and formatting for the PDF document.

The parsed HL7 message data, including details such as the patient's ID, name, date of birth, gender, address, contact details, description of the observation, collection time, result time, result status, composition description, observation value, observation unit, reference range, and conclusion, are then added to the PDF document in a structured format using the Paragraph class.

After adding all the required information to the PDF document, it is closed and saved to a file using the **PdfWriter** class. The generated PDF file is then read as a byte array using the **Files.readAllBytes()** method.

Returning the PDF Report:

To return the PDF report as the response, the code constructs a new **ResponseEntity** object and sets the byte array of the PDF content as the response body. It also sets appropriate HTTP headers, including the content type as **MediaType.APPLICATION_PDF** and the content disposition as an attachment named **"result.pdf"**.

Error Handling:

The code handles various potential exceptions, such as **IOException**, **DocumentException**, and **HL7Exception**, by catching them and returning an appropriate error response with an HTTP status code of 500 (Internal Server Error). This ensures that any unexpected issues during PDF generation or HL7 processing are properly handled and reported to the client.

Dependencies:

The project includes several dependencies managed by Maven. Noteworthy dependencies are:

Spring Boot Starter Data JPA: Provides support for integrating JPA (Java Persistence API) with Spring Data repositories.

Spring Boot Starter Web: Enables the development of web applications using Spring Web MVC.

HAPI Base and Structures for HL7 v2.3: HAPI libraries required for parsing HL7 messages and accessing the HL7 message structure.

iTextPDF: The iText library for generating PDF documents and manipulating their content.

Spring Boot DevTools: Offers development-time features for increased productivity, such as automatic application restarts.

Spring Boot Starter Test: Provides support for testing Spring Boot applications. Overall, the provided code demonstrates an implementation for parsing HL7 messages, extracting relevant information, generating a structured laboratory report in PDF format, and returning it as a response to the client.

ADVANTAGES:

Enhanced Data Accessibility: By converting HL7 messages into PDF format, this project improves the accessibility and readability of laboratory reports. PDFs are widely supported and can be easily viewed and shared across different devices and platforms.

Standardized Reporting: The project ensures standardized laboratory reporting by extracting specific segments and fields from HL7 messages. This allows for consistent formatting and presentation of information, making it easier for healthcare professionals to interpret and analyze the data.

Efficient Workflow: The REST API implementation enables seamless integration with existing healthcare systems. This streamlines the process of generating laboratory reports, reducing manual effort and improving overall workflow efficiency.

Flexibility and Customization: The project provides flexibility in generating PDF reports by allowing customization of fonts, colors, and formatting options. This enables healthcare organizations to align the reports with their branding guidelines and specific requirements.

Scalability and Performance: Leveraging the Spring Framework and industry-standard libraries, the project ensures scalability and high performance. It can handle a large volume of HL7 messages and generate PDF reports efficiently, catering to the needs of healthcare facilities of varying sizes.

Technology Compatibility: The project utilizes well-established technologies such as Spring Boot, HAPI, and iTextPDF, which have a strong community support and are widely used in the healthcare industry. This ensures compatibility with existing infrastructure and facilitates future enhancements and integrations.

INPUT SEGMENT

Input String (HL7 Message):

```
MSH|^~\&|SendingApp|SendingFacility|ReceivingApp|ReceivingFacility|20230620120000||
ORU^R01|MSG001|P|2.3
```

PID|1|54321|12345^^^HospitalID||Akash^Chowdhury||20010512|M||Agartala^799002| |(555)555-5555||S||123456789|98765432

OBR 1 1234567^^^HospitalID 9876543^^^LabID CBC^Complete	Blood
Count R 20230620110000 PhysicianID^Doe^John^^Dr.^^^ 20230620120000	

OBX	1	NM	WBC^White Blood Cell Count^LN	6.7	10.3/uL	4.5-11.0	N		F
-----	---	----	-------------------------------	-----	---------	----------	---	--	---

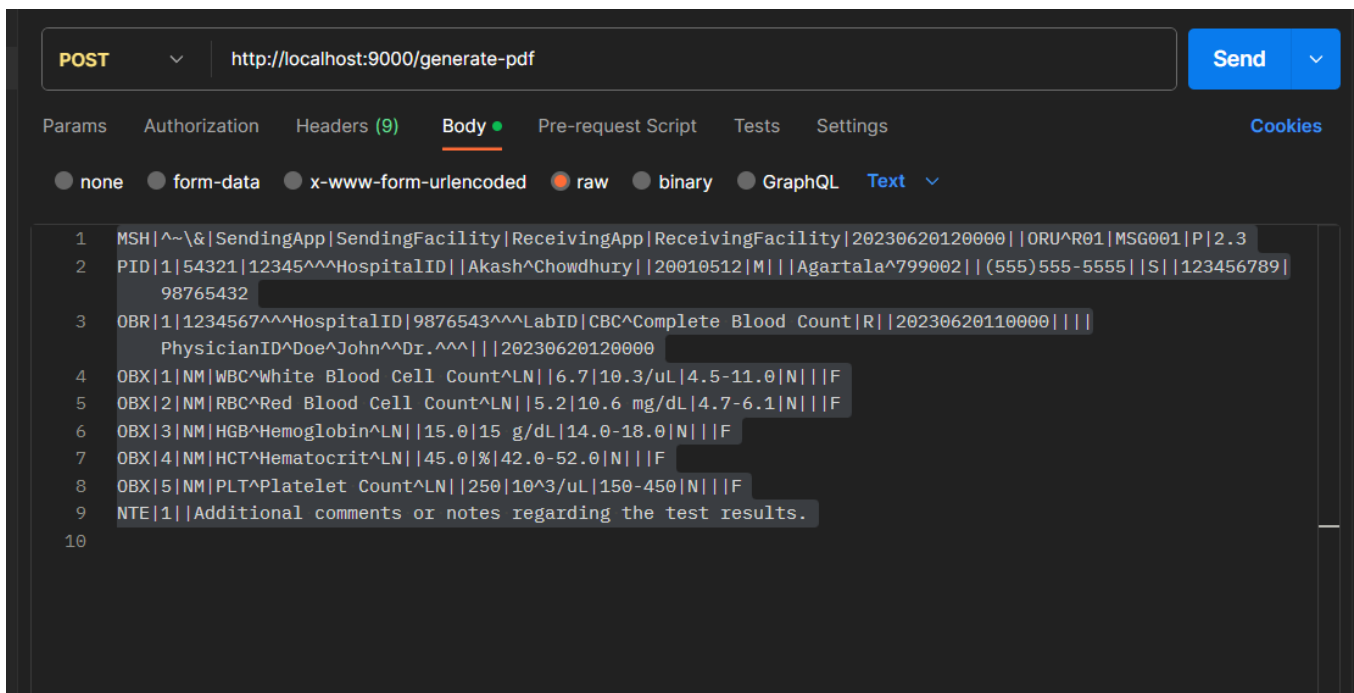
OBX 2 NM RBC^Red Blood Cell Count^LN	5.2	10.6 mg/dL	4.7-6.1	N		F
--------------------------------------	-----	------------	---------	---	--	---

OBX 3 NM HGB^Hemoglobin^LN 15.0 15 g/dL 14.0-18.0 N F
--

OBX		4		NM		HCT^Hematocrit^LN			45.0		%		42.0-52.0		N			F
-----	--	---	--	----	--	-------------------	--	--	------	--	---	--	-----------	--	---	--	--	---



OBX	5	NM	PLT^Platelet Count^LN	250	10^3/uL	150-450	N		F
-----	---	----	-----------------------	-----	---------	---------	---	--	---

NTE | 1 | Additional comments or notes regarding the test results.






OUTPUT SEGMENT

//Report.pdf (path: D:\Downloads\report.pdf)

POST  **http://localhost:9000/generate-pdf** **Send** 

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings Cookies

Body Cookies Headers (6) Test Results  200 OK 45 ms 1.77 KB  Save as Example 

Agartala Government Medical College Hospital

Government of Tripura, Agartala, Tripura West ,799006

LABORATORY OBSERVATION REPORT

Paitent ID : 54321

Patient Name: AKASH CHOWDHURY

DOB : 12/05/2001

Gender : M

Address: Agartala,799002

Contact: (555)555-5555

Description: Complete Blood Count

Hospital Id : 9876543

Lab Id : 1234567

Collect Time : 11:00, 20/06/2023

Result Time : 12:00, 20/06/2023

Composition_Description: White Blood Cell Count

Observation_Value : 6.7

Observation_Unit : 10.3/uL

Reference_Range : 4.5-11.0

Conclusion : N

Composition_Description: Red Blood Cell Count

Observation_Value : 5.2

Observation_Unit : 10.6 mg/dL

Reference_Range : 4.7-6.1

Conclusion : N

Agartala Government Medical College Hospital

Government of Tripura, Agartala, Tripura West ,799006

LABORATORY OBSERVATION REPORT

Patient ID : 54321

Patient Name: AKASH CHOWDHURY

DOB : 12/05/2001

Gender : M

Address: Agartala,799002

Contact: (555)555-5555

Description: Complete Blood Count

Hospital Id : 9876543

Lab Id : 1234567

Collect Time : 11:00, 20/06/2023

Result Time : 12:00, 20/06/2023

Composition_Description: White Blood Cell Count

Observation_Value : 6.7

Observation_Unit : 10.3/uL

Reference_Range : 4.5-11.0

Conclusion : N

Composition_Description: Red Blood Cell Count

Observation_Value : 5.2

Observation_Unit : 10.6 mg/dL

Reference_Range : 4.7-6.1

Conclusion : N

Composition_Description: Hemoglobin

Observation_Value : 15.0

Observation_Unit : 15 g/dL

Reference_Range : 14.0-18.0

Conclusion : N

#This is computer generated report, here no signature required. It is verified by Hospital admin

CONCLUSION

In conclusion, this project successfully addresses the need for converting HL7 messages to PDF format, enabling healthcare professionals to generate readable laboratory reports. By leveraging the Spring Framework, HAPI library for HL7 parsing, and **iTextPDF** for PDF generation, a robust REST API was implemented. The project's methodology involved parsing HL7 messages, extracting relevant information, and generating well-formatted PDF reports. With a carefully chosen technology stack and thorough dependency management using **Maven**, the solution ensures efficiency, reliability, and scalability. This project serves as an effective tool for enhancing healthcare workflows and facilitating the seamless conversion of HL7 messages into easily accessible PDF documents.

THANKYOU