**Project Structure**

    my-spring-boot-app

    |-- src

    |   |-- main

    |       |-- java

    |           |-- com.example

    |               |-- MySpringBootApplication.java

    |               |-- model

    |                   |-- PersonalInfo.java

    |                   |-- ProfessionalInfo.java

    |               |-- repository

    |                   |-- PersonalInfoRepository.java

    |                   |-- ProfessionalInfoRepository.java

    |               |-- service

    |                   |-- InfoService.java

    |                   |-- ReportGenerationService.java

    |                   |-- PDFExporter.java

    |               |-- controller

    |                   |-- InfoController.java

    |                   |-- ReportGenerationController.java

    |-- src

    |   |-- main

    |       |-- resources

    |           |-- application.properties

    |-- pom.xml

1. **MySpringBootApplication.java:**
   * Main class with the main method to run the Spring Boot application.
2. **model Package:**
   * Contains entity classes representing personal and professional information.
3. **repository Package:**
   * Spring Data JPA repositories for personal and professional information.
4. **service Package:**
   * InfoService: Service interface for retrieving information.
   * ReportGenerationService: Service interface for generating PDF reports.
   * PDFExporter: Component responsible for PDF generation using Apache PDFBox.
5. **controller Package:**
   * InfoController: Controller for handling requests related to information retrieval.
   * ReportGenerationController: Controller for handling requests related to PDF generation.
6. **resources Directory:**
   * application.properties: Configuration properties file.
7. **pom.xml:**
   * Maven Project Object Model file specifying project dependencies, plugins, and configurations.

The PDFExporter component in the service package is responsible for generating PDF reports using Apache PDFBox. The controllers (**InfoController** and **ReportGenerationController**) handle HTTP requests related to information retrieval and PDF generation, respectively.

Ensure to add the Apache PDFBox dependency to your pom.xml:

<dependency>

    <groupId>org.apache.pdfbox</groupId>

    <artifactId>pdfbox</artifactId>

    <version>2.0.31</version> <!-- Use the latest version available -->

</dependency>

**Detailed Flow for PDF Generation (using PDFBox):**

1. **Client Sends Request:**
   * The client sends an HTTP request to the ReportGenerationController to initiate the PDF generation process.
2. **ReportGenerationController:**
   * The ReportGenerationController receives the request and delegates the request to the ReportGenerationService.
3. **ReportGenerationService:**
   * The ReportGenerationService is responsible for orchestrating the PDF generation process.
   * It interacts with the InfoService to retrieve necessary data (personal and professional information).
4. **InfoService:**
   * The InfoService communicates with the InfoRepository to fetch personal and professional information from the database.
   * It returns the retrieved data to the ReportGenerationService.
5. **ReportGenerationService (contd.):**
   * The ReportGenerationService has obtained the required data.
   * It invokes the PDFExporter component to generate the PDF report.
6. **PDFExporter:**
   * The PDFExporter is a specialized component responsible for PDF generation using Apache PDFBox.
   * It creates a PDF document, adds content, and saves it to a byte array.
7. **PDFExporter (contd.):**
   * The PDFExporter utilizes Apache PDFBox to create a PDF document.
   * It adds content to the document based on the retrieved data from the InfoService.
   * The PDF content is then saved to a byte array.
8. **ReportGenerationService (contd.):**
   * The ReportGenerationService receives the generated PDF content from the PDFExporter.
9. **ReportGenerationController (contd.):**
   * The ReportGenerationController receives the generated PDF content and prepares the HTTP response.
10. **HTTP Response:**
    * The ReportGenerationController sends an HTTP response to the client.
    * The response includes the generated PDF content as a downloadable file.
11. **Client Receives PDF:**
    * The client receives the HTTP response with the PDF file.
    * The client can download and view the PDF document.

**Notes:**

* **PDF Exporter (using PDFBox):**
  + Apache PDFBox is a Java library for working with PDF documents.
  + In the PDFExporter component, you can use PDFBox to create a PDDocument, add content to it, and convert it to a byte array.
  + Ensure to handle exceptions and cleanup resources properly.
* **PDF Content Structure:**
  + The structure of the PDF content, such as headers, tables, or other elements, is defined within the PDFExporter component based on your requirements.