NIC TRAINING PROGRAM

Name : Arijit Das
Enrollment no. : 20UCS007
College : NIT Agartala
Group : 2A

Ehospital LIS Image OCR API

The objective was to create a microservice to extract text data from a lab report file in such a format that data becomes easily accessible to the doctors in the hospital.

We have developed an **API** that allows users to easily extract text from image files by performing **Optical Character Recognition** (OCR). This API is implemented using the Spring Boot framework and leverages the powerful **Tesseract OCR engine**.

With our API, users can upload an image file containing text, such as scanned **documents, images with embedded text**, or any other **readable image**. The API then processes the image and returns the extracted **text** or a **JSON** representation of the text, depending on the user's preference.

By providing this API, we offer a convenient and efficient way for developers and users to integrate OCR functionality into their applications or services. It eliminates the need for manual text extraction and allows for automation of tasks that require extracting text from images.

Brief description of the code we have developed :

1. The code imports necessary dependencies from the `net.sourceforge.tess4j` library, which is used for performing OCR using Tesseract, and other Spring Boot dependencies.

2. The `OCRcontroller` class is annotated with `@Controller`, indicating that it is a controller component in the Spring MVC framework.
3. The controller has two `@PostMapping` methods, `jsonOcrApi` and `imageOcrApi`, which handle the requests to `/image-to-json` and `/image-to-text`, respectively.
4. Both methods accept a `MultipartFile` parameter named `image`, which represents the uploaded image file.
5. Inside each method, the first check is performed to ensure that the image file is not empty. If the image is empty, a response with an error message is returned.
6. If the image is not empty, the code proceeds to extract the text from the image using the `imageToText` method.
7. The extracted text is then either converted to JSON format using the `textToJson` method (in the case of `jsonOcrApi`) or returned as plain text (in the case of `imageOcrApi`).
8. The resulting text or JSON is converted to a byte array and wrapped in an `InputStreamResource` for the response.
9. The appropriate HTTP headers and content type are set based on the API endpoint (`application/json` for `/image-to-json` and `text/plain` for `/image-to-text`).
10. The response entity with the processed image text is returned.

We have utilized various **Technologies and Frameworks** to build this API. The core components include **Java, Spring Boot, and Tesseract OCR**. Java provides the foundation for the code, while Spring Boot simplifies the development of web applications. Tesseract OCR is a robust and widely-used library for extracting text from images.

Additionally, the code uses the **Spring MVC framework**, which follows the Model-View-Controller architectural pattern. This ensures a clean separation of concerns and allows for efficient handling of HTTP requests and responses. We have also incorporated JSON processing capabilities using the `org.json.JSONObject` class. This enables easy conversion of the extracted text into a JSON format, providing a structured representation of the extracted data.

Throughout the development process, we have employed Maven as a build automation and dependency management tool. Maven helps ensure smooth project setup and handles the management of project dependencies.

Our API allows users to effortlessly extract text from image files using OCR. **It provides a user-friendly and efficient solution for applications that require text extraction from various types of images.** By leveraging technologies like Java, Spring Boot, Tesseract OCR, and JSON processing, **we have built a powerful tool that simplifies the integration of OCR functionality into other applications or services.**

Libraries and Tools Used

The following libraries and tools are used in this API:

- Language : Java (JDK 17)
- Framework: SpringBoot (3.0.8)
- IDE: IntelliJ

Dependencies

Maven Dependencies used:

- **tess4j**: The library provides optical character recognition (OCR) support for:TIFF, JPEG, GIF, PNG, and BM(provided in this project)
- **org.json**: JSON is a light-weight, language independent, data interchange format
- **spring-boot-starter-web**: Starter for building web, including RESTful, applications using Spring MVC. Uses Tomcat as the default embedded container