

Integración de OCR con el backend

Propósito y alcance

Este documento explica cómo la API de backend de Spring Boot se integra con el servicio externo de OCR de Python para procesar imágenes de recibos y tickets. Esto incluye la implementación de la capa de servicio, la configuración de los endpoints REST, los patrones de flujo de datos y los mecanismos de gestión de errores que permiten la comunicación del backend con el servidor de OCR de Python.

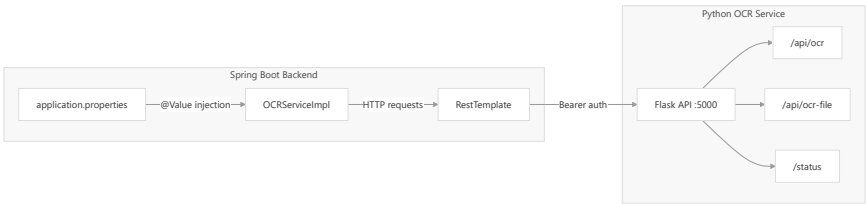
Para obtener información sobre la implementación del servidor OCR de Python, consulte [Servidor OCR de Python](#) . Para obtener más información sobre los componentes frontend que utilizan estas API, consulte [Interfaz de gestión de gastos](#) .

Configuración y configuración de la conexión

El backend se conecta al servicio OCR de Python mediante las propiedades de configuración definidas en [Nombre del servicio] `application.properties` . El sistema utiliza comunicación basada en HTTP con autenticación por token de portador.

Configuración de la conexión

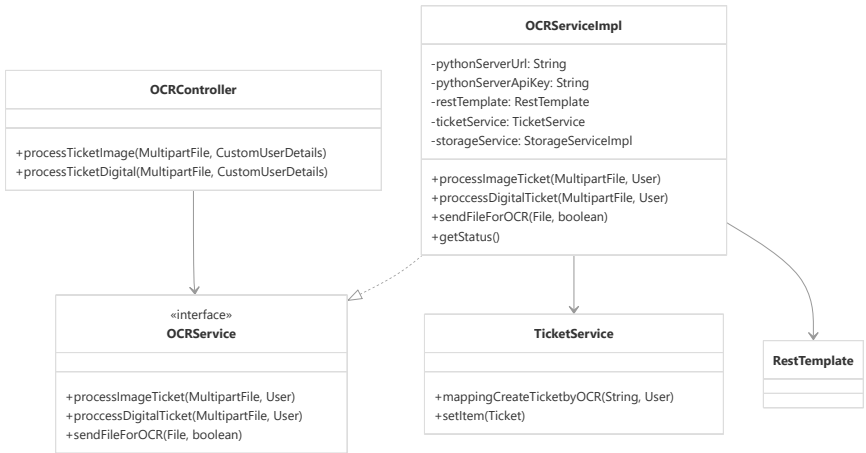
| Propiedad de configuración | Valor predeterminado | Objetivo |
|-----------------------------------|------------------------------------|---|
| <code>python.server.url</code> | <code>http://localhost:5000</code> | URL base para el servicio OCR de Python |
| <code>python.server.apiKey</code> | <code>CREAR_API_KEY</code> | Token portador para autenticación |



Arquitectura de servicios

La integración de OCR se implementa a través de una arquitectura en capas con una clara separación de preocupaciones entre los controladores REST, la capa de servicio y la comunicación externa.

Componentes principales



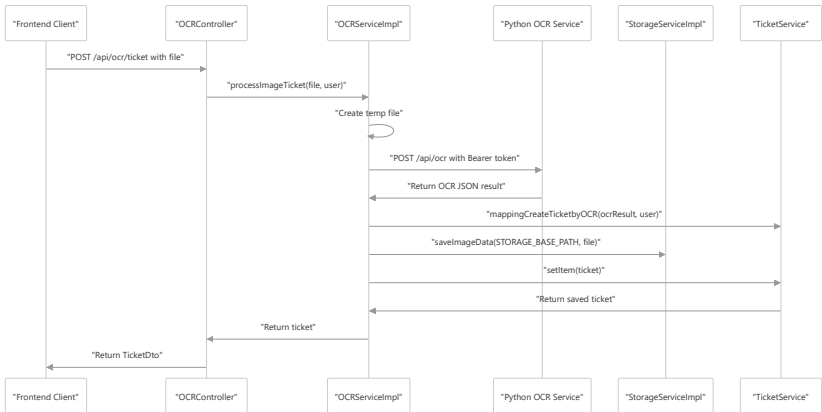
Integración de API REST

El backend expone dos puntos finales de OCR principales que aceptan cargas de archivos multiparte y delegan el procesamiento al servicio Python.

Puntos finales de OCR

| Punto final | Método | Objetivo | Tipo de contenido |
|------------------------|--------|--|---------------------|
| /api/ocr/ticket | CORREO | Procesar imágenes de tickets | multipart/form-data |
| /api/ocr/ticketdigital | CORREO | Procesar archivos de tickets digitales | multipart/form-data |

Patrón de flujo de solicitud



Flujo de procesamiento de datos

La integración de OCR gestiona dos flujos de trabajo de procesamiento distintos: tickets basados en imágenes y tickets de archivos digitales. Ambos flujos de trabajo siguen patrones similares, pero utilizan diferentes puntos de conexión de Python.

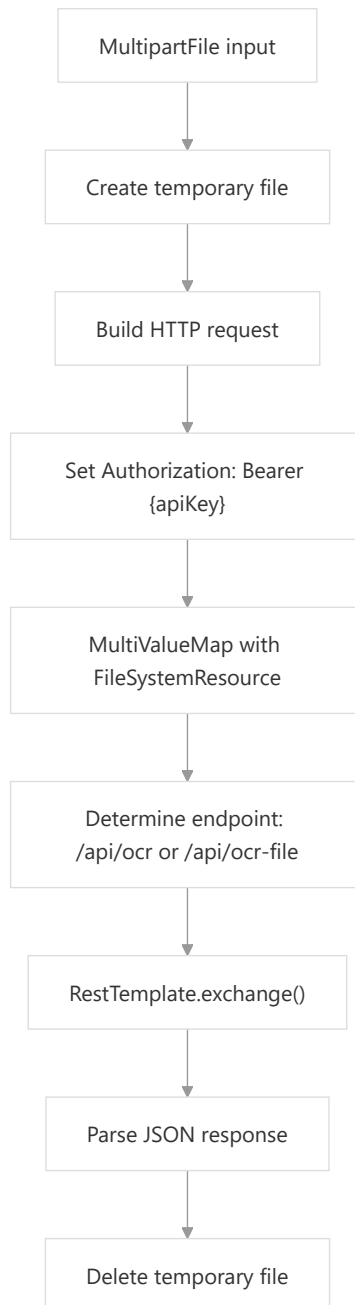
Flujo de trabajo de procesamiento de imágenes

El `processImageTicket` método `OCRServiceImpl` orquesta todo el proceso de procesamiento:

| | | |
|---------------------------------------|--|-------|
| 1. Creación de archivos temporales : | <code>ServidorServicioAPI/GestorAPI/src/main/java/Proyecto/GestorAPI/servicesimpl/OCRServiceImpl.java</code> | 65-66 |
| 2. Procesamiento de OCR : | <code>ServidorServicioAPI/GestorAPI/src/main/java/Proyecto/GestorAPI/servicesimpl/OCRServiceImpl.java</code> | 70 |
| 3. Mapeo de resultados : | <code>ServidorServicioAPI/GestorAPI/src/main/java/Proyecto/GestorAPI/servicesimpl/OCRServiceImpl.java</code> | 74 |
| 4. Almacenamiento de imágenes : | <code>ServidorServicioAPI/GestorAPI/src/main/java/Proyecto/GestorAPI/servicesimpl/OCRServiceImpl.java</code> | 78 |
| 5. Persistencia de la base de datos : | <code>ServidorServicioAPI/GestorAPI/src/main/java/Proyecto/GestorAPI/servicesimpl/OCRServiceImpl.java</code> | 82 |

Detalles de la comunicación HTTP

El `sendFileForOCR` método maneja la comunicación HTTP de bajo nivel con el servicio Python:



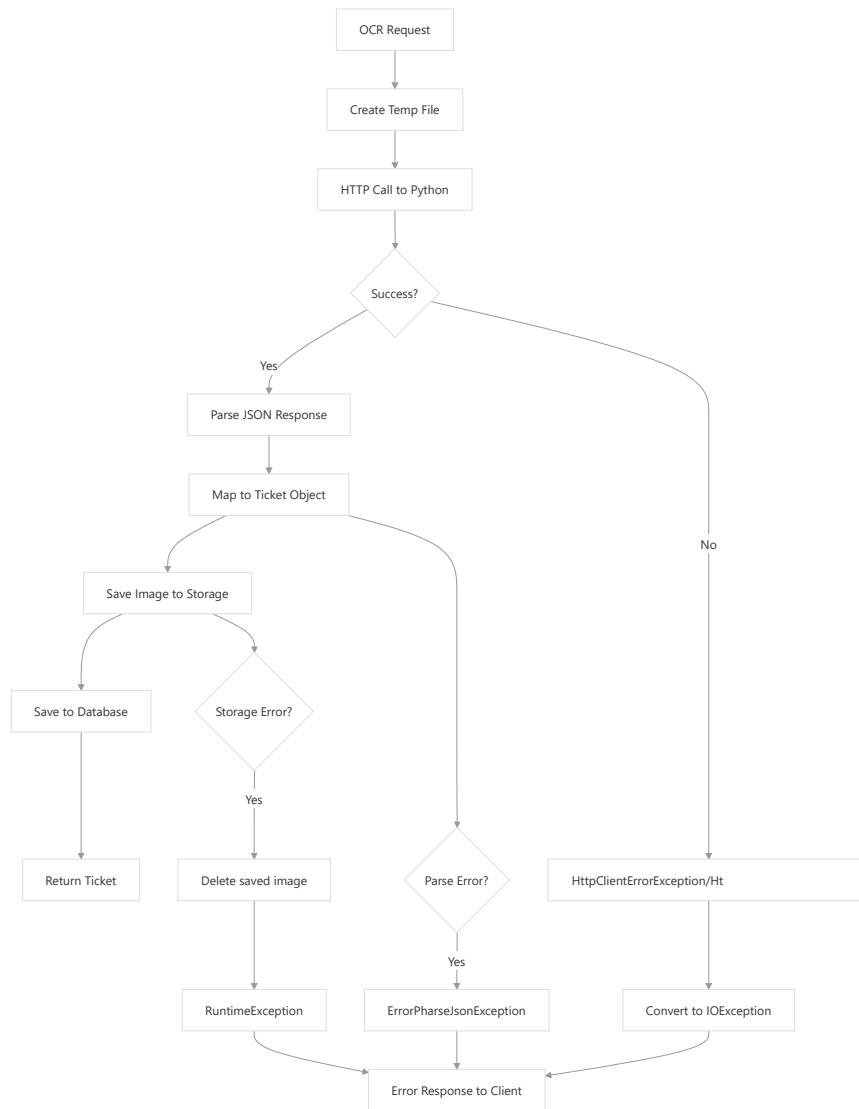
Mapeo de resultados de OCR

| Campo OCR | Propiedad del boleto | Valor predeterminado | Validación |
|-----------------|----------------------|--------------------------|-----------------------|
| establecimiento | store , name | "Desconocido" , "Ticket" | Comprobación de nulo |
| total | total | 0.0 | Manejo de excepciones |
| iva | iva | 0.0 | Manejo de excepciones |
| articulos | productsJSON | "[]" | serialización JSON |
| fecha , hora | expenseDate | Fecha/hora actual | Análisis de fechas |

Manejo y recuperación de errores

La integración implementa un manejo integral de errores en múltiples niveles para garantizar la confiabilidad del sistema.

Estrategia de manejo de excepciones



Tipos de respuesta de error

| Tipo de error | Estado HTTP | Patrón de mensaje de respuesta |
|-------------------------------|--------------------------------|--|
| Archivo vacío | 400 Solicitud incorrecta | "El archivo no puede estar vacío" |
| Error de procesamiento de OCR | 500 Error interno del servidor | "Error en el procesamiento OCR: {message}" |
| Error de análisis de JSON | 500 Error interno del servidor | A través de <code>ErrorParseJsonException</code> |
| Error de comunicación HTTP | 500 Error interno del servidor | "Error al procesar OCR. Código: {status} - {body}" |

Integración de comprobación de salud

El sistema incluye un mecanismo de verificación de estado para supervisar el estado del servicio OCR de Python a través del `getStatus` método.

Monitoreo de estado

El `StatusServerResponse` objeto rastrea múltiples estados de servicio:



