

# API Legacy

En el contexto de la migración del proyecto SICOF, se ha identificado la necesidad de desarrollar una API Legacy que permita la comunicación entre las aplicaciones legacy de la empresa (SICOF, BPP y Rentas) con los nuevos servicios y microfrontend. Esta API actuará como un puente entre el pasado y el futuro, facilitando la integración de datos y procesos entre los diferentes entornos tecnológicos.

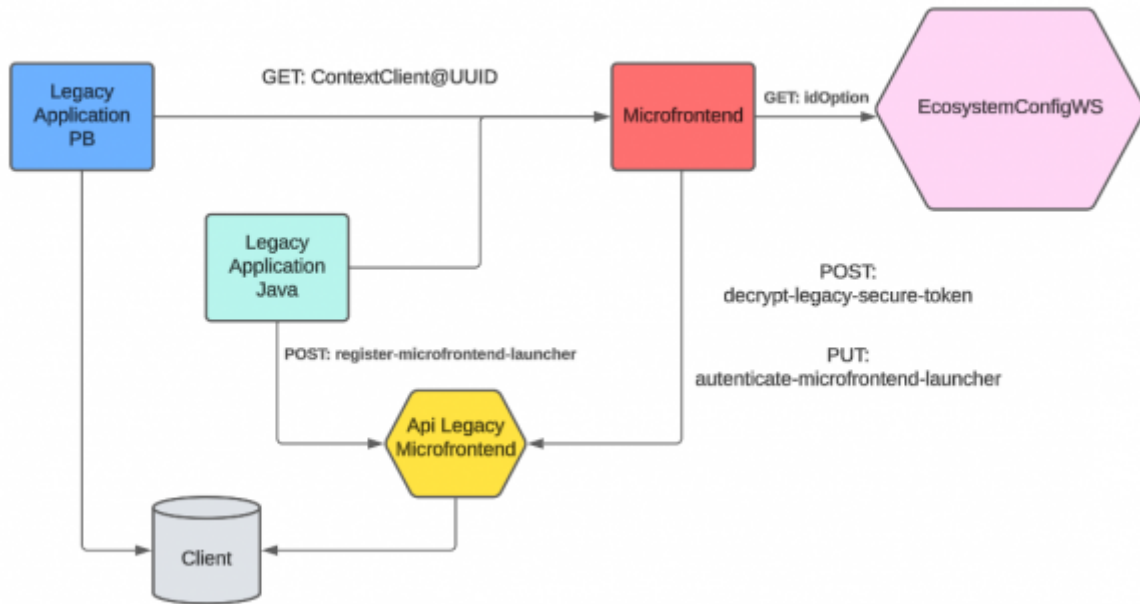
## Objetivos

Los objetivos principales del desarrollo de la API Legacy son:

- **Facilitar la migración de datos:** La API proporcionará un mecanismo estandarizado para la transferencia de datos de sesión entre las aplicaciones legacy y los nuevos sistemas microfrontend, minimizando la necesidad de reescritura de código y simplificando el proceso de migración.
- **Exponer funcionalidades legacy:** La API permitirá exponer las funcionalidades clave de las aplicaciones legacy a los nuevos sistemas, permitiendo su reutilización y evitando la necesidad de replicarlas desde cero.
- **Promover la integración:** La API servirá como base para la integración de las aplicaciones legacy con los nuevos sistemas, permitiendo una comunicación fluida y el intercambio de información en tiempo real.

## Arquitectura

La API Legacy se basará en una arquitectura RESTful, siguiendo los principios de diseño de APIs REST. Esto permitirá una fácil integración con las aplicaciones legacy y los nuevos sistemas, utilizando protocolos HTTP estándar y formatos de datos como JSON.



A continuación se describe el modelo utilizado de forma general:

- **LegacyApplicationPB:** Son las aplicaciones desarrolladas en Powerbuilder las cuales integran un api propia en el core que genera la información de integración posteriormente consume la url del microfrontend por medio de un método GET.
- **LegacyApplicationJava:** Son las aplicaciones desarrolladas en Java pero a diferencia de las anteriores si hace un registro previo de la información de integración por medio del api-legacy y por último consume la url del microfrontend por medio de un método GET.
- **Api Legacy Microfrontend:** Es la encargada de registrar la información de integración, encriptar y desencriptar tokens y autorizar los consumos de las aplicaciones legacy.
- **Microfrontend:** Es la funcionalidad front migrada a las nuevas tecnologías utiliza el api-legacy para autentificar y autorizar las peticiones realizadas por las aplicaciones legacy.
- **EcosystemConfigWS:** Entre sus configuraciones provee un metodo que devuelve las rutas que utilizará el microfrontend.
- **Client:** Origen de datos del cliente (DB).

## Componentes

La API Legacy estará compuesta por los siguientes componentes:

### Capa de acceso a datos

Esta capa encapsulará el acceso a las bases de datos de las aplicaciones legacy, proporcionando una interfaz única y simplificada para la consulta y manipulación de datos.

Se define el siguiente modelo.

```
/**
```

```
* The Class FrontendConsumingUrlDto.
*/
@Schema(description = "Entidad que almacena los registros de tokens heredados para consumo microfrontend.")
@Data
public class FrontendConsumingUrlDto implements Serializable{

    /** The Constant serialVersionUID. */
    private static final long serialVersionUID = 1L;

    /** The id. */
    @Schema(description = "Identificador del registro.")
    private Long id;

    /** The microfrontend url. */
    @Schema(description = "url del microfrontend desplegado.")
    private String microfrontendUrl;

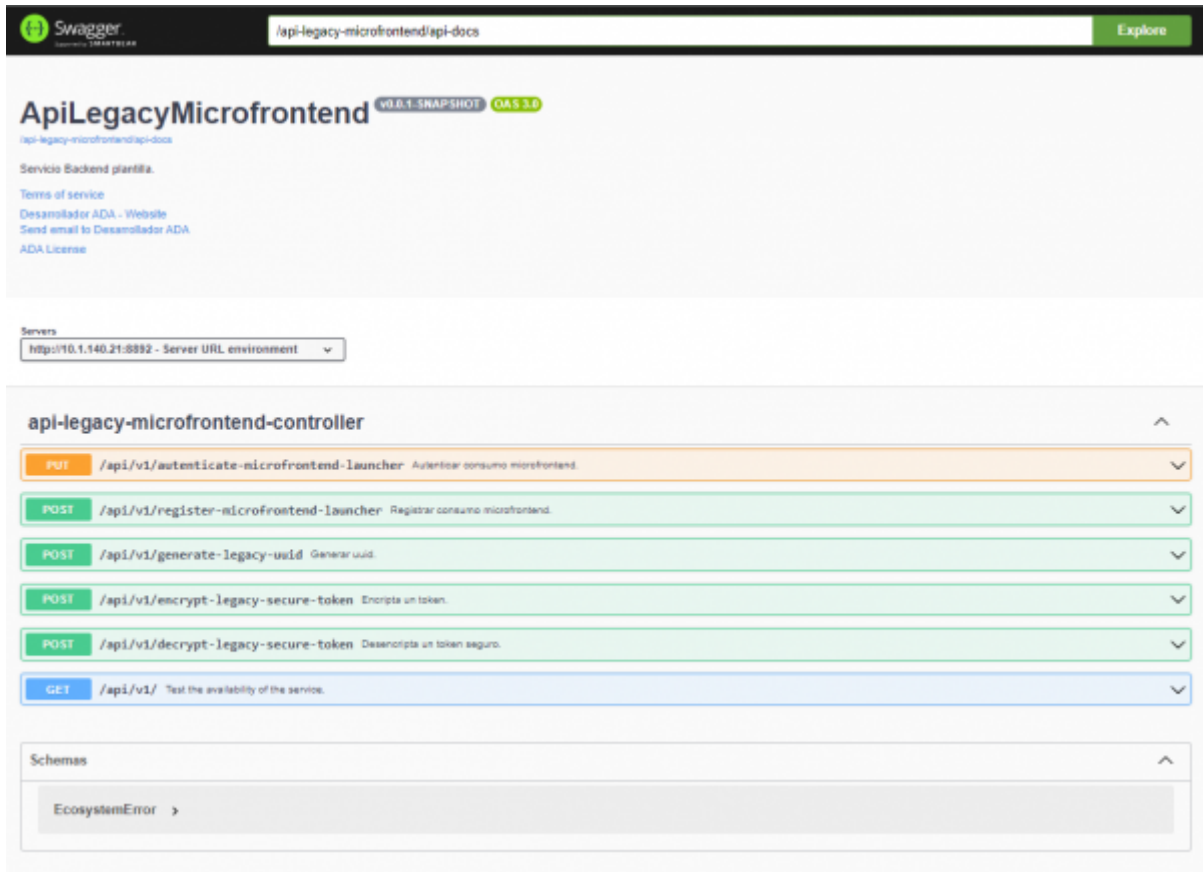
    /** The path param. */
    @Schema(description = "Parámetro que contiene la cadena de la sesión encriptada que será utilizada en la inicialización del microfrontend.")
    private String pathParam;

    /** The consume status. */
    @Schema(description = "Estado del registro. Es utilizado para validar si el token heredado está vigente.")
    private String consumeStatus;

    /** The uuid. */
    @Schema(description = "Identificador del token heredado.")
    private String uuid;
}
```

## Capa de lógica de negocio

Esta capa implementará la lógica de negocio específica de las aplicaciones legacy, exponiendo las funcionalidades de estas aplicaciones a través de la API.



La documentación del api se expone en los siguientes endpoint.

- [Documentación OpenAPI](#): JSON de documentación del servicio.
- [Documentación Swagger](#): Visor Swagger de documentación del servicio.

## Seguridad

La seguridad de la API Legacy será una prioridad absoluta. Se implementarán medidas de seguridad como autenticación, autorización, cifrado y validación de datos para proteger la información confidencial y garantizar la integridad de las transacciones.

## Integración del API Legacy: Escenarios y Tipos

El API Legacy se integrará con las aplicaciones legacy y los nuevos sistemas de diversas maneras, abriendo un abanico de posibilidades para la migración y el desarrollo futuro. A continuación, se describen algunos de los escenarios de integración más comunes:

- [Integración Microfrontend](#)

[←Regresar](#)

From:  
<http://wiki.adacsc.co/> - Wiki

Permanent link:  
<http://wiki.adacsc.co/doku.php?id=ada:howto:sicoferp:factory:new-migracion-sicoferp:apilegacy&rev=1718294492>

Last update: **2024/06/13 16:01**

