

Fábrica - Servicios Web - Integraciones - SicofWS

Este servicio expone las siguientes operaciones que afectan Sicof ERP:

Method	Path	Description
Get	/	Verifica el estado del servicio
Post	/facturacion	Generación de facturas (Cuentas por Cobrar)
Post	/recaudo	Generación de recaudos(Comprobantes de Ingresos)
Post	/cambiold	Actualiza la identificación de los tercero
Get	/tercero/search	Realiza la búsqueda de un tercero por medio de su Nit
Post	/tercero/guardarTercero	Crea la creación de terceros por medio de un listado de estudiantes
Get	/factura/search	Realiza la búsqueda de facturas por medio de su referencia
Get	/plantilla/search	Realiza la búsqueda de plantillas por medio de su código

Descripción del Servicio

Nombre	Descripción	Tipo Servicio	Tecnología	Lenguaje	Tipo Desarrollo	Versión Desarrollo	Archivo de Configuración
SicofWS	Servicio que provee las funcionalidades soportadas en la app mobile de Sicof ERP.	Rest	SpringBoot 2.5.5	Java 11.0.315	Migrado de JSP	11.0.1.0	application.properties
Tipo Empaquetado	Gestor de Dependencias	Servidor de Despliegue	IDE Proyecto	Tipo Repositorio	Url Desarrollo	Url Calidad	Url Producción
war	Maven 3.8.1	Tomcat 6	SpringTools 4.6	Subversión	SVN branches	SVN trunk	SVN tags
Dependencias							
groupid	artifactId	version	scope	optional	Descripción		
org.springframework.boot	spring-boot-starter-actuator	N/A	N/A	N/A	API de actualización del servicio		
org.springframework.boot	spring-boot-starter-data-jpa	N/A	N/A	N/A	API de persistencia		
org.springframework.boot	spring-boot-starter-data-rest	N/A	N/A	N/A	API de mapeo de rutas		
org.springframework.boot	spring-boot-starter-web	N/A	N/A	N/A	Core Servicios Web		
org.springframework.boot	spring-boot-devtools	N/A	runtime	true	Recarga de aplicaciones		
org.projectlombok	lombok	N/A	N/A	true	API de anotaciones para reducción de código		
org.springframework.boot	spring-boot-starter-test	N/A	test	N/A	Core Springboot		
org.springframework.boot	spring-boot-starter-data-jdbc	N/A	N/A	N/A	API JDBC para conexión a Base de Datos		
org.apache.tomcat.embed	tomcat-embed-jasper	N/A	provided	N/A	Servidor embebido tomcat		
org.springframework.boot	spring-boot-starter-tomcat	N/A	provided	N/A	Core Spring		
org.apache.maven.plugins	maven-compiler-plugin	3.8.1	N/A	N/A	Gestor de dependnecias		
io.springfox	springfox-swagger2	2.9.2	N/A	N/A	Core Documentación Web Services RESTful		
io.springfox	springfox-swagger-ui	2.9.2	N/A	N/A	GUI Documentación Web Services RESTful		
Operaciones							
Tipo Petición Http	Nombre operación	Descripción				Documentación	
Get	GetServicio	Método orquestador que procesa las peticiones de las funcionalidades soportadas.				La documentación del consumo se debe consultar.	
Ambientes de Despliegue							
Tipo	Url	Observaciones					
Desarrollo:	http://10.1.20.84:3131/SicofMobileServiceADA/api/	Ambiente de desarrollo					
Calidad:		Ambiente de validación					
Producción:		Ambiente de producción					

Información Adicional

Despliegue

Requisitos Técnicos Servidor

- Servidor Wildfly 20 o superior
- Java 11 o OpenJDK 11
- S.O. Linux
- Ram 4GB o superior
- Disco Duro 64 GB
- Publicar la URL y Puerto del WildFly

Pasos Previos: Generar war (IDE Eclipse y Spring Tools)

El proyecto está configurado para generar compilable war. Siga los siguientes pasos para generarlo¹⁾.

- Ubíquese en la raíz del proyecto, click derecho → Run as... → Maven builds...
- En la casilla **Goals** ingrese el comando: **clean install** para generar el war. También puede utilizar en su lugar la instrucción **clean install -DskipTests** si no desea ejecutar las pruebas unitarias.
- Al finalizar el proceso si no hay errores se generará el war en el directorio target del proyecto.

Nota

Para ubicar el war en el directorio del equipo desde el proyecto seleccione el war con click derecho y vaya a propiedades al visualizar la ventana emergente se verá la ubicación del archivo.

Despliegue en el servidor

Actualmente solo hay servidor de desarrollo:

1. Acceder al servidor de despliegue por WSCP o SSH: IP: **10.1.20.84**, Usuario:**wildfly23**, Password:**vwd5ELGR5j7v**
2. Borre el archivo SicofMobileServiceADA.war y espere que el archivo de despliegue quede inválido.
3. Borre el archivo de despliegue SicofMobileServiceADA.war.undeploy
4. Copie el nuevo war en la ruta **/opt/wildfly23/standalone/deployments** se recomienda utilizar WSCP por su interfaz gráfica. Debe esperar mientras se sube el despliegue al servidor el cual terminará correctamente si se crea el archivo SicofMobileServiceADA.war.deployed Si falla este proceso debe ir al paso 2.
5. Por último desde un navegador ingrese la url **<http://10.1.20.84:3131/SicofMobileServiceADA/api/>** y deberá visualizar el mensaje

Online!!! indicando que el despliegue fué realizado correctamente.

Notas

- Debe contar con VPN
- Ejemplo de Consumo Login Multiempresa:
http://10.1.20.84:3131/SicofMobileServiceADA/api/GetServicio?ping=desarrollo&nombreServicio=Login&callback=__ng_jsonp__.__req0.finished&separador=,¶metros=SICOF,1234,null,null,null&codigoMEmpresa=9999999999
- Ejemplo de consumo Empresas:
http://10.1.20.84:3131/SicofMobileServiceADA/api/GetServicio?ping=fosemed&nombreServicio=GetEmpresas&callback=__ng_jsonp__.__req0.finished&separador=,¶metros=

[←Volver atrás](#)

1)

Se asume que el lector tiene conocimientos previos del uso del lenguaje Java e IDE utilizado

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

Permanent link:
<http://wiki.adacsc.co/doku.php?id=ada:howto:sicoferp:factory:integrations:sicofws&rev=1637339106>

Last update: **2021/11/19 16:25**

