

Migración SICOF ERP - Proceso: Guía de Dependencias en Microservicios

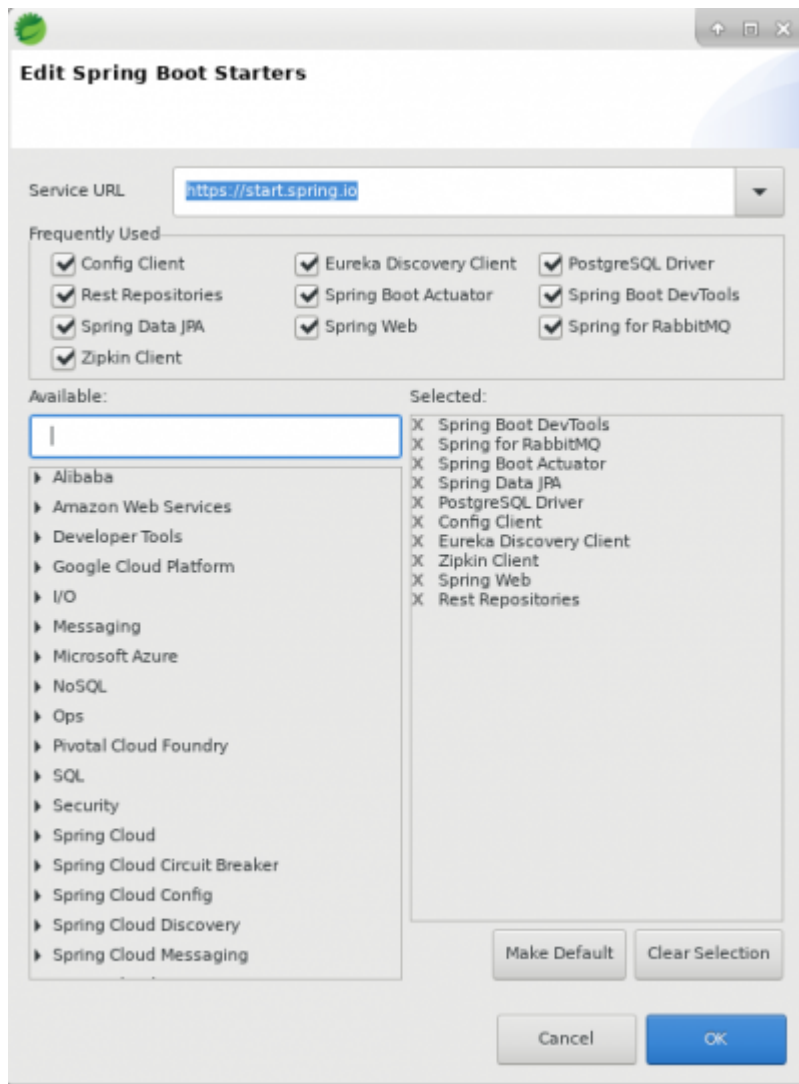
La siguiente sección define las dependencias básicas que se deben tener presente y utilizar en el desarrollo de los microservicios de la fábrica según la necesidad de los mismos.

Consideraciones Previas

- Solo las dependencias básicas son requeridas
- Las dependencias opcionales solo deben ser consideradas si el microservicio aplica para el caso de la dependencia
- Se asume conocimiento previo de la tecnología Maven / Spring Cloud / Netflix OSS
- Si un microservicio requiere configuraciones especiales, estas deben ser validadas con los líderes de desarrollo (Pablo Quintana, Daberson Henao, Carlos Torres, Gersain Castañeda).

Escenario estandar de dependencias

La siguiente imagen define el escenario estandar de dependencias que pueden ser utilizados en un microservicio. Posteriormente se describirá cada una de acuerdo al grupo al que pertenece.



Clasificación de dependencias

Se definen las siguientes clasificaciones de dependencias:

- Dependencias básicas
- Dependencias de traza
- Dependencias de clientes
- Dependencias de datos

Dependencias básicas

Son aquellas dependencias que deben estar presentes en todos los microservicios ya que su uso aplica la arquitectura desea en la fabrica de desarrollo. A continuación se define esas dependencias:

Spring Boot DevTools

Spring Boot DevTools¹⁾ es la herramienta de Spring Boot que nos permite reiniciar de forma

automática nuestras aplicaciones cada vez que se produce un cambio en nuestro código. Útil para la fase de desarrollo y pruebas.

Spring Boot Actuator

Spring Boot Actuator²⁾ es una librería que nos proporciona herramientas de monitorización y administración de nuestra API REST de una manera bastante sencilla.

Spring Cloud Config Client

Spring Cloud Config³⁾ es el enfoque cliente / servidor de Spring para almacenar y servir configuraciones distribuidas en múltiples aplicaciones y entornos.

Este almacén de configuración se versiona idealmente bajo el control de versión de Git y se puede modificar en el tiempo de ejecución de la aplicación.

Spring Cloud Netflix Eureka Discovery Client

Eureka⁴⁾ es un servicio REST, utilizándose principalmente en la nube de AWS, a la cuál está estrechamente ligado. Eureka se comporta como servidor, cuyo objetivo es registrar y localizar microservicios existentes, informar de su localización, su estado y datos relevantes de cada uno de ellos. Además, nos facilita el balanceo de carga y tolerancia a fallos.

La dependencia cliente permite que los microservicios se registren automáticamente al arrancar su instancia.

Spring Web

Contiene los recursos necesarios (conjunto de dependencias) para el desarrollo web⁵⁾.

Dependencias de traza

Son aquellas dependencias que permiten obtener información asociada al microservicio como:

- Estado
- Memoria
- Trazas de rendimiento
- Log de errores

Se define las siguientes dependencias de traza:

Zipkin Client

Zipkin⁶⁾ es un proyecto de código abierto que proporciona mecanismos para enviar, recibir, almacenar y visualizar trazas. Esto nos permite correlacionar la actividad entre servidores y obtener una imagen mucho más clara de lo que está sucediendo exactamente en nuestros servicios.

La dependencia cliente permite integrar el registro en el servidor Zipkin.

Spring for RabbitMQ

RabbitMQ es un servidor AMQP. Es decir acepta y reenvía mensajes.

Dependencias de clientes

Son aquellas dependencias que proveen clientes para el consumo de servicios como Feign o RestController

OpenFeign

Feign⁷⁾ es un cliente java http ligero. Que se al stack Spring Boot para simplificar el consumo de los servicios.

Dependencias de datos

Son aquellas relacionadas a la persistencia de la información o a la gestión y exposición de los datos. Entre ellas están definidas:

- Spring Data JPA
- RestRepository
- PostgreSQL Driver
- MySQL Driver
- Oracle Driver

El uso de estas dependencias dependerá del origen de datos.

[←Volver atrás](#)

1)

<https://www.arquitecturajava.com/spring-boot-devtools-y-recarga-de-aplicaciones/>

2)

<https://openwebinars.net/blog/como-anadir-spring-boot-actuator-tu-proyecto/>

3)

<https://www.baeldung.com/spring-cloud-configuration>

4)

<https://blog.bi-geek.com/arquitecturas-spring-cloud-netflix-eureka/>

5)

<https://www.adictosaltrabajo.com/2016/12/22/aplicaciones-web-con-spring-boot-capa-a-cap/>

6)

<https://www.baeldung.com/tracing-services-with-zipkin>

7)

<https://github.com/OpenFeign/feign>

From:

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

Permanent link:

<http://wiki.adacsc.co/doku.php?id=ada:howto:sicoferp:factory:migracionsicoferp:process:backend:guiadependenciasmaven> 

Last update: **2020/05/05 20:40**