

12

Features of the framework for building SOAP Web Service

Java Backend Developer I

Objetivos

Comprender los conceptos:

- What is Spring Web Service
- Why Contract First?
- Supports WS-Security
- Integrates with Spring



Agenda

Revisión de los siguientes conceptos :

- What is Spring Web Service
- Why Contract First?
- Supports WS-Security
- Integrates with Spring



What is Spring Web Service

Concepto

Spring Web Services (Spring-WS) es un producto de la comunidad Spring enfocado en la creación de servicios web basados en documentos.

Spring Web Services tiene como objetivo facilitar el desarrollo de servicios SOAP de primer contrato, lo que permite la creación de servicios web flexibles utilizando una de las muchas formas de manipular cargas útiles XML.

El producto se basa en Spring, lo que significa que puede utilizar los conceptos de Spring, como la inyección de dependencias, como parte integral de su servicio web.



What is Spring Web Service

Características

- **Hace de las mejores prácticas una práctica sencilla:** Spring Web Services facilita la aplicación de las mejores prácticas. Esto incluye prácticas como el perfil básico WS-I, el desarrollo Contract-First y un acoplamiento flexible entre contrato e implementación.
- **Asignaciones potentes:** puede distribuir solicitudes XML entrantes a cualquier objeto, según la carga útil del mensaje, el encabezado de acción SOAP o una expresión XPath.
- **Compatibilidad con API XML:** los mensajes XML entrantes se pueden manejar en API JAXP estándar como DOM, SAX y StAX, pero también JDOM, dom4j, XOM o incluso tecnologías de clasificación.



What is Spring Web Service

Características

- **Marshalling XML flexible:** el módulo Object / XML Mapping en la distribución de Spring Web Services es compatible con JAXB 1 y 2, Castor, XMLBeans, JiBX y XStream. Y debido a que es un módulo separado, también puede usarlo en código de servicios no web.
- **Reutiliza su experiencia en Spring:** Spring-WS usa contextos de aplicación Spring para todas las configuraciones, lo que debería ayudar a los desarrolladores de Spring a ponerse al día de manera rápida y agradable. Además, la arquitectura de Spring-WS se parece a la de Spring-MVC.
- **Soporta WS-Security:** WS-Security le permite firmar mensajes SOAP, cifrarlos y descifrarlos, o autenticarse contra ellos.



What is Spring Web Service

Características

- **Se integra con Acegi Security:** la implementación WS-Security de Spring Web Services proporciona integración con Spring Security. Esto significa que también puede usar su configuración existente para su servicio SOAP.
- **Creado por Maven:** esto le ayuda a reutilizar eficazmente los artefactos de Spring Web Services en sus propios proyectos basados en Maven.
- **Licencia de Apache.** Puede utilizar Spring-WS con confianza en su proyecto.



Why Contract First?

- Al crear servicios web, existen dos estilos de desarrollo: Contract Last o Contract First. Cuando utiliza un enfoque de último contrato, comienza con el código Java y deja que el contrato de servicio web (WSDL , consulte la barra lateral) se genere a partir de eso. Cuando usa el contrato primero, comienza con el contrato WSDL y usa Java para implementar dicho contrato.

¿Qué es WSDL?

- WSDL significa Lenguaje de descripción de servicios web. Un archivo WSDL es un documento XML que describe un servicio web. Especifica la ubicación del servicio y las operaciones (o métodos) que expone el servicio.



Why Contract First?

- Además de los problemas de asignación de objetos / XML que tiene el contract last, existen otras razones para preferir un estilo de desarrollo de contrato primero.
- A continuación se revisa el porqué se prefiere Contract First.

Fragilidad

- Como se mencionó anteriormente, el estilo de desarrollo del último contrato da como resultado que su contrato de servicio web (WSDL y su XSD) se genere a partir de su contrato de Java (generalmente una interfaz). Si utiliza este enfoque, no tendrá ninguna garantía de que el contrato se mantenga constante a lo largo del tiempo. Cada vez que cambie su contrato de Java y lo vuelva a implementar, puede haber cambios posteriores en el contrato de servicio web.



Why Contract First?

- Además, no todas las pilas de SOAP generan el mismo contrato de servicio web a partir de un contrato de Java. Esto significa que cambiar su pila SOAP actual por una diferente (por el motivo que sea), también podría cambiar su contrato de servicio web.
- Cuando cambia un contrato de servicio web, los usuarios del contrato deberán recibir instrucciones para obtener el nuevo contrato y posiblemente cambiar su código para adaptarse a cualquier cambio en el contrato.
- Para que un contrato sea útil, debe permanecer constante durante el mayor tiempo posible. Si un contrato cambia, deberá comunicarse con todos los usuarios de su servicio e indicarles que obtengan la nueva versión del contrato.



Why Contract First?

Actuación

- Cuando Java se transforma automáticamente en XML, no hay forma de estar seguro de lo que se envía a través del wire. Un objeto puede hacer referencia a otro objeto, que se refiere a otro, etc. Al final, la mitad de los objetos de la máquina virtual JAVA podrían convertirse a XML, lo que resultará en tiempos de respuesta lentos.
- Cuando se utiliza Contract First, se describe explícitamente qué XML se envía a dónde, asegurándose así de que sea exactamente lo que desea.



Why Contract First?

Reutilización

- Definir su esquema en un archivo separado le permite reutilizar ese archivo en diferentes escenarios. Si define un AirportCode en un archivo llamado airline.xsd, así:

```
<simpleType name = "AirportCode">  
  <base de restricción = "cadena">  
    <patrón valor = "[AZ] [AZ] [AZ]" />  
  </restriction>  
</simpleType>
```

Control de versiones

- Aunque un contrato debe permanecer constante durante el mayor tiempo posible, a veces *es* necesario cambiarlo. En Java, esto normalmente da como resultado una nueva interfaz Java, como AirlineService2, y una (nueva) implementación de esa interfaz.



Why Contract First?

- Por supuesto, el servicio anterior debe mantenerse, porque puede haber clientes que aún no hayan migrado.
- Si usamos el Contract First, podemos tener un acoplamiento más flexible entre el contrato y la implementación. Un acoplamiento tan flexible nos permite implementar ambas versiones del contrato en una clase. Podríamos, por ejemplo, usar una hoja de estilo XSLT para convertir cualquier mensaje de "estilo antiguo" en mensajes de "estilo nuevo".



Supports WS-Security

- WS-Security (Seguridad en Servicios Web) es un protocolo de comunicaciones que suministra un medio para aplicar seguridad a los Servicios Web. En abril de 2004 el estándar WS-Security 1.0 fue publicado por Oasis-Open. En 2006 fue publicada la versión 1.1.
- Originalmente desarrollado por IBM, Microsoft, y VeriSign, el protocolo es ahora llamado oficialmente WSS y está desarrollado por un comité en Oasis-Open.
- El protocolo contiene especificaciones sobre cómo debe garantizarse la integridad y seguridad en mensajería de Servicios Web. El protocolo WSS incluye detalles en el uso de SAML y Kerberos, y formatos de certificado tales como X.509.



Supports WS-Security

La Integridad de datos y confidencialidad podrían también garantizarse sobre Servicios Web a través del uso de la Transport Layer Security (TLS), por ejemplo enviando mensajes sobre HTTPS.

WS-Security incorpora características de seguridad en el encabezado de un mensaje SOAP, trabajando en la capa aplicación. Así asegura seguridad extremo a extremo.



Integrates with Spring

Actualmente la mayoría de empresas tienen Servicios SOAP debido a que hace 10 años o más esta tecnología era la mejor especificación para exponer a clientes de aplicaciones web o de escritorio.

Spring Boot permite hacer llamadas al webservices SOAP, para ello debemos tener presente:

- Para poder hacer la invocación de un servicio SOAP necesitamos generar su cliente. Esto implica que en nuestro servicio nuevo debemos conocer la implementación del SOAP que queremos invocar. Si ya has trabajado con servicios SOAP anteriormente probablemente ya has trabajado también con programas como Metro para la generación de clientes SOAP a partir de un WSDL
- Generar clases java a partir de la especificación WSDL



Integrates with Spring

- Generar clase cliente: la clase en la que hacemos la invocación del servicio
- Configurar el marshalling: El marshalling se refiere a la conversión entre XML y objetos Java en tiempo de ejecución, esto es, los requests que enviamos (Java a XML) y los responses que recibimos (XML a Java).
- Invocar el cliente SOAP
- Setear autenticación
- Setear cabeceras HTTP adicionales



Integrates with Spring

- Si bien hoy en día ya prácticamente no se están generando nuevos servicios SOAP, existen muchas empresas y organizaciones que aun no renuevan esta parte de su infraestructura tecnológica, por lo que el camino para muchos será escribir integraciones contra estos servicios.
- Afortunadamente, con Spring Boot se puede observar que no es complicado desarrollar estas integraciones. Es más, podemos desarrollarlas de forma bastante limpia y modular.



Lecturas adicionales

Para obtener información adicional, puede consultar los siguientes enlaces:

- [https://spring.io/projects/spring-ws#:~:text=Spring%20Web%20Services%20\(Spring%2DWS,ways%20to%20manipulate%20XML%20payloads.](https://spring.io/projects/spring-ws#:~:text=Spring%20Web%20Services%20(Spring%2DWS,ways%20to%20manipulate%20XML%20payloads.)
- https://docs.oracle.com/cd/E23943_01/web.1111/b32511/intro_security.htm#WSSEC1112
- <https://es.wikipedia.org/wiki/WS-Security>
- <https://docs.spring.io/spring-integration/reference/html/ws.html>



Resumen

En este capítulo, usted aprendió:

- What is Spring Web Service
- Why Contract First?
- Supports WS-Security
- Integrates with Spring

