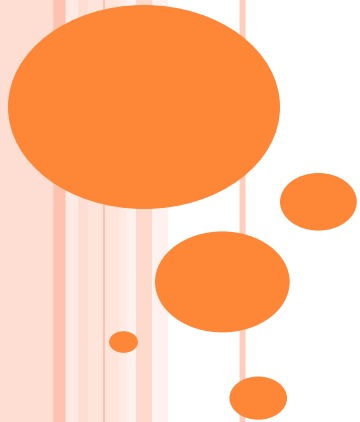


IMPLEMENTACIÓN DE SERVICIOS WEB XML CON SPRING



Dependencias

➤ **Spring simplifica la implementación de servicios SOAP, encargándose del mapeo Java/XML y de la generación de modelo de datos y WSDL**

➤ **Se requieren las dependencias:**

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web-services</artifactId>
</dependency>
<dependency>
    <groupId>wsdl4j</groupId>
    <artifactId>wsdl4j</artifactId>
</dependency>
```



Archivo xsd

➤ Define los datos utilizados por el servicio. Se añadirá a la carpeta de recursos:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://www.cursoindara/academy"
targetNamespace="http://www.cursoindara/academy" elementFormDefault="qualified">

  <xs:element name="CursoDetalleRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="CursoDetalleResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Curso" type="tns:Curso"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="Curso">
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="duration" type="xs:int"/>
      <xs:element name="price" type="xs:double"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```



Generación de clases

➤ Para generar las clases Java a partir del modelo de datos, será necesario registrar el siguiente plugin en **pom.xml**:

```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>jaxb2-maven-plugin</artifactId>
  <version>1.6</version>
  <executions>
    <execution>
      <id>xjc</id>
      <goals>
        <goal>xjc</goal>
      </goals>
    </execution>
  </executions>
  <configuration>
    <schemaDirectory>${project.basedir}/src/main/resources/</schemaDirectory>
    <outputDirectory>${project.basedir}/src/main/java</outputDirectory>
    <clearOutputDir>>false</clearOutputDir>
  </configuration>
</plugin>
```

➤ Después, run as->maven build->compile



Endpoint

➤ **Representa el punto de entrada a las peticiones:**

```
@Endpoint
public class CursosEndpoint {
    private static final String NAMESPACE_URI = "http://www.cursoindara/academy";
    private CursosRepository cursosRepository;
    @Autowired
    public CursosEndpoint(CursosRepository cursosRepository) {
        this.cursosRepository = cursosRepository;
    }
    @PayloadRoot(namespace = NAMESPACE_URI, localPart = "CursoDetalleRequest")
    @ResponsePayload
    public CursoDetalleResponse getCurso(@RequestPayload CursoDetalleRequest request) {
        CursoDetalleResponse response = new CursoDetalleResponse();
        response.setCurso(cursosRepository.findByName(request.getName()));
        return response;
    }
}
```

Lógica de negocio

Objeto con mensaje de
respuesta

Objeto con mensaje de
llamada



Clase de configuración

➤ Objetos de configuración de Spring:

```
@EnableWs
@Configuration
public class WebServiceConfig extends WsConfigurerAdapter {
    // bean definitions
    @Bean
    public ServletRegistrationBean messageDispatcherServlet(ApplicationContext applicationContext) {
        MessageDispatcherServlet servlet = new MessageDispatcherServlet();
        servlet.setApplicationContext(applicationContext);
        servlet.setTransformWsdLocations(true);
        return new ServletRegistrationBean(servlet, "/ws/*");
    }
    @Bean(name = "cursos")
    public DefaultWsd11Definition defaultWsd11Definition(XsdSchema countriesSchema) {
        DefaultWsd11Definition wsdl11Definition = new DefaultWsd11Definition();
        wsdl11Definition.setPortTypeName("CursoPort");
        wsdl11Definition.setLocationUri("/ws");
        wsdl11Definition.setTargetNamespace("http://www.cursoindara/academy");
        wsdl11Definition.setSchema(countriesSchema);
        return wsdl11Definition;
    }
    @Bean
    public XsdSchema countriesSchema() {
        return new SimpleXsdSchema(new ClassPathResource("curso.xsd"));
    }
}
```

Probando el servicio

➤ Creamos un documento xml soap con la petición:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:gs="http://www.cursoindara/academy">
  <soapenv:Header/>
  <soapenv:Body>
    <gs:CursoDetalleRequest>
      <gs:name>Java</gs:name>
    </gs:CursoDetalleRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

➤ Y lanzamos la llamada con curl:

```
>curl --header "content-type: text/xml" -d @request.xml http://localhost:8080/ws
```

