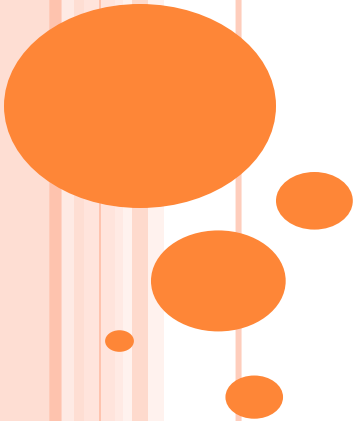
Aplicación cliente de un servicios Web XML con Spring



Dependencias

- **Spring simplifica la implementación de aplicaciones clientes de servicios SOAP, encargándose del mapeo Java-XML a partir del documento WSDL**

- **Se requieren las dependencias:**

```
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter</artifactId>
</dependency>
<dependency>
        <groupId>org.springframework.ws</groupId>
        <artifactId>spring-ws-core</artifactId>
</dependency>
```

Generación de clases

- **Para generar las clases Java a partir del documento WSDL, será necesario registrar el siguiente plugin en pom.xml:**

```
<plugin>
   <groupId>org.jvnet.jaxb2.maven2</groupId>
   <artifactId>maven-jaxb2-plugin</artifactId>
   <version>0.14.0</version>
   <executions>
      <execution>
         <goals>
            <goal>generate</goal>
         </goals>
      </execution>
   </executions>
   <configuration>
      <schemaLanguage>WSDL</schemaLanguage>
      <generateDirectory>${project.basedir}/src/main/java</generateDirectory>
      <generatePackage>com.example.gen</generatePackage>
      <schemas>
         <schema>
            <url>http://localhost:8080/ws/cursos.wsdl</url>
         </schema>
      </schemas>
   </configuration>
</plugin>
```

Creación de cliente

- **Para crear un cliente se debe extender WebServiceGatewaySupport :**

```
public class CursosClient extends WebServiceGatewaySupport{
   private static final Logger log = LoggerFactory.getLogger(CursosClient.class);

   public CursoDetalleResponse getCurso(String name) {

      CursoDetalleRequest request = new CursoDetalleRequest();
      request.setName(name);

      log.info("Requesting location for " + name);

      CursoDetalleResponse response = (CursoDetalleResponse) getWebServiceTemplate()
            .marshalSendAndReceive("http://localhost:8080/ws/cursos", request,
                  new SoapActionCallback(
                        "http://www.cursoindara/academy/CursoDetalleRequest"));

      return response;
   }
}
```

clases generadas por el plugin JAXB

Clase de configuración

- **Objetos de configuración de Spring:**

```
@Configuration
public class CursosConfiguration {
   @Bean
   public Jaxb2Marshaller marshaller() {
      Jaxb2Marshaller marshaller = new Jaxb2Marshaller();
      // this package must match the package in the <generatePackage> specified in
      // pom.xml
      marshaller.setContextPath("com.example.gen");
      return marshaller;
   }
   @Bean
   public CursosClient countryClient(Jaxb2Marshaller marshaller) {
      CursosClient client = new CursosClient();
      client.setDefaultUri("http://localhost:8080/ws");
      client.setMarshaller(marshaller);
      client.setUnmarshaller(marshaller);
      return client;
   }
}
```

Ejecución del cliente

- **Se prueba directamente la aplicación desde la clase lanzadora mediante CommandLineRunner:**

```
@SpringBootApplication(scanBasePackages =
{"com.example.client","com.example.demo","com.example.gen"})
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
    @Bean
    CommandLineRunner lookup(CursosClient quoteClient) {
        return args -> {
            String name = "Java";
            CursoDetalleResponse response = quoteClient.getCurso(name);
            System.out.println(response.getCurso().getDuration());
        };
    }

}
```