



INSTITUTO TECNOLÓGICO DE NUEVO LEÓN

ALUMNA:

EDITH MENDOZA CONTRERAS #16480063

MATERIA:

LENGUAJES Y AUTOMATAS 1

INVESTIGACION JAX

----corrección de esta investigación----

Que es:

Jax es un compilador léxico creado en lenguaje java, que genera un escáner a partir de expresiones regulares que existen por defecto en un archivo de java. Jax procesa estas expresiones regulares y genera un fichero java que pueda ser compilado por jaca y así crear el escáner. Los escáneres generados por jax tienen entradas de búfer de tamaño arbitrario, y es al menos más conveniente para crear las tablas de tokens, jax utiliza solo 7 bits de caracteres ASCII, y no permite código unario.

Cuántas funciones implementa.

JAX-RPC

La especificación Java API for XML-based RPC (JAX-RPC) permite desarrollar clientes de servicios web y servicios web interoperativos y portátiles basados en SOAP. JAX-RPC 1.1 proporciona API básicas para el desarrollo y el despliegue de servicios web en una plataforma Java™ y es una parte de la plataforma de servicios web para Java EE (Java Platform, Enterprise Edition). La plataforma Java EE permite desarrollar servicios web portátiles. WebSphere Application Server implementa estándares JAX-RPC 1.1.

El estándar JAX-RPC cubre el modelo de programación y los enlaces para la utilización de WSDL (Web Services Description Language) para servicios web en lenguaje Java. JAX-RPC simplifica el desarrollo de servicios web al protegerle de la complejidad subyacente de la comunicación SOAP. En la superficie, JAX-RPC parece otra creación de instancias de RMI (Remote Method Invocation).

Básicamente, JAX-RPC permite a los clientes acceder a un servicio web como si el servicio web fuera un objeto local correlacionado con el espacio de direcciones del cliente, aunque el proveedor del servicio web se encuentre en otra parte del mundo. JAX-RPC se ejecuta utilizando el protocolo SOAP basado en XML, que se ejecuta normalmente encima de HTTP.

JAX-RPC define las correlaciones entre los tipos de puerto WSDL y las interfaces Java, así como entre el lenguaje Java y los tipos de esquema XML (Extensible Markup Language). Se puede crear un servicio web de JAX-RPC a partir de una implementación de JavaBeans o de Enterprise bean. Puede especificar los procedimientos remotos definiendo métodos remotos en una interfaz Java. Sólo necesita codificar una o varias clases que implementen los métodos.

Las demás clases y los otros artefactos se generan mediante las herramientas del proveedor del servicio web. A continuación, se proporciona un ejemplo de una interfaz de servicios web:

Ejemplo:

```
package com.ibm.mybank.ejb;
import java.rmi.RemoteException;
import com.ibm.mybank.exception.InsufficientFundsException;
/**
 * Interfaz remota para Enterprise Bean: Transfer
 */
public interface Transfer_SEI extends java.rmi.Remote {
    public void transferFunds(int fromAcctId, int
toAcctId, float amount)
                                throws
java.rmi.RemoteException;
}
```

JAX-WS

(Java™ API for XML-Based Web Services) es la siguiente generación del modelo de programación de servicios web que complementa la infraestructura que proporciona el modelo de programación JAX-RPC (Java API for XML-based RPC). Con JAX-WS, se simplifica el desarrollo de servicios web y clientes con una mayor independencia de la plataforma para las aplicaciones Java mediante el uso de anotaciones Java y proxies dinámicos. JAX-WS es un modelo de programación que simplifica el desarrollo de aplicaciones mediante el soporte de un modelo basado en anotaciones estándar para desarrollar clientes y aplicaciones de servicio web.

La tecnología JAX-WS se alinea estratégicamente con las tendencias actuales del sector hacia un modelo de mensajería más centrado en los documentos y sustituye el modelo de programación de llamada a procedimiento remoto tal como se define en JAX-RPC.

Aunque el modelo de programación JAX-RPC y las aplicaciones siguen recibiendo soporte en este producto, JAX-RPC tiene limitaciones y no da soporte a varios servicios complejos centrados en los documentos. JAX-WS es el modelo de programación estratégico para desarrollar servicios web, además de ser una parte necesaria de Java Platform, Enterprise Edition 6 (Java EE 6). JAX-WS también se

conoce como JSR 224. La especificación JAX-WS 2.2 sustituye e incluye funciones de la especificación JAX-WS 2.1. JAX-WS 2.2 añade soporte del lado cliente para utilizar anotaciones relacionadas con `WebServiceFeature` como `@MTOM`, `@Addressing` y las anotaciones `@RespectBinding`. JAX-WS 2.1 anteriormente ha agregado soporte para estas anotaciones en el servidor. También existe ahora la posibilidad de habilitar y configurar el soporte de WS-Addressing en un cliente o servicio añadiendo aserciones WS-Policy en el documento WSDL. Además, los servicios web para la especificación Java EE 1.3 introducen soporte para estas anotaciones relacionadas con `WebServiceFeature`, así como soporte para utilizar los elementos del descriptor de despliegue para configurar estas características tanto en el cliente como en el servidor. JAX-WS 2.2 exige Java Architecture for XML Binding (JAXB) Versión 2.2 para el enlace de datos. La implementación del estándar de programación JAX-WS proporciona las siguientes mejoras para desarrollar clientes y servicios web:

- Independencia de plataforma ampliada para aplicaciones Java. Con las API de JAX-WS el desarrollo de los servicios y los clientes web se simplifica con una independencia de plataforma ampliada para aplicaciones Java. JAX-WS

aprovecha el mecanismo de proxy dinámico para proporcionar un modelo de delegación formal con un proveedor conectable. Esto constituye una mejora respecto a JAX-RPC, que se basa en la generación de objetos stub específicos del proveedor para la invocación.

- Anotaciones JAX-WS proporciona el soporte para anotar clases Java con metadatos para indicar que la clase Java es un servicio web. JAX-WS da soporte al uso de anotaciones basándose en la especificación del recurso de metadatos del lenguaje de programación Java (JSR 175), la especificación de metadatos de servicios web para la plataforma Java (JSR 181) y las anotaciones definidas por la especificación JAXWS 2.2. Utilizando las anotaciones del origen Java y la clase Java se simplifica el desarrollo de los servicios web. Utilice las anotaciones para definir la información que se suele especificar en los archivos del descriptor de despliegue, los archivos WSDL o en la correlación de metadatos de archivos XML y WSDL con los artefactos de origen. Por ejemplo, puede incorporar un código `@WebService` simple en el origen Java para exponer el bean como servicio web.

```
@WebService
public class QuoteBean implements StockQuote {
    public float getQuote(String sym) { ... }
}
```

Para los servicios Web JAX-WS, el uso del descriptor de despliegue `webservices.xml` es opcional porque puede utilizar anotaciones para especificar toda la información incluida dentro del archivo del descriptor de despliegue. Puede

utilizar el archivo del descriptor de despliegue para aumentar o alterar temporalmente las anotaciones de JAX-WS existentes. Cualquier información que defina en el descriptor de despliegue webservices.xml altera temporalmente la información correspondiente especificada por las anotaciones. Por ejemplo, si la clase de implementación de servicio del servicio web JAX-WS incluye lo siguiente:

la anotación @WebService:

```
o @WebService(wsdlLocation="http://myhost.com/location/of/the/wsdl/ExampleService.wsdl")
```

el archivo webservices.xml especifica un nombre de archivo distinto para el documento WSDL, como se indica a continuación:

```
<webservices>
<webservice-description>
<webservice-description-name>ServicioEjemplo</webservice-description-name>
<wsdl-file>META-INF/wsdl/ExampleService.wsdl</wsdl-file>
...
</webservice-description>
</webservices>
```

EJEMPLO

Generar un web service con metro.

```
62.983 activation.jar
      291.817 FastInfoset.jar
      21.839 gmbal-api-only.jar
      82.265 http.jar
      104.554 jaxb-api.jar
      876.738 jaxb-impl.jar
    3.105.074 jaxb-xjc.jar
      54.341 jaxws-api.jar|
    1.484.080 jaxws-rt.jar
      520.641 jaxws-tools.jar
      23.346 jsr173_api.jar
       7.993 jsr181-api.jar
       6.165 jsr250-api.jar
      41.429 management-api.jar
      38.772 mimepull.jar
     156.212 policy.jar
       68.177 resolver.jar
       18.774 saaj-api.jar
     288.529 saaj-impl.jar
       11.001 stax-ex.jar
       59.771 streambuffer.jar
     505.825 woodstox.jar
```