UF4.
Serveis web. Pàgines dinàmiques interactives.
Webs Híbrids

M07. Desenvolupament web en entorn servidor

2.
SERVEIS WEB
SOAP

JESUÏTES El Clot
Escola del Clot
Sergi Grau

Introducció

Sergi Grau
sergi.grau@fje.edu

- SOAP (sigles de Simple Object Access Protocol) és un protocol estàndard que defineix com dos objectes en diferents processos poden comunicar-se per mitjà d'intercanvi de dades XML .
- Aquest protocol deriva d'un protocol creat per Dave Winer el 1998, anomenat XML-RPC.
- SOAP va ser creat per Microsoft , IBM i altres. Està actualment sota l'auspici de la W3C . És un dels protocols utilitzats en els serveis web .

Característiques

Sergi Grau
sergi.grau@fje.edu

Bàsicament SOAP És un paradigma de missatgeria d'una direcció sense estat, que pot ser utilitzat per formar protocols més complexos i complets segons les necessitats de les aplicacions que l'implementen. Pot formar i construir la capa base d'una " pila de protocols de web service ", oferint un framework de missatgeria bàsica en el qual els web services es poden construir. Aquest protocol està basat en XML i es conforma de tres parts:

- Sobre (envelope): el qual defineix què hi ha al missatge i com s'ha de processar.
- Conjunt de regles de codificació per expressar instàncies de tipus de dades.
- La Convenció per a representar crides a procediments i respostes.

Característiques

Sergi Grau
sergi.grau@fje.edu

El protocol SOAP té tres característiques principals:

- Extensibilitat (seguretat i WS-routing són extensions aplicades en el desenvolupament).
- Neutralitat (SOAP pot ser utilitzat sobre qualsevol protocol de transport com HTTP , SMTP , TCP o JMS).
- Independència (SOAP permet qualsevol model de programació).

Característiques

Sergi Grau
sergi.grau@fje.edu

Com a exemple de com el model SOAP pot ser utilitzat, considerarem un missatge SOAP que podria ser enviat a un web service per realitzar la recerca d'algun preu en una base de dades, indicant per a això els paràmetres necessitats en la consulta. El servei podria retornar un document en format XML amb el resultat, un exemple, preus, localització o característiques. Tenint les dades de resposta en un format estandarditzat processable (en anglès "parseable"), aquest pot ser integrat directament en un lloc web o aplicació externa.

L'arquitectura SOAP està formada per diverses capes d'especificació: MEP (Message Exchange Patterns) per al format del missatge, enllaços subjacents del protocol de transport, el model de processament de missatges, i la capa d'extensibilitat del protocol. SOAP és el successor de XML-RPC , tot i que pren el transport i la neutralitat de la interacció, així com l'envelope / header / body, d'altres models (probablement de WDDX).

Història

Sergi Grau
sergi.grau@fje.edu

La preocupación por los sistemas distribuidos y de cómo diferentes máquinas podían comunicarse entre sí surgió en la década de los 90. Hasta ese momento, era suficiente con que las aplicaciones de un mismo ordenador pudieran establecer una comunicación.

- En 1990, surgieron los modelos COM y CORBA. El primero, Component Object Model fue creado por Microsoft, y el segundo, CORBA, por el Object Management Group. No obstante, estos dos modelos presentaban un hándicap muy importante: no eran fácilmente interoperables ya que las dos máquinas que llevaran a cabo la comunicación debían soportar COM o CORBA, por tanto únicamente se podía utilizar con dos máquinas COM o dos máquinas CORBA. Más adelante, Microsoft creó DCOM y Sun, RMI (Remote Method Invocation).

Història

Sergi Grau
sergi.grau@fje.edu

- Aunque estos métodos permitían establecer una conexión entre ordenadores a través de la red, tampoco eran interoperables ya que RMI está disponible únicamente para Java, y por tanto, es dependiente del lenguaje de programación. Por todo ello, Microsoft empezó a interesarse por la computación distribuida basada en XML en el año 1997. Su objetivo era terminar con los problemas de interoperabilidad de las soluciones anteriores y permitir que las aplicaciones se conectaran mediante RPCs (Remote Procedure Calls), utilizando los estándares de comunicación XML y HTTP.
- SOAP fue diseñado como un protocolo de acceso a objetos en 1998 por Dave Winer, Don Box, Bob Atkinson y Mohsen Al-Ghosein por Microsoft, donde Atkinson y Al-Ghosein trabajaban en aquel entonces. La especificación SOAP actualmente es mantenida por el XML Protocol Working Group del World Wide Web Consortium.

Història

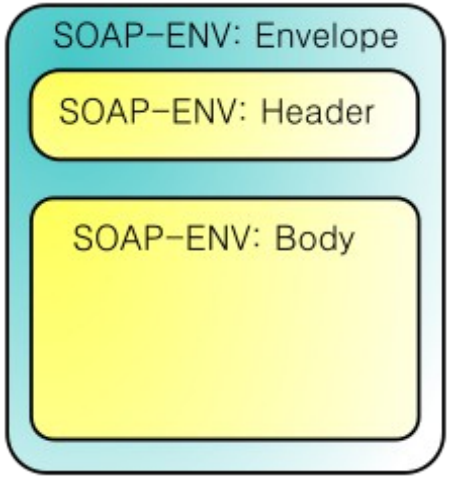
Sergi Grau
sergi.grau@fje.edu

- La versión SOAP 1.1 se presentó en el año 2000 e IBM participó en su creación. Esta participación resultó muy positiva ya que se produjeron cambios significativos y cruciales para su posterior uso: se diseñó de una forma más modular y escalable, eliminando los problemas derivados de una tecnología propietaria, en este caso de Microsoft. Además, IBM llevó a cabo una implementación de SOAP en Java y SOAP se integró en Web Services J2EE.

Estructura

Sergi Grau
sergi.grau@fje.edu

- Un mensaje SOAP es un documento XML ordinario con una estructura definida en la especificación del protocolo. Dicha estructura la conforman las siguientes partes:
- Envelope (obligatoria): raíz que de la estructura, es la parte que identifica al mensaje SOAP como tal.



Estructura

Sergi Grau
sergi.grau@fje.edu

- Header: esta parte es un mecanismo de extensión ya que permite enviar información relativa a como debe ser procesado el mensaje. Es una herramienta para que los mensajes puedan ser enviados de la forma más conveniente para las aplicaciones. El elemento "Header" se compone a su vez de "Header Blocks" que delimitan las unidades de información necesarias para el header.
- Body (obligatoria): contiene la información relativa a la llamada y la respuesta.
- Fault: bloque que contiene información relativa a errores que se hayan producido durante el procesado del mensaje y el envio desde el "SOAP Sender" hasta el "Ultimate SOAP Receiver"

Model de processament

Sergi Grau
sergi.grau@fje.edu

El modelo de procesado de SOAP está definido como un sistema distribuido, en el que intervienen diferentes nodos. En un escenario básico, los nodos SOAP se comunican uno asumiendo el rol de SOAP Sender y SOAP Receiver. Aún así, la especificación define diferentes tipos de nodos en función del rol que asumen en el envío del mensaje:

- SOAP Sender: nodo que transmite un mensaje SOAP.
- SOAP Receiver: nodo que acepta un mensaje.
- SOAP message path: conjunto de nodos por los cuales debe pasar un mensaje SOAP, incluyendo el nodo inicial, cero o más nodos intermediarios y el SOAP Receiver definitivo.
- Initial SOAP sender: el "sender" que origina el mensaje y que es el punto de inicio del camino que seguirá el mensaje.

Model de processament

Sergi Grau
sergi.grau@fje.edu

- SOAP intermediary: el intermediario actua como SOAP receiver y como SOAP sender, ya que primero recibe el mensaje para después reenviarlo al siguiente nodo en el camino.
- Ultimate SOAP receiver: destino final del mensaje SOAP, es el responsable de procesarlo. Cabe destacar que el mensaje podría no llegar al receptor definitivo debido a que problemas en los intermediarios hagan que se pierda.
- Un nodo SOAP puede actuar con uno o varios roles, cada uno de los cuales se encuentra definido mediante una URI conocida como el nombre de rol. Los roles asumidos por un nodo son invariantes durante el envío de un mensaje, teniendo en cuenta la especificación el procesado individual de mensajes. Tal y como se ha comentado, una aplicación puede crear protocolos de comunicación más complejos como capas superiores sobre SOAP, pudiendo definir sus propios roles para poder cumplir con sus necesidades.

Petició SOAP

Sergi Grau
sergi.grau@fje.edu

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
     <getProductDetails xmlns="http://warehouse.example.com/ws">
       <productId>827635</productId>
     </getProductDetails>
   </soap:Body>
</soap:Envelope>
```

Resposta SOAP

Sergi Grau
sergi.grau@fje.edu

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
     <getProductDetailsResponse
xmlns="http://warehouse.example.com/ws">
       <getProductDetailsResult>
         <productName>Toptimate 3-Piece Set</productName>
         <productId>827635</productId>
         <description>3-Piece luggage set.  Black
Polyester.</description>
         <price>96.50</price>
         <inStock>true</inStock>
       </getProductDetailsResult>
     </getProductDetailsResponse>
   </soap:Body>
 </soap:Envelope>
```

Estructura

Sergi Grau
sergi.grau@fje.edu

- Les aplicacions web basades en dades REST necessiten coneixer els recursos disponibles. Per a fer-ho tracten un recull de resultats de cerca com un altre tipus de recurs, el que requereix que els dissenyadors de l'aplicació coneguin URL addicionals per mostrar o buscar cada tipus de recurs.
- Per exemple, una petició GET HTTP sobre l'URL http://www.exemple.cat/localitzacio/ca/barcelona" podria tornar un enllaç a una llista de fitxers en XML amb totes les localitzacions possibles a barcelona.

Avantatges

Sergi Grau
sergi.grau@fje.edu

- A causa al ÚS d'XML permet Invocar Procediments Remots de molts llenguatges, Per Tant, Presenta Una gran Interoperabilitat.
- En utilitzar Una Comunicació via HTTP és fàcilment escalable ,: A més de Ser gairebé per sempre Permès Pels tallafocs.
- Pot Ser Implementat utilitzant qualsevol Llenguatge i executat en qualsevol Plataforma.
- És Possible utilitzar Mitjançant usuari anònim i mitjà autenticació.
- És Possible transmetre Mitjançant qualsevol protocol sanitari de transport Capaç de Transmetre text, típicament HTTP o SMTP .

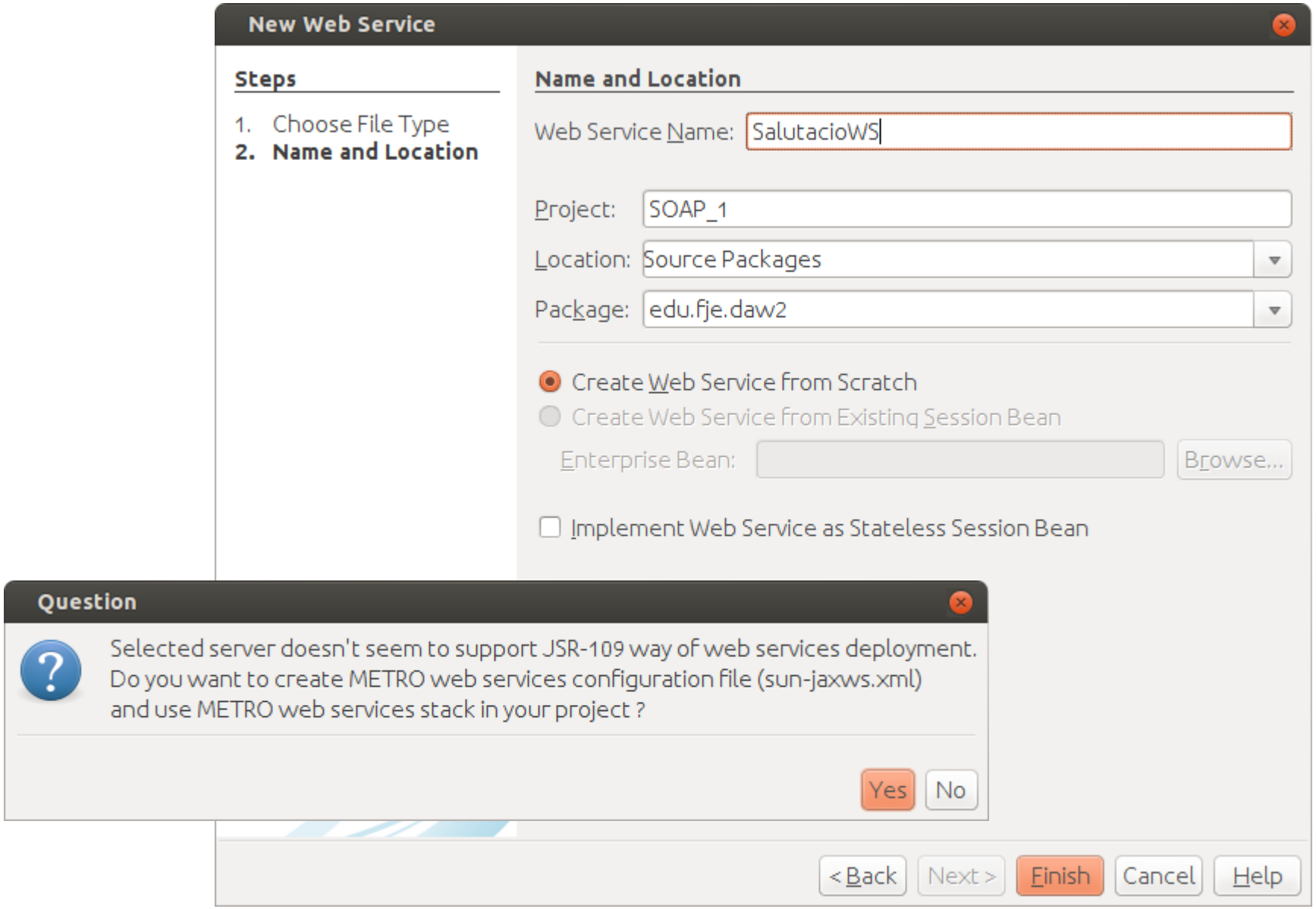
Inconvenients

Sergi Grau
sergi.grau@fje.edu

- Per l'ús de XML paràgraf del Pas de Missatges, SOAP és considerablement més lent que altres middleware com CORBA ja que les dades binaries es codifiquen Com a text.
- Depèn del WSDL (Web Services Description Language).
- Al contrari que Java, PHP o Python, alguns llenguatges no ofereixen suport. El mateix passa amb els IDEs

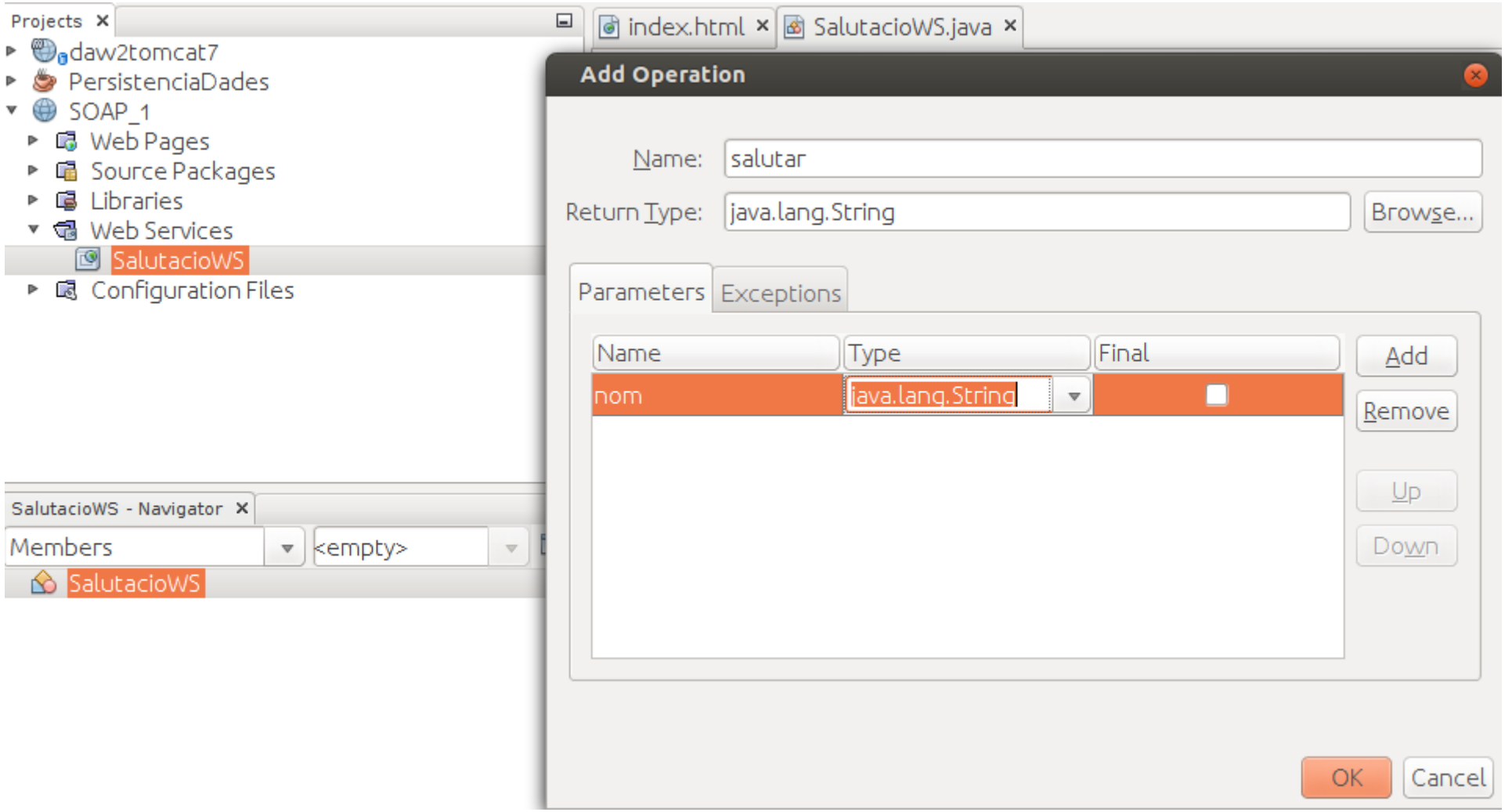
Creació de WS SOAP amb NetBeans

Sergi Grau
sergi.grau@fje.edu



Creació de WS SOAP amb NetBeans

Sergi Grau
sergi.grau@fje.edu



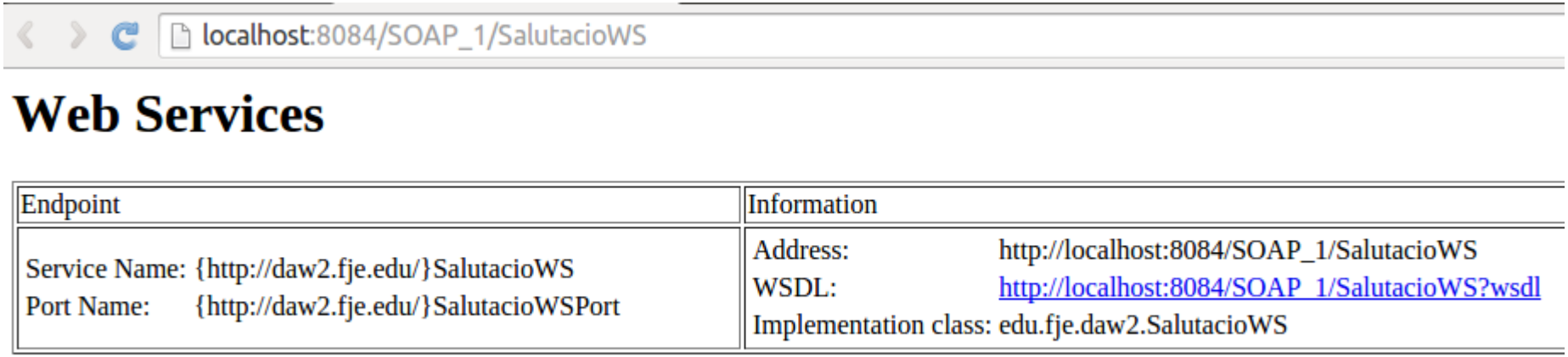
Exemple de codi del WS amb JWS

Sergi Grau
sergi.grau@fje.edu

```
package edu.fje.daw2;

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

@WebService(serviceName = "SalutacioWS")
public class SalutacioWS {

    /**
     * Operació del servei web
     */
    @WebMethod(operationName = "salutar")
    public String salutar(@WebParam(name = "nom") String nom) {
        return "hola" + nom;
    }


}
```

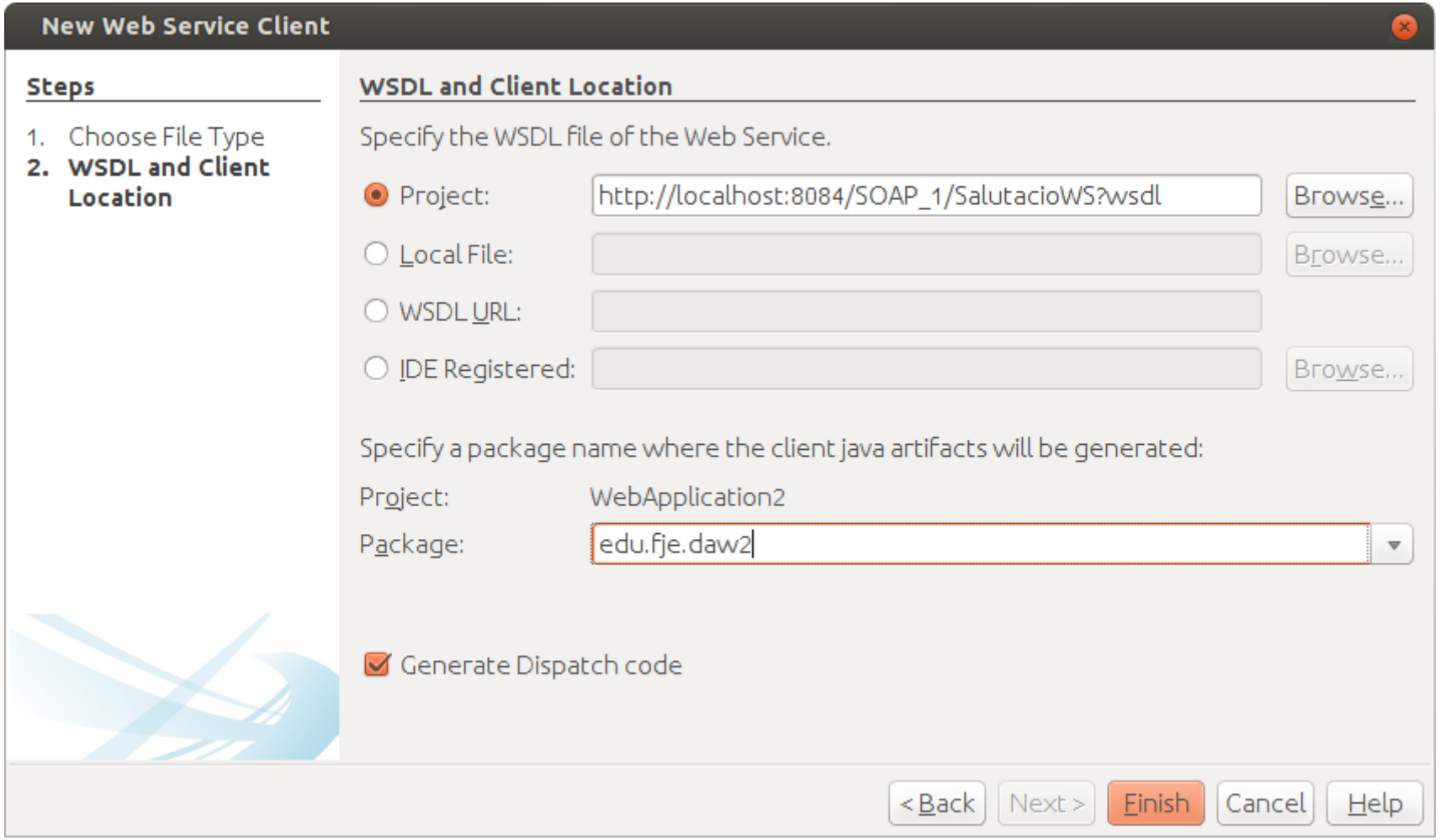
Test del WS

Sergi Grau
sergi.grau@fje.edu



Consumidor WS

Sergi Grau
sergi.grau@fje.edu



Consumidor WS amb Java

Sergi Grau
sergi.grau@fje.edu

```
public class ConsumidorWSEscriptori {

    public static void main(String[] args) {
        try {

            String resultat = salutar("sergi");
            System.out.println("Resultat = " + resultat);
        } catch (Exception ex) {
            System.out.println("Exception: " + ex);
        }
    }

    /**
     * Mètode privat que accedeix a un servei web
     *
     * @param i
     * @param j
     * @return retorna el resultat de la suma
     */
    private static String salutar(String n) {
        edu.fje.daw2.SalutacioWS_Service service = new
edu.fje.daw2.SalutacioWS_Service();
        edu.fje.daw2.SalutacioWS port = service.getSalutacioWSPort();
        return port.salutar(n);
    }
```

Consumidor WS amb PHP

Sergi Grau
sergi.grau@fje.edu

```php
<?php

$client = new SoapClient("http://localhost:8084/SOAP_1/SalutacioWS?
wsdl");
$params = array(
  "nom" => "sergi",
);
$response = $client->__soapCall("salutar", array($params));
var_dump($response);

?>
```

WSDL

Sergi Grau
sergi.grau@fje.edu

```
<definitions xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://daw2.fje.edu/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://daw2.fje.edu/" name="SalutacioWS">
<types>
<xsd:schema>
<xsd:import namespace="http://daw2.fje.edu/"
schemaLocation="http://localhost:8084/SOAP_1/SalutacioWS?xsd=1"/>
</xsd:schema>
</types>
<message name="salutar">
<part name="parameters" element="tns:salutar"/>
</message>
```

WSDL

Sergi Grau
sergi.grau@fje.edu

```
<message name="salutarResponse">
<part name="parameters" element="tns:salutarResponse"/>
</message>
<portType name="SalutacioWS">
<operation name="salutar">
<input
wsam:Action="http://daw2.fje.edu/SalutacioWS/salutarRequest"
message="tns:salutar"/>
<output
wsam:Action="http://daw2.fje.edu/SalutacioWS/salutarResponse"
message="tns:salutarResponse"/>
</operation>
</portType>
<binding name="SalutacioWSPortBinding" type="tns:SalutacioWS">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http"
style="document"/>
<operation name="salutar">
<soap:operation soapAction=""/>
<input>
```

WSDL

Sergi Grau
sergi.grau@fje.edu

```
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
</binding>
<service name="SalutacioWS">
<port name="SalutacioWSPort"
binding="tns:SalutacioWSPortBinding">
<soap:address
location="http://localhost:8084/SOAP_1/SalutacioWS"/>
</port>
</service>
</definitions>
```