Generalitat de Catalunya
Centre de Telecomunicacions
i Tecnologies de la Informació

Framework Corporatiu J2EE

Servei d'HTTP

Versió 1.1

Barcelona, 2 / octubre / 2006

Històric de modificacions

Data	Autor	Comentaris	Versió
13/01/2006	Atos Origin, sae openTrends	Versió inicial del document	1.0
29/09/2006	Atos Origin	Versió 1.1 (no hi ha canvis respecte versió 1.0.X)	1.1

Llegenda de Marcadors

Índex

1. Introducció

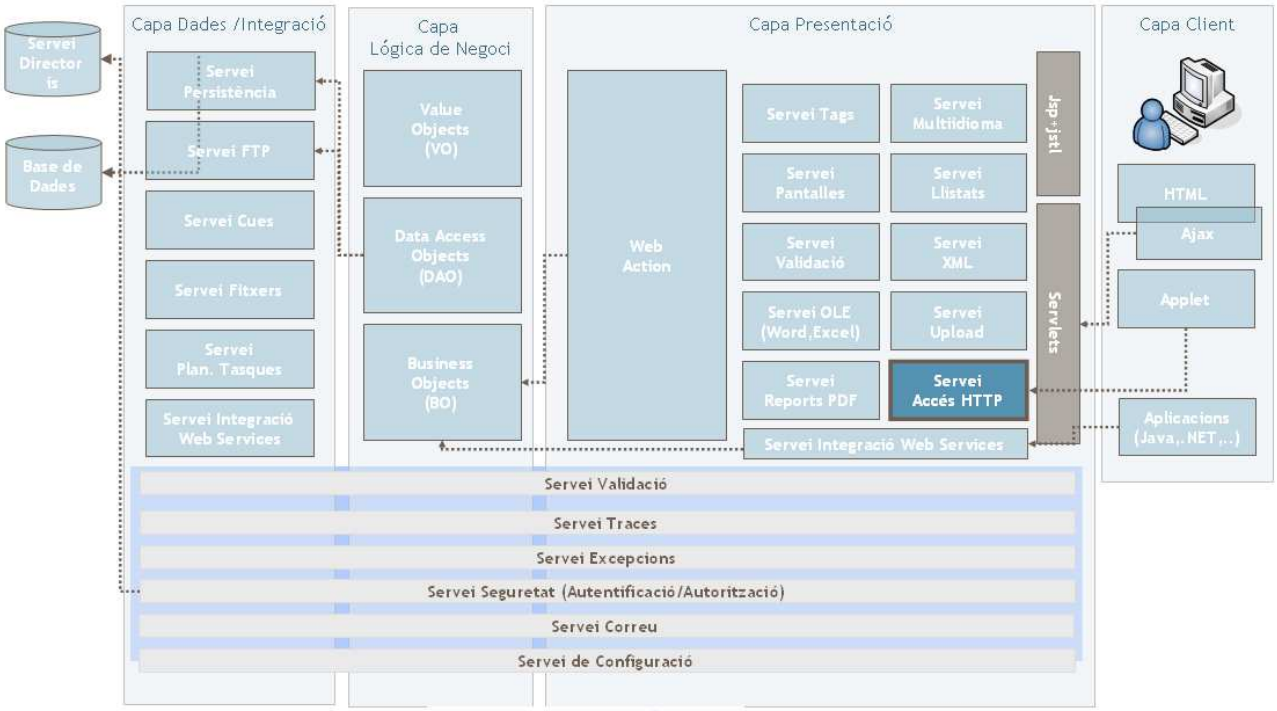
1.1. Propósit

Aquest servei permet un accés remot mitjançant HTTP als serveis o classes que es trobin instal·lades al servidor. Per lo general, aquest accés remot es realitzarà des de Applets Java, però es pot realitzar també des d'altres tipus de clients que vulguin fer ús del protocol HTTP sense fer ús de WebServices.

Els Serveis podran ser publicats per HTTP per tal que els clients puguin accedir de forma senzilla.

1.2. Context i Escenaris d'Ús

El Servei d'Accés HTTP es troba ubicat dins els serveis proporcionats a la Capa de Presentació.



1.3. Versions i Dependències

1.3.1. Versions:

No hi ha cap diferència entre la versió 1.0.X i la versió 1.1.

1.3.2. Dependències Bàsiques

En el present apartat es mostren quines són les versions i dependències necessàries per fer ús del Servei.

Dins la llista de dependències es mostren diferenciades:

- Dependències bàsiques. Llibreries necessàries per fer ús del servei.
- Dependències addicionals. Aquestes dependències són necessàries per poder fer ús de característiques concretes del servei o per l'ús dels tests unitaris proporcionats amb el servei.
-

Nom	Tipus	Versió	Descripció
commons-logging	jar	1.0.4	http://jakarta.apache.org/commons
log4j	jar	1.2.12	
hessian	jar	3.0.1	
spring	jar	1.2.5	http://www.springframework.org

1.3.3. Dependències Addicionals

- Proves Unitàries del Servei

Nom	Tipus	Versió	Descripció
httpunit	jar	1.6	

Veure l'apartat 'Instal·lació i configuració' per a més detall.

1.4. A qui va dirigit

Aquest document va dirigit als següents perfils:

- Programador. Per conèixer l'ús del servei.
- Arquitecte. Per conèixer quins són els components i la configuració del servei.
- Administrador. Per conèixer cóm configurar el servei en cadascun dels entorns en cas de necessitat.

1.5. Documents i Fonts de Referència

[1] Hessian http://www.caucho.com/hessian/

[2] Spring i Hessian http://static.springframework.org/spring/docs/1.2.x/reference/remoting.html#d0e12449

1.6. Glossari

2. Descripció Detallada

2.1. Arquitectura i Components

Actualment, el Servei HTTP es basa en Hessian, un protocol *open-source* binari sobre HTTP.
Per a més informació sobre Hessian, veure la plana http://www.caucho.com/hessian/

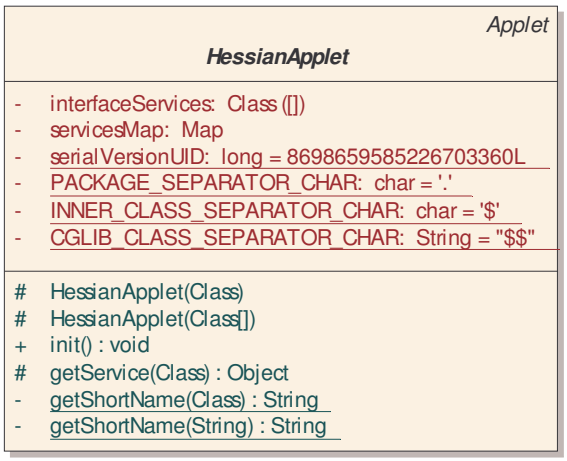
2.1.1. Procés de Comunicació amb el Servidor

En el següent gràfic es mostra quin és el flux d'una petició des d'un Applet fins un servei funcional (anomenat 'Business Service') que s'hagi publicat.

1. En primer lloc, des de la pàgina HTML es fa una crida a l'applet o bé la própia interfície de l'applet rep la petició de l'usuari. Tots els mètodes públics de l'applet són accessibles des de la pàgina que el conté
2. La inicialització de l'applet Java ha permés d'obtenir una referència al servei "Business Service" per mitjà del protocol Hessian. El jar de l'applet ha de contenir una còpia de la interfície "Business Service" per tal que el protocol Hessian pugui funcionar.
3. El Servlet "remoting " rep la petició i la remet al contenidor de Spring.
4. El contenidor de Spring injecta la implementació de la interfície "Business Service" en el servei
5. La resposta de la crida és retornada a l'Applet iniciador. Aquest ha realitzat la crida a la interfície del servei sense saber que realment s'ha realitzat una crida remota.

openFrame proporciona una classe abstracta per desenvolupar Applets basats en aquest protocol. Aquesta classe és 'net.opentrends.openframe.services.http.HessianApplet'.

Component	Package	Descripció
HessianApplet	net.opentrends.openframe.services.http	**Classe abstracta que han d'heretar els applets que vulguin usar el servei HTTP**



Veure l'apartat 'Utilització del Servei' per a més referència.

2.2. Instal·lació i Configuració

2.2.1. Instal·lació

La instal·lació del servei requereix de la utilització de la llibreria 'openFrame-services-http' i les dependències indicades a l'apartat 'Introducció - Versions i Dependències'.

2.2.2. Configuració

La configuració del Servei d'Accés HTTP implica 3 pasos:

1) Definir el servlet remot que rep les peticions i encapsula i desencapsula les dades rebudes
2) Definir els serveis que es volen exportar

Definició del Servlet Remot

```
<servlet
      ...>
<servlet-mapping>
```

Fitxer de configuració: web.xml

Ubicació: <PROJECT_ROOT>/src/main/webapp/WEB-INF

Definir el següent codi:

```
<servlet>
        <servlet-name>remoting</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-
class>
        <load-on-startup>1</load-on-startup>
</servlet>
```

```
<servlet-mapping>
        <servlet-name>remoting</servlet-name>
        <url-pattern>/remoting/*</url-pattern>
</servlet-mapping>
```

Definició dels Serveis a Exportar

```
<bean id=""
      ...>
```

Fitxer de configuració: remoting-servlet.xml

Ubicació: <PROJECT_ROOT>/src/main/webapp/WEB-INF

En el fitxer "remoting-servlet.xml", definirem tots els serveis funcionals que volem exportar. Per a cada servei, definirem dos beans, un bean que implementa la interfície del servei i un altre bean per a exportar el servei amb la classe HessianServiceExporter.

Per cadascun dels serveis que volguem publicar definirem un bean amb la següent informació:

- name. Url (amb '/' inicial) que identificarà el servei des de clients HTTP
- class. Usar el valor 'org.springframework.remoting.caucho.HessianServiceExporter'

Per aquesta classe configurarem les següents propietats:

Propietat	Requerit	Descripció
service	Sí	Referència al bean implementació la interfície del servei
serviceInterface	Sí	Indicar en el value la classe implementació específica que es vol usar de la interfície definida en el bean indicat a 'service'

Exemple:

```
<bean id="serverHelloWorldService"
class="net.opentrends.samples.http.business.impl.HelloWorldServiceServerImpl"/>

<bean     name="/HelloWorldService"
          class="org.springframework.remoting.caucho.HessianServiceExporter">
   <property name="service" ref="remoteHelloWorldService" />
   <property name="serviceInterface"
   value="net.opentrends.samples.http.business.HelloWorldService" />
</bean>
```

Aquest servei serà exportat per Spring amb la url següent:
http://servidor:8080/remoting/HelloWorldService

2.3. Utilització del Servei

La Utilització del Servei, una vegada realitzada la configuració, implica els següents pasos:

1) Crear el Applet
2) Crear el Jar del Applet
3) Crear la plana que contindrà el Applet i incorporar el tag <applet>
4) Especificar la comunicació amb Javascript des de de la plana al Applet contingut

2.3.1. Creació del Applet

Per a usar el Servei realitzarem els següents pasos:

1) Crear una classe que hereti de la classe net.opentrends.opentrends.services.http.HessianApplet.
2) Definir com a a atribut la interfície del servei que volem cridar
3) Definir el constructor buit i cridar al super amb el nom de la classe de la interfície del servei
4) Sobreescriure el mètode init() de la classe Applet per a inicialitzar els camps de tipus servei. En aquest mètode cridar al super.init del pare i inicialitzar la variable que conté el servei cridant el mètode 'getService(nomInterficie)' (on nomInterficie correspon a la interfície del servei)
5) Si el Applet és accedit des d'una funció Javascript de la pàgina definirem una funció públic amb el pas dels paràmetres. Aquesta funció farà la crida al servei.

Exemple:

```
import net.opentrends.samples.http.business.HelloWorldService;
import net.opentrends.openframe.services.http.HessianApplet;


public class HelloWorldApplet extends HessianApplet❶
{

    private HelloWorldService service❷;

    public HelloWorldApplet()
    {
        super(HelloWorldService.class)❸;
    }

    public void init() {
      super.init();
      this.service =
(HelloWorldService)super.getService(HelloWorldService.class)❹;
    }

    public String sayHello(String yourName)❺
    {
        return service.sayHello(yourName);
    }

}
```

2.3.2. Creació del Jar del Applet

La creació del jar del Applet és necessària per incloure les classes necessàries del Servei d'Accés HTTP. Cal incloure les llibreries següents:

- hessian-x.x.x.jar
- openFrame-services-http-x.x.jar

On x.x.x correspon a la versió utilitzada (veure l'apartat 'Versions i Dependències').

Aquestes llibreries les ubicarem al directori 'lib' del projecte per tal de crear el jar.

A continuació s'ofereix un exemple de fitxer de Ant per a crear aquest jar:

```
<?xml version="1.0"?>
<project name="Hessian" default="deploy" basedir=".">

   <property name="src.dir" value="src/main/java" />

   <property name="lib.dir" value="lib" />
   <property name="target.dir" value="target/applet" />
   <property name="jar.file" value="services-client-applet.jar" />

   <path id="classpath">
          <pathelement location="${lib.dir}/hessian-3.0.1.jar" />
          <pathelement location="${lib.dir}/openFrame-services-http-1.0.jar" />
   </path>

   <target name="compile">
          <delete dir="${target}/classes"/>
          <mkdir dir="${target.dir}/classes"/>

          <javac srcdir="${src.dir}" destdir="${target.dir}/classes">
                 <classpath refid="classpath" />
                 <exclude name="**/*Impl.java" />
          </javac>
   </target>

   <target name="jar" depends="compile">
          <jar jarfile="${target.dir}/${jar.file}" basedir="${target.dir}/classes"
/>
   </target>

   <target name="deploy" depends="jar">
          <copy file="${target.dir}/${jar.file}" todir="src/main/webapp" />
          <copy todir="src/main/webapp" overwrite="yes">
                 <fileset dir="${lib.dir}">
                        <include name="*.jar" />
                 </fileset>
          </copy>
   </target>
</project>
```

2.3.3. Creació de la Plana JSP/HTML Contenidora del Applet

Usar el tag <applet> indicant les següents propietats:

Propietat	Requerit	Descripció

Propietat	Requerit	Descripció
id	Sí	Nom de l'element en el document HTML. Aquest identificador ens permetrà accedir de forma directa des de Javascript
code	Sí	Nom complet de la classe Applet
width,height	No	Tamany de l'àrea del Applet a la plana. Especificar '0,0' per fer que sigui invisible
archive	Sí	Llista de jars de l'Applet. Indicar per defecte: • services-client-applet.jar. Jar que conté les classes específiques del Applet i les interfícies dels serveis cridats • hessian-x.x.x.jar • openFrame-services-http-x.x.jar

A l'applet cal especificar com a mínim el paràmetre 'hostUrl', on s'especificarà el host en el qual es troba la implementació del servei. Per defecte usar el següent codi:

```
   <param name="hostUrl" value="<%
out.print(request.getRequestURL().toString()); %>" />
```

Mitjançant aquest codi es generarà com a url la del host des del qual s'ha descarregat la pàgina.

Exemple:

```
<applet id="service"
          code="net.opentrends.samples.http.client.HelloWorldApplet.class"
          width="0" height="0" mayscript="true"
          archive="services-client-applet.jar,hessian-3.0.1.jar,openFrame-
services-http-1.0.jar">

   <param name="hostUrl" value="<% out.print(request.getRequestURL().toString());
%>" />
</applet>
```

2.3.4. Comunicació Javascript amb el Applet

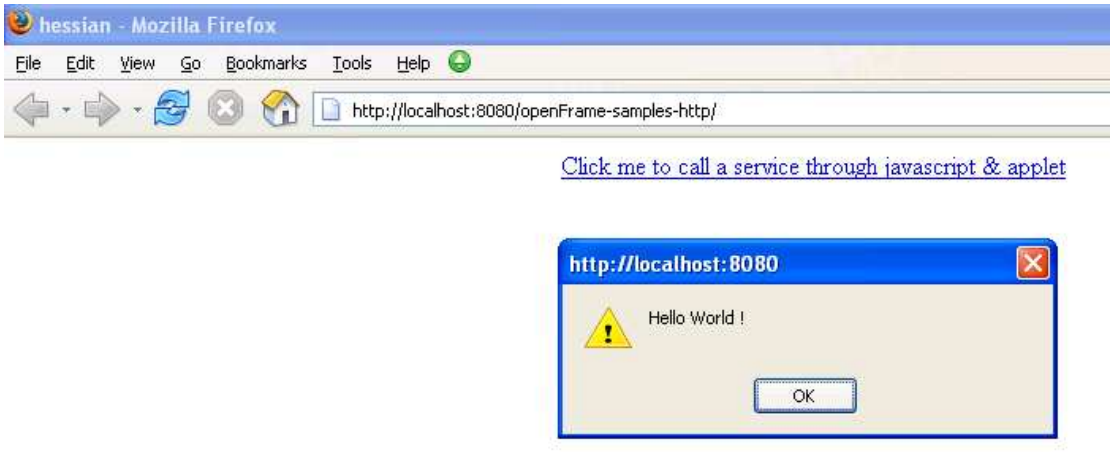
Mitjançant la funció document.getElementById() podem accedir a l'Applet java. Com a paràmetre de 'getElementById' usarem el identificador definit a la propietat 'name' del tag <applet>.

Després, tots els mètodes públics de l'Applet Java són accessibles com si l'objecte Javascript fos un objecte java normal.

Exemple:

```
<script language="JavaScript">
function helloWorld()
{
   var appletService=document.getElementById("appletService");
   alert(appletService.sayHello("World"));
}
</script>

<center><a href="javascript:helloWorld()">Click me to call a service through
javascript and an applet</a></center>
```



2.4. Eines de Suport

2.5. Integració amb Altres Serveis

2.6. Preguntes Freqüènts

3. Exemples

4. Annexos