

## Configuració de Webservices MTOM Mitjançant el Framework Canigó amb CXF

### A qui va dirigit

Aquest how-to va dirigit a tots aquells que hagin de desenvolupar una aplicació Canigó que **transfereixi dades en Base64 mitjançant webservices**.

### Versió de Canigó

Els passos descrits en aquest document són aplicables a la versió 2.3.12, 3.x i superiors del Framework de Canigó.

### Introducció

[MTOM](#) (*Message Transmission Optimization Mechanism*) es una optimització del protocol SOAP que tracta exclusivament els tipus de dades *base64Binary* de manera externa al missatge SOAP, d'aquesta manera s'estalvia aproximadament un 35% del tamany de les peticions que inclouen aquests tipus de dades.

No es recomana l'ús de MTOM sobre peticions petites amb un volum inferior a 8 KB (<http://msdn.microsoft.com/en-us/library/aa395209.aspx>) ja que el canvi de content-type que s'aplica al missatge HTTP acaba fent que el volum total del *payload* augmenti.

### Serveis Web mitjançant Apache CXF

Des de Canigó 2.3.12 s'ha utilitzat XFire com a API de Webservice. Apache CXF es la evolució d'aquesta tecnologia i es completament compatible amb el Framework Canigó 2 i 3 per la seva integració nativa amb Spring.

### Activar MTOM en un Servei Web servidor.

La configuració dun servei web en CXF té 3 parts destacades: Interfície d'Operacions, Implementació del Servei i l'arxiu de Configuració. A continuació s'indiquen els canvis que s'ha de realitzar en cadascun d'aquests elements.

1. Els canvis a la Interfície de les Operacions es redueixen a transformar el que anteriorment era un `byte[]` en un objecte del tipus `DataHandler` i afegir l'anotació `XmlMimeType` a aquest paràmetre. A continuació s'inclou un exemple del mètode:

```
...
@WebMethod(operationName="aplanarPDF")
@WebResult(name="Result")
public Result flatteningPDF(
    @WebParam(name="configCall")
        ConfigCall config,
    @WebParam(name="pdf", targetNamespace="http://eforms.canigo.ctti.gencat.cat")
    @XmlMimeType("application/octet-stream")
        DataHandler pdf);
...
```

Figura 1

## Configuració de Webservices MTOM Mitjançant el Framework Canigó amb CXF

- Un cop realitzat això, la Implementació del Servei passarà a utilitzar l'objecte `DataHandler` que es pot convertir fàcilment com s'indica a continuació:

```
import org.apache.commons.io.IOUtils;

...

@Trace
//public Result flatteningPDF(ConfigCall config, byte[] pdf) {
public Result flatteningPDF(ConfigCall config, DataHandler pdf) {
    log.info("ServeisSOAPImpl.flatteningPDF - INICI");

    //Result result = output.getFlattenPDF(pdf);
    byte[] byteArray = IOUtils.toByteArray(pdf.getInputStream());
    Result result = output.getFlattenPDF(byteArray);

    return result;
}
```

Figura 2

- A nivell de configuració només cal afegir la propietat `mtom-enabled` amb valor `true` al bean del endpoint del servei del contexte de Spring:

```
...
<!-- Service endpoint -->
<jaxws:endpoint id="serveisSOAP12"
    implementorClass="cat.gencat.ctti.canigo.eforms.services.impl.ServeisSOAPImpl"
    implementor="#serveisSOAPImpl"
    address="/serveis/ServeisInvocacio"
    >
    <jaxws:serviceFactory>
        <ref bean="jaxws-and-aegis-service-factory"/>
    </jaxws:serviceFactory>
    <jaxws:binding>
        <soap:soapBinding version="1.2" />
    </jaxws:binding>
    <jaxws:properties>
        <entry key="mtom-enabled" value="true"/>
    </jaxws:properties>
</jaxws:endpoint>
...
```

Figura 3

## Configuració de Webservices MTOM Mitjançant el Framework Canigó amb CXF

### Activar MTOM en un Servei Web client.

Per tal de realitzar crides amb MTOM a un servei web mitjançant un client generat amb CXF només cal activar la propietat MTOMEnabled del SOAPBinding del servei.

```
...
private static final QName SERVICE_NAME = new QName(
    "http://eforms.canigo.ctti.gencat.cat",
    "ServeisSOAPImplService");

public static void main(String args[]) throws Exception {
    URL wsdlURL = ServeisSOAPImplService.WSDL_LOCATION;
    if (args.length > 0) {
        File wsdlFile = new File(args[0]);
        try {
            if (wsdlFile.exists()) {
                wsdlURL = wsdlFile.toURI().toURL();
            } else {
                wsdlURL = new URL(args[0]);
            }
        } catch (MalformedURLException e) {
            e.printStackTrace();
        }
    }

    ServeisSOAPImplService ss = new ServeisSOAPImplService(
        wsdlURL, SERVICE_NAME);
    IServeisSOAP port = ss.getServeisSOAPImplPort();

    BindingProvider bp = (BindingProvider) port;
    SOAPBinding binding = (SOAPBinding) bp.getBinding();
    binding.setMTOMEnabled(true);

    System.out.println("Invoking aplanarPDF...");
    cat.gencat.ctti.canigo.eforms.services.objects.ConfigCall configCall = null;
    byte[] pdf = new byte[0];
    cat.gencat.ctti.canigo.eforms.services.objects.Result return =
    port.flattenPDF(configCall, pdf);
    System.out.println("aplanarPDF.result=" + return);
}
```

Figura 4