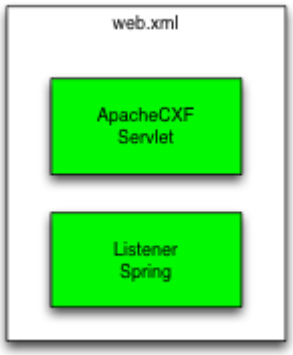
En este post vamos a configurar Apache CXF para que sea capaz de publicar un servicio web REST . Para ello necesitaremos realizar varias operaciones . En primer lugar modificar el fichero web.xml para añadir el servlet de Apache CXF y los listener de Spring .Ya que recordemos Apache CXF se apoya de manera fuerte en este framework.

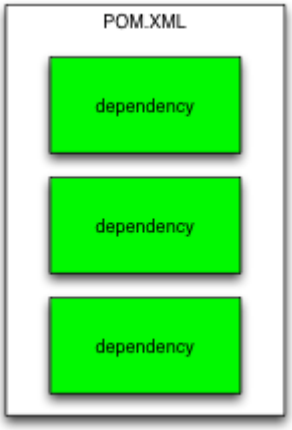


A continuación se muestra el contenido de este fichero

```
&amp;amp;lt;?xml version="1.0" encoding="UTF-8"?&amp;amp;gt;
&amp;amp;lt;web-app version="2.5"
xmlns="http://java.sun.com/xml/ns/javaee"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"&amp;amp;gt;
 &amp;amp;lt;listener&amp;amp;gt;
 &amp;amp;lt;listener-
class&amp;amp;gt;org.springframework.web.context.ContextLoaderListener
&amp;amp;lt;/listener-class&amp;amp;gt;
 &amp;amp;lt;/listener&amp;amp;gt;

&amp;amp;lt;servlet&amp;amp;gt;
 &amp;amp;lt;servlet-name&amp;amp;gt;CXFServlet&amp;amp;lt;/servlet-
name&amp;amp;gt;
```

```
 &amp;amp;lt;servlet-
class&amp;amp;gt;org.apache.cxf.transport.servlet.CXFServlet&amp;amp;l
t;/servlet-class&amp;amp;gt;
 &amp;amp;lt;load-on-startup&amp;amp;gt;1&amp;amp;lt;/load-on-
startup&amp;amp;gt;
 &amp;amp;lt;/servlet&amp;amp;gt;

&amp;amp;lt;servlet-mapping&amp;amp;gt;
 &amp;amp;lt;servlet-name&amp;amp;gt;CXFServlet&amp;amp;lt;/servlet-
name&amp;amp;gt;
 &amp;amp;lt;url-pattern&amp;amp;gt;/rest/*&amp;amp;lt;/url-
pattern&amp;amp;gt;
 &amp;amp;lt;/servlet-mapping&amp;amp;gt;
&amp;amp;lt;/web-app&amp;amp;gt;
```

Una vez tenemos configurado el fichero web.xml necesitaremos configurar también las dependencias en este caso nos apoyaremos en Maven y en un fichero pom.xml.



Vamos a ver el contenido a nivel de dependencias del fichero POM.

<dependencies>

<dependency>

<groupId>org.springframework</groupId
>

<artifactId>spring-
core</artifactId>

<version>3.2.2.RELEASE</version&a
mp;gt;

</dependency>

<dependency>

<groupId>org.springframework</groupId
>

<artifactId>spring-
context</artifactId>

<version>3.2.2.RELEASE</version&a
mp;gt;

</dependency>

<dependency>

<groupId>org.springframework</groupId

>

<artifactId>spring-web</artifactId>

<version>3.2.2.RELEASE</version>

</dependency>

<dependency>

<groupId>org.apache.cxf</groupId>

<artifactId>cxf-rt-frontend-jaxws</artifactId>

<version>2.7.4</version>

</dependency>

<dependency>

<groupId>org.apache.cxf</groupId>

<artifactId>cxf-rt-transports-http</artifactId>

<version>2.7.4</version>

</dependency>

<dependency>

<groupId>org.apache.cxf</groupId>

<artifactId>cxf-rt-frontend-jaxrs</artifactId>

<version>2.7.4</version>

</dependency>

<dependency>

<groupId>org.apache.cxf</groupId>

<artifactId>cxf-rt-rs-extension-providers</artifactId>

<version>2.6.0</version>

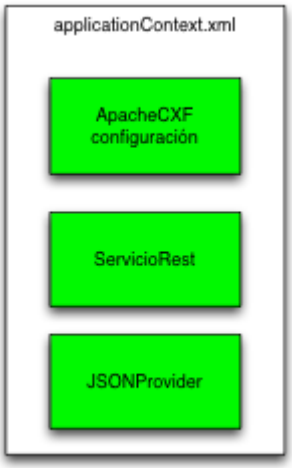
</dependency>

<dependency>

<groupId>javax.xml.bind</groupId>

```
&amp;amp;lt;artifactId&amp;amp;gt;jaxb-
api&amp;amp;lt;/artifactId&amp;amp;gt;

&amp;amp;lt;version&amp;amp;gt;2.2.9&amp;amp;lt;/version&amp;amp;gt;

&amp;amp;lt;/dependency&amp;amp;gt;

&amp;amp;lt;dependency&amp;amp;gt;

&amp;amp;lt;groupId&amp;amp;gt;org.codehaus.jackson&amp;amp;lt;/groupI
d&amp;amp;gt;

&amp;amp;lt;artifactId&amp;amp;gt;jackson-
jaxrs&amp;amp;lt;/artifactId&amp;amp;gt;

&amp;amp;lt;version&amp;amp;gt;1.9.0&amp;amp;lt;/version&amp;amp;gt;

&amp;amp;lt;/dependency&amp;amp;gt;

&amp;amp;lt;/dependencies&amp;amp;gt;
```

Realizadas estas operaciones nos queda configurar el fichero de Spring para que pueda funcionar con servicios REST sin problemas



El contenido del fichero de applicationContext.xml será el siguiente

```
&amp;amp;lt;?xml version="1.0" encoding="UTF-8"?&amp;amp;gt;
&amp;amp;lt;beans xmlns="http://www.springframework.org/schema/beans"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:jaxrs="http://cxf.apache.org/jaxrs"
 xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://cxf.apache.org/jaxrs
http://cxf.apache.org/schemas/jaxrs.xsd"&amp;gt;
&amp;amp;lt;import resource="classpath:META-INF/cxf/cxf.xml"
/&amp;amp;gt;
 &amp;amp;lt;import resource="classpath:META-INF/cxf/cxf-servlet.xml"
/&amp;amp;gt;
&amp;amp;lt;jaxrs:server id="servicioRest" address="/"&amp;amp;gt;
 &amp;amp;lt;jaxrs:serviceBeans&amp;amp;gt;
 &amp;amp;lt;bean
class="com.arquitecturajava.serviciosexternos.ServicioPersonas"
/&amp;amp;gt;
```

```
 &amp;amp;lt;/jaxrs:serviceBeans&amp;amp;gt;
 &amp;amp;lt;jaxrs:providers&amp;amp;gt;
 &amp;amp;lt;ref bean="jsonProvider"/&amp;amp;gt;
 &amp;amp;lt;/jaxrs:providers&amp;amp;gt;
 &amp;amp;lt;jaxrs:features&amp;amp;gt;
 &amp;amp;lt;/jaxrs:features&amp;amp;gt;
 &amp;amp;lt;/jaxrs:server&amp;amp;gt;
 &amp;amp;lt;bean id="jsonProvider"
class="org.codehaus.jackson.jaxrs.JacksonJsonProvider"/&amp;amp;gt;
&amp;amp;lt;/beans&amp;amp;gt;
```

Como podemos ver en un primer lugar aparecen las etiquetas <import> que se encargan de dar de alta los ficheros de configuración de Apache CXF.

```
&amp;amp;lt;import resource="classpath:META-INF/cxf/cxf.xml"
/&amp;amp;gt;
&amp;amp;lt;import resource="classpath:META-INF/cxf/cxf-servlet.xml"
/&amp;amp;gt;
```

Una vez tenemos esto dado de alta añadimos un servidor de JAXRS y el bean que deseamos gestionar a traves de él.

```
&amp;amp;lt;jaxrs:server id="servicioRest" address="/"&amp;amp;gt;
 &amp;amp;lt;jaxrs:serviceBeans&amp;amp;gt;
&amp;amp;lt;bean
class="com.arquitecturajava.serviciosexternos.ServicioPersonas"/&amp;a
mp;gt;
 &amp;amp;lt;/jaxrs:serviceBeans&amp;amp;gt;
```

```
&amp;amp;lt;jaxrs:providers&amp;amp;gt;
&amp;amp;lt;ref bean="jsonProvider"/&amp;amp;gt;
&amp;amp;lt;/jaxrs:providers&amp;amp;gt;
&amp;amp;lt;jaxrs:features&amp;amp;gt;
&amp;amp;lt;/jaxrs:features&amp;amp;gt;
&amp;amp;lt;/jaxrs:server&amp;amp;gt;
```

Por último añadimos un proveedor de JSON para que sea capaz de transformar una lista de personas a una estructura JSON sin problemas

```
&amp;amp;lt;bean id="jsonProvider"
class="org.codehaus.jackson.jaxrs.JacksonJsonProvider" /&amp;amp;gt;
&amp;amp;lt;/beans&amp;amp;gt;
```

Realizados estos cambios tenemos Apache CXF configurado y podremos realizar las invocaciones REST necesarias.