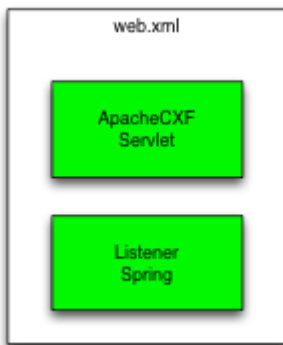


En este post vamos a configurar Apache CXF para que sea capaz de publicar un servicio web REST . Para ello necesitaremos realizar varias operaciones . En primer lugar modificar el fichero web.xml para añadir el servlet de Apache CXF y los listener de Spring .Ya que recordemos Apache CXF se apoya de manera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



A continuación se muestra el contenido de este fichero

```
&lt;?xml version="1.0" encoding="UTF-8"?>
<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">
    <listener>
    <listener-
class>org.springframework.web.context.ContextLoaderListener
    </listener-class>
    </listener>
    <servlet>
    <servlet-name>CXFServlet</servlet-
name>
```

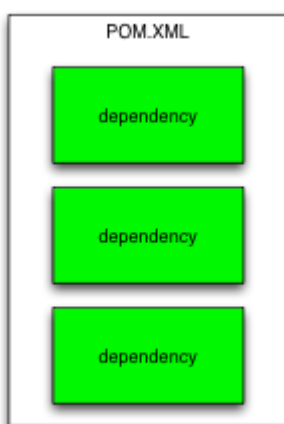
```

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

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

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>3.2.2.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

&gt;

&lt;artifactId&gt;spring-
web&lt;/artifactId&gt;

&lt;version&gt;3.2.2.RELEASE&lt;/version&a
mp;gt;

&lt;/dependency&gt;

&lt;dependency&gt;

&lt;groupId&gt;org.apache.cxf&lt;/groupId&a
mp;gt;

&lt;artifactId&gt;cxf-rt-frontend-
jaxws&lt;/artifactId&gt;

&lt;version&gt;2.7.4&lt;/version&gt;

&lt;/dependency&gt;

&lt;dependency&gt;

&lt;groupId&gt;org.apache.cxf&lt;/groupId&a
mp;gt;

&lt;artifactId&gt;cxf-rt-transport-
http&lt;/artifactId&gt;

&lt;version&gt;2.7.4&lt;/version&gt;

</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>

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

```
&lt;version&gt;2.2.9&lt;/version&gt;
```

```
&lt;/dependency&gt;
```

```
&lt;dependency&gt;
```

```
&lt;groupId&gt;org.codehaus.jackson&lt;/groupId&gt;
```

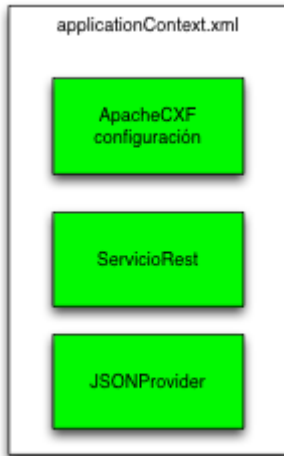
```
&lt;artifactId&gt;jackson-  
jaxrs&lt;/artifactId&gt;
```

```
&lt;version&gt;1.9.0&lt;/version&gt;
```

```
&lt;/dependency&gt;
```

```
&lt;/dependencies&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

```
&lt;?xml version="1.0" encoding="UTF-8"?>
&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">
  &lt;import resource="classpath:META-INF/cxf/cxf.xml"
/>
  &lt;import resource="classpath:META-INF/cxf/cxf-servlet.xml"
/>
  &lt;jaxrs:server id="servicioRest" address="/">
    &lt;jaxrs:serviceBeans>
      &lt;bean
class="com.arquitecturajava.servicioexternos.ServicioPersonas"
/>
    &lt;/jaxrs:serviceBeans>
  &lt;/jaxrs:server>
```

```

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

```

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

```

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

```

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

```

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

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

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

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