

## 3.12.2 Exposer le service REST

### Sommaire :

- [Configuration](#)
  - [Maven](#)
  - [Spring](#)

### Configuration

#### Maven

Le module métier de votre application doit contenir la dépendance suivante :

```
<dependency>
  <groupId>org.esupportail</groupId>
  <artifactId>esup-commons2-rs-cxf</artifactId>
  <version>${esupcommons.version}</version>
  <type>pom</type>
</dependency>
```

#### Spring

Afin d'exposer la couche service sous forme de service REST il y a quelques éléments de configuration CXF à mettre dans la configuration Spring de la couche métier (**properties/domain/domain.xml**) :

```
<import resource="classpath:META-INF/cxf/cxf.xml" />
<import resource="classpath:META-INF/cxf/cxf-servlet.xml" />

<jaxrs:server id="domainServiceRest" address="/rest">
  <jaxrs:serviceBeans>
    <ref bean="domainService" />
  </jaxrs:serviceBeans>
  <jaxrs:providers>
    <ref bean="jsonProvider" />
  </jaxrs:providers>
</jaxrs:server>

<bean id="jsonProvider" class="org.codehaus.jackson.jaxrs.JacksonJaxbJsonProvider" />
```

Explications :

- Les imports spécifiques à CXF. A noter l'utilisation du préfixe **classpath:** qui permet de rechercher les fichier XML dans les jar du projet (celui de **CXF** ici)
- L'espace de nom **jaxrs:** est défini dans la balise racine du fichier *Spring* :

```
xmlns:jaxrs="http://cxf.apache.org/jaxrs"
xsi:schemaLocation=".../...
  http://cxf.apache.org/jaxrs http://cxf.apache.org/schemas/jaxrs.xsd"
```

- La balise **jaxrs:server** définit un point d'entrée (il est possible d'en définir plusieurs) de type REST. La balise a une propriété **address** qui sera à utiliser dans l'URL d'accès au service (Ex : `/cxf/rest/domainService/users`)  
La balise a comme enfants :
  - Une balise **jaxrs:serviceBeans** qui permet de référencer le bean correspondant au service métier à exposer
  - Une balise **jaxrs:providers** qui pointe vers le bean servant à sérialiser sous forme JSON les objets Java