# Apache CXF Client

**Overview**

[Apache CXF](http://cxf.apache.org/) is a JAX-WS fully compliant framework.

On top of features defined by JAX-WS standards, Apache CXF provides the capability of conversion between WSDL and Java classes, APIs used to manipulate raw XML messages, the support for JAX-RS, integration with the Spring Framework, etc.

**Requirement of JAX-WS client**

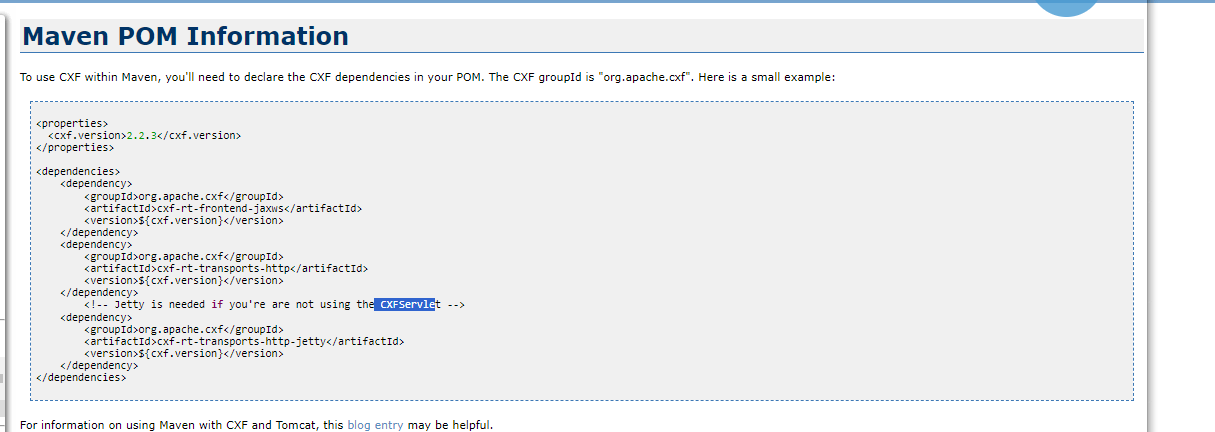
1. Whenever we need to consume existing soap service.
2. Whenever we want to create wrapper service. Where user will call Rest API for getting response and server internally will call soap services to get the response based on given request body.

Apache CFX Website Important Link

<https://cxf.apache.org/docs/how-do-i-develop-a-client.html>

<https://cxf.apache.org/docs/using-cxf-with-maven.html>

<https://cxf.apache.org/docs/maven-cxf-codegen-plugin-wsdl-to-java.html>



**Note:-**  We followed here codegen-plugin wsdl2java with xjc extension.

**Prerequisites**

**:-** As we are generating a client so there should be a server. For this example I used the SOAP-UI exposed service known as calculator service

<URL:-> <http://www.dneonline.com/calculator.asmx?wsdl>

**Project Setup**

Pom.xml

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.7.16</version>

<relativePath /> <!-- lookup parent from repository -->

</parent>

<groupId>com.javainformatic</groupId>

<artifactId>cfxclinet</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>cfxclinet</name>

<description>Apache cfx client example</description>

<properties>

<java.version>1.8</java.version>

<cxf.version>2.2.3</cxf.version>

<cxf-xjc.version>2.6.0</cxf-xjc.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

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

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

<version>${cxf.version}</version>

</dependency>

<dependency>

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

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

<version>${cxf.version} </version>

</dependency>

<!-- Jetty is needed if you're are not using the CXFServlet -->

<dependency>

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

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

<version>${cxf.version}</version>

</dependency>

<dependency>

<groupId>org.apache.cxf.xjc-utils </groupId>

<artifactId>cxf-xjc-runtime </artifactId>

<version>${cxf-xjc.version}</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

<plugin>

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

<artifactId>cxf-codegen-plugin</artifactId>

<version>${cxf.version}</version>

<executions>

<execution>

<id>generate-sources</id>

<phase>generate-sources</phase>

<configuration>

<sourceRoot>${basedir}/src/main/java</sourceRoot>

<wsdlOptions>

<wsdlOption>

<wsdl>

${basedir}/src/main/resources/wsdl/calculator.wsdl</wsdl>

<extraargs>

<extraarg>-xjc-Xts</extraarg>

</extraargs>

</wsdlOption>

</wsdlOptions>

</configuration>

<goals>

<goal>wsdl2java</goal>

</goals>

</execution>

</executions>

<dependencies>

<dependency>

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

<artifactId>cxf-xjc-ts</artifactId>

<version>${cxf-xjc.version}</version>

</dependency>

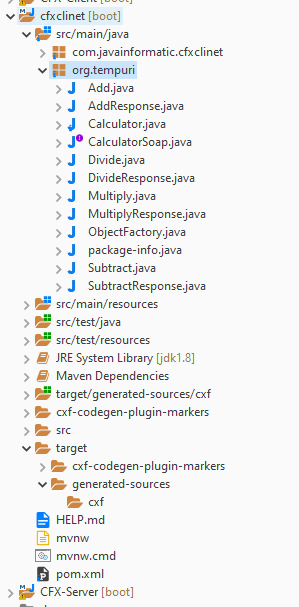
</dependencies>

</plugin>

</plugins>

</build>

</project>



CalculatorClient.java

**package** com.javainformatic.cfxclinet;

**import** java.net.MalformedURLException;

**import** java.net.URL;

**import** org.tempuri.Calculator;

**import** org.tempuri.CalculatorSoap;

**public** **class** CalculatorClient {

**public** **static** **void** main(String[] args) {

**try** {

Calculator calculator = **new** Calculator(**new** URL("http://www.dneonline.com/calculator.asmx?wsdl"));

CalculatorSoap calculatorSoap = calculator.getCalculatorSoap();

**int** sum = calculatorSoap.add(2, 3);

System.***out***.println("sum : "+ sum);

**int** multiply = calculatorSoap.multiply(2, 3);

System.***out***.println("multiply :" + multiply);

} **catch** (MalformedURLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}