CXF SOAP Without Spring

By admin | October 03, 2012 | Updated : March 27, 2016 🏷️ *cxf, soap,* 💬 *5 Comments*

CXF SOAP Without Spring explains about step by step details of Creating / Developing Web service using Apache CXF without any spring dependencies

Apache CXF is a free and open source project, and a fully featured Webservice framework. It helps you building webservices using different front-end API's, like as JAX-RS and JAX-WS.

Most of the tutorial related to Apache CXF is developed using spring framework, this is because of easy integration with spring. But it is not a necessity.

In this, we are extending **CXFNonSpringServlet** in order to remove spring dependency.
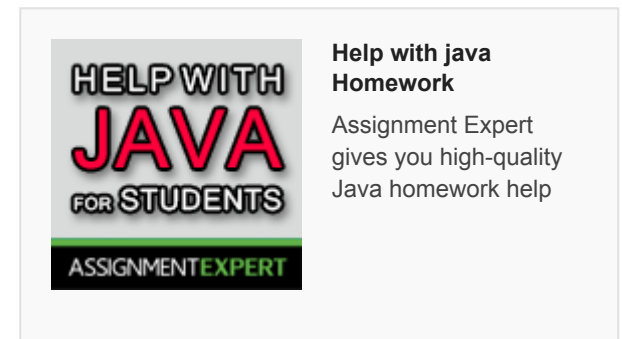
Required Libraries

You need to download

1. JDK 6
2. Eclipse 3.7
3. CXF-2.7.3
4. Tomcat 7

Following jar must be in ClassPath

1. commons-logging-1.1.1.jar
2. cxf-2.7.3.jar
3. httpasyncclient-4.0-beta3.jar
4. httpclient-4.2.1.jar
5. httpcore-4.2.2.jar
6. httpcore-nio-4.2.2.jar
7. neethi-3.0.2.jar
8. wsdl4j-1.6.2.jar
9. xmlschema-core-2.0.3.jar

CXF Without Spring

I am creating a sample web service project that pass Student object and return with some changes on that object. The service is using simple POJO (Plain Old Java Object) bean.

Firstly create a Dynamic Web Project (**File**->**New**->**Dynamic Web Project**) named "CXFTutorial" according to following screenshot

New	Alt+Shift+N	▶		Web Application Project
Open File...				JPA Project
Close	Ctrl+W			Enterprise Application Project
Close All	Ctrl+Shift+W			Dynamic Web Project
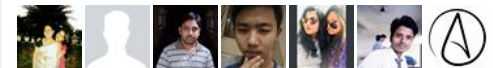
Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name: CXFTutorial

Project location

☑ Use default location

Location: C:\Documents and Settings\Melbin\workspace\CXFTutorial   Browse...

Target runtime

Apache Tomcat v7.0   ⬍   New Runtime...

Dynamic web module version

3.0   ⬍

Configuration

Default Configuration for Apache Tomcat v7.0   ⬍   Modify...

A good starting point for working with Apache Tomcat v7.0 runtime. Additional facets can later be installed to add new functionality to the project.

EAR membership

☐ Add project to an EAR

EAR project name: EAR   ⬍   New Project...

Working sets

☐ Add project to working sets

Working sets: ⬍   Select...

⦿   < Back   Next >   Finish   Cancel

```java
package com.student;

public class Student {
  private String name;
  public String getName() {
    return name;
  }
  public void setName(String name) {
    this.name = name;
  }
}
```

Related Posts

- CXF With JBoss Tutorial
- Trace SOAP request/response using JAX-WS
- Apache CXF And Embedded Tomcat
- CXF Webservice Using Embedded Jetty
- Add HTTP Headers To SOAP Request Using CXF
- CXF WS-Discovery Client
- CXF WS-Discovery Example
- CXF Asynchronous Client
- Asynchronous Web Service Using CXF
- CXF java.net.SocketTimeoutException: Read timed out
- Consume JAX-WS Web Service Using Eclipse

- Create JAX-WS Web Service Using Eclipse
- Mock Web Service With soapUI
- SoapUI Tutorial For Testing Web Service (REST & SOAP)
- PrettyPrint With CXF
- Download Text File Using CXF REST / JAX-RS
- Download Image File Using CXF REST / JAX-RS
- Download Excel File Using CXF REST / JAX-RS
- Download DOC File Using CXF REST / JAX-RS
- Download PDF File Using CXF REST / JAX-RS

You might also like following tutorials :

5 Responses to "CXF SOAP Without Spring"

Roman

2012-10-04 08:10:47.0

Great tutorial. Thanks. I used it in my application and got this error

SRVE0242I: [oLine] [/oline22] [SimpleCXFNonSpringServlet]: Initialization successful.
[11/6/14 14:39:23:956 CET] 00000026 E UOW=null source=com.ibm.ws.webcontainer.servlet class=com.ibm.ws.webcontainer.servlet.ServletWrapper method=service org=null prod=null component=null thread=[WebContainer : 0]
SRVE0068E: Uncaught exception created in one of the service methods of the servlet SimpleCXFNonSpringServlet in application oLine. Exception created : java.lang.NullPointerException
at org.apache.cxf.transport.servlet.ServletController.invoke(ServletController.java:15

at org.apache.cxf.transport.servlet.ServletController.invoke(ServletController.java:14

at org.apache.cxf.transport.servlet.CXFNonSpringServlet.invoke(CXFNonSpringServl

at org.apache.cxf.transport.servlet.AbstractHTTPServlet.handleRequest(AbstractHTT

at javax.servlet.http.HttpServlet.service(HttpServlet.java:718)
at
org.apache.cxf.transport.servlet.AbstractHTTPServlet.service(AbstractHTTPServle

at
com.ibm.ws.webcontainer.servlet.ServletWrapper.service(ServletWrapper.java:16

at
com.ibm.ws.webcontainer.servlet.ServletWrapper.service(ServletWrapper.java:15

at
com.ibm.ws.webcontainer.filter.WebAppFilterChain.doFilter(WebAppFilterChain.j

at
com.ibm.ws.webcontainer.filter.WebAppFilterChain._doFilter(WebAppFilterChain.

at
com.ibm.ws.webcontainer.filter.WebAppFilterManager.doFilter(WebAppFilterMar

at
com.ibm.ws.webcontainer.servlet.ServletWrapper.handleRequest(ServletWrappe

at
com.ibm.ws.webcontainer.servlet.ServletWrapper.handleRequest(ServletWrappe

at

```
at
com.ibm.ws.webcontainer.webapp.WebApp.handleRequest(WebApp.java:3954)
at
com.ibm.ws.webcontainer.webapp.WebGroup.handleRequest(WebGroup.java:27

at
com.ibm.ws.webcontainer.WebContainer.handleRequest(WebContainer.java:942

at
com.ibm.ws.webcontainer.WSWebContainer.handleRequest(WSWebContainer.ja

at
com.ibm.ws.webcontainer.channel.WCChannelLink.ready(WCChannelLink.java:18

at
com.ibm.ws.http.channel.inbound.impl.HttpInboundLink.handleDiscrimination(H

at
com.ibm.ws.http.channel.inbound.impl.HttpInboundLink.handleNewRequest(Htt

at
com.ibm.ws.http.channel.inbound.impl.HttpInboundLink.processRequest(HttpIn

at
com.ibm.ws.http.channel.inbound.impl.HttpInboundLink.ready(HttpInboundLink

at
```

at com.ibm.ws.tcp.channel.impl.NewConnectionInitialReadCallback.complete(NewC

at com.ibm.ws.tcp.channel.impl.AioReadCompletionListener.futureCompleted(AioR

at com.ibm.io.async.AbstractAsyncFuture.invokeCallback(AbstractAsyncFuture.java:

at com.ibm.io.async.AsyncChannelFuture.fireCompletionActions(AsyncChannelFutu

at com.ibm.io.async.AsyncFuture.completed(AsyncFuture.java:138)
at com.ibm.io.async.ResultHandler.complete(ResultHandler.java:204)
at com.ibm.io.async.ResultHandler.runEventProcessingLoop(ResultHandler.java:775

at com.ibm.io.async.ResultHandler$2.run(ResultHandler.java:905)
at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:1646)

Could anyone tell me how to fix it? I have no idea why it does not work. I deployed the application on WAS.

admin

Sorry, I am too clueless & can't help you as I don't have WebSphere application server for deploy it. This tutorial is only tested wilth Tomcat.

A better choice will be asking to CXF forum mailing list. See the link below

http://cxf.547215.n5.nabble.com/cxf-user-f547216.html

Roman

2012-10-06 08:10:47.0

Thanks for quick reply. I would like to try your example. Could you give me the wsdl file please?

Roman

2012-10-07 08:10:47.0

When I put your example on Tomcat, following error occures:

javax.servlet.ServletException: Servlet.init() for servlet SimpleCXFNonSpringServlet threw exception org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.ja

org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:100)

org.apache.coyote.http11.AbstractHttp11Processor.process(AbstractHttp11Proce

org.apache.coyote.AbstractProtocol$AbstractConnectionHandler.process(Abstrac

org.apache.tomcat.util.net.JIoEndpoint$SocketProcessor.run(JIoEndpoint.java:313

java.util.concurrent.ThreadPoolExecutor$Worker.runTask(ThreadPoolExecutor.ja

java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:9(

java.lang.Thread.run(Thread.java:662)
root cause

java.lang.IllegalArgumentException: Cannot create URL for this address
/ChangeStudent
com.sun.xml.internal.ws.transport.http.server.EndpointImpl.publish(EndpointImp

com.sun.xml.internal.ws.spi.ProviderImpl.createAndPublishEndpoint(ProviderImp

javax.xml.ws.Endpoint.publish(Endpoint.java:170)
com.student.SimpleCXFNonSpringServlet.loadBus(SimpleCXFNonSpringServlet.ja

org.apache.cxf.transport.servlet.CXFNonSpringServlet.init(CXFNonSpringServlet.ja

org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.ja

org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:100)

```
org.apache.coyote.http11.AbstractHttp11Processor.process(AbstractHttp11Proce

org.apache.coyote.AbstractProtocol$AbstractConnectionHandler.process(Abstrac

org.apache.tomcat.util.net.JIoEndpoint$SocketProcessor.run(JIoEndpoint.java:313

java.util.concurrent.ThreadPoolExecutor$Worker.runTask(ThreadPoolExecutor.ja

java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:90

java.lang.Thread.run(Thread.java:662)
```

sanjeev

2012-10-08 08:10:47.0

@Roman,

I have created an example with CXF 2.7.12 package.

As of 2.7.12 you need too have following additional libraries

1) asm-3.3.1.jar
2) woodstox-core-asl-4.4.0.jar
3) stax2-api-3.1.4.jar

Just change the extension from zip to war. It also contains the source code and required libraries too

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Please answer this simple math question

5 + 24 =

Submit Comment

Blogroll

- Privacy Policy
- Sitemap

- Contact
- Guest Post
- Advertise

About Us

Javatips.net provides unique and complete articles about java and other related technologies

Recent Posts

- How to Avoid Liferay DXP Upgrading Pitfalls?
- 7 Reasons Why You Should Learn Python
- Anti Cross-Site Scripting (XSS) For Java Web Application
- Iterate HashMap Using Java
- Trace SOAP message Using Eclipse IDE
- Jersey Bean Validation Example
- Convert Java InputStream to String
- What Is Garbage Collection Log? How To Enable & Analyze?

Connect with Us

Your email Address...   Subscribe