

# Index

## Apache CXF™: An Open-Source Services Framework

### OVERVIEW

Apache CXF™ is an open source services framework. CXF helps you build and develop services using frontend programming APIs, like JAX-WS and JAX-RS. These services can speak a variety of protocols such as SOAP, XML/HTTP, RESTful HTTP, or CORBA and work over a variety of transports such as HTTP, JMS or JBI.

### NEWS

#### **March 5, 2019 - Apache CXF 3.3.1 released!**

The Apache CXF team is proud to announce the availability of our latest patch release! Over 10 JIRA issues were fixed in this release.

This is mostly a patch release to fix problems and issues that users have encountered. Downloads are available [here](#).

#### **January 28, 2019 - Apache CXF 3.3.0 released!**

The Apache CXF team is proud to announce the availability of our latest release! There are several enhancements and new features such as Java 11 support, MicroProfile Rest Client 1.2, draft HTTP Signatures, etc... More details can be found in the [3.3 Migration Guide](#). Downloads are available [here](#)

#### **January 28, 2019 - Apache CXF 3.2.8 and 3.1.18 released!**

The Apache CXF team is proud to announce the availability of the latest patch releases. Over 15 JIRA issues were fixed for 3.2.8.

This is mostly a patch release to fix problems and issues that users have encountered. Downloads are available [here](#).

## April 16, 2018 - 10 years of CXF - Happy Birthday!

It's time to celebrate: 10 years ago, on April 16th in the year 2008, CXF graduated from the Apache incubator as a merge of the Objectweb Celtix project and the Codehaus XFire project (see <http://incubator.apache.org/projects/cxf.html> for more details).

Today we have lots of more features, support for additional specifications and a large user base.

Thanks to anyone who made this possible and looking forward to another 10 years :-)

### Features

CXF includes a broad feature set, but it is primarily focused on the following areas:

- **Web Services Standards Support:** CXF supports a variety of web service standards including SOAP, the WS-I Basic Profile, WSDL, WS-Addressing, WS-Policy, WS-ReliableMessaging, WS-Security, WS-SecurityPolicy, WS-SecureConversation, and WS-Trust (partial).
- **Frontends:** CXF supports a variety of "frontend" programming models.

CXF implements the JAX-WS APIs. CXF JAX-WS support includes some extensions to the standard that make it significantly easier to use, compared to the reference implementation: It will automatically generate code for request and response bean classes, and does not require a WSDL for simple cases.

It also includes a "simple frontend" which allows creation of clients and endpoints without annotations. CXF supports both contract first development with WSDL and code first development starting from Java.

For REST, CXF also supports a JAX-RS frontend.

- **Ease of use:** CXF is designed to be intuitive and easy to use. There are simple APIs to quickly build code-first services, Maven plug-ins to make tooling integration easy, JAX-WS API support, Spring 2.x XML support to make configuration a snap, and much more.
- **Binary and Legacy Protocol Support:** CXF has been designed to provide a pluggable architecture that supports not only XML but also non-XML type bindings, such as JSON and CORBA, in combination with any type of transport.

To get started using CXF, check out the downloads, the user's guide, or the mailing lists to get more information!

## GOALS

### General

- High Performance
- Extensible
- Intuitive & Easy to Use

### Support for Standards

### JSR Support

- JAX-WS - Java API for XML-Based Web Services (JAX-WS) 2.0 - JSR-224
- Web Services Metadata for the Java Platform - JSR-181
- JAX-RS - The Java API for RESTful Web Services - JSR-311
- SAAJ - SOAP with Attachments API for Java (SAAJ) - JSR-67

## WS-\* and related Specifications Support

- Basic support: WS-I Basic Profile 1.1
- Quality of Service: WS-Reliable Messaging
- Metadata: WS-Policy, WSDL 1.1 - Web Service Definition Language
- Communication Security: WS-Security, WS-SecurityPolicy, WS-SecureConversation, WS-Trust (partial support)
- Messaging Support: WS-Addressing, SOAP 1.1, SOAP 1.2, Message Transmission Optimization Mechanism (MTOM)

## OpenAPI Specification (OAS) Support

- OAS 2.0 (classic Swagger specification)
- OAS 3.0.x (new revised specification)

## Multiple Transports, Protocol Bindings, Data Bindings, and Formats

- Transports: HTTP, Servlet, JMS, In-VM and many others via the Camel transport for CXF such as SMTP/POP3, TCP and Jabber
- Protocol Bindings: SOAP, REST/HTTP, pure XML
- Data bindings: JAXB 2.x, Aegis, Apache XMLBeans, Service Data Objects (SDO), JIBX
- Formats: XML Textual, JSON, FastInfoset
- Extensibility API allows additional bindings for CXF, enabling additional message format support such as CORBA/IIOP

## Flexible Deployment

- Lightweight containers: deploy services in Jetty, Tomcat or Spring-based containers
- JBI integration: deploy as a service engine in a JBI container such as ServiceMix, OpenESB or Petals
- Java EE integration: deploy services in Java EE application servers such as Apache Geronimo, JOnAS, Redhat JBoss, OC4J, Oracle WebLogic, and IBM WebSphere
- Standalone Java client/server

## Support for Multiple Programming Languages

- Full support for JAX-WS 2.x client/server programming model
- JAX-WS 2.x synchronous, asynchronous and one-way API's
- JAX-WS 2.x Dynamic Invocation Interface (DII) API
- JAX-RS for RESTful clients
- Support for wrapped and non-wrapped styles
- XML messaging API
- Support for JavaScript and ECMAScript 4 XML (E4X) - both client and server
- Support for CORBA
- Support for JBI with ServiceMix

## Tooling

- Generating Code: WSDL to Java, WSDL to JavaScript, Java to JavaScript
- Generating WSDL: Java to WSDL, XSD to WSDL, IDL to WSDL, WSDL to XML
- Adding Endpoints: WSDL to SOAP, WSDL to CORBA, WSDL to service

- [Generating Support Files: WSDL to IDL](#)
- [Validating Files: WSDL Validation](#)

## **GETTING INVOLVED**

Apache CXF is currently under heavy development. To get involved you can subscribe to the mailing lists. You can also grab the code from the Source Repository. You also need to read about [Building CXF](#). For Eclipse users, you should read about [Setting up Eclipse](#).