Building REST Services with Apache CXF

Desert Code Camp 2016

Presenters

Aaron Hoffer

Travis McMahon

Connexta

Things we will demo

- Running REST services
- JAX-RS
- REST CRUD: Get, Post, Put, Delete
- Changing resource representation between XML and JSON
- Autogenerating API documentation
- Enabling SSL/TLS
- REST HATEOAS (*not implemented yet*)

Pieces and Parts

What is CXF?

- Framework for service-oriented applications
- Supports multiple protocols (styles?) for a service
- Supports multiple transports (HTTP, message queues, others)
- Standards driven
  - JAX-WS
  - JAX-RS
  - Lots of OASIS WS-* standards
- CXF is part of *Service Mix*
  - Suite of Apache software to create an enterprise service bus

What is JAX-RS

- Java API for RESTful Web Services
- Add annotations to create services
- It's a standard
  - JSR-339 specification
- Implementations include CXF, Jersey, RESTEasy, others

Spring Boot (1/2)

- Wait, what?
  - Spring Boot seen as CXF "competitor"
- CXF needs a web container like Jetty or Tomcat
- CXF can integrate with an application server
  - WebSphere
  - JBoss
  - Glassfish
  - Spring Boot
  - ... others...

Spring Boot (2/2)

- This presentation started with Jetty, but it was easier to get SSL/TLS working with Spring Boot
- "Why didn't you Sprint Boot instead of CXF?"
  - Simple answer: *Our company uses Service Mix, so we did a presentation on CXF.*
  - Our customers like standards based development and CXF implements standards.

What else will you see today?

- **Maven** to manage dependencies
- **Maven Sprint Boot plugin** to make building and running easy
- **Swagger** to create a nice summary of our web service API

About JAX-RS

- Use annotations to control:
  - When a method is invoked
  - What is passed into the method
  - What method should return
- JAX-RS Annotation you will see today
  - @Path and @GET, @PUT, @POST, ...
  - @Consumes, @PathParam
  - @Produces
- Annotations you won't see today
  - @QueryParam, @CookieParam, @Context
- Install new providers/mappers to serve new MIME types
  - JAXB to convert between Java objects to XML
  - Jackson to convert between Java objects and JSON

Demo Progression (speakers notes)

Getting started

1. Create a new project in IntelliJ (Travis: Maybe we just have some skeleton in Github that we clone down?)
2. Copy in the POM file contents
3. Try building the project. Look at log output for port #

Demo Progression (speakers notes)

"Hello World"

1. Create the CodeCamp service interface with "hello world method"
2. Create an implementation of the interface
3. Create the application and application configuration classes
4. Run it. Look at log output for CXF Servlet path
5. Test it with Postman
6. Modify application properties for /services path

Demo Progression (speakers notes)

Get rolling with RESTful resources

1. Add Customer, Product, Order, and Database classes
2. Add GET methods to service interface and implementation for Customer
3. Test getting customer with Postman
4. Test getting non-existent customer
5. Enhance GET method to return different status code
6. Change Accepts header to switch between XML and JSON

Demo Progression (speakers notes)

HTTP Verbs

1. Add POST, PUT, DELETE for customer
2. Demonstrate POST
3. Demonstrate PUT
4. Demonstrate DELETE

Demo Progression (speakers notes)

HATEAOS

1. Add getOrderHateoas resource
2. Add link headers for customer and product
3. Test it
4. Add link headers for cancel
5. Test it

Things about CXF we left out

- CXF Interceptors (reading and writing), Filters
  - Useful for authentication, authorization steps
- Streaming input and output
- Alternate transports like JMS
- Other "styles" of web services like SOAP and POX
- Integration with Apache Camel for ESB
- Generating WSDLs for SOAP or WADLs for REST services
- Using CXF as a client instead of a server