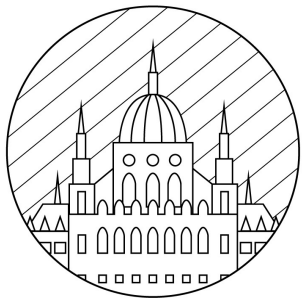


JAX-RS 2.0 With Apache CXF Continued

Sergey Beryozkin, Talend



APACHECON
EUROPE

CORINTHIA HOTEL
BUDAPEST, HUNGARY
— NOVEMBER 17-21, 2014 —



What is Apache CXF

- Production quality Java based framework for developing REST and SOAP services
- Major focus on the interoperability, security, compatibility
- Philosophy: what works best for the user wins
- JAX-RS 2.0, JAX-WS 2.1, OAuth2, WS-Security, SSO (Fediz), WebSockets, JOSE
- Complete web services platform
- Busy and healthy project environment



Brief History of JAX-RS

- JAX-RS 1.0 (JSR 311): Oct 2008, Paul Sandoz and Mark Hadley, Sun; followed by JAX-RS 1.1, Nov 2009
- JAX-RS: streamlines the HTTP Services development
- JAX-RS RI is Jersey (Sun/Oracle), RestEasy (RedHat/JBoss) and Apache CXF – two other well-known projects supporting JAX-RS; Apache Wink
- Concerns about JAX-RS after Oracle buys Sun...
- JAX-RS 2.0 (JSR-339): May 2012, Marek Potociar, Santiago Perico-Geertsen, new super leads.
- JAX-RS 2.1 (JSR-370): work is due to start shortly



JAX-RS 1.0: Core Features

- `@Path`, `@Consumes`, `@Produces`, `@GET/@POST/etc` mark classes as root resources, methods as JAX-RS resource methods or sub-locators
- `@PathParam/@QueryParam/etc` map URI parts or HTTP headers to method parameters
- Contexts such as `UriInfo`, `Request` or `HttpHeaders` capture the request specific information
- Data are usually Beans or Response (Beans and Headers), `MessageBodyReader` or `Writer` handle IO
- Code example...



The Best Of JAX-RS 1.0

- Conditional HTTP support with the help of Request context: tricky to do right with Servlet API, JAX-RS 1.0 makes it nearly trivial to implement
- Initial streaming support with StreamingOutput, provides for the immediate push of the application data back to the consumer as they become available
- Pluggability of custom readers/writers
- UriBuilder: real effort at simplifying building URIs
- Code example...



JAX-RS 2.0: What is New ?

- Fluent Client API
- Support for Suspended Invocations
- Container and Client request and response filters
- Container and Client reader and writer interceptors
- Http Error Exception Hierarchy
- Optional Bean Validation
- Static and dynamic filter or interceptor bindings
- Capturing request parameters and contexts into BeanParam beans



JAX-RS 2.0: Client API

- Client->WebTarget->Invocation.Builder->Call
- Client is Configurable with filters, interceptors, message body readers/writers and properties.
- JAX-RS Response or Typed responses
- Synchronous and asynchronous invocations
- Http errors can be handled by new exception classes
- ClientRequestFilter, ClientResponseFilter, the former can block a request (useful for testing, client side caches); ReaderInterceptor, WriterInterceptor
- Code Example...



JAX-RS 2.0: Suspended invocations

- Allows for the request be suspended and resumed with the response when it is ready or the timeout expires; 'AsyncResponse' is a continuation manager
- The request can be suspended multiple times with the help of 'TimeoutHandler' until the response is ready, allows to have multiple short timeouts and hence the more effective response provisioning
- Applications can be notified when the server has completed sending a response or if the client has disconnected
- Code Example...



JAX-RS 2.0: Filters and Interceptors

- ContainerRequestFilter (PreMatch or PostMatch) and ContainerResponseFilter, ClientRequestFilter and ClientResponseFilter, the runtime calls each filter in turn, request filters can block
- WriterInterceptor and ReaderInterceptor (client, server) can delegate to the next handler or block
- Filters and interceptors can be ordered by @Priority
- Filters and interceptors can be bound to specific methods only statically (name bindings) or dynamically (DynamicFeature)
- Code Example...



The Best Of JAX-RS 2.0

- Suspended invocations with `AsyncResponse`: top design, rich and elegant support for managing complex and subtle mechanics of continuations
- Dynamic binding of filters – maximum flexibility in deciding which JAX-RS method may need to be pre or post processed
- Client API: type safe collection processing with `GenericType`, reuse of `Response`, mapping of HTTP errors to new exception classes, and so on...
- All of JAX-RS 2.0 is great !



JAX-RS 2.1: The Process Continues

- JSR-370: Enhancing JAX-RS further
- Server Side Events (SSE) binding
- Non-blocking IO in providers
- Optional integration with MVC frameworks
- Java 8 features into the API (improving asynchronous client API, etc)
- Hypermedia related API enhancements
- Etc...



Additional Resources

- JAX-RS 2.0:
<http://jax-rs-spec.java.net/nonav/2.0-rev-a/apidocs/index.html>
- JAX-RS 2.1:
<https://www.jcp.org/en/jsr/detail?id=370>
- JAX-RS in Apache CXF:
<http://cxf.apache.org/docs/jax-rs.html>
- Examples:
<https://github.com/sberyozkin/jaxrs20examples>

The image features a dark purple gradient background. In the center, there is a white silhouette of a mosque. The mosque has a large central dome and several minarets with pointed tops. The word "Questions ?" is written in white, sans-serif font across the middle of the mosque's silhouette.

Questions ?



Thank You