



# **REST WEB SERVICES IN JAVA EE 6 AND SPRING 3**

**Srini Panchikala**

**Austin Java User Group**

**March 30, 2010**

# ABOUT THE SPEAKER

- Security Architect
- Certified SCRUM Master
- Writer, Editor (InfoQ)
- Detroit Java User Group Leader
- Working with Java since 1996, JEE (2000), SOA (2006) & PPT since 09/2009
- Current: Agile & Lean Architectures, Domain-Driven Design, Architecture Enforcement, MDD
- Future: Role of DSL in Architecture Enforcement



# GOALS

- Overview of REST Web Services Support in Java EE 6
- JAX-RS 1.1 v. Spring 3



# SCOPE

- JAX-RS Discussion
- How JEE6 compares with Spring3
- Demo(s)



# FORMAT

- Interactive
- Demos
- Duration: ~75 minutes
- Q & A



# AGENDA

- RESTful Web Services in Java EE 6
- Simple JAX-RS Web Service
- Content Types
- Handling Request Data
- Preparing Response
- Exception Handling
- REST Frameworks
- REST Services Security
- Conclusions



# AGENDA

- **RESTful Web Services in Java EE 6**
- Simple JAX-RS Web Service
- Content Types
- Handling Request Data
- Preparing Response
- Exception Handling
- REST Frameworks
- REST Services Security
- Conclusions



# RESTFUL WEB SERVICES IN JAVA EE 6

- JAX-RS 1.1 (JSR 311)
- POJO based framework
- Annotations to make Java class methods as web resources
- Integration with CDI, EJB 3.1, Servlet 3, and Web Fragments



# REST SERVICES IN SPRING 3

- Server Side: Extension to Spring MVC programming model
- Client Side: RestTemplate
- Fits nicely with other Spring projects (Spring Security)
- HttpConverters
- Object to XML Mapping (OXM)



# JAX-RS CORE ANNOTATIONS

- @Path
- @GET
- @POST
- @Produces
- @Consumes



# MAP REQUESTS TO METHODS

## **JAX-RS:**

- `@Path("SimpleRestService")`
- `@Path("/customers/{customerId}")`
- `@GET`

## **Spring3:**

- `@RequestMapping(value="/simplerestspring")`
- `@RequestMapping(value="/customers/{customerId}")`
- `@RequestMapping(value="/simplerestspring", method=RequestMethod.GET)`



# AGENDA

- RESTful Web Services in Java EE 6
- **Simple JAX-RS Web Service**
- Content Types
- Handling Request Data
- Preparing Response
- Exception Handling
- REST Frameworks
- REST Services Security
- Conclusions



# SIMPLE JAX-RS WEB SERVICE

- SimpleRestWebService.java
- web.xml
- pom.xml



# DEMO

- Demo



# SPRING 3 REST WEB SERVICE

- SimpleRestWebServiceSpring.java
- web.xml
- rest-spring-servlet.xml



# AGENDA

- RESTful Web Services in Java EE 6
- Simple JAX-RS Web Service
- **Content Types**
- Handling Request Data
- Preparing Response
- Exception Handling
- REST Frameworks
- REST Services Security
- Conclusions



# CONTENT TYPES

- Plain Text
- XML
- HTML
- JSON
- Image
- Multi-Part Form
- MediaType Utility Class



# DEMO

- RestWebServiceContentType .java
- Demo
- UsersResource.java
- JSON Demo



# UTILITY CLASSES

- UriInfo
- UriBuilder
- HttpHeaders
- Response
- Response.ResponseBuilder



# DEMO

- RestWebServiceParams.java



# AGENDA

- RESTful Web Services in Java EE 6
- Simple JAX-RS Web Service
- Content Types
- **Handling Request Data**
- Preparing Response
- Exception Handling
- REST Frameworks
- REST Services Security
- Conclusions



# HANDLING REQUEST DATA

- RequestParam
- QueryParam
- HeaderParam
- CookieParam
- PathParam
- FormParam
- MatrixParam



# DEMO

- RestWebServiceParams.java



# REQUEST PARAMS - JAX-RS V. SPRING3

- **JAX-RS:**

```
getHttpParams (@QueryParam("q") String q,  
  @FormParam("f") String f,  
  @CookieParam("c") String c,  
  @HeaderParam("h") String h,  
  @MatrixParam("m") m)
```

- **Spring3:**

```
getHttpParams (@RequestParam String r,  
  @CookieValue String c, @PathVariable  
  String p, @RequestHeader h)
```



# WORKING WITH REQUEST DATA

## **JAX-RS:**

- `MessageBodyReader`

## **Spring3:**

- `@RequestBody`
- Spring OXM
- `HttpMessageConverter`



# REQUEST DATA CONVERSION IN SPRING

```
<bean
  class="org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter" >
  <property name="messageConverters"
    ref="marshallingConverter"/>
</bean>
```

```
<bean id="marshallingConverter"
  class="org.springframework.http.converter.xml.MarshallingHttpMessageConverter">
  <constructor-arg ref="jaxb2Marshaller"/>
  <property name="supportedMediaTypes"
    value="application/vnd.stocks+xml"/>
</bean>
```

```
<oxm:jaxb2-marshaller id="jaxb2Marshaller"/>
```



# AGENDA

- RESTful Web Services in Java EE 6
- Simple JAX-RS Web Service
- Content Types
- Handling Request Data
- **Preparing Response**
- Exception Handling
- REST Frameworks
- REST Services Security
- Conclusions



# RESPONSE

## **JAX-RS:**

- MessageBodyWriter

## **Spring 3:**

- @ResponseBody



# STATUS CODES

## JAX-RS

### ○ Chained API:

- `Response.noContent().build(); // 204 (No Content)`

### ○ UriBuilder:

- `URI accountLocation =  
uriInfo.getAbsolutePathBuilder().path(account.getUsername()).build(); return  
Response.created(accountLocation).build();  
;`



# STATUS CODES - 2

## Spring3

### ○ Response Code Annotation:

- `@ResponseStatus(HttpStatus.NO_CONTENT)`



# AGENDA

- RESTful Web Services in Java EE 6
- Simple JAX-RS Web Service
- Content Types
- Handling Request Data
- Preparing Response
- **Exception Handling**
- REST Frameworks
- REST Services Security
- Conclusions



# EXCEPTION HANDLING

## **JAX-RS:**

- NotFoundException
- WebApplicationException
- ExceptionMapper

## **Spring3:**

- @ExceptionHandler({NoResultException.class})



# AGENDA

- RESTful Web Services in Java EE 6
- Simple JAX-RS Web Service
- Content Types
- Handling Request Data
- Preparing Response
- Exception Handling
- **REST Frameworks**
- REST Services Security
- Conclusions



# REST FRAMEWORKS

- Jersey
- RESTEasy
- Apache CXF
- Restlet
- Spring 3



# AGENDA

- RESTful Web Services in Java EE 6
- Simple JAX-RS Web Service
- Content Types
- Handling Request Data
- Preparing Response
- Exception Handling
- REST Frameworks
- **REST Services Security**
- Conclusions



# REST SERVICES SECURITY

- Authentication methods
  - HTTP Basic (no SSL)
  - HMAC-based authentication
- Authorization methods supported
  - OAuth
  - Custom
- Same attack vectors as a standard web application



# AGENDA

- RESTful Web Services in Java EE 6
- Simple JAX-RS Web Service
- Content Types
- Handling Request Data
- Preparing Response
- Exception Handling
- REST Frameworks
- REST Services Security
- **Conclusions**

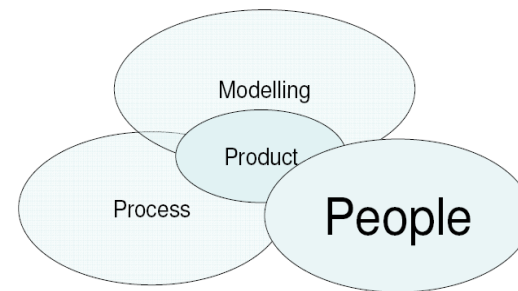


# CONCLUSIONS

- Light-weight WS solution
- More style than standard
- Annotation Based Support in JEE 6
- Separation on Concerns in the Architecture
- Same security concerns as a Web Application
- Avoid rolling your own security
- JEE6/JAX-RS1.1 v. Spring3



- Q & A



*Non-Linear First-Order Components*  
Alistair Cockburn



# CONTACT INFORMATION

- Domain-Driven Design and Enterprise Architecture articles on InfoQ
-  website: <http://www.infoq.com>
-  [srinipenchikala@gmail.com](mailto:srinipenchikala@gmail.com)
-  @srinip
-  <http://srinip2007.blogspot.com>



# RESOURCES

- JSR 311: JAX-RS: The Java API for RESTful Web Services (<http://jcp.org/en/jsr/detail?id=311>)
- API (<https://jsr311.dev.java.net/nonav/releases/1.1/index.html>)
- JAX-RS v1.1 Spec (<https://jsr311.dev.java.net/nonav/releases/1.1/spec/spec.html>)
- Changelog for JSR 000311 - JAX-RS 1.1 (<http://jcp.org/aboutJava/communityprocess/maintenance/jsr311/311changelog.1.1.html>)
- JEE6 Article (<http://java.sun.com/developer/technicalArticles/JavaEE/JavaEE6Overview.html>)
- JEE 6 Tutorial (<http://java.sun.com/javae/6/docs/tutorial/doc/>)
- Spring 3 Documentation (<http://static.springsource.org/spring/docs/3.0.x/spring-framework-reference/html/>)



# RESOURCES - 2

## Frameworks:

- Jersey (<https://jersey.dev.java.net/>)
- RESTEasy (<http://jboss.org/resteasy>)
- Apache CXF (<http://cxf.apache.org/>)
- Restlet (<http://www.restlet.org/>)

## Tools:

- REST Client (<http://code.google.com/p/rest-client/>)

