RESTful web services with JAX-RS and RESTEasy

2010-03-18/StephaneEpardaud@www.lunatech.com
Outline

# A word about the author

# RESTful web services

> An introduction

# JAX-RS and RESTEasy

> Demo

# Conclusion

# Questions
About the author

#Stéphane Épardaud
#Regional Director Lunatech France
#Senior Software Architect
#RESTEasy committer
#jax-doclets author
Outline

RESTful web services
What is REST?

# Representation State Transfer

> Roy Fielding (HTTP) thesis

> Applying HTTP principles to internet applications

# Addressable resources

> ID, URI

# Limited interface

> Standard protocol methods (GET, POST, PUT, DELETE...)
What is REST?

# Different representations per resource

> Decouple formats from resources (XML, JSON…)

# Stateless communication

> Scales well
Addressability

# URIs

> Well-known

> Linkable

# Example

<order id="12">
  <customer ref="/customer/42"/>
</order>
URI Parameters

# Path

>/customer/{id}

# Query

>/customers?country=FR

# Matrix

>/customers;start=45;count=20
Limited interface

# Limit the number of operations
> Define them well

# In HTTP:
> POST, GET, PUT, DELETE

# CRUD:
> CREATE, READ, UPDATE, DELETE

# SQL:
> INSERT, SELECT, UPDATE, DELETE
Limited interface

# GET
> Idempotent read, no side-effect

# PUT
> Idempotent update, can be repeated

# DELETE
> Idempotent deletion

# POST
> Not idempotent, everything goes
Limited interface

# Simple
  > No API, no dependency

# Portable
  > HTTP supported everywhere, format decoupling

# Interoperable
  > HTTP stable and well-known

# Familiar
  > Securing and caching HTTP well-known

# Easy to debug
Example API

/orders
- GET – list every order
- POST – create a new order
/orders/{order-id}
- GET – get an order representation
- PUT – update an order
- DELETE – cancel an order
/orders/average-sale
- GET – calculate the average sale
/customers
- GET – list every customer
- POST – create a new customer
/customers/{cust-id}
- GET – get a customer representation
- DELETE – delete a customer
/customers/{cust-id}/orders
- GET – list every order for a customer
No state

# No session/conversation on the server

> Does not mean no storage

# Maintain state on the client

> Reify it as a resource

# Scales well
Outline

JAX-RS/RESTEasy demo
Features

# @Path

# @PathParam, @QueryParam, @FormParam, @CookieParam, @HeaderParam

>@DefaultValue

# @Form

# Response
Features

@Produces, @Consumes

JAXB, JSON

Type Mappings

@Wrapped

@Context

HttpHeaders, SecurityContext, UriInfo, Request, HttpServletRequest, HttpServletResponse, ServletConfig, ServletContext
Features

# LastModified

> Request.evaluatePreconditions

# @MultiPartForm

# Cache support

# Servlet 3.0 Asynchronous support

# Security

> @RolesAllowed, OAuth
Features

# @Provider
  > Exception mapping
  > Pluggable marshallers

# GZIP

# Interceptors
Clients

#Java

#JavaScript
Conclusion

# REST is simple and powerful

> Real paradigm shift

# JAX-RS is easy and extendable

# RESTEasy provides many extensions

> http://www.jboss.org/resteasy

# Jax-doclets for documenting JAXB and JAX-RS

> http://www.lunatech-labs.com/open-source/jax-doclets
Questions
stephane.epardaud@lunatech.com
www.lunatech.com
employment@lunatech.com