



# RESTful Web Services

**[Jakub.Podlesak@Sun.COM](mailto:Jakub.Podlesak@Sun.COM)**

**<http://blogs.sun.com/japod>**



# Goal

- To show you how to develop and run RESTful web services using Java



## Yahoo! Live REST Web Services



### AWS::S3

A Ruby Library for Amazon's Simple Storage Service's (S3) REST API.

[Browse code](#) | [Docs](#) | [Downloads](#) | [Bugs & Patches](#) | [Mailing list](#)

Download with RubyGems

```
sudo gem i aws-s3
```



# Demonstration 1

# REpresentational State Transfer

- Is an architectural style
- Set of constraints
- Applied to the architecture of a distributed system

# Key Concepts

- Resources identified by IDs (URIs)
- Uniform service interface
- Hypermedia exchange
- Stateless communication

# Why apply REST?

- It works :-)
- Simplicity
- Loose coupling
- Scalability
- Service Composition & Orchestration

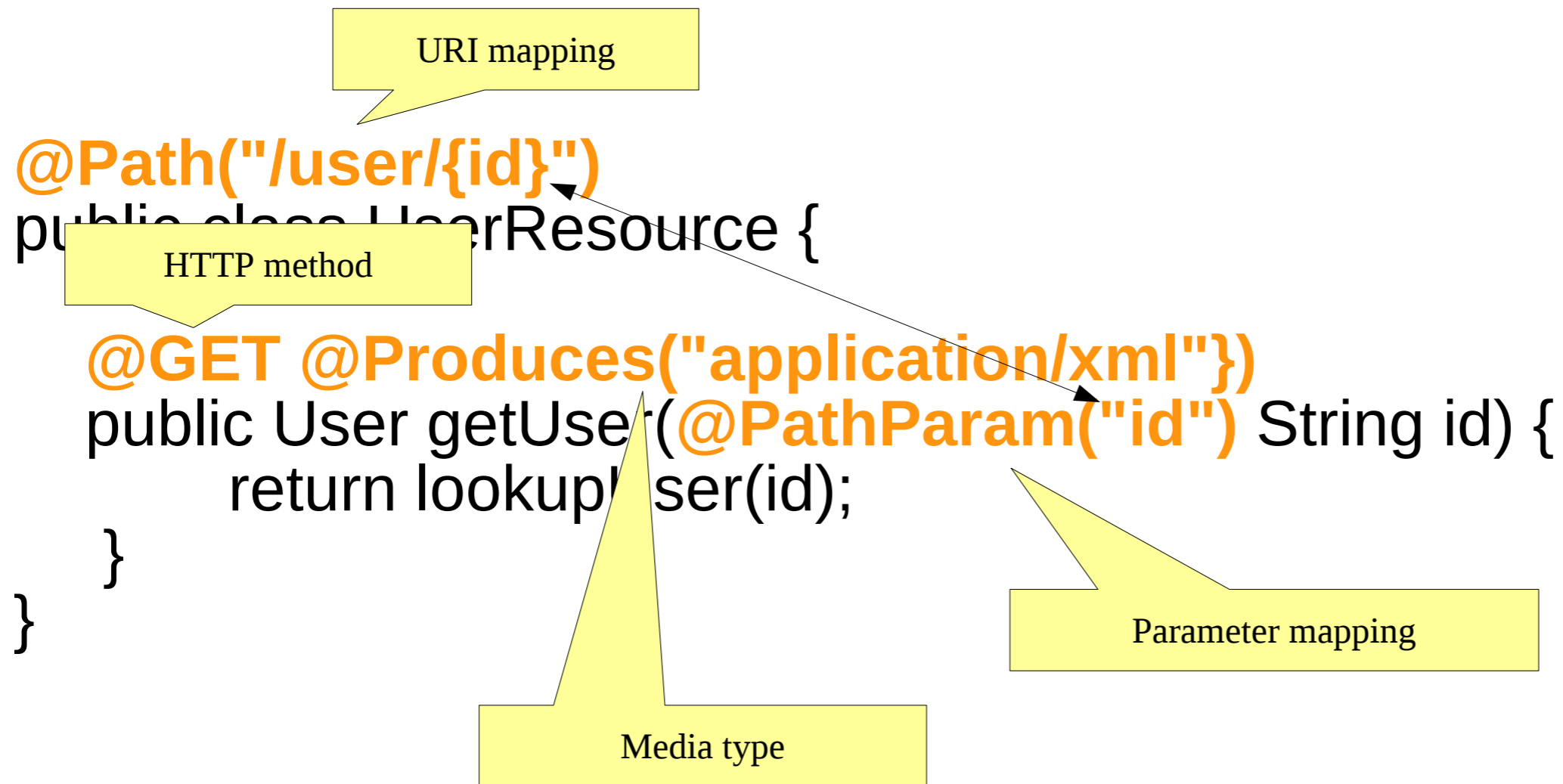
# JAX-RS (1/2)

- Standardized in the JCP: JSR-311
- Server side centric
- Part of Java EE 6

# JAX-RS (2/2)

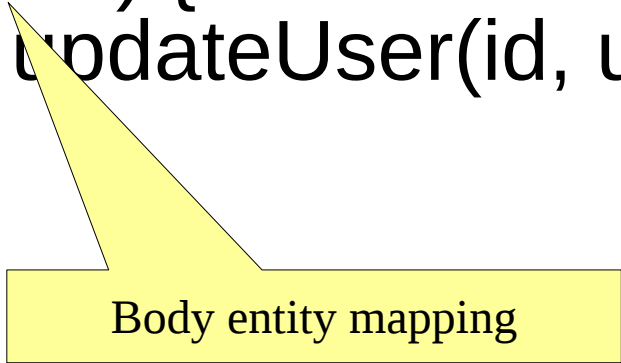
- POJO based
- Annotation driven
- Container independent

# JAX-RS Example (1/2)



# JAX-RS Example (2/2)

```
@Path("/user/{id}")  
public class UserResource {  
  
    @PUT @Consumes("application/xml")  
    public void putUser(@PathParam("id") String id,  
                        User u) {  
        return updateUser(id, u);  
    }  
}
```



Body entity mapping

# Jersey

- Open source
- Production quality
- Reference implementation of JAX-RS
- 1.1.3-ea released in October

# Quick start

- Maven archetypes
  - > `mvn -DarchetypeCatalog=http://download.java.net/maven/2/archetype-catalog.xml archetype:generate`
- GlassFish
  - > 3.0 Preview (Jersey core libs integrated directly)
  - > 2.1 (Jersey libs and apidocs and examples at UC repo)
- NetBeans 6.7
- Examples
  - > <http://download.java.net/maven/2/com/sun/jersey/samples/jersey-samples/1.1.3-ea/jersey-samples-1.1.3-ea-project.zip>

# Demonstration 2

# Jersey features

- Jersey client API
- Embedded Grizzly container
- Embedded GlassFish
- JSON with JAXB
- MVC with JSP
- Multipart MIME
- Simple injection

# Jersey features (contd...)

- Spring integration
- Guice integration
- Java client API using Apache HTTP client
- Comet support

# GET more information

- <https://jsr311.dev.java.net/>
- <https://jersey.dev.java.net/>
- **Blogs**
  - > <http://blogs.sun.com/sandoz>
  - > <http://weblogs.java.net/blog/mhadley/>
  - > <http://blogs.sun.com/japod>
  - > <http://blogs.sun.com/theaquarium/>

# Q&A

<http://blogs.sun.com/japod>

# Thank You!

And do not forget to visit:

<https://jersey.dev.java.net>

# BACKUP

# Jersey features demo

- Twitter client
- Twitter as a message bus