



Java is a trademark of Sun Microsystems, Inc.

# JavaOne<sup>SM</sup>

## Developing RESTful Web Services with Jersey and JAX-RS

Craig McLanahan  
Jakub Podlesak  
Paul Sandoz  
Sun Microsystems, Inc

## Agenda

- > REST, JAX-RS & Jersey
- > Paul: Stuff using Jersey
- > Craig: Multipart, Atom, Cloud
- > Jakub: Clients to services
- > Q & A

# REST is an Architectural Style

Set of constraints you apply to the architecture of a distributed system to induce desirable properties

# RESTful Web Services

Application of REST architectural style to services that utilize Web standards (URIs, HTTP, HTML, XML, Atom, RDF etc.)

## Java API for RESTful Web Services (JAX-RS)

Standard annotation-driven API that aims to help developers build RESTful Web services in Java

## Jersey

Open source, production quality, reference  
implementation  
... and more ...

## HEAD /Jersey

- > <http://jersey.dev.java.net>
- > <mailto:users@jersey.dev.java.net>
- > 1.0.3 (JAX-RS 1.0) released on April 15<sup>th</sup>
- > 1.1.0-ea (JAX-RS 1.1) released on May 4<sup>th</sup>
  - GET [started](#)
  - GET the [User guide](#)
  - GET the [dependencies](#)
  - GET the [samples](#)
- > Blogs: [Craig](#), [Marc](#), [Jakub](#), [Paul](#), [Aquarium](#)

## GET /Jersey/GlassFish

- > GlassFish v2.x
  - Available from the update center
  - Will be distributed with binaries
- > GlassFish v3 Prelude
  - Available from the update center
- > GlassFish v3 Preview
  - Stable releases distributed with binaries
  - Samples/docs available from the update center

## GET /Jersey/NetBeans

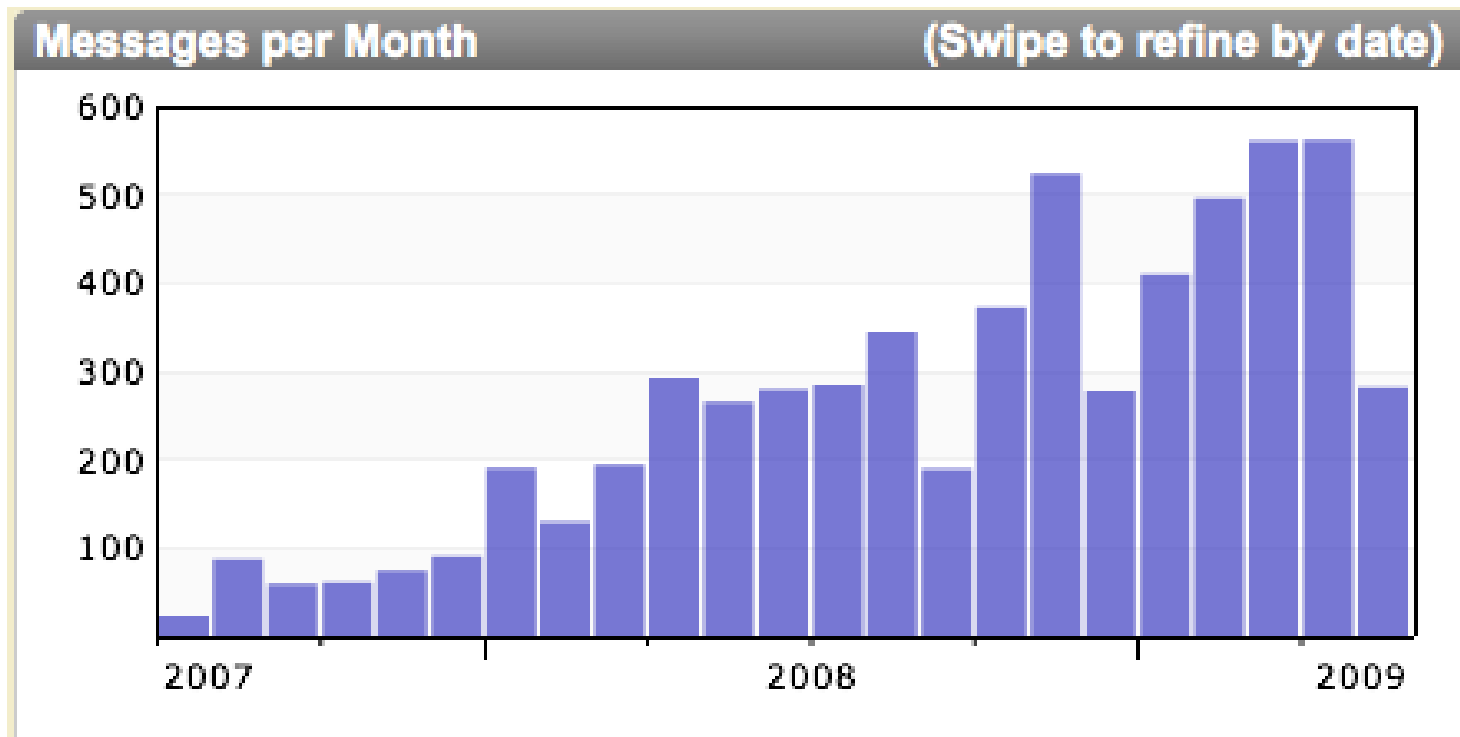
- > NetBeans 6.5
- > NetBeans 6.7
  - Maven support

## GET /Samples

- > Many samples are provided with the release
  - Atom, JAXB, JSON, Scala, Spring, WADL
  - Using GlassFish (+embedded) and Grizzly
- > Download the **1.1.0-ea samples**
  - Samples are maven-based
  - Works with NetBeans 6.x + maven plugin
  - Individual sample zip files are also available
    - e.g. **Sparklines, Mandel**

## GET /Involved

- > Ask/answer questions, propose fixes/features
  - <mailto:users@jersey.dev.java.net>



## GET /Features

- > Client API
- > JSON with JAXB
- > Spring and Guice integration
- > MIME multipart API
- > Atom with Abdera API
- > WADL
- > Grizzly and Simple servers

 Download Jersey  
NOW 1.0.3 

 Download Jersey  
NOW 1.1.0-ea 

[Jersey User Guide](#)

Client, Server/Grizzly, WADL & JSON

[API](#)

Spring integration

[API](#)

Guice integration

[API](#)

MIME multipart

[API](#)

Client & Apache HTTP client

[API](#)

Atom & Apache Abdera

[API](#)

Simple HTTP server

[API](#)

## Jersey being used by...

- > UShareSoft  
<https://www.usharesoft.com/>
- > Java App Store  
<http://java.sun.com/warehouse/>
- > RestMQ: Jersey, Scala and Lift  
<http://restmq.fusesource.org/>

# Jersey Multipart Support

## Goals

- > Support multipart media types:
  - multipart/mixed, multipart/form-data
- > Leverage JAX-RS providers for body parts
- > Operate both server side and client side
- > Convenience methods for ease of use:
  - Builder pattern constructors
  - Access body parts by index or name
  - Files as body parts
- > Stream large body parts to disk as necessary

# Jersey Multipart Support

## Create and Consume MultiPart Instances

### > Create with String and XML body parts:

```
MyBean bean = ...; // JAXB-annotated bean class
String metadata = "Metadata about this bean";
MultiPart multiPart =
    new MultiPart().
        bodyPart(metadata, MediaType.TEXT_PLAIN_TYPE).
        bodyPart(bean, MediaType.TEXT_XML_TYPE);
```

### > Consume at other end:

```
String metadata =
    multiPart.getBodyParts().get(0).
    getEntityAs(String.class);
MyBean bean =
    multiPart.getBodyParts().get(1).
    getEntityAs(MyBean.class)
```

# Jersey Atom Abdera Support

## Goals

- > Easily produce and consume Atom messages
- > Operate both server side and client side
- > Support embedding custom data:
  - Via XML namespaces and custom elements
  - Via Atom's "content" element
- > Leverage JAX-RS providers for content element
- > Make use of Apache Abdera's model classes

# Jersey Atom Abdera Support

## Create and return an Atom Feed

```
@RolesAllowed("user")
@GET
@Produces({"application/atom+xml"})
public Response get() {
    Feed feed = abdera.newFeed();
    // NOTE - RFC 4287 requires a feed
    // to have id, title, and updated properties
    feed.setId("contacts");
    feed.setTitle("Contacts System Contacts for
user '" + username + "'");
    feed.addLink(uriInfo.
    getRequestUriBuilder().build().toString(),
    "self");
    ...
}
```

# Jersey Atom Abdera Support

## Create and return an Atom Feed

```
        feed.setUpdated(  
            Database.contactsUpdated.get(username) );  
        List<Contact> contacts =  
            Database.contacts.get(username) ;  
        for (Contact contact : contacts) {  
            Entry entry = contact.asEntry() ;  
            String uri =  
uriInfo.getRequestUriBuilder().path(contact.getId()).  
build().toString() ;  
            entry.addLink(uri, "self") ;  
            entry.addLink(uri, "edit") ;  
            feed.addEntry(entry) ;  
        }  
        return Response.ok(feed).build() ;  
    }
```

# Jersey in the Sun Cloud

## Server Side Usage

- > Storage Administration Service:
  - Create and delete storage volumes
  - Create snapshots
  - Create clone (new volume from snapshot)
- > Identity Service (internal):
  - Authenticate users
  - Authorize users
- > Various other internal services

# Jersey in the Sun Cloud

## Server Side Code Snippet

### > Storage Administration Service – Resource:

```
@GET
@Produces({"application/xml", "text/xml",
    "application/json", "x-network/resource+xml",
    "x-network/resource+json"})
public Response find(
    @PathParam("volumeName") String volumeName,
    @QueryParam("metadata") String metadata) {
    return getUserVolume(volumeName, null,
        metadata);
}
```

# Jersey in the Sun Cloud

## Additional Jersey Features


### > Server Side Filters:

- Authentication and access control
- Client version compatibility

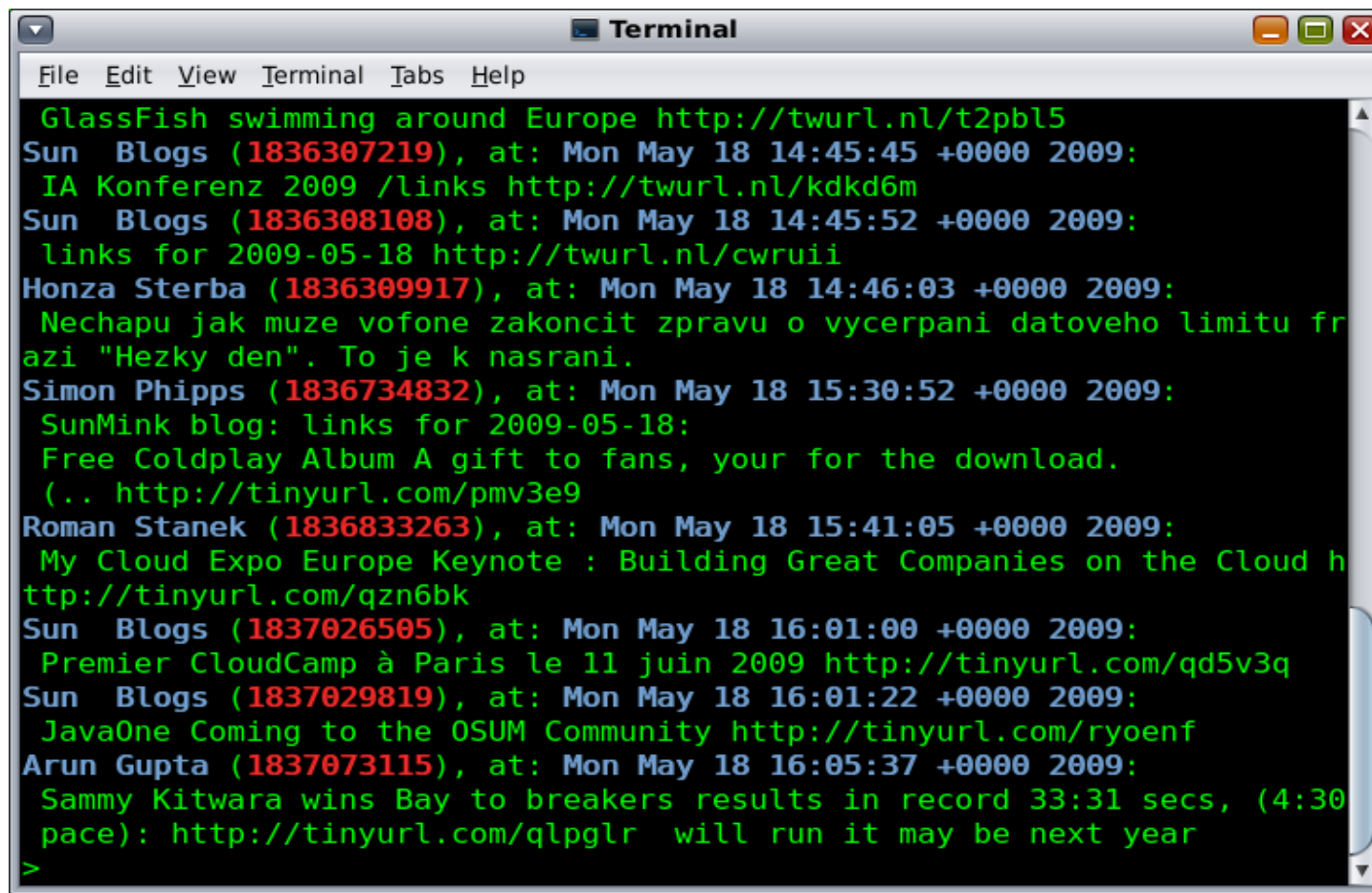
### > Client Libraries:

- Storage Administration Service
- Cloud API
- Clients for various internal services
- Copied API style for Ruby and Python clients :-)

## Jersey based clients to real-life web services

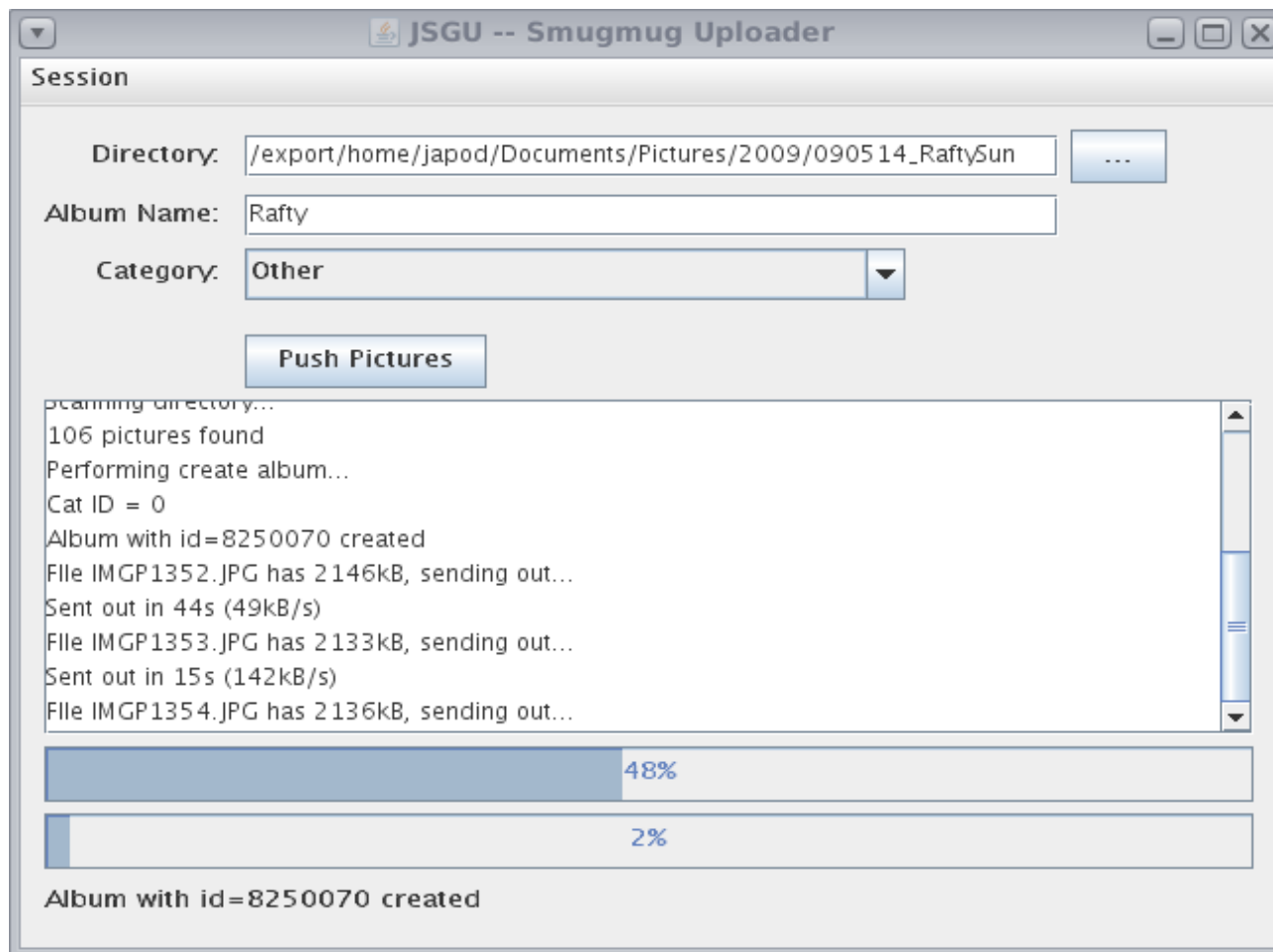
- **twitter** (communication service)
  - HTTP Basic Authentication
  - HTTPS
  - JSON for data exchange
- **SmugMug**  (file-sharing service)
  - Custom authentication mechanism
  - HTTPS again

# Twitter Client



```
Terminal
File Edit View Terminal Tabs Help
GlassFish swimming around Europe http://twurl.nl/t2pb15
Sun Blogs (1836307219), at: Mon May 18 14:45:45 +0000 2009:
IA Konferenz 2009 /links http://twurl.nl/kdkd6m
Sun Blogs (1836308108), at: Mon May 18 14:45:52 +0000 2009:
links for 2009-05-18 http://twurl.nl/cwruii
Honza Sterba (1836309917), at: Mon May 18 14:46:03 +0000 2009:
Nechapu jak muze vofone zakoncit zpravu o vycerpani datoveho limitu fr
azi "Hezky den". To je k nasrani.
Simon Phipps (1836734832), at: Mon May 18 15:30:52 +0000 2009:
SunMink blog: links for 2009-05-18:
Free Coldplay Album A gift to fans, your for the download.
(.. http://tinyurl.com/pmv3e9
Roman Stanek (1836833263), at: Mon May 18 15:41:05 +0000 2009:
My Cloud Expo Europe Keynote : Building Great Companies on the Cloud h
ttp://tinyurl.com/qzn6bk
Sun Blogs (1837026505), at: Mon May 18 16:01:00 +0000 2009:
Premier CloudCamp à Paris le 11 juin 2009 http://tinyurl.com/qd5v3q
Sun Blogs (1837029819), at: Mon May 18 16:01:22 +0000 2009:
JavaOne Coming to the OSUM Community http://tinyurl.com/ryoenf
Arun Gupta (1837073115), at: Mon May 18 16:05:37 +0000 2009:
Sammy Kitwara wins Bay to breakers results in record 33:31 secs, (4:30
pace): http://tinyurl.com/qlpglr will run it may be next year
>
```

# SmugMug Client





# JavaOne<sup>SM</sup>

# Thank You

[Craig.McClanahan@Sun.Com](mailto:Craig.McClanahan@Sun.Com)

[Jakub.Podlesak@Sun.Com](mailto:Jakub.Podlesak@Sun.Com)

[Paul.Sandoz@Sun.Com](mailto:Paul.Sandoz@Sun.Com)

[users@jersey.dev.java.net](mailto:users@jersey.dev.java.net)

