

# jax-doclets

## JavaDoc doclets for Java Extension APIs



**LUNATECH**  
LABS

1

**Stéphane Épardaud**

---

<sup>1</sup> <http://www.lunatech-labs.com>

---

## jax-doclets: JavaDoc doclets for Java Extension APIs



# LUNATECH

LABS

1

Stéphane Épardaud

0.7

Copyright © 2009, 2010 Lunatech Labs

### Abstract

jax-doclets is a set of JavaDoc doclets for the Java Extension APIs.

---

<sup>1</sup> <http://www.lunatech-labs.com>

---

---

## Table of Contents

Preface .....	iv
1. Overview .....	1
1.1. Information .....	1
1.2. Example .....	1
2. Running the jax-doclets .....	5
2.1. Running jax-doclets in standalone .....	5
JAXB doclet .....	5
JAX-RS doclet .....	5
2.2. Running jax-doclets in ant .....	5
JAXB doclet .....	5
JAX-RS doclet .....	5
2.3. Doclet parameters .....	6
Generic parameters .....	6
JAX-RS doclet parameters .....	6
3. JAX-RS doclet documentation .....	7
3.1. Where should you write JavaDoc .....	7
3.2. Supported standard JavaDoc tags .....	7
3.3. Supported specific JavaDoc tags .....	7
3.4. Supported JAX-RS annotations .....	7
3.5. Supported RESTEasy JAX-RS extension annotations .....	8
4. JAXB doclet documentation .....	9
4.1. Where should we write JavaDoc .....	9
4.2. Supported standard JavaDoc tags .....	9
4.3. Supported specific JavaDoc tags .....	9
4.4. Supported JAXB annotations .....	9
4.5. Mapping Java types to XML types .....	9
5. License .....	11

---

# Preface

This is pre-release software, please bear with it.

---

# 1. Overview

jax-doclets allows you to generate JavaDoc<sup>1</sup> documentation for specific Java annotation-based extensions such as:

- JAX-RS<sup>2</sup>: the RESTful API for Java
- JAXB<sup>3</sup>: the XML binding API for Java

The goal of jax-doclets is to let you write documentation for your JAX-RS API and JAXB structures in JavaDoc, where it belongs, where it is maintainable, and produce a quality JavaDoc-style documentation.

## 1.1 Information

jax-doclets is an open-source project maintained by Lunatech Labs<sup>4</sup>.

Home page	<a href="http://www.lunatech-labs.com/content/jax-doclets">http://www.lunatech-labs.com/content/jax-doclets</a>
Download	<a href="http://code.google.com/p/jax-doclets/downloads/list">http://code.google.com/p/jax-doclets/downloads/list</a>
Issue Tracker	<a href="http://code.google.com/p/jax-doclets/issues/list">http://code.google.com/p/jax-doclets/issues/list</a>
Source Control Management	<a href="http://code.google.com/p/jax-doclets/source/checkout">http://code.google.com/p/jax-doclets/source/checkout</a>

## 1.2 Example

Here is an example of documented JAX-RS and JAXB code:

---

<sup>1</sup> <http://java.sun.com/j2se/javadoc/>

<sup>4</sup> <http://www.lunatech-labs.com>

**Figure 1.1. Example of documented JAX-RS and JAXB code**

```
package com.lunatech.doclets.jax.test;

import javax.ws.rs.*;
import javax.xml.bind.annotation.*;

/**
 * An example JAX-RS resource
 */
@Path("/example")
@Produces( { "application/xml", "application/*+xml" })
public class JAXRSExample {

    /**
     * An example resource
     */
    @XmlRootElement
    public static class JAXBExample {

        /**
         * The resource ID
         */
        @XmlID
        @XmlElement
        String id;

        /**
         * The example contents
         */
        @XmlValue
        String contents;

        /**
         * An optional attribute
         */
        @XmlAttribute
        String type;
    }

    /**
     * Gets an example resource
     *
     * @param id
     *         the example id
     * @param type
     *         the type of resource we prefer
     * @param startIndex
     *         the start index
     * @return an example resource suitable for the given parameters
     * @HTTP 404 if there is no such example resource
     * @RequestHeader X-Example-Auth the authentication header
     * @ResponseHeader Location a pointer to the example details
     */
    @Path("/{id}")
    @GET
    public JAXBExample getExample(@PathParam("id") String id,
                                  @MatrixParam("type") String type,
                                  @QueryParam("start") int startIndex) {

        return new JAXBExample();
    }
}
```

Figure 1.2. Result of documented JAX-RS code



[Overview](#) [Index](#) [Root resource](#)

SUMMARY: RESOURCE | [METHOD](#)

DETAIL: [METHOD](#)

**Path: / [rest](#) / [example](#) / {id}**

Gets an example resource

**Path parameters:**

**id** - the example id

**Method Summary**

Resource	Description
<a href="#">GET /rest/example/{id}; type=...?start=...</a>	Gets an example resource

**Method Detail**

**HTTP Example:**

GET /rest/example/{id}; type=...?start=...

**API Example:**

```
JAXRSExample.getExample({'type': /* type
the type of resource we prefer */
'start': /* startIndex the start index
*/
'id': /* id the example id */});
```

Gets an example resource

**Output:**

[JAXRSExample.JAXBExample](#) - an example resource suitable for the given parameters

**Query parameters:**

**start** - the start index

**Matrix parameters:**

**type** - the type of resource we prefer

**Produces:**

application/xml  
application/\*+xml

**HTTP return codes:**

**404** - if there is no such example resource

**HTTP response headers:**

**Location** - a pointer to the example details

**HTTP request headers:**

**X-Example-Auth** - the authentication header

[Overview](#) [Index](#) [Root resource](#)

SUMMARY: RESOURCE | [METHOD](#)

DETAIL: [METHOD](#)

Generated by [Lunatech Labs jax-doclets](#) v0.7

Figure 1.3. Result of documented JAXB code



**Overview**

DETAIL: [ELEMENT](#) | [ATTRIBUTE](#) | [VALUE](#)

**Name: JAXBExample**

An example resource

**XML Example:**

```
<JAXBExample
  type="xsd:string">
  <id>xsd:ID[xsd:string]</id>
  xsd:string
</JAXBExample>
```

**JSON Example:**

```
{'JAXBExample':
  {
    '@type': String,
    'id': String /* ID */,
    String,
  }
}
```

**ID**

[id](#)

Elements		
Name	Type	Description
id	xsd:ID[xsd:string]	The resource ID

Attributes		
Name	Type	Description
type	xsd:string	An optional attribute

Value		
Type	Description	
xsd:string	The example contents	

**Overview**

DETAIL: [ELEMENT](#) | [ATTRIBUTE](#) | [VALUE](#)

Generated by [Lunatech Labs jax-doclets](#) v0.7

---

## 2. Running the jax-doclets

The jax-doclets are run by JavaDoc either as standalone or via ant.

*Since the JAX-RS supports links to JAXB documentation, you should first run the JAXB doclet, then the JAX-RS doclet using the `-link` parameter.*

### 2.1 Running jax-doclets in standalone

#### JAXB doclet

You can use the following command to run the JAXB doclet on your code:

#### Figure 2.1. Running the JAXB doclet

```
javadoc -doclet com.lunatech.doclets.jax.jaxb.JAXBDoclet -docletpath lib/jax-doclets-0.7.jar
com.lunatech.doclets.jax.test
```

#### JAX-RS doclet

You can use the following command to run the JAX-RS doclet on your code:

#### Figure 2.2. Running the JAX-RS doclet

```
javadoc -doclet com.lunatech.doclets.jax.jaxrs.JAXRSDoclet -docletpath lib/jax-doclets-0.7.jar
com.lunatech.doclets.jax.test
```

### 2.2 Running jax-doclets in ant

#### JAXB doclet

You can use the following ant XML to run the JAXB doclet on your code:

#### Figure 2.3. Ant XML for running the JAXB doclet

```
<target name="doc-jaxb" depends="jars" description="Run the JAXB doclet">
  <javadoc doclet="com.lunatech.doclets.jax.jaxb.JAXBDoclet"
    docletpath="lib/jax-doclets-0.7.jar">
    <package name="com.lunatech.doclets.jax.test.*"/>
  </javadoc>
</target>
```

#### JAX-RS doclet

You can use the following ant XML to run the JAX-RS doclet on your code:

#### Figure 2.4. Ant XML for running the JAX-RS doclet

```
<target name="doc-jaxrs" depends="jars" description="Run the JAXRS doclet">
  <javadoc doclet="com.lunatech.doclets.jax.jaxrs.JAXRSDoclet"
    docletpath="lib/jax-doclets-0.7.jar">
    <package name="com.lunatech.doclets.jax.test.*"/>
  </javadoc>
</target>
```

## 2.3 Doclet parameters

### Generic parameters

These parameters are valid for all jax-doclets

**Table 2.1. jax-doclet generic parameters**

<b>Parameter</b>	<b>Function</b>
-stylesheet	The CSS stylesheet to copy and use.
-header	The header which is inserted on every page header.
-footer	The footer which is inserted on every page footer.
-charset	The charset to use for source files and produced HTML documentation.
-link	Path to another JavaDoc documentation. This is used to produce links to other package's documentation, either regular JavaDoc or to JAXB documentation in the case of the JAX-RS doclet.

### JAX-RS doclet parameters

These parameters are only valid for the JAX-RS doclet

**Table 2.2. JAX-RS doclet parameters**

<b>Parameter</b>	<b>Function</b>
-jaxrscontext	The URL path to your RESTful API, if there is a prefix prepended to it on your deploy site.

---

## 3. JAX-RS doclet documentation

The JAX-RS doclet generates documentation for your RESTful service based on JAX-RS annotations and JavaDoc comments on your JAX-RS resource methods.

### 3.1 Where should you write JavaDoc

JavaDoc is read either on the JAX-RS resource methods, or their interface. Only method-level JavaDoc is used. Documentation for a given RESTful URL is taken from the method annotated with @GET, @HEAD, @POST, @PUT or @DELETE for that URL (in order of preference).

JAX-RS resource locators are supported.

*Since the JAX-RS supports links to JAXB documentation, you should first run the JAXB doclet, then the JAX-RS doclet using the `-link` parameter.*

### 3.2 Supported standard JavaDoc tags

The following standard JavaDoc tags are supported on resource methods:

**Table 3.1. Standard JavaDoc tags**

Tag	Function
@param {name} {doc}	This is used to document the corresponding resource method parameters annotated with @PathParam, @QueryParam or @MatrixParam. Can be used at most once per parameter name.
@return {doc}	Documents the entity returned from this resource method. Can only be used once.

### 3.3 Supported specific JavaDoc tags

The following specific JavaDoc tags are supported on resource methods:

**Table 3.2. Specific JavaDoc tags**

Tag	Function
@HTTP {code} {doc}	This is used to document the codes that the method can return. Can be used multiple times.
@inputWrapped {fq-classname}	Specifies the real type of input when declared as a String parameter. Can only be used once.
@returnWrapped {doc} {fq-classname}	Used in place of @return when output type is String, void or Response to specify the real type of output and documentation for each possible type. Can be used multiple times.
@RequestHeader {header} {doc}	This is used to document HTTP request headers. Can be used multiple times.
@ResponseHeader {header} {doc}	This is used to document HTTP response headers. Can be used multiple times.

### 3.4 Supported JAX-RS annotations

The following standard JAX-RS annotations are supported on resource methods or classes:

### **Figure 3.1. Standard JAX-RS annotations**

- `@Path`
- `@PathParam`
- `@FormParam`
- `@CookieParam`
- `@HeaderParam`
- `@QueryParam`
- `@MatrixParam`
- `@Produces`
- `@Consumes`
- `@Context` (ignored)

### **3.5 Supported RESTEasy JAX-RS extension annotations**

The following RESTEasy annotations are supported on resource methods or classes:

### **Figure 3.2. RESTEasy annotations**

- `@Form`

---

## 4. JAXB doclet documentation

The JAXB doclet generates documentation for your XML schema based on JAXB annotations and JavaDoc comments on your JAXB classes.

### 4.1 Where should we write JavaDoc

JavaDoc is read either on the JAXB properties (getter methods or fields), or their interface (only for the getters) as well as on the JAXB classes.

*Since the JAX-RS supports links to JAXB documentation, you should first run the JAXB doclet, then the JAX-RS doclet using the `-link` parameter.*

### 4.2 Supported standard JavaDoc tags

There are no standard JavaDoc tags supported. Everything comes from JavaDoc comments.

### 4.3 Supported specific JavaDoc tags

There are no specific JavaDoc tags supported.

### 4.4 Supported JAXB annotations

The following standard JAXB annotations are supported on properties or classes:

#### Figure 4.1. Standard JAXB annotations

- `@XmlAccessorType`
- `@XmlRootElement`
- `@XmlElement`
- `@XmlElementWrapper`
- `@XmlAttribute`
- `@XmlValue`
- `@XmlID`
- `@XmlIDREF`
- `@XmlTransient` (ignored)

### 4.5 Mapping Java types to XML types

The following Java types have a special mapping in XML:

**Table 4.1. Standard Java types mapping to XML**

Type	XML mapping
<code>java.lang.String</code>	<code>xsd:string</code>
<code>java.lang.Character</code> , <code>char</code>	<code>xsd:string</code>
<code>java.lang.Date</code>	<code>xsd:datetime</code>

<b>Type</b>	<b>XML mapping</b>
<code>java.lang.Integer, int</code>	<code>xsd:int</code>
<code>java.lang.Long, long</code>	<code>xsd:long</code>
<code>java.lang.Short, short</code>	<code>xsd:short</code>
<code>java.lang.Byte, byte</code>	<code>xsd:byte</code>
<code>java.lang.Float, float</code>	<code>xsd:float</code>
<code>java.lang.Double, double</code>	<code>xsd:double</code>
<code>java.lang.Boolean, boolean</code>	<code>xsd:boolean</code>
<code>java.lang.Object</code>	<code>xsd:any</code>
<code>java.lang.Enum</code>	List of enum values
<code>java.util.Collection</code>	<code>xsd:list</code>

Any other type is taken to be either a Java type or a JAXB type, for whom proper links will be generated.

---

## 5. License

jax-doclets is distributed under the LGPL license. It does not distribute any thirdparty libraries that are GPL. It does ship thirdparty libraries licensed under Apache ASL 2.0 and LGPL.