

# JCP, Adopt-a-JSR & You

*Heather VanCura*  
*February 2015*  
*<http://jcp.org>*  
*Twitter @heathercv*

# For the *community*, by the *community*



- The Java Community Process (JCP) is the open, inclusive process to develop and revise Java technology specifications (JSRs), reference implementations (RI), and technology compatibility kits (TCK).
  - The JCP program now has over 800 corporate, individual and Java User Group (JUG) members.
  - More than 370 Java technology specifications are in development in the JCP program; ~two thirds have reached Final Release.
    - Java EE 7 Final Release 2013.
    - Java SE 8 Final Release 2014.
    - Java ME 8 Final Release 2014.



- Celebrating 15 years!



# Levels of participation



- Public Observer.
- Members.
  - Review specs, may participate in Expert Groups, vote in Executive Committee elections; flexible time commitment.
- Expert Group members.
  - Create JSRs (write the spec, develop the RI and TCK); expert in field, high time commitment, help drive the direction of the JSR; moderate time commitment.
- Specification Leads.
  - Responsible for leading the JSR & Expert Group (EG), delivery of spec, RI & TCK; a leading expert in area of JSR; considerable time commitment and recognition.
- Executive Committee members.
  - Define JCP governance, processes, and contractual terms of membership. Vote on JSRs at defined stages through the process. Moderate time commitment.



# JCP is now more open than before



- Public JSR schedule--current, and updated regularly.
- Public read and/or write page/wiki for the JSR.
- Public discussion for the work of the JSR (archived).
- Public feedback mechanism.
- Public issue tracker.
- One EC votes on all JSRs 3x in life of JSR with public results.
- All JSRs have 3 open public review periods (30 days).
  - Release early and often!

# Get involved!



# What do you get?



- As a corporation:
  - Influence over the technologies that drive your market.
  - Savings - it's cheaper than developing everything yourself.
  - Opportunities for your developers.
  - Increase in the size of your market.
  - Competitive advantage (ship new products faster.)
- As an individual:
  - Acquire knowledge: Learn from experts, early access.
  - Better specifications based on real world experience.
  - Experience and skill development.
  - Professional Visibility: Curriculum, articles, presentations.
  - Fame and fortune – grow your reputation and network.



# How will you participate?



- As an individual – OK.
- As part of a team – better.
- Work through your JUG or employer.
  - Help each other.
  - Teach each other.
  - Work with each other.



# Adopt a JSR!



Thank you **LJC** and **SOU Java**  
sociedade de usuários java

# What is it?



- A JUG-lead initiative to improve Java standards.
- Groups of JUG members work on JSRs.
- See <http://java.net/projects/adoptajsr> for full details.

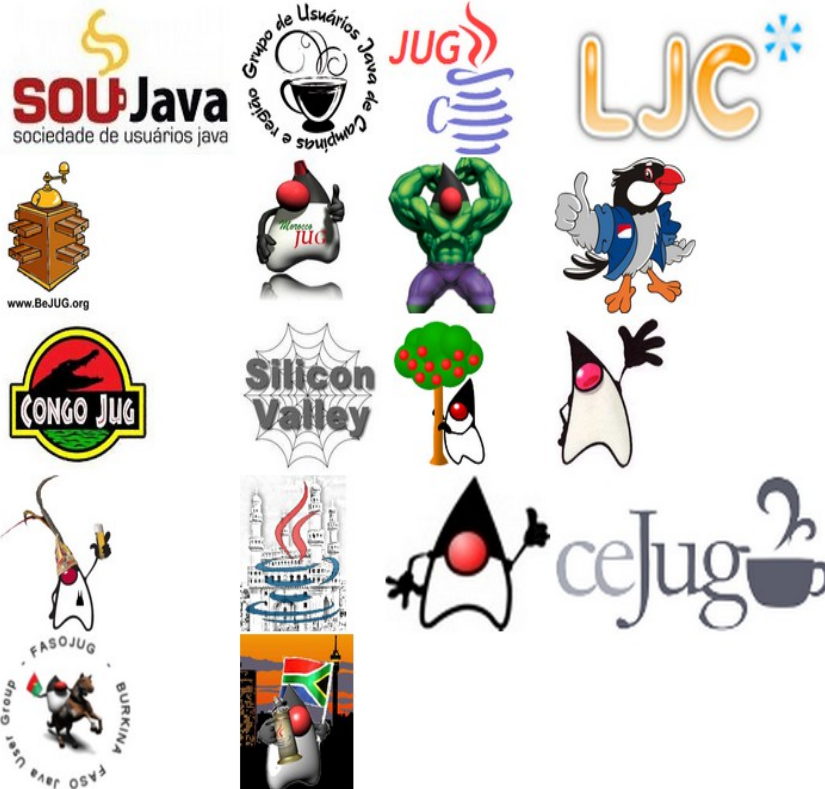


# How Will it Help your JUG?



- Receive attention from industry.
  - Big vendors, local companies see the value of all your efforts.
- Become more professional.
  - Participation requires organization and commitment—this helps promote focus in your group.
- Attract people.
  - Developers working together has a global impact, and reaps personal benefits.
- Collaborate with other JUGs.
  - Be part of something larger!

# JUGS Around the World!



# Adopt-a-JSR Adopters



- 28+ JUGS participating:
  - South America: Campinas JUG (Brazil), ceJUG (Brazil), GOJava (Brazil), SouJava (Brazil), Brasilia, DF JUG (Brazil), GUJ Londrina (Brazil) Peru JUG, Guadalajara JUG, GUJavaSC (Brazil).
  - North America: Chicago JUG, Silicon Valley JUG, Houston JUG (USA), Toronto JUG (Canada).
  - Europe: London Java Community (UK), BeJUG (Belgium), Madrid JUG (Spain), JUG Cologne (Germany), Istanbul JUG (Turkey).
  - Asia: Indonesia JUG, JUG JogLoSemar (Indonesia), Hyderabad JUG (India), JUG Chennai (India).
  - Africa & Middle East: EGJUG (Egypt), Morocco JUG, Faso JUG (Africa), Jozi JUG (South Africa), Mbale (Uganda).

# 2 Way Street



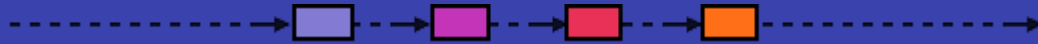
# Adopter Experience



- Java User Group Leaders and Members
  - Experience with Adopt-a-JSR
    - Gain knowledge
    - Experience with new technologies
    - Network opportunities
    - Make better standards

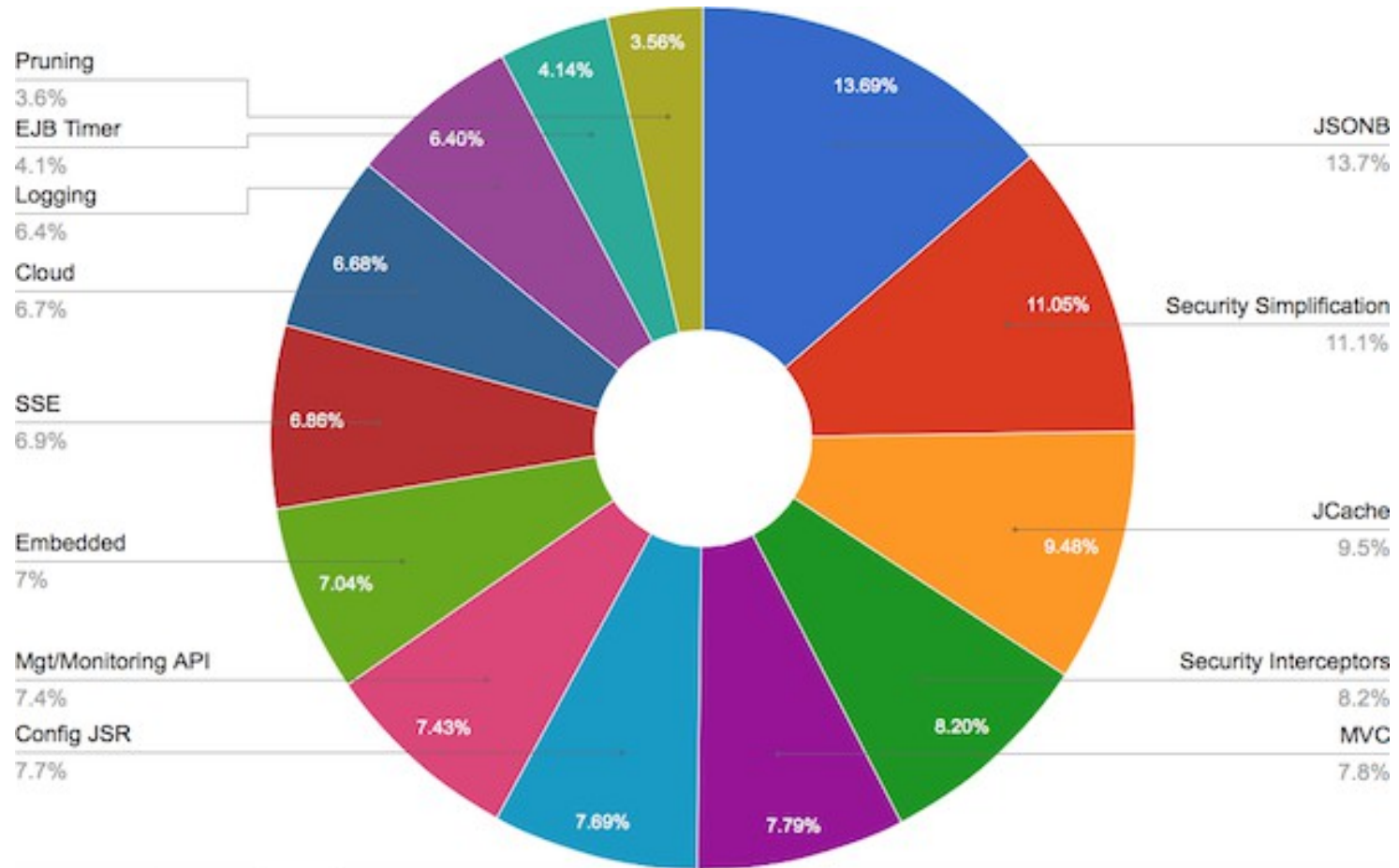


# Use Cases of Adopters



- BeJUG: JSR 356, Java API for WebSocket - Hack day in March 2013/developed Tic-Tac-Toe game included in Java EE 7 SDK.
- Chennai JUG: JSR 338, Java Persistence 2.1, JSR 344, JSF 2.2, JSR 352, Batch, JSR 353, JSON, JSR 356, WebSocket - Urban Traveler application; JSR 354, Money & Currency API.
- London Java Community: JSR 310, Date and Time API and JSR 335, Lambda Expressions (Martijn Verburg); myriad hack days, talks and events organized!
- SouJava: JSRs 310, 339, 344, and other JSRs; talks, books, articles, classes, code dojos:
  - <http://soujava.org.br/servicos/adopt-a-jsr/>

# Java EE 8 Community Survey



[Glassfish.org/survey](http://Glassfish.org/survey)

# Java EE 8 Possibilities



- Web Standards/HTML5 Alignment
  - HTTP2, SSE, JSON-B, action-oriented web framework
- Cloud
  - Simple security providers, multitenancy, REST management/monitoring
- CDI Alignment
  - CDI 2, EJB services outside EJB, security interceptors, EJB pruning
- Enterprise
  - JCache, Configuration, JMS
- Java SE 8 alignment

# Java EE 8 JSR Status



## Already Here

- Java EE 8 (JSR 366)
- CDI 2 (JSR 365)
- JSON-B (JSR 367)
- JMS 2.1 (JSR 368)
- Servlet 4 (JSR 369)
- JAX-RS 2.1 (JSR 370)
- MVC (JSR 371)
- JSF 2.3 (JSR 372)

## Forthcoming/

- Security (JSR 375)
- Java EE Management (JSR 373)
- JSON-P 1.1 (JSR 374)
- WebSocket 1.x
- JTA 1.3
- Batch 1.1
- JPA 2.2
- Concurrency 1.1

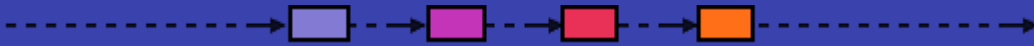
# Adopt-a-JSR for Java EE 8



- Make Java EE 8 the most community driven platform to-date!
- Share ideas and feedback, possibly by entering issues in public issue trackers.
- Follow expert group discussions and chime in if needed.
- Read early versions of specifications and Javadocs.
- Try writing applications using early builds of the reference implementation.
- Write or speak about the technology.
- Encourage others to participate.
- A number of JUGs have already signed up!

**<http://glassfish.org/adoptajsr/>**

# Putting the *community* back into the JCP



- No more barriers to participation.
- All members of the Java community can participate:
  - In the implementation of the platform through OpenJDK and GlassFish.
  - In the evolution of the platform through the JCP and Adopt-a-JSR programs.
- If you care about the future of Java...participate!

OpenJDK

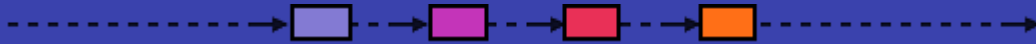


# How to get started on Adopt-a-JSR



- Join the JCP as a JUG.
  - Not required, but recommended. You need to do this in order to contribute code, and you get voting rights and other benefits with this membership!
- Join the Adopt-a-JSR project:
  - <http://java.net/projects/adoptajsr/pages/Home>
  - Subscribe to [members@adoptajsr.java.net](mailto:members@adoptajsr.java.net) (archive).
- Update the wiki to indicate your interest in a particular JSR and create a wiki page to provide updates on your plans/progress.
- Communicate directly with the Specification Lead/Expert Group (links on [jcp.org](http://jcp.org)) or the [java.net](http://java.net) users alias for the specification.
- Mail your JUG members to gather interest.
- Outline what the JSR is and why the JUG should get involved; identify areas that you want to work on; coordinate this with the Spec Lead
- Share success on the wiki, list, twitter, etc.

# Java.net wiki



## Project Features

- [Mailing Lists](#)
- [Downloads](#)
- [Issue Tracking](#)
- [Wiki](#)
- [WikiHomePage](#)

## Project Links

- [JSR Stages](#)
- [Glassfish/JEE AdoptAJSR](#)
- [GitHub](#)
- [Facebook Page](#)

## About this Project

Adopt a JSR is a subproject of Java Users Groups (JUG) Community, was started in December 2011 and has 77 members. The project administrators are Bruno Souza, Martijn Verburg, rwarburton, jcp\_pmo, Frank Nikola, Tonya Moore, and KRadm\_first KRadm\_last.

[Join This Project](#)

[java.net](#) > [projects](#) > [adoptajsr](#) > [wiki](#) > [Home](#)

Last updated 2 weeks ago, by [taman](#)



## Adopt a JSR Program



### Contents [\[hide\]](#)

- [1 Introduction](#)
- [2 Who's Adopting JSRs?](#)
- [3 Getting Started](#)
  - [3.1 Sign Up](#)
  - [3.2 Initial Steps](#)
  - [3.3 Leading an Adopt a JSR effort](#)
  - [3.4 What do we work on for a JSR?](#)
- [4 Talks, Presentations and Resources](#)
- [5 Projects and Code](#)
- [6 Success Stories](#)

## Introduction

This program is intended to encourage JUG members, individuals and other organisations to get involved in Java Specification Requests (JSRs). This means more grass roots, developer level participation in existing and emerging Java standards. JSRs cover all aspects of the Java ecosystem such as the new Date and Time API coming into Java 8, the coming JavaEE 8 APIs for the cloud and much more!

Also see [Benefits](#) and [Laser-like focus](#)

## Who's Adopting JSRs?

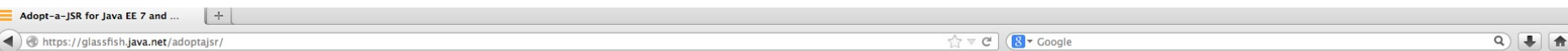
See the [Active JSRs](#), [Status of active JSRs](#) and [Who Is Adopting JSRs](#).

## Getting Started





# Java EE – specific requests from Spec Leads



 **GlassFish** - World's first Java EE 7 Application Server

About ▾

## Adopt-a-JSR for Java EE 7 and beyond



**Adopt-a-JSR** is an initiative by JUG leaders to encourage JUG members to get involved in a JSR and to evangelize that JSR to their JUG and the wider Java community, in order to increase grass roots participation. The typical question in Adopt-a-JSR is "how do I get started?".

In general, the **Starter Level** tasks are helpful at all stages of a JSR. The platform is already released and several JUGs already contributed to improve the quality.

The Java EE 7 platform was released on Jun 12, 2013 can be [downloaded from here](#). You can continue to provide feedback on different JSRs and influence the future direction:

- [Java API for WebSocket 1.0 \(JSR 356\)](#)
- [Java API for JSON Processing 1.0 \(JSR 353\)](#)
- [Batch Applications for the Java Platform 1.0 \(JSR 352\)](#)
- [Concurrency Utilities for Java EE 1.0 \(JSR 236\)](#)
- [Java API for RESTful Web Services 2.0 \(JSR 339\)](#)
- [Java Message Service 2.0 \(JSR 343\)](#)
- [Expression Language 3.0 \(JSR 341\)](#)
- [JavaServer Faces 2.2 \(JSR 344\)](#)
- [Servlet 3.1 \(JSR 340\)](#)
- [Java Persistence API 2.1 \(JSR 338\)](#)
- [Enterprise JavaBeans 3.2 \(JSR 345\)](#)
- [Contexts and Dependency Injection 1.1 \(JSR 346\)](#)
- [Interceptors 1.2 \(JSR 318\)](#)
- [Bean Validation 1.1 \(JSR 349\)](#)
- [Java Transaction API 1.2 \(JSR 907\)](#)

## General Areas for Providing Feedback

Each specification can take general feedback in the following areas:

- Build sample applications using the implementations integrated in GlassFish 4.
- Ensure all issues in JIRA are:
  - reproducible with the latest codeline.
  - have a short, self-contained, compilable, executable reproducer
  - have the correct metadata (component, environment, version, etc).
- Read the final version of the specification (available from the JSR page) and provide feedback. You don't need to necessarily review the entire specification and instead can focus on specific chapters and sections.
- Review the [javadocs](#) for clarity and completeness.
- Explore integration of one/multiple technologies with your existing Java EE applications.
- File bugs on the corresponding spec or RI project.

# Translated wiki - francais

## Adopt a JSR Resources

- [Community Homepage](#)
- [Mailing Lists](#)
- [Wiki](#)
- [Downloads](#)
- [Issue Tracking](#)
- [RSS Feeds](#)

Join #adoptajsr on irc for discussion.

## Community Leaders

- Richard Warburton ([@RichardWarburto](#))
- Martijn Verburg ([@karianna](#))
- Bruno Souza ([@brjavaman](#))

## Adopter une JSR

### Introduction

Ce programme vise à encourager les membres des Java User Groups (JUGs) à participer dans la création des Java Specification Requests (JSRs) et à diffuser l'information sur ces JSRs ainsi que leurs développements à la communauté Java grâce aux réunions des JUGs, conférences et autres événements. Les JSRs concernent toutes les technologies Java, par exemple Date and Time API sera inclus dans Java 8, les APIs de Java EE 7 en faisaient partie aussi et bien plus encore! Ce programme permet de:

- Donner des retours plus tôt sur les standards Java. Les standards seront plus proches des exigences des développeurs grâce à leurs retours.
- Fournir aux standards Java l'avis des experts aussi bien développeurs qu'utilisateurs
- Aider pour que les implémentations de référence des standards Java soient développées plus rapidement
- Aider à gérer des projets open source qui surgissent autour d'une JSR

Les avantages pour les membres des JUGs qui contribuent à une JSR sont évidents. Juste pensez à l'ajouter à votre CV.

La JSR 348 apporte une transparence au processus de création des nouvelles JSRs. Cette ouverture est une opportunité pour aider l'écosystème Java à être plus innovant.



### Un but bien defini

Ce programme a pour but de regrouper l'énergie et le talent de la communauté Java à travers le monde entier et d'influencer les technologies Java au sein même des JSRs. Par exemple, les membres du programme seront encouragés de tester la complexité de Java 8 en signalant les problèmes des sorties précoces, et en

## Adopt a JSR Projects

- [Adopt a JSR](#)
- [JSR Transparency](#)
- [Adopt OpenJDK](#)
- [London Java Community \(London JUG\)](#)
- [Brazilian Java Users Society](#)

# List of Active JSRs



The Java Community Process(S... | +

jcpr.org/en/jsr/all?status=Active&activeMonths=12 | Google

Press Room | Get Java Here | Search JSRs

Java Community Process | Community Development of Java Technology Specifications

### JSRs: Java Specification Requests

List of all JSRs | By Number | By Date

Search JSRs

> JSRs by Platform  
> JSRs by Technology  
> JSRs by Stage  
> JSRs by Committee  
> List of All JSRs

#### My JCP

User ID:   
Password:   
Register for Site  
Having problems logging in?

#### JCP Info

> About JCP  
> Get Involved  
> Community Resources  
> Community News  
> FAQ  
> Contact Us

Download Java Software for Your Computer

107 **JCACHE - Java Temporary Caching API**  
Description: Specifies API and semantics for temporary, in memory caching of Java objects, including object creation, shared access, spooling, invalidation, and consistency across JVM's.  
Status: Active  
Latest Stage:     
Spec Lead: Gregory Luck, Luck, Gregory Robert  
Spec Lead: Brian Oliver, Oracle  
Spec Lead: Cameron Purdy, Oracle

109 **Implementing Enterprise Web Services**  
Description: This specification defines the programming model and runtime architecture for implementing web services in Java.  
Status: Active  
Latest Stage:     
Spec Lead: Jitendra Kotamraju, Oracle

115 **Java™ Authorization Contract for Containers**  
Description: This JSR seeks to define a contract between containers and authorization service providers that will result in the implementation of providers for use by containers.  
Status: Active  
Latest Stage:     
Spec Lead: Ron Monzillo, Oracle

181 **Web Services Metadata for the Java™ Platform**  
Description: This JSR defines an annotated Java™ format that that uses Java™ Language Metadata (JSR 175) to enable easy definition of Java Web Services in a J2EE container.  
Status: Active  
Latest Stage:     
Spec Lead: Alan Mullendore, Oracle

196 **Java™ Authentication Service Provider Interface for Containers**  
Description: This JSR seeks to define a standard interface by which authentication modules may be integrated with containers and such that these modules may establish the authentication identities used by containers.  
Status: Active  
Latest Stage:     
Spec Lead: Ron Monzillo, Oracle

206 **Java™ API for XML Processing (JAXP) 1.3**  
Description: JAXP 1.3 is the next version of JAXP, an implementation independent portable API for processing XML with Java™.

# JSR page example



Java Community Process

Community Development of Java Technology Specifications

JSR 310: Date and Time API

Stage	Access	Start	Finish
Early Draft Review 2	Download page	14 Sep, 2012	14 Oct, 2012
Early Draft Review	Download page	26 Feb, 2010	28 Mar, 2010
Expert Group Formation		13 Feb, 2007	
JSR Review Ballot	View results	30 Jan, 2007	12 Feb, 2007

Status: Active  
JCP version in use: 2.9  
Java Specification Participation Agreement version in use: 2.0

Description:  
This JSR will provide a new and improved date and time API for Java.

Expert Group Transparency:  
Public Communications  
Issue Tracking

Team  
Specification Leads

List of Active JSRs (posting in last 12 months):

<http://jcp.org/en/jsr/all?status=Active&activeMonths=12> .

# Jira example



The screenshot shows a Jira issue page for the project 'ADOPTAJSR-2'. The issue key is 'ADOPTAJSR-2' and the title is 'Javadoc 7 help for JSF'. The issue is of type 'Task', status 'Open', and priority 'Major'. The assignee is 'Thomas Modeneis' and the reporter is 'karianna'. There is 1 vote and 0 watchers. The issue is currently in the 'Open' status.

**Issue Details:**  
Key: ADOPTAJSR-2  
Type: Task  
Status: Open  
Priority: Major  
Assignee: Thomas Modeneis  
Reporter: karianna  
Votes: 1  
Watchers: 0

**Available Workflow Actions:**  
 Resolve Issue  
 Close Issue

**Operations:**  
 Go to Planning Board  
 Assign this issue (to me)  
 Clone this issue  
 Comment on this issue  
 Create sub-task  
 Delete this issue  
 Edit this issue  
 Link this issue to another issue  
 Move this issue  
 Convert to sub-task

**Issue Information:**  
adoptajsr  
Javadoc 7 help for JSF  
Created: 12/Oct/12 06:16 PM Updated: 14/Nov/12 10:29 PM  
Component/s: JSR-344 (JSF 2.2)  
Affects Version/s: None  
Fix Version/s: None  
Time Tracking: Not Specified  
Tags: javadoc EDIT  
Participants: Ed Burns, karianna and Thomas Modeneis

**Description:**  
I sure could use some help in bringing the new Javadoc 7 style to the generated portion of the JSF spec [1]. Can anyone from the Adopt-a-JSR program help me out here? The task would be very hands-on, as the generated portion of the spec uses several different kinds of documents as its source inputs.

Ed




[1] <https://maven.java.net/service/local/repositories/snapshots/archive/javafx/faces/javafx-faces-api/2.2-SNAPSHOT/javafx-faces-api-2.2-20121011.174655-72-javadoc.jar/javadocs/index.html>

This is not a permalink so it will probably not resolve once a new nightly gets published. If it does not resolve, just go to <http://maven.java.net/> and put in `javafx:faces:faces-api:2.2-SNAPSHOT` into the search box, select the `-javadoc.jar`, and view it in the archive browser. Instructions are here: <http://weblogs.java.net/blog/edburns/archive/2012/05/17/how-view-absolute-latest-jsf-22-spec-snapshot>

# Share Success



## Project Features

Downloads  
Mailing Lists  
 [Adopt a JSR Archive](#)  
 [Adote uma JSR - Portugues A...](#)  
 [Issues Archive](#)  
Issue Tracking  
Wiki  
WikiHomePage

## Wiki Controls

Show Page  
Edit Page  
Show Page as Text  
Show Page Revision History

## Jump to Page:


List all Pages  
Manage Images

## Create New Page:

## Project Links

JSR Stages  
[Checklist for Adopt a JSR](#)

java.net > projects > adoptajsr > wiki > SuccessStories

Last updated 3 weeks ago, by Martijn Verburg 

### Contents [hide]

- 1 [Java EE 7 in general](#)
- 2 [Java in General](#)
- 3 [JSR 310 \(Date and Time\)](#)
- 4 [JSR 335 \(Lambdas\)](#)
- 5 [JSR 348 and 355 \(JCP.next\)](#)

## Java EE 7 in general

- Several projects have been donated to GitHub in which users can run test applications against new specifications. This has helped:
  - Fix bugs and tweak APIs to be more developer friendly
  - Raised awareness of Java EE 7
  - Some projects are part of the RI sample kit
- There have been a multitude of Java EE 7 related presentations made at JUG meetings and major conferences

## Java in General

- See the [Adopt OpenJDK programme](#) 

# Enter activity in the matrix

https://java.net/projects/adoptajsr/pages/CompletedJSRs

[Show Page](#)  
[Edit Page](#)  
[Show Page as Text](#)  
[Show Page Revision History](#)

Jump to Page:

[List all Pages](#)  
[Manage Images](#)

Create New Page:  
 [GO](#)

Project Links

- [JSR Stages](#)
- [Glassfish/JEE AdoptAJSR](#)
- [GitHub](#)
- [Facebook Page](#)

About this Project

Adopt a JSR is a subproject of Java Users Groups (JUG) Community, was started in December 2011 and has 70 members. The project administrators are Bruno Souza, Martijn Verburg, rwarburton, jcp\_pmo, and Frank Nikola.

[Manage This Project](#)

These JSRs are now standards, thanks for helping improve Java! There's still plenty of work that can be done on these, especially if they are new. There's still educational efforts (workshops), working on alternative implementations, working on maintenance fixes and more.

### Live JSRs (<= #340)

JUG/Individual/ Organisation	JSR-236 (Concurrency utilities for Java EE)	JSR-321 (Trusted Computing)	JSR-331 (Constraints Programming)	JSR-338 (JPA 2.1)	JSR-339 (JAX-RS 2.0)
<a href="#">Chennai</a>		<a href="#">Chennai</a>			
<a href="#">Houston</a>		<a href="#">321 Link TBA</a>			
<a href="#">Hyderabad JUG</a>			<a href="#">Hyderabad JUG</a>		
<a href="#">LJC</a>					<a href="#">339 Page</a>
<a href="#">MoroccoJUG</a>					<a href="#">339 page</a>
<a href="#">PeruJUG</a>					<a href="#">339 Link TBA</a>
<a href="#">SouJava</a>					<a href="#">339 Link TBA</a>

### Live JSRs (#340 --> #349)

JUG/Individual/ Organisation	JSR-340 (Servlet 3.1)	JSR-341 (EL 3.0)	JSR-342 (JavaEE 7)	JSR-343 (JMS 2.0)	JSR-344 (JSF 2.2)	JSR-345 (EJB 3.2)	JSR 346 (CDI 1.1)	JSR-348 (JCP.next)	JSR-349 (Bean Validation 1.1)
<a href="#">FASOJUG</a>				<a href="#">343 Link</a>					
<a href="#">JUGC</a>					<a href="#">344 page</a>		<a href="#">346 page</a>		
<a href="#">LJC</a>		<a href="#">341 Page</a>	<a href="#">342 Page</a>	<a href="#">343 Page</a>	<a href="#">344 Page</a>	<a href="#">345 Page</a>	<a href="#">346 Page</a>	<a href="#">LJC JSR-348 Page</a>	<a href="#">349 Page</a>
<a href="#">PeruJUG</a>				<a href="#">343 Link TBA</a>					
<a href="#">SouJava</a>					<a href="#">344 Link TBA</a>			<a href="#">348 Link TBA</a>	<a href="#">349 Link TBA</a>

### Live JSRs (#350 --> #359)

# Add resources to GitHub

GitHub

Search or type a command

Explore Features Enterprise Blog

Sign up

Sign in



**Adopt a JSR**  
Adopt-a-JSR

Worldwide

members@adoptajsr.java.net

http://adoptajsr.java.net

Joined on Jan 22, 2013

**9** **15**  
public repos members

Repositories Members

Find a Repository...

All Sources Forks Mirrors

**java-8-benchmarks**

Java ★ 3 🍴 5

Last updated 3 days ago

**JEE7-Hack-Sessions**

Java ★ 1 🍴 1

This repository contains all my hack sessions, projects and conferences materials.

Last updated 8 days ago

**UrbanTravellers**

JavaScript ★ 0 🍴 11

forked from jugchennaiadoptjava/UrbanTravellers

Last updated 24 days ago

**cdiex-palindrome-jsf**

Java ★ 2 🍴 2

CDI examples - CDI + JSF palindrome application

Last updated 2 months ago

**cdiex-datastore**


Java ★ 2 🍴 2

CDI examples - simple datastore

Last updated 2 months ago



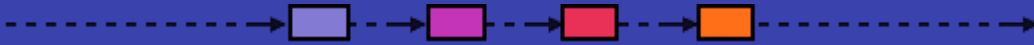
# Resources: find out more

- 
- JCP & Adopt-a-JSR online meeting-materials and recordings available: <http://jcp.org/en/resources/multimedia>
  - Follow the JCP online: <http://JCP.org>
    - Twitter @jcp\_org
    - [blogs.oracle.com/jcp](http://blogs.oracle.com/jcp)
    - Facebook: Java Community Process
  - Adopt-a-JSR online: <http://adoptajsr.java.net>.
    - Twitter @adoptajsr
    - Facebook: AdoptaJSR
    - [members@adoptajsr.java.net](mailto:members@adoptajsr.java.net) - the global mailing list
    - IRC: #adoptajsr channel on [irc.freenode.net](http://irc.freenode.net)
    - GitHub: <https://github.com/Adopt-a-JSR>



thank you, dziekuje, tak, tack,  
merci, grazie, gracias, danke,  
obrigado, bedankt, kiitos, xie xie,  
ありがとう .

# How can you help - easy



- Help to moderate the mailing lists; steer towards helpful conversations
- Evangelize the JSR through social media, blogging or lightning talks.
  - Give a talk at your user group at events, or online.
- Help with documentation.
  - Translate into your native language.
- Help to maintain an FAQ/Wiki.
- Improve project infrastructure & JSR visibility.
  - Help with Search Engine Optimization of website.
  - Ensure that downloads, mailing lists, issue trackers are easy to find.

# How else can you help - moderate



- Organize Hack days.
- Test the early RI builds; use them to find pain points, report bugs, suggest feature enhancements.
  - Help triage issues; reproduce issues, erase/merge duplicates, set priorities/categories etc.
- Give feedback on design; discuss issues and deliver feedback, think about how you would use the JSR as a developer.
- Help build the RI; get coding with the actual implementation of the spec or build sample applications.
- Help build the TCK; all implementations must pass the test suite – this is a great way to gain test experience.