Adopt-a-JSR July Meeting

Special Guest: Arun Gupta
Bruno Souza,
Heather VanCura,
Martijn Verburg

July 2013
You expanded wiki into eight languages: Arabic, Chinese, English, French, German, Portuguese, Spanish, Turkish...you choose who will translate next language!
You expanded wiki into eight languages: Arabic, Chinese, English, French, German, Portuguese, Spanish, Turkish...you choose who will translate next language!
How to start in your language?

- Translate top page, ‘Sign up’, ‘Get started’ ‘Ways to get involved’; plus, a few slides to present to your JUG about Adopt-a-JSR.
- Participate on JSR mailing lists in English and then bring back topics into the JUG in native language.
- Assign people that will present in meetings on Adopt-a-JSR; promote, report and talk about Adopt-a-JSR activities at the JUG meetings.
- Sign up people—take a look at this JSR, find a JSR of interest, participate in JSR mailing lists--track on your wiki, mail list, etc.
- It is a long term effort! Have patience—do not get nervous things are not happening immediately. Enter reminders to follow up weekly (if things are very active) or monthly, more the normal cadence.
- Talk about JCP at every meeting-JSR activity, EC Meeting topics, get comfortable with subject of the JCP--explain what it is, how it works, then get into practical aspects, and how members can get involved.
- Evangelize and communicate results.
Adopt-a-JSR!

- A JUG-lead *global* initiative to improve Java standards.
- Groups of JUG members work on JSRs.
- See http://adoptajsr.java.net for full details.
JUGs around the world!
How can you join Adopt-a-JSR?

• Contact your local JUG leader, or...someone on this call to let us know of your interest.

• Go to http://adoptajsr.java.net and follow the simple set-up steps described there:
  • Join the Adopt-a-JSR java.net project.
  • Join the mailing list: http://java.net/projects/adoptajsr/lists.
How to start an Adopt-a-JSR project

- Set up a wiki page to coordinate your activities e.g.: http://java.net/projects/ljc-london-jug/pages/JSR-310.
- Link that into the matrix at: http://java.net/projects/adoptajsr.
- Contact the Spec Lead & Expert Group - find links on JSR pages e.g.: http://jcp.org/en/jsr/detail?id=310.
- 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.
- Organize talks and hackdays.
- Communicate results!!
Enter activity in the matrix
List of Active JSRs (posting in last 12 months)
http://jcp.org/en/jsr/all?status=Active&activeMonths=12
List of Active JSRs

JSRs: Java Specification Requests

List of all JSRs

<table>
<thead>
<tr>
<th>JSR</th>
<th>Description</th>
<th>Status</th>
<th>Latest Stage</th>
<th>Spec Lead</th>
<th>BSpec Lead</th>
</tr>
</thead>
<tbody>
<tr>
<td>187</td>
<td>JCACHE - Java Temporary Caching API</td>
<td>Active</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>116</td>
<td>Implementing Enterprise Web Services</td>
<td>Active</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>115</td>
<td>Java Authorization Contract for Containers</td>
<td>Active</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>181</td>
<td>Web Services Metadata for the Java™ Platform</td>
<td>Active</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>186</td>
<td>Java™ Authentication Service Provider Interface for Containers</td>
<td>Active</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>206</td>
<td>Java™ API for XML Processing (JAXP) 1.3</td>
<td>Active</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 JVMs.
Status: Active
Spec Lead: Gregory Luck, Oracle
BSpec Lead: Brian Oliver, Oracle
Cameron Purdy, Oracle

Implementing Enterprise Web Services
This specification defines a programming model and runtime architecture for implementing web services in Java.
Status: Active
Latest Stage: | Maintenance Release 3 | Download page | Start: 2013-06-07 |
Spec Lead: Jhendry Kotsmanou, Oracle

Java™ Authorization Contract for Containers
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: | Maintenance Release 3 | Download page | Start: 2013-06-10 |
Spec Lead: Ron Moroño, Oracle

Web Services Metadata for the Java™ Platform
This JSR defines an annotated Java™ format that uses Java™ Language Metadata (JLS 175) to enable easy definition of Java Web Services in a J2EE container.
Status: Active
Latest Stage: | Maintenance Release 2 | Download page | Start: 2013-06-16 |
Spec Lead: Alan Mynderse, Oracle

Java™ Authentication Service Provider Interface for Containers
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: | Maintenance Release 2 | Download page | Start: 2013-06-12 |
Spec Lead: Ron Moroño, Oracle

Java™ API for XML Processing (JAXP) 1.3
JAXP 1.3 is the next version of JAXP, an implementation independent portable API for processing XML with Java™.
Java EE 7 JSRs

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 363)
- 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 4.1 (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 518)
- Bean Validation 1.1 (JSR 349)
- Java Transaction API 1.2 (JSR 807)

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 codebase.
  - 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.
Share Successes

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
Add resources to GitHub

- **java-8-benchmarks**
  - Last updated 3 days ago

- **JEE7-Hack-Sessions**
  - This repository contains all my hack sessions, projects and conferences materials.
  - Last updated 8 days ago

- **UrbanTravellers**
  - Forked from jugosmnin/adoptjava/UrbanTravellers
  - Last updated 24 days ago

- **cdix-palindrome-jsf**
  - CDI examples - CDI + JSF palindrome application
  - Last updated 2 months ago

- **cdix-datastore**
  - CDI examples - simple datastore
  - Last updated 2 months ago

---

Adopt a JSR
Adopt-a-JSR

Worldwide
- members@adoptjsr.java.net
- http://adoptjsr.java.net
- Joined on Jan 22, 2013

- **GitHub**
  - Search or type a command
  - Explore
  - Features
  - Enterprise
  - Blog
  - Sign up
  - Sign in

---

14
Guidelines for feedback

• The specification lead and the Expert Group has the final decision on incorporating the feedback.

• For specifications in Public Drafts, new feature requests may not be considered for this release. For specifications in Early Draft, new features may be accepted if they are within the existing scope. If new features are not accepted for this release, then they may be considered for the next release.

• Multiple JUGs can and should collaborate on a particular JSR. There is always plenty of work and going through the material multiple times only makes it that more better.

• When communicating with the EG on mailing lists, include "Adopt-a-JSR" in the subject line.

• When filing issues/bugs, use "adoptajsr" tag.
Jira example
• Transparency sub-project of the Adopt-a-JSR program hosted on java.net.
  • http://www.java.net/form/jcp-2829-transparency?force=408.
• JUGs or individuals do transparency checks or audits on JSRs.
• Provide qualitative and quantitative data to the community, Spec Leads, and the Program Office on the JSR’s compliance with the JCP’s transparency and participation requirements.
• A matrix of results will be provided for each JSR actively operating under JCP 2.8 and above.
  • http://java.net/projects/jsr-transparency/pages/Home.
• Once responses are received :-)

Transparency sub-project
JCP 2.8/2.9 Transparency

This is a transparency sub-project introduced as part of the Adopt-a-JSR program. We need JUGs or individuals to do transparency checks or audits on JSRs. Provide qualitative data to the community, Spec Leads and Program Office on the compliance to JCP 2.8/2.9 and above transparency and participation requirements.

Please complete online survey below. Your input will go into the Adopt-a-JSR Transparency sub-project with a matrix of results for each JSR actively operating under JCP 2.8/2.9 and above.

For JCP 2.7 and prior please use the JCP 2.7 Transparency Survey.

JSR #: *
- Select -

Is the schedule for the JSR publicly available, current, and updated regularly?:

- [ ] yes
- [ ] no

Discussion Board

Is there a publicly accessible discussion board for the JSR that is responded to regularly and are there community contributions on this mechanism:

- [ ] publicly accessible discussion board
- [ ] discussion board responded to regularly
- [ ] community contributions on this mechanism

How does the EG and Spec Lead interact and communicate on a public mailing list:
Nominate for JCP Awards-extended!

2013 JCP Annual Awards

The Java Community Process (JCP) program celebrates success. Members of the community nominate worthy participants, Spec Leads, and Java Specification Requests (JSRs) in order to cheer on the hard work and creativity that produces ground-breaking results for the community and industry in the Java Standard Edition (SE), Java Enterprise Edition (EE), or Java Micro Edition (ME) platforms.

The community gets together every year at the JavaOne conference to applaud in person the winners of four awards: JCP Member/Participant of the Year, Outstanding Spec Lead, Most Significant JSR and Outstanding Adopt-a-JSR Participant. This year’s unveiling will occur at JavaOne in September 2013.

The nomination period is open until July 12, 2013!

Description of the JCP Award Categories

JCP Member/Participant Of The Year
This award recognizes the corporate or individual member (either Member or Participant) who has made the most significant positive impact on the community in the past year. Leadership, investment in the community, and innovation are some of the qualities that EC Members look for in voting for this award.

Outstanding Spec Lead
The role of Spec Lead is not an easy one, and the person who takes that responsibility must be, among other things, technically savvy, able to build consensus in spite of diverse corporate goals, and focused on efficiency and execution. This award recognizes the person who has brought together those qualities the best in the past year. In leading a JSR for the Java community (Java SE, Java EE or Java ME).

Most Significant JSR
Specification development is key to the success of the JCP program and helps ensure we remain a fresh and vibrant community. This award recognizes the Spec Lead and Expert Group that have contributed either in progress or final the most significant JSR for the Java community (Java SE, Java EE or Java ME) in the past year.

Outstanding Adopt-a-JSR Participant
This award recognizes the Java User Group (JUG) that has made the most exemplary contribution through the Adopt-a-JSR program in the past year. The London Java Community and SouJava initiated, and are thereby implicitly recognized, in this effort for JUGs around the world to become more involved in the work of JSRs. Innovation, community engagement and technical impact are some of the characteristics that EC members look for in voting for this award.

Submit your Nomination

Award Category: *
- Select -
thank you, tak, tack, merci, grazie, gracias, danke, obrigado, bedankt, kiitos, xie xie, ありがとう．

We will meet again in June...
Find out more

• 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
  • Facebook: Java Community Process

• Adopt-a-JSR online: http://adoptajsr.java.net.
  • Twitter @adoptajsr
  • Facebook: AdoptaJSR
  • members@adoptajsr.java.net - the global mailing list
  • IRC: #adoptajsr channel on irc.freenode.net
  • GitHub: https://github.com/Adopt-a-JSR