JSR-369 Review
9 May 2017

Edward Burns
Shing-Wai Chan

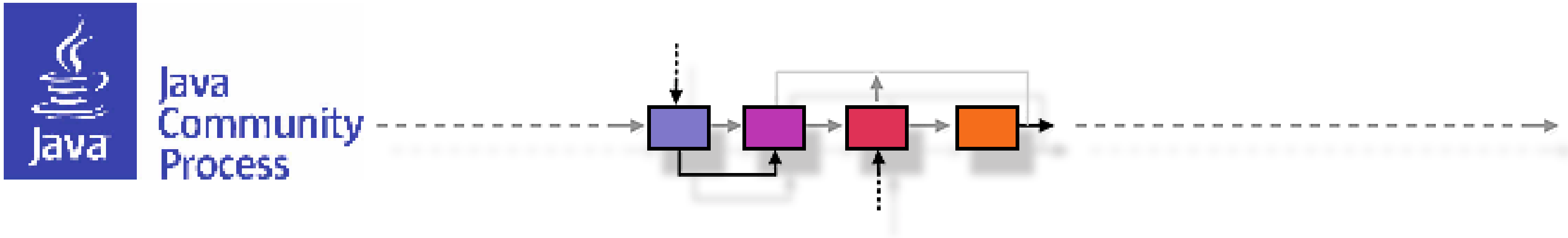
Agenda

- About this JSR
- Introduction
  - History, Ecosystem, Scope, Schedule
- Community Interaction
  - EG, Publicity, Collaboration
- Process
  - Deliverables, Implementations, IP Flow, Development
- Summary and Next Steps

About JSR-369

About JSR-369

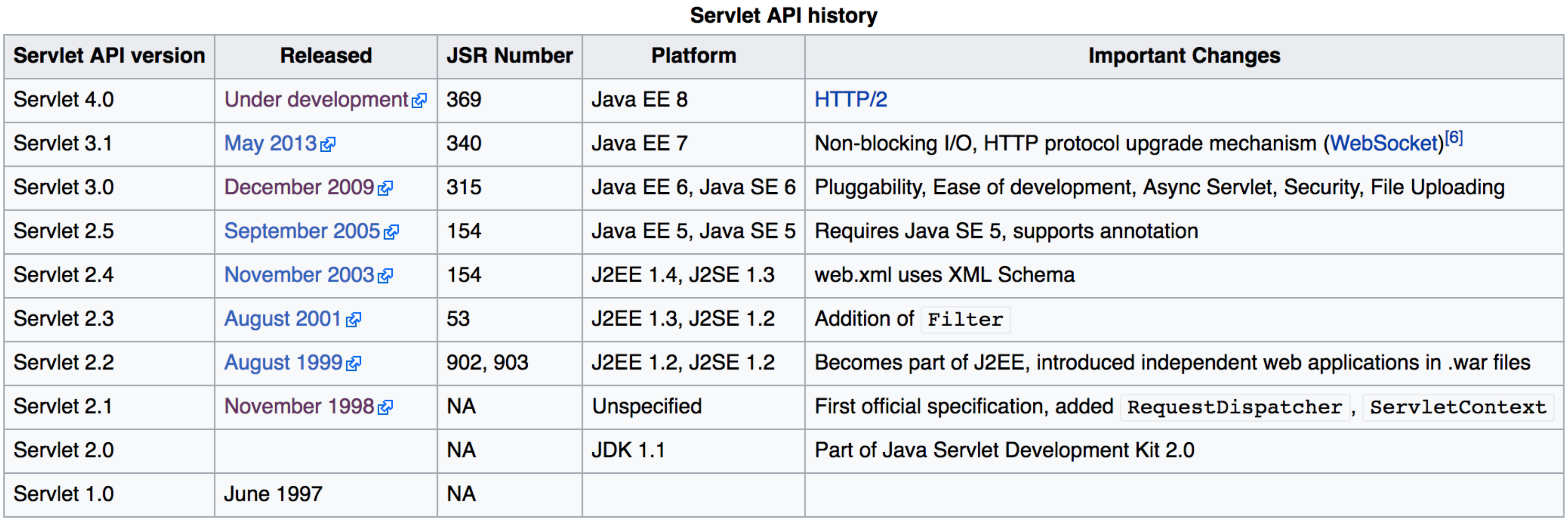
- “This JSR aims to expose the latest advances in HTTP to users of JavaEE as well as providing a vehicle for continual improvement in this foundational Java EE API.”
  - Incremental evolution
  - Incorporate revolutionary change incrementally where possible
    - HTTP/2
    - Reactive Ideas
    - Community driven features



Introduction

How does JSR-369 Fit In to Java?

- Servlet is so old that it predates even J2EE.
- It was in J2EE and now Java EE
- Wikipedia page Java_servlet



Servlet API history

Servlet API version	Released	JSR Number	Platform	Important Changes
Servlet 4.0	Under development	369	Java EE 8	HTTP/2
Servlet 3.1	May 2013	340	Java EE 7	Non-blocking I/O, HTTP protocol upgrade mechanism (WebSocket)[6]
Servlet 3.0	December 2009	315	Java EE 6, Java SE 6	Pluggability, Ease of development, Async Servlet, Security, File Uploading
Servlet 2.5	September 2005	154	Java EE 5, Java SE 5	Requires Java SE 5, supports annotation
Servlet 2.4	November 2003	154	J2EE 1.4, J2SE 1.3	web.xml uses XML Schema
Servlet 2.3	August 2001	53	J2EE 1.3, J2SE 1.2	Addition of `Filter`
Servlet 2.2	August 1999	902, 903	J2EE 1.2, J2SE 1.2	Becomes part of J2EE, introduced independent web applications in .war files
Servlet 2.1	November 1998	NA	Unspecified	First official specification, added `RequestDispatcher`, `ServletContext`
Servlet 2.0		NA	JDK 1.1	Part of Java Servlet Development Kit 2.0
Servlet 1.0	June 1997	NA		

Business/Marketing/Ecosystem Justification

- Why do we need JSR-369?
  - Community Need: Deliver HTTP/2 to Java EE
  - This need is not met by any other specifications
  - 8 of the top 10 Alexa sites support HTTP/2
    - 2017-05-06 using chrome://net-internals
    - Outliers: Baidu and Qq, both from China
  - The idea is ready for standardization
    - HTTP/2 is final
    - Community ideas are incremental improvements

JSR-369 History

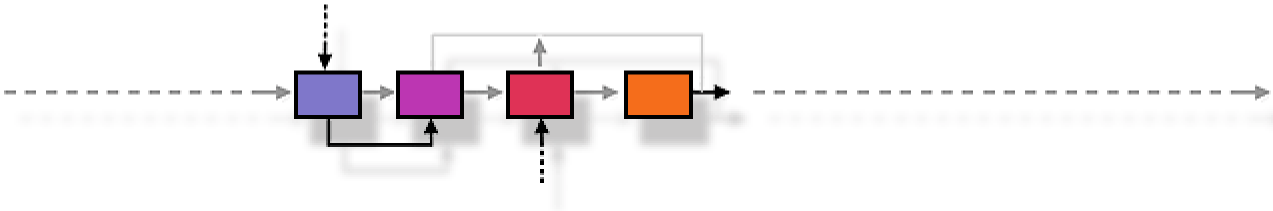
Date	Milestone
2014-09-22	Complete JSR review ballot
2015-05-07	EG formation
2015-12-15	Delivered Early Draft Review
2017-01-30	Completed Renewal Ballot
2017-04-20	Started Public Review

Technical Scope and Features

- HTTP/2 Features
  - Most are hidden from the API
    - ALPN and HPACK
  - Server Push is exposed to the API
- Java SE 8 alignment
  - Default methods
- Mapping Discovery
- Mark some methods @Deprecated
- Explicit support for HTTP Trailers

Schedule

- 2017-04-20 Start 30 Day Public Review
- 2017-05-20 End 30 Day Public Review
- 2017-05-23 Start Public Review Ballot
- 2017-05-30 TCK Functionally Complete
- 2017-06-05 End Public Review Ballot
- 2017-06-06 Start 21 Day Proposed Final Draft
- 2017-06-26 End 21 Day Proposed Final Draft
- 2017-06-27 Hand Final Approval Ballot Materials to JCP
- 2017-07-11 Start 14 Day Final Approval Ballot
- 2017-07-24 End 14 Day Final Approval Ballot
- 2017-07-26 General Availability Release



Community Interaction

Expert Group

- Is the Expert Group representative of the community and the Java ecosystem?
  - Active participants
    - Greg Wilkins, Jetty
    - Stuart Douglas, Undertow
    - Mark Thomas, Tomcat
    - Martin Mulholland, WebSphere
  - Less active participants
    - Neil Griffin, Liferay
    - Wenbo Zhu, Google
    - Eiugen Chung, Tmaxsoft
    - Justin Lee, MongoDB
  - JCP Contributor status: Reza Rahman

Publicity

- How does the EG operate?
  - Nearly 100% on email list only
  - Occasional conference calls
- Publicity: Numerous conference presentations
  - JavaOne
    - https://www.youtube.com/watch?v=3uHhx-TMV8o&t=13s
    - https://www.youtube.com/watch?v=702S0XENdKY
    - https://www.youtube.com/watch?v=T9HPLLXjGzI&t=1s
  - Devoxx
    - https://www.youtube.com/watch?v=yJU4VJjwBOY
  - DevNexus
    - https://www.youtube.com/watch?v=DSu9RZIU-OE
  - GeekOut
  - JavaLand

Collaboration with the Community

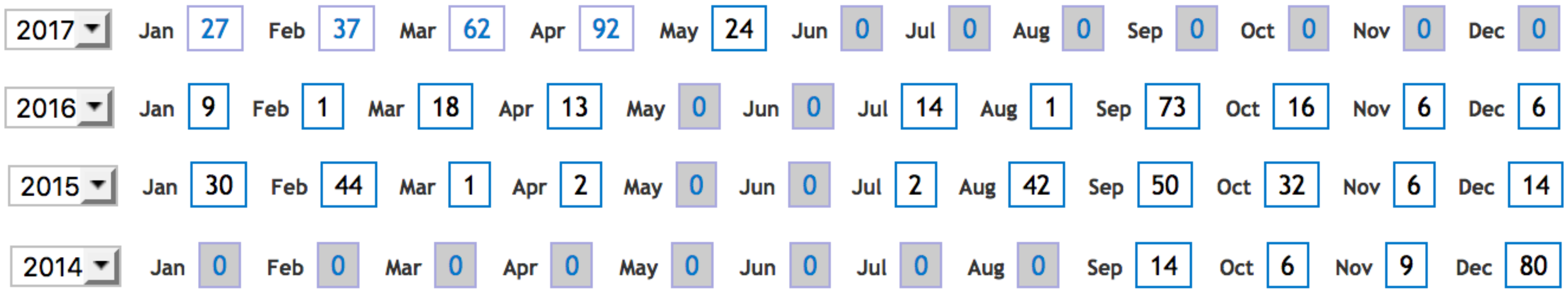
- Are you working with other community groups?
  - Semi-regular phone calls with Java EE Guardians representative Reza Rahman
- Adopt-a-JSR
  - Reza put Ed in touch with an individual who was interested. Ed gave him a concrete task to do for testing Server Push.

Participation and Transparency

- JSR Page
  - https://jcp.org/en/jsr/detail?id=369
- Moved to GitHub!
  - https://github.com/javaee/javax.servlet
    Name will change to servlet-spec.

Participation and Transparency

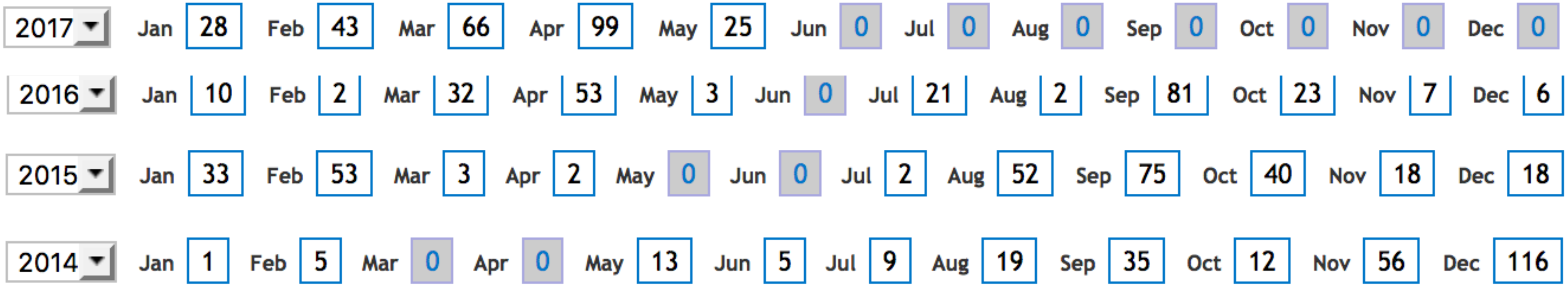
- Mailing Lists
  - Will be migrating away from java.net, still being discussed
    - jsr369-experts
      - 731 messages, 16 members, ~83 discussion threads
      - Estimated 25% spec lead 75% EG members
      - Summary of traffic



2017	Jan	27	Feb	37	Mar	62	Apr	92	May	24	Jun	0	Jul	0	Aug	0	Sep	0	Oct	0	Nov	0	Dec	0
2016	Jan	9	Feb	1	Mar	18	Apr	13	May	0	Jun	0	Jul	14	Aug	1	Sep	73	Oct	16	Nov	6	Dec	6
2015	Jan	30	Feb	44	Mar	1	Apr	2	May	0	Jun	0	Jul	2	Aug	42	Sep	50	Oct	32	Nov	6	Dec	14
2014	Jan	0	Feb	0	Mar	0	Apr	0	May	0	Jun	0	Jul	0	Aug	0	Sep	14	Oct	6	Nov	9	Dec	80

Participation and Transparency

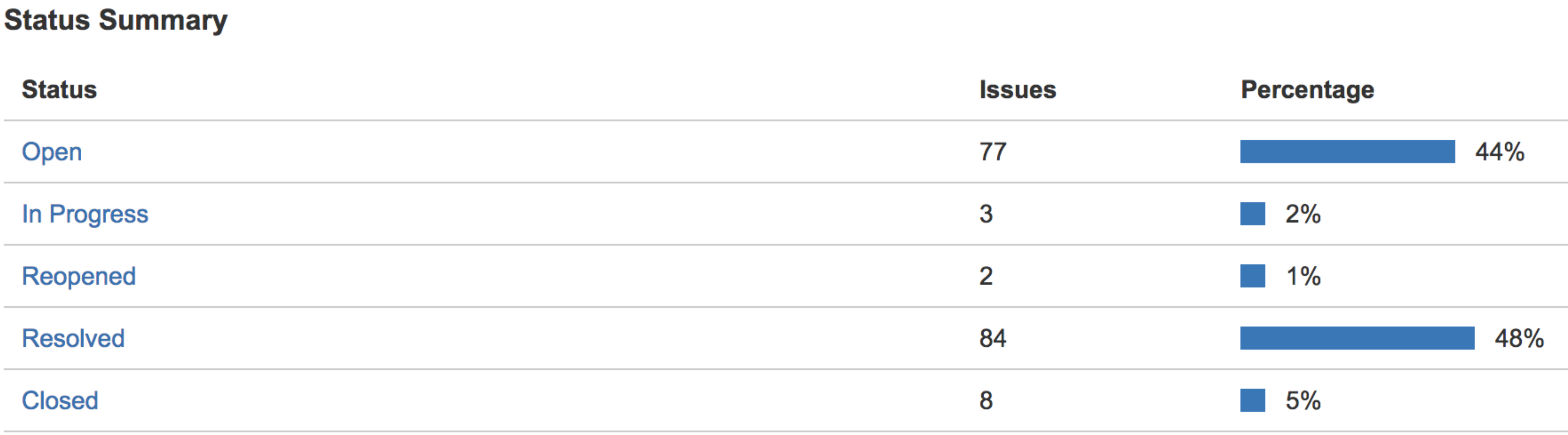
- Mailing Lists
  - Will be migrating away from java.net, still being discussed
    - users
      - 1035 messages, 117 subscribers members, ~32 discussion threads
      - Estimated 10% spec lead, 75% EG members, 15% non-EG members
      - Summary of traffic



	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2017 ▾	28	43	66	99	25	0	0	0	0	0	0	0
2016 ▾	10	2	32	53	3	0	21	2	81	23	7	6
2015 ▾	33	53	3	2	0	0	2	52	75	40	18	18
2014 ▾	1	5	0	0	13	5	9	19	35	12	56	116

Issue Tracker

- Moved to GitHub!
  - https://github.com/javaee/javax.servlet/issues



Status Summary

Status	Issues	Percentage
Open	77	44%
In Progress	3	2%
Reopened	2	1%
Resolved	84	48%
Closed	8	5%

Document Archive

- Still in progress to move from java.net
  - https://java.net/projects/servlet-spec/downloads
  - Contains audio of teleconferences. Minutes published to EG.
  - Contains drafts of specifications and javadocs
- No other transparency and participation tools used.

Process

Deliverables

- Spec, RI and TCK only
  - No other deliverables: no user’s guide, no additional documentation
  - RI: Glassfish 5.x https://github.com/javaee/glassfish/
- Non-RI implementations
  - Tomcat 9.x http://tomcat.apache.org/whichversion.html
  - Jetty 9.x http://www.eclipse.org/jetty/download.html
  - Undertow 2.x http://undertow.io/downloads.html
  - WebSphere Liberty > 17.x https://developer.ibm.com/wasdev/downloads/download-latest-stable-websphere-liberty-runtime/

IP Flow

- Spec license https://jcp.org/aboutJava/communityprocess/licenses/jsr369/JSR_369-Spec_License-Java_Servlet-4.0-8.18.14.pdf
- RI license
  - Commercial Use
    - The RI will be available for commercial use under the CDDL 1.1 open source license, the GPLv2 with Classpath Exception open source license, or this RI license.
  - Non-commercial Use
    - The RI will be available for non-Commercial use under the CDDL 1.1 open source license or the GPLv2 with Classpath Exception open source license.

IP Flow

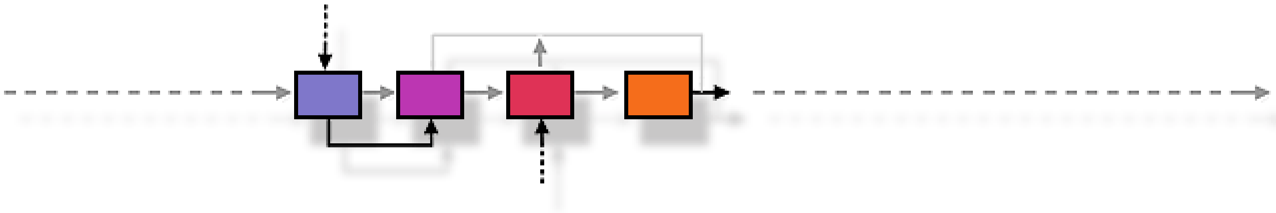
- Spec license
  https://jcp.org/aboutJava/communityprocess/licenses/jsr369/JSR_369-Spec_License-Java_Servlet-4.0-8.18.14.pdf
- TCK license
  - Commercial Use
    - The TCK will be available for commercial use under this TCK license.
  - Non-commercial Use
    - As required by the Java Specification Participation Agreement (JSPA), the TCK will be licensed at no charge without support to qualified not-for-profit. The Compatibility Testing Scholarship Program will verify such qualification. Support may also be provided at no charge with approval of the scholarship board. For more information, please refer to: http://www.oracle.com/technetwork/java/index-137188.html

IP Flow

- Handling contributions from non JCP members
  1. Give the content to a JCP member who accepts responsibility and ownership of it and contributes it on their behalf.
  2. Follow the Oracle Contributor Agreement process and give it to the RI project, which the spec lead can then contribute to JCP.
- Collaboration Tool Terms of Use
  - https://java.net/terms_of_use

RI and TCK Development

- RI is developed as a part of GlassFish, using the GlassFish open source process and licenses.
- TCK is developed internally at Oracle.
- RI Source code is available at
  - https://github.com/javaee/javax.servlet
  - https://github.com/javaee/glassfish/
- RI Binaries available at
  - http://download.oracle.com/glassfish/5.0/index.html



Conclusion and Summary

Conclusion and Summary

- JSR-369, Servlet 4.0, brings HTTP/2 to Java EE 8
  - It exposes the Server Push feature of HTTP/2
  - It delivers community driven features
- We aim to complete Servlet 4.0 by August 2017

Backup Slides

•Servlet 4.0 Small C

Community Features

- **Add Java SE 8 default methods**
  - `ServletContextAttributeListener, ServletContextListener, ServletRequestAttributeListener, ServletRequestListener, HttpSessionActivationListener, HttpSessionAttributeListener, HttpSessionBindingListener, HttpSessionListener`
- **Add `default` to `Filter#init`, `#destroy`**
- **Add `GenericFilter` and `HttpFilter`**
- **`<default-context-path> element in web.xml`**
- **`Add Cookie[] getCookie(String name) to HttpServletRequest`**

•Servlet 4.0 Small C

Community Features

- Discover current Active Mapping
  - `Mapping javax.servlet.http.HttpSer`
    `getMapping()`
  - `javax.servlet.http.Mapping`
    - `MappingMatch getMatchType()`
    - `String getMatchValue()`
    - `String getPattern()`
  - `javax.servlet.http.MappingMatch enum`
    - `CONTEXT_ROOT, DEFAULT, EXACT, EXTENSION, IMPLICIT, PATH, UNKNOWN`

•Servlet 4.0 Small C

Community Features

@Deprecate some methods and classes

- ServletContext: getServlet(), getServlets(), getServletNames()
- ServletRequestWrapper: getRealPath()
- SingleThreadModel
- UnavailableException
- HttpServletResponse: encodeUrl(), encodeRedirectUrl(), setStatus()
- HttpServletResponseWrapper: encodeUrl(), encodeRedirectUrl(), setStatus()
- HttpSession: getSessionContext(), getValue(), getValueNames(), putValue(), removeValue()
- HttpSessionContext

•Servlet 4.0 Small Change

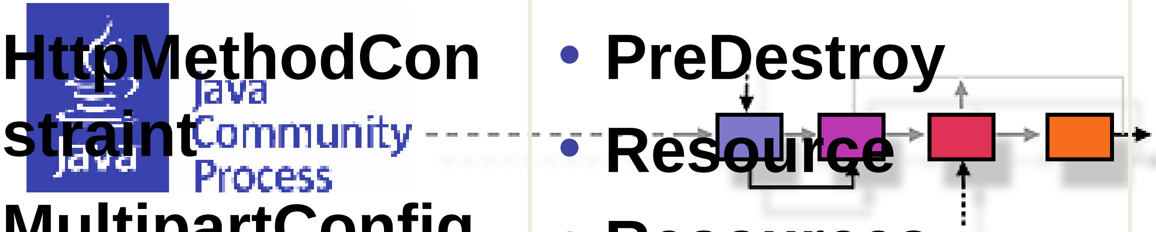
Community Features

Clarify exactly which annotations are *not* scanned when metadata-complete="true"

- HttpConstraint
- HttpMethodConstraint
- MultipartConfig
- ServletSecurity
- WebFilter
- WebInitParam
- WebListener
- WebServlet
- …

- PostConstruct
- PreDestroy
- Resource
- Resources
- DeclareRoles
- RunAs
- DataSourceDefinition
- DataSourceDefinitions
- …

- AroundConstruct
- AroundInvoke
- AroundTimeout
- ExcludeClassInterceptors
- ExcludeDefaultInterceptors
- Interceptors
- …



Goals

Information to be gathered

Business/marketing/ecosystem justification

- Why do this JSR?
- What's the need?
- How does it fit in to the Java ecosystem?
- Is the idea ready for standardization?

History

- List the significant dates in the history of the JSR.
  - When submitted?
  - What stages reached?
  - What drafts of the Spec published?
- We have this info in our database. Need to supply to the Spec Lead automatically.

Technical scope and features

- Provide a high-level summary of technical features.
  - No more than 2 or 3 slides.

The Expert Group

- [How diverse is the EG? Is it representative of the ecosystem?]
- Who is on the EG and who do they represent?
  - Industry, JUGs, individuals, open-source groups, academia, etc.
  - We have this info in our database. Need to supply to the Spec Lead automatically.
- How does the EG operate? How often does it meet and how (teleconferences, online, f2f?)
- What collaboration tools are used to facilitate EG communications?

Other deliverables

- Other than Spec, RI, and TCK are you delivering, for example:
  - Additional documentation?
  - User's guide?
  - Sample code?
  - FAQ?
  - Other artifacts?

Publicity

- Provide links to FAQs, tutorials, conference proceedings, etc.

Collaboration with other community groups

- Are you working with other community groups or organizations?
- If so, describe the collaboration activities and the results (with URLs where appropriate.)

Implementations

- How many implementations (apart from the RI) exist?
  - Provide URLs.

Schedule

- Indicate the expected schedule to completion of the JSR, with milestones and JSR stages.

IP flow

- Provide pointers to the licenses for the the Spec, RI, and TCK.
  - We have this info in our database. Need to supply to the Spec Lead automatically.
- How are you handling contributions from non JCP members?
- What Terms of Use apply to your collaboration tools?
- Do you have a Contributor Agreement?
- Any legal issues or concerns?

RI and TCK development

- How are you developing the RI and TCK?
- If collaboratively (through an open-source project)
  - How many committers and who?
  - How many apart from the Spec Lead (organization)?
- Is the RI available for public download? (If so, provide URL.)
- Is the TCK available for public download? (If so, provide URL.)
- Do you have a source-code repository? (If so, provide URL.)

Participation and transparency

- Provide a pointer to the JSR page on JCP.org
  - [Reviewers: check that is this up to date. Does it point to the JSR’s project page and/or explain how to participate?]
- Provide a pointer to the “JSR project website" (eg, on Java.net.)
  - [Reviewers: how much content is here (how many pages)? Is the online project easy to navigate? Does it clearly explain how to participate?].

Adopt-a-JSR

- Are you participating in the Adopt-a-JSR program?
- If not, why not?
- If so, what kind of contributions/help have you received and from whom/which JUGs?
  - Be specific/provide metrics.

Mailing lists or forums

- How are you communicating with the public and how can they communicate with you?
- Provide pointers to public mailing list(s) and/or forum(s)
- Total number of messages, threads?
- Total number of participants (EG members, non-EG members?)
- How many messages per month (from Spec Lead, EG members, and non-EG members?)

Issue tracker

- Total number of issues?
- How many in each state (open, closed, deferred, etc?)
- Average number of issues logged per month?
- Average number of issues resolved per month?
- How many different people logged them?
- How does this break down between Spec Lead, EG members, and non-EG members?

Document archive

- Provide a pointer to your document archive.
- Are meeting minutes and materials published?
- What other materials are available for download?
- Total number of files available for download?
- Average number of new files available for for download each month?

Other transparency and participation metrics

- Provide information – including metrics – about any additional transparency and participation mechanisms you use.

Implementation notes

Implementation notes

- Create a single template to be used (possibly in subset form) for JSR submission, Public Review, Final Release, and for any intermediate Spec Lead presentations to the EC.
- Provide the template in both document and presentation forms.
- Wherever we have data in our database, supply this to the Spec Lead for copying into the template.

Issues

Issues

- This is a lot of data to ask for. How can we persuade Spec Leads to cooperate?
  - Public recognition?
  - Tie to Star Spec Lead program?

Questions, discussion, next steps

Thank you!

http://jcp.org