JSR – Java API for JSON Binding (JSON-B)

Title: *

Java API for JSON Binding (JSON-B)

Summary: *

A standard binding layer (metadata & runtime) for converting Java objects to/from JSON messages.

Section 1: Identification

JCP Member submitting this proposal: *

Oracle Corporation

Name of Contact Person: *

Martin Grebac

E-Mail Address: *

martin.grebac@oracle.com

Telephone Number: *

+420 221 438 700

Fax Number: *

+420 221 438 700

Name of Specification Lead: *

Martin Grebac

E-Mail Address: *

martin.grebac@oracle.com

Telephone Number: *

+420 221 438 700

Fax Number: *

+420 221 438 700

Initial Group Membership:

Oracle

Supporting this JSR: *

Oracle

Section 2: Request

2.1 Please describe the proposed Specification: *

JSON has grown considerably as a portable means to represent objects and data. It is used by more and more systems and layers, ranging from web services to configuration data.

The recently established Java API for JSON Processing (JSON-P) specification defines a standard API for parsing and generating JSON data. The next logical step is to standardize a way to convert JSON into Java objects and vice versa. JSON-B will leverage JSON-P and provide a conversion layer above it.

JSON-B will be the standard binding layer for converting Java objects to/from JSON messages. A default mapping algorithm will be defined for converting existing Java classes to JSON. The default mappings can be customized through the use of Java annotations and will be leveraged by the JSON-B runtime to convert Java objects to/from JSON.

JSON-B will be useful to a number of the upper layers, such as JAX-RS. The requirements of these layers may be complex and include such features as bidirectional mappings and composite key support. Such requirements, as well as others that may be discussed, will be considered and incorporated as time permits.

As JAXB has been a successful platform for XML binding, JSON-B will have a similar feel. This will make it easier for JSON-B to be adopted by Java SE and Java EE developers.

2.2 What is the target Java platform? (i.e., desktop, server, personal, embedded, card, etc.) *

This JSR is targeted for Java SE 7 or higher and Java EE 7 or higher platforms.

2.3 The Executive Committees would like to ensure JSR submitters think about how their proposed technology relates to all of the Java platform editions. Please provide details here for which platform editions are being targeted by this JSR, and how this JSR has considered the relationship with the other platform editions.*

This JSR will work with Java EE 7 and will be targeted for inclusion in Java EE 8.

2.4 What need of the Java community will be addressed by the proposed specification? *

- There is currently a need in JAX-RS to convert objects to/from JSON messages. The community is currently forced to do one of the following:
    - Do the conversion themselves
    - Use JAXB with libraries like Jettison to get a “good enough” conversion.
    - Use proprietary tools, resulting in non-portable applications.
- Many services (Twitter, Google Geocoding API, etc) are offering data via JSON messages that Java developers need to interact with as Java objects. Some such as Twitter are even dropping the XML version of these APIs.

2.5 Why isn't this need met by existing specifications? *

Java API for JSON Processing (JSON-P) JSR

- The above JSR provides a generic model to represent a JSON message. JSON-B will bind the JSON message to a real object domain model.

Java API for XML Binding (JAXB) JSR

The JSON databinding need is currently being addressed by the JAXB specification together with libraries that expose JSON messages via StAX APIs. However, this approach faces following problems:

- Some of the JAXB mapping concepts have no JSON equivalent: namespaces, substitution groups, etc. Problems can arise when these mappings are used to produce JSON messages.
- It is difficult to produce JSON messages with collections of size one. This is due to a difference in the way JSON and XML represent collections.
- The JAXB approach assumes that the XML and JSON messages have same general structure and names.

2.6 Please give a short description of the underlying technology or technologies: *

Java API for JSON Processing (JSON-P) will be leveraged to parse the JSON messages

2.7 Is there a proposed package name for the API Specification? (i.e., javapi.something, org.something, etc.) *

javax.json.bind

2.8 Does the proposed specification have any dependencies on specific operating systems, CPUs, or I/O devices that you know of? *

No

2.9 Are there any security issues that cannot be addressed by the current security model? *

No

2.10 Are there any internationalization or localization issues? *

This specification uses the I18N support in Java SE

2.11 Are there any existing specifications that might be rendered obsolete, deprecated, or in need of revision as a result of this work? *

No

2.12 Please describe the anticipated schedule for the development of this specification: *

Q3 2014 Expert group formed

Q1 2014 Early draft

Q3 2015 Public review

Q1 2016 Proposed final draft

Q3 2016 Final Release

2.13 Please describe the anticipated working model for the Expert Group working on developing this specification.: *

The primary means of communication will be email, with weekly conference calls and a possibility of face-to-face meetings.

2.14 Provide detailed answers to the transparency checklist, making sure to include URLs as appropriate.: *

- Is the schedule for the JSR publicly available, current, and updated regularly?
  - Schedule will be available on the java.net project page for the JSR: https://java.net/projects/jsonb-spec/
- Can the public read and/or write to a wiki for the JSR?
  - Public mailing list will be used for comments (users@jsonb-spec.java.net)
- Is there a publicly accessible discussion board for the JSR that you read and respond to regularly?
  - Public mailing list will be used for comments (users@jsonb-spec.java.net)
- Have you spoken at conferences and events about the JSR recently?
  - No.
- Are you using open-source processes for the development of the RI and/or the TCK?
  - Yes.
- What are the Terms of Use required to use the collaboration tools you have prepared to use with the Expert Group, so that prospective EG members can judge whether they are compatible with the JSPA?
  - Terms of use of java.net: http://www.java.net/javanet-web-site-terms-use
- What is the location of your publicly-accessible Issue list? In order to enable EC members to judge whether Issues have been adequately addressed, the list must make a clear distinction between Issues that are still open, Issues that have been deferred, and those that are closed, and must indicate the reason for any change of state.
  - JIRA is part of the JSR java.net project, will be linked from there; https://java.net/jira/browse/JSONB_SPEC
- What is the mechanism for the public to provide feedback on your JSR?
  - Public mailing list will be used for comments (users@jsonb-spec.java.net)
- Where is the publicly-accessible document archive for your Expert Group?
  - At the java.net project page for the JSR.
- Does the Community tab for my JSR have links to and information about all public communication mechanisms and sites for the development of my JSR?
  - Yes, it will.
- Do you have a Twitter account or other social networking feed which people can follow for updates on your JSR?
  - Public user list and project site will be used to provide updates on the JSR.
- Which specific areas of feedback should interested community members (such as the Adopt-a-JSR program) provide to improve the JSR (please also post this to your Community tab)?
  - Technical feedback for individual parts of the JSR is welcome.

2.15 Please describe how the RI and TCK will be delivered, i.e. as part of a profile or platform edition, or stand-alone, or both. Include version information for the profile or platform in your answer.: *

The RI and TCK will be provided stand-alone, and will be included in the Java EE 8 RI and TCK.

2.16 Please state the rationale if previous versions are available stand-alone and you are now proposing in 2.16 to only deliver RI and TCK as part of a profile or platform edition (See sections 1.1.5 and 1.1.6 of the JCP 2 document).: *

N/A

2.17 Please provide the full text of the licenses that will apply to your Final Release Specification, Reference Implementation and Technology Compatibility Kit, or provide links to the same.: *

TBD

2.18 Please describe the communication channel you have established for the public to observe Expert Group deliberations, provide feedback, and view archives of all Expert Group communications.

JSR Java.net project will contain publicly available mailing lists, archives, and document download areas. For more details, please refer to the transparency section above.

2.19 What is the URL of the Issue Tracker that the public can read, and how does the public log issues in the Issue Tracker?

https://java.net/jira/browse/JSONB_SPEC/ URL will be also provided at the JSR java.net project page. JIRA use is described in JIRA documentation
https://confluence.atlassian.com/display/JIRA/Creating+an+Issue

2.20 Please provide the location of the publicly accessible document archive you have created for the Expert Group:

https://java.net/projects/jsonb-spec/downloads The URL will also be linked from the JSR project web page.

Section 3: Contributions

3.1 Please list any existing documents, specifications, or implementations that describe the technology. Please include links to the documents if they are publicly available. *

- JSON - http://json.org/
- Open Source Object-to-JSON Implementations
    - EclipseLink MOXy (http://www.eclipse.org/eclipselink/moxy.php)
    - Jackson (http://jackson.codehaus.org/)
    - Google GSON (http://code.google.com/p/google-gson/)
    - XStream (http://xstream.codehaus.org/)
    - Genson (http://code.google.com/p/genson/)
- JSON-P (JSR-353) - http://jcp.org/en/jsr/detail?id=353

3.2 Explanation of how these items might be used as a starting point for the work. *

There is now enough industry experience with object-to-JSON binding that the lessons learned from current implementations can be leveraged to form a Java standard that will add value to the Java stack.

Section 4: Additional Information

4.1 This section contains any additional information that the submitting Member wishes to include in the JSR.

There are several open source object-to-JSON binding solutions available today. The leads of these projects will be invited to join this expert group.