JSR 367 (JSON Binding) Review

September 15 2016

Dmitry Kornilov
Agenda

• Goals
• Information to be gathered
• Next Steps
• Issues
• Q&A
Goals
• Support binding (serialization and deserialization) for all RFC 7159-compatible JSON documents.

• JSON-related specifications will be surveyed to determine their relationship to JSON-Binding.

• Maintain consistency with JAXB and other Java EE and SE APIs where appropriate.

• Define default mapping of Java classes and instances to JSON document counterparts.

• Allow customization of the default mapping definition.
Goals (2/2)

• Default use of the APIs should not require prior knowledge of the JSON document format and specification.

• Define or enable integration with JSR 374: Java API for JSON Processing (JSON-P) 1.1.
Information to be gathered
JSON-B is standard binding layer for converting Java objects to/from JSON documents

```java
public class Customer {
    public int id = 1;
    public String firstName = “John”;
    public String lastName = “Doe”;
    ....
}
```
About this JSR

Usage in Other Java EE Frameworks

JAX-RS

Objects

JAXB

JSON-B

XML

JSON
Introduction

• Supports Java EE 7
• Targeted for inclusion in Java EE 8
• One of the core parts of Java EE 9
Which of these APIs do you think is important to be included into Java EE 8?

- JCACHE (JSR 107): 67.4% Yes, 6.3% No, 26.3% Not sure
- Data Grid API (JSR 347): 41.6% Yes, 14.3% No, 44.1% Not sure
- State Management API (JSR 350): 36.7% Yes, 12.3% No, 51% Not sure
- Identity API (JSR 351): 47.2% Yes, 9.5% No, 43.3% Not sure
- Java API for JSON Binding: 78.8% Yes, 5.6% No, 15.6% Not sure
- Java EE Configuration: 63.7% Yes, 6.9% No, 29.4% Not sure
Business/marketing/ecosystem justification

• Many services are offering data via JSON messages that Java developers need to interact with as Java objects.
• There is a need in other frameworks like JAX-RS to convert objects to/from JSON
• JAXB analog for JSON
• Standardize current technology
History

- 26 Aug 2014 – JSR Submitted
- 23 Sep 2014 – Expert Group Formation
- 20 Aug 2015 – Early Draft Review
- 26 July 2016 – Public Draft Ballot
Technical scope and features

Default Mapping
- Basic Types
- Specific Types
- Dates
- Classes
- Collections/Arrays
- Enumerations
- JSON-P types

Customized Mapping
- Property names
- Property order
- Ignoring properties
- Null handling
- Custom instantiation
- Custom visibility
- Date/Number Formats
- Binary Handling
- Adapters
- Serializers/Deserializers
Technical scope and features

- Annotation based customizations
- Easy runtime customizations using JsonbBuilder
- Full support of generic types
- Integration with JSON-P
- I-JSON support
The Expert Group

- Dmitry Kornilov (Oracle)
- Przemyslaw Bielicki
- Eugen Cepoi
- Martin Vojtek (Datlowe)
- Rick Curtis (IBM)
- Nathan Rauh (IBM)
- Roman Grigoriadi (Oracle)

- Alexander Salvanos
- Hendrik Saly
- Otavio Santana
- Inderjeet Singh
- Kyung Koo Yoon (TmaxSoft)
- Romain Manni-Bucau (Tomitribe)
- Gregor Zurowski
The Expert Group

• Main communication channel is experts mailing list
• If needed we setup a videoconference using Zoom, Skype or Hangouts
• Face to face conversations
Other deliverables

- JSONB web site (jsonb.java.net)
- Samples is a part of the spec
- Sample projects on GitHub
Publicity

- JavaOne 2015
- EclipseCon 2016 NA
- JavaOne 2016
Collaboration with other community groups

- Presentation for CZ JUG
- Presentation for Bentonville JUG
- Faso JUG Adopts JSR 367
  - [https://github.com/pandaconstantin/adopjsrfasojug](https://github.com/pandaconstantin/adopjsrfasojug)
- First look on JSR-367 (Turkish)
- JSONB test-drive using JAX-RS
  - [https://abhirockzz.wordpress.com/2016/08/24/json-b-test-drive-using-jax-rs/](https://abhirockzz.wordpress.com/2016/08/24/json-b-test-drive-using-jax-rs/)
IP flow

• Spec, RI and TCK licenses:

• Code contributions from non JCP members:
  – Accepted on GitHub or Eclipse.org
  – Reviewed and merged
  – We didn't have any yet

• Terms of Use of our collaboration tools:
  – https://www.java.net/javanet-web-site-terms-use

• No legal issues so far
RI and TCK development

• RI is developed as an open-source project
  – http://git.eclipse.org/gitroot/eclipselink/eclipselink.runtime.git

• Binary download
  – https://oss.sonatype.org/content/repositories/snapshots/org/eclipse/persistence/jsonb-ri/1.0-SNAPSHOT/

• Main committers:
  – Roman Grigoriadi (Oracle)
  – Dmitry Kornilov (Oracle)
  – David Kral (Oracle)

• TCK development is not started yet
Participation and transparency

• JCP.org page

• Specification project page
  – https://jsonb-spec.java.net
Adopt-a-JSR

- We are participating in Adopt-a-JSR program
  - Presentation for CZ JUG
  - Presentation for Bentonville JUG
  - Faso JUG Adopts JSR 367
    - https://github.com/pandaconstantin/adopjsrfasojug
- Where is no contribution to the project received
Mailing lists or forums

• Mailing lists:
  – Experts, Users, Issues
  – https://java.net/projects/jsonb-spec/lists

• Total number of messages: 562

• Average per Month: 25.5
Mailing lists or forums

Mailing lists messages

Experts
Users

Java Community Process
Mailing lists or forums

Emails in mailing lists by type

- Spec Lead: 100
- Experts: 145
- Users: 317
Issue tracker

- [http://java.net/jira/browse/JSONB_SPEC](http://java.net/jira/browse/JSONB_SPEC)
Next Steps
Schedule

- Start TCK Q4 2016
- Proposed Final Draft Q1 2017
- Final Release Q2 2017
Next Steps

• RI
  – Performance testing
  – Performance comparison
  – Performance optimization
  – Test on real life use cases

• TCK

• Evangelism
  – Samples, guides, manuals
  – Blog articles
  – Conferences
Next Steps

• JSONB 1.1
  – Integration with other Java EE frameworks (JAX-RS, JPA)
  – JSON Pointer
  – Partial Mapping
Issues
Issues

• Integration with JSON-P
• Resources for TCK
Questions?
Thank you!
http://jcp.org