
JavaOne™

ORACLE

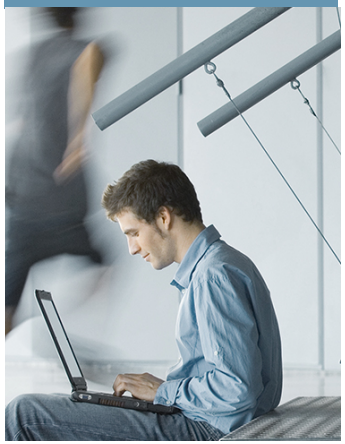
Java API for JSON

Jitendra Kotamraju
Oracle

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Program Agenda

- Overview
- JAX-RS Usage
- Standardization
- API



JavaOne™ ORACLE®

Overview

JSON

- JSON is a light-weight data exchange format
  - Easy for humans/machines to read and write
  - For e.g.

```
{"name":"Bob", "age":20, "phone":["276 1234", "123 4567"]}
```

- JSON is used by popular web sites in their RESTful web services
  - Facebook, Twitter, Amazon, …
  - Twitter Streaming API discontinues XML

JavaOne™ ORACLE®

Overview

JSON usages

- Policy in Amazon SQS

```
{
    ...
    "Statement": {
        "Effect": "Allow",
        "Principal": { "AWS": "123456789012” },
        "Action": "sqs:SendMessage",
        "Resource": "/987654321098/queue1"
    }
}
```


JavaOne™ ORACLE®

Overview

JSON usages

- Followers in Twitter API

```
{
    "previous_cursor": 0,
    "ids": [143206502, 143201767, 777925],
    "previous_cursor_str": "0",
    ...
}
```


JavaOne™ ORACLE®

JAX-RS

XML Usage

- JAX-RS applications handle XML using JAXP API

```
@Produces("application/xml")
public Source getBook(String id) {
    return new StreamSource(…);
}
```


JavaOne™ ORACLE®

JAX-RS

XML Usage

- JAX-RS applications handle XML using JAXB API

```
@Produces("application/xml")
public Book getBook(String id) {
    return new Book(…);
}
```

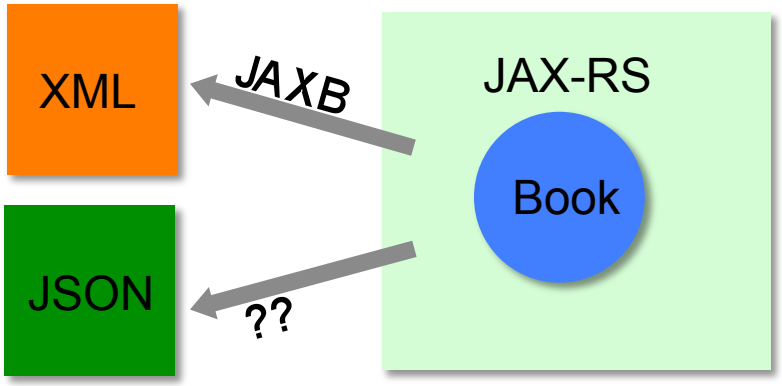

JavaOne™ ORACLE®

JAX-RS

DataBinding

- JAX-RS content negotiation

```
@Produces({"application/xml", "application/json"})
public Book getBook(String id) {
    return new Book();
}
```



 | |

JAX-RS

JSON Solutions & Limitations

- A custom MessageBodyWriter converts Book object to JSON
  - Book is of type JSONObject (For e.g. json.org's API)
  - JAXB → StAX → JSON (For e.g. using jettison)
  - POJO/JAXB → JSON (For e.g. using jackson, eclipseLink etc.)
- No standard API
- Some solutions have technical limitations
- Applications/Frameworks need to bundle the libraries

JavaOne™ ORACLE®

Standard API

Advantages

- Application can use standard types
- Leaner, portable applications

Standard API

Contents

- Streaming API to produce/consume JSON
  - Similar to StAX API in XML world
- Object model API to represent JSON
  - Similar to DOM API in XML world
- Data binding : JSON text <-> Java Objects

Standardization

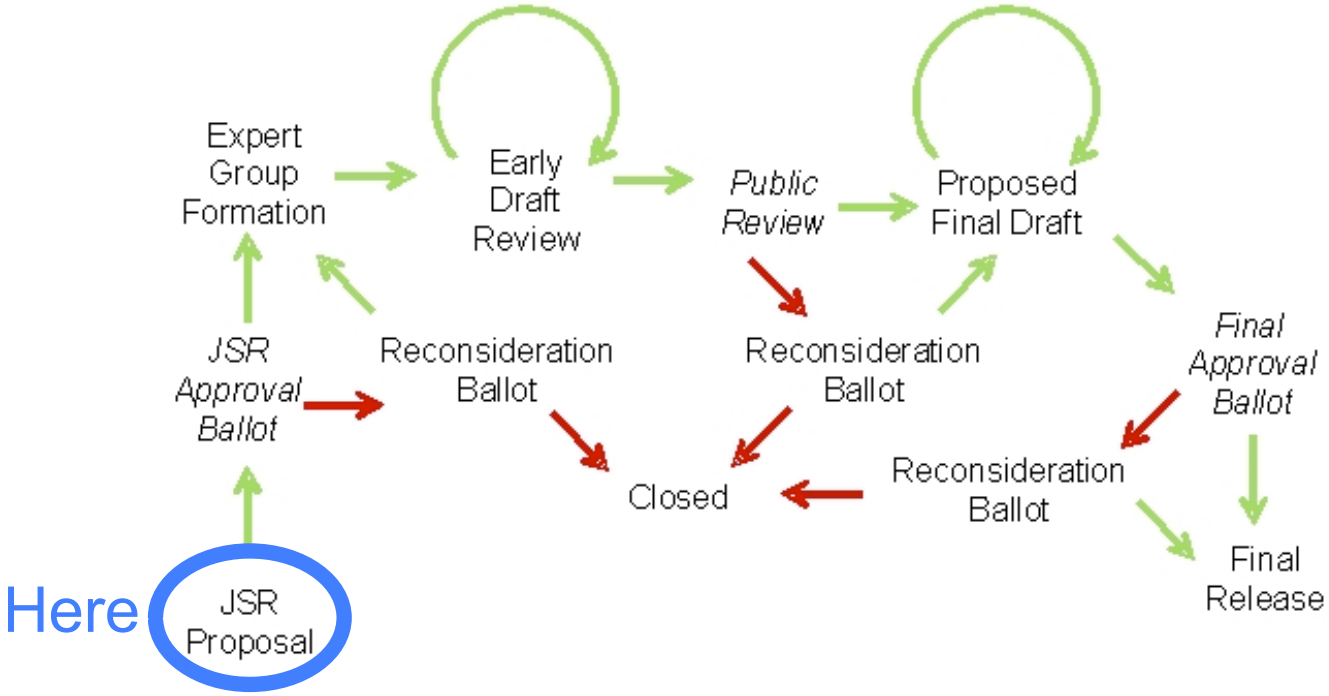
JSR

- Perhaps two JSRs: Processing/Parsing, Binding
  - Similar to JAXP and JAXB
  - Close collaboration between the two
  - Parsing JSR align with Java EE 7 schedules
- Can learn from existing implementations
  - json.org/java, google-gson, Jackson, JavaFx, …
- Processing/Parsing JSR Supporters
  - Tatu Saloranta(Jackson), Doug Crockford(json.org)

JavaOne™ ORACLE®

Standardization

JSR State Diagram



Source: http://blogs.oracle.com/darcy/entry/pictorial_jcp

Parsing API

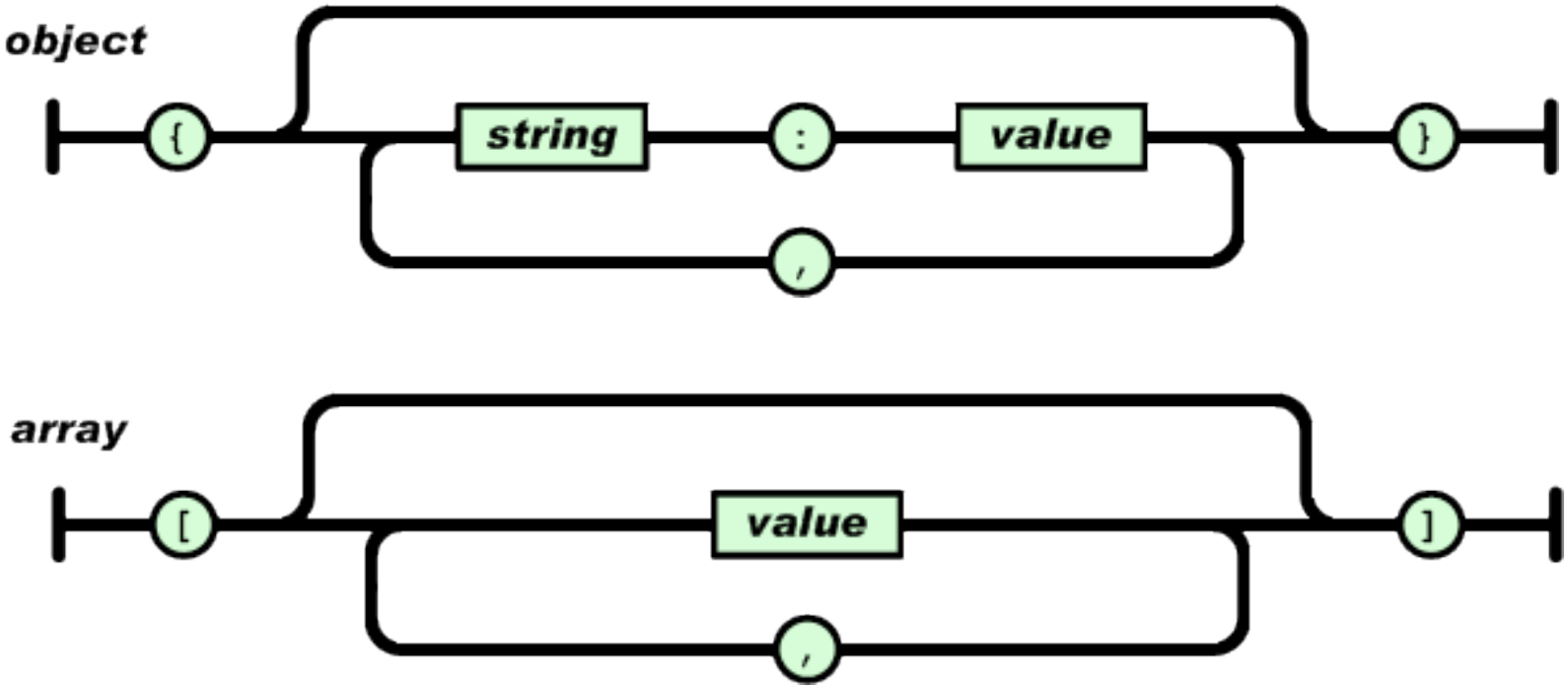
JSR

- My experimental API
  - Based visitor pattern (similar to ASM, JSR 269 API, …)
  - Not reviewed internally also
  - Still evolving
  - Of course, EG may have a different take !

 | |

Parsing API

JSON Grammar

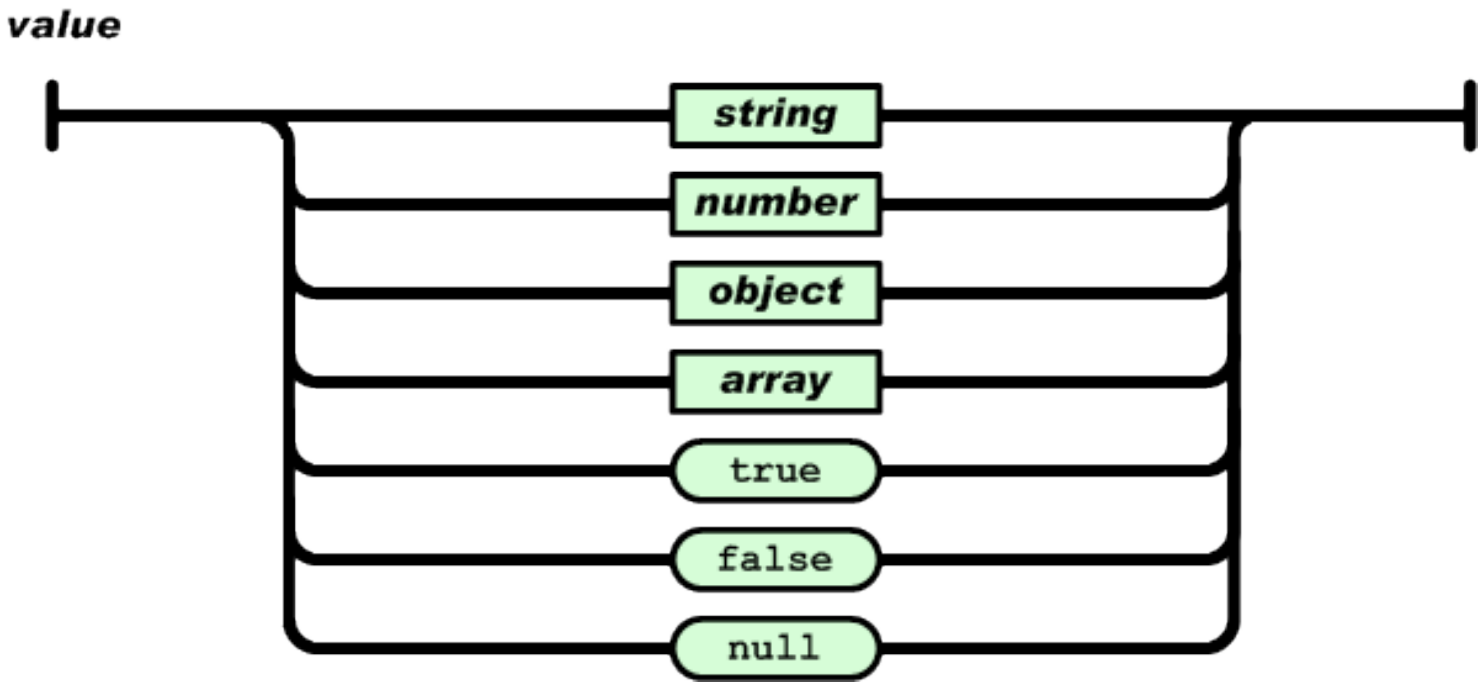


Source: http://json.org

 | |

Parsing API

JSON Grammar



Source: http://json.org

 | |

API

Resources

- http://weblogs.java.net/blog/jitu/archive/2011/03/15/json-jsrpre-jcp-filed-draft

Q&A

JavaOne™ ORACLE®


JavaOne™