

JCP EC Meeting January 2013

Java Money and Currency API

Anatole Tresch/Werner Keil

Summary

- This JSR will provide a money and currency API for Java, targeted at all users of currencies and monetary amounts in Java.
- The API will provide support for standard ISO-4217 and custom currencies, and a portable representation of a monetary amount.
- It will support currency arithmetic, even across different currencies, and will support foreign currency exchange.
- Additionally, implementation details surrounding serialization and thread safety are to be considered.

Why is this needed?

- Monetary values are a key feature of many applications
- The existing `java.util.Currency` class is strictly a structure used for representing ISO-4217 standard currencies
- No standard value type to represent a monetary amount
- No support for currency arithmetic

Challenges

- Keep it simple, but powerful
- Several usage scenarios - low latency applications, such as High Frequency Trading apps, high precision, financial calculations, simple web shops
- Precision - There are differing precisions for arithmetic, currency exchange, externalization and formatting. Precision may also vary depending on use case!
- Ensure interoperability in a multi-tier context
- Complex formatting (grouping, regional particularities, natural language support, multiple usage scenarios – display, technical, printout etc.)
- Formatting and access determined by other than a `Locale`, e.g., regions, territories or custom groupings
- Support for non-standard rounding rules

JSR Participants

Spec Lead

- Credit Suisse

Expert Group Members

- Stephen Colebourne
- Benjamin Cotton
- Jeremy Davies
- Tony Jewell
- Werner Keil
- Bob Lee
- Christopher Pheby
- Jefferson Prestes
- Arumugam Swaminathan

Supporting this JSR

- Credit Suisse
- Stephen Colebourne
- London Java Community
- Caxton Associates
- Goldman Sachs
- JP Morgan/Chase
- Werner Keil

New Schedule and Status

- Targeted to Java 9, parts (core) of it also for CLDC 8
- With back-port to previous versions

- Early Draft Review until end of March 2013
- Status:
 - Wiki updated
 - API stable until end of January
 - Discussions is focusing, consensus on several key points
 - Core abstractions stable
 - Extended areas (conversion, formatting) ongoing
 - SPIs stable until end of February
 - Work on EDR spec document also started in parallel
 - Running continuous build of Spec/RI code

References

- The JSR: <http://jcp.org/en/jsr/summary?id=354>
- Wiki (java.net): <http://java.net/projects/javamoney/>
- Mailing lists: jcurrency_mail@javamoney.java.net
- Current Source Code: <https://github.com/JavaMoney/javamoney>

- JMaghreb 1.0 presentation: <http://www.slideshare.net/keilw/jsr-354-money-and-currency-api-for-java>
- Prezi of initial presentation: <http://prezi.com/no48uqcsyhjy/jsr-354/>
- Minutes of JCP EC F2F, Princeton 2009:
<http://jcp.org/aboutJava/communityprocess/ec-public/materials/2009-09-1011/Money09-EC-F2F.pdf>

Appendix

Example of Solutions Identified

- Model `CurrencyUnit`, `MonetaryAmount` as interfaces
- Allow multiple currency *namespaces*
- Define basic arithmetic operations on `MonetaryAmount`, similarly to `BigDecimal`
- Encapsulate representation type for numeric part of `MonetaryAmount`:
 - Allowing different implementations for SE/ME
 - Allowing different implementations for different use cases
- *AmountAdjusters* allow amount manipulation not covered by the spec
 - Rounding is a special case of an adjustment
- Precision must be distinguished into
 - *Internal* precision used for arithmetic
 - *External* precision, when converting to other types, e.g. `float`, `double`
 - *Formatting* precision, for representing as `String`