

Go for the Money

Introduction to JSR 354



Go for the money - JSR 354 – <http://javamoney.org>

May 2015

Bio

Anatole Tresch @atsticks

- Consultant, Coach
- Credit Suisse
- Technical Architect
- Specification Lead JSR 354
- Apache Tamaya
- EG Member JSR 354, 365, 377



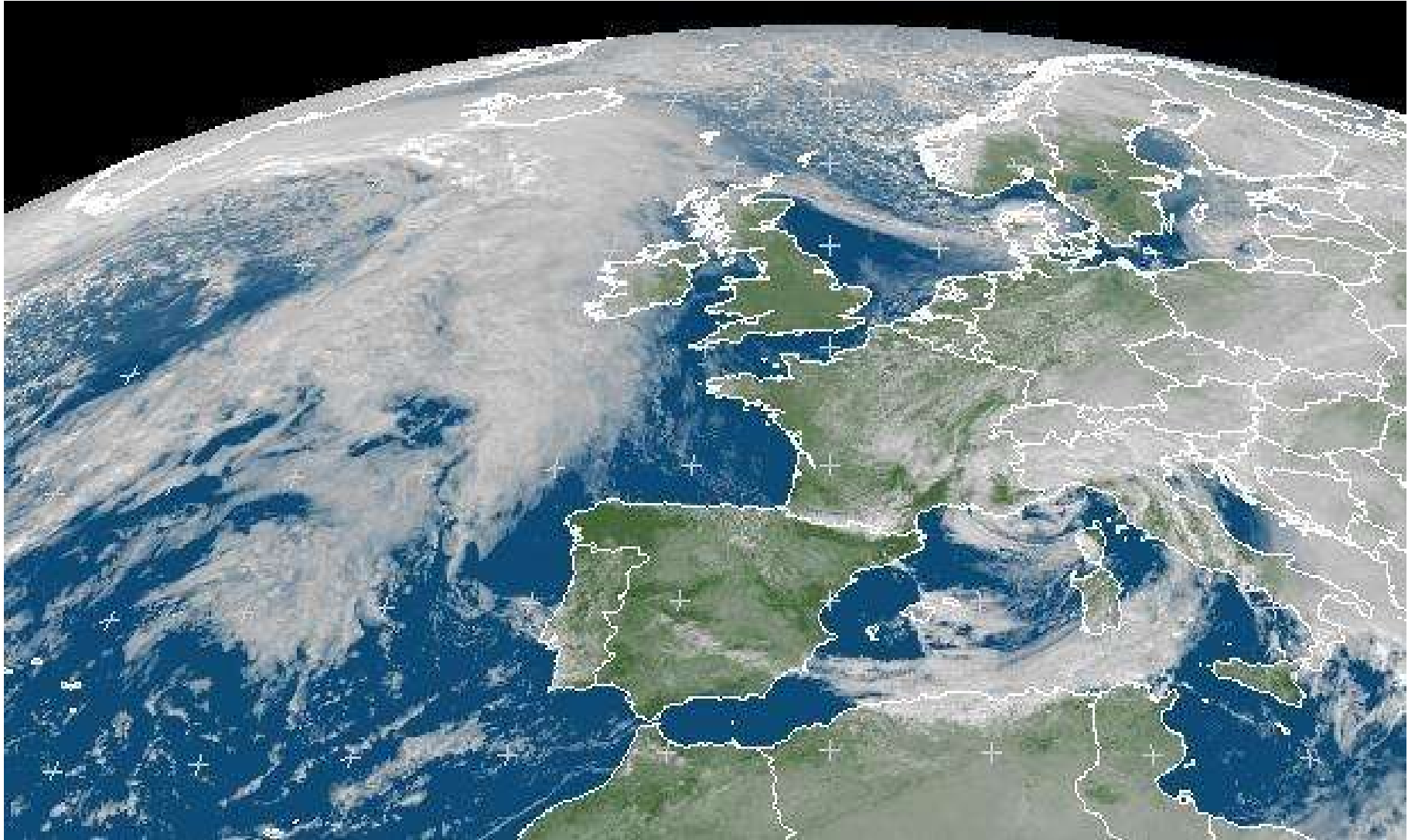
- Twitter/Google+: @atsticks
- atsticks@java.net

Agenda

- Overview
- Currencies and Amounts
- Currency Conversion
- Formatting (& Parsing)
- JavaMoney OSS Project Family



Overview



Motivation

- Monetary values are a key feature
- Constraints of `java.util.Currency`
- No standard amount value type
- No support for monetary arithmetic or currency conversion
- Lacking features of JDK Formatting

Scope & Schedule

- Started in 2012, **final beginning this week!**
- Standalone scope
- Double Platform Release for Java 8 and Java 7
- API compatible with Java 6!

General Considerations

- Minimal, but extendible API
- Complete API
- Reimplementable Value Types defined by interfaces
- Singleton Accessor
- Flexible SPI making it suitable for SE and EE
- OSGI Compatible

Core Artifacts

■ API:

- **Core (javax.money)**: CurrencyUnit, MonetaryAmount, MonetaryException, MonetaryOperator, MonetaryQuery, Monetary

- **Conversion (javax.money.conversion)**: ExchangeRate, ExchangeRateProvider, CurrencyConversion, MonetaryConversions

- **Format (javax.money.format)**: MonetaryAmountFormat, MonetaryFormats;

■ RI (org.javamoney.moneta) :

Money, FastMoney, MonetaryFunctions, BuildableCurrencyUnit

■ TCK (org.javamoney.tck)

Currencies and Amounts (Core)

javax.money



Design Decisions & Objectives

Currencies

- Allow arbitrary currency codes
- Provide a flexible and easy SPI
- Fluent API using builders
- Supporting also historic currencies
- Compatible with existing `java.util.Currency`

```
public interface CurrencyUnit {  
    String getCurrencyCode();  
    int getNumericCode();  
    int getDefaultFractionDigits();  
    CurrencyContext getCurrencyContext();  
}
```

Design Decisions & Objectives

Monetary Amounts

- What is a Monetary Amount?
 - CurrencyUnit +
 - Numeric Value +
 - Arithmetic Functions +
 - Comparison Functions +
 - MetaData (Capabilities)

Design Decisions & Objectives

Monetary Amounts (continued)

- Support multiple numeric representations
- Use functional design for extendible functionality
(`MonetaryOperator`, `MonetaryQuery`)
 - Used for Rounding, Conversion, Formatting, Financial Calculations etc.
- Define Recommendations
 - E.g. precision/scale capabilities should be inherited to its operational results.

Examples

Currencies & Amounts

Demo:

- Currencies & Amounts
- Extension Points

Currency Conversion

`javax.money.convert.*`



Currency Conversion

ExchangeRate

- A `ExchangeRate` models a conversion between two currencies:
 - A base `CurrencyUnit`
 - A term/target `CurrencyUnit`
 - A `Provider`
 - A Conversion Factor f , where
$$M(\text{term}) = M(\text{base}) * f$$
 - Optional attributes (`ConversionContext`)
 - An rate chain (multiple for composite rates)
- Rates may be *direct* or *derived* (composite rates)
- Rates are provided by a `ExchangeRateProvider`

Examples

Currency Conversion

Demo:

■ Currency Conversion

Formatting and Parsing

javax.money.format.*

Portfolio									
Cash: 64102.56 € Market: FRA									
Symbol	Company	Price	Change	% Change	Shares	Open	Volume	Current Value *	Gain/Loss
IBM	"IBM"	115.43	-0.37	-32%	50	115.80	2,655,471	3699.68 €	-15.98
JAVA	"JAVA"	16.56	0.44	273%	200	16.12	5,750,460	2123.08 €	545.90
DELL	"DELL"	19.52	0.08	41%	200	19.44	14,293,015	2502.56 €	82.30
GOOG	"GOOG"	426.88	1.62	38%	100	425.26	5,523,309	27363.97 €	38.05
MSFT	"MSFT"	28.58	0.20	71%	100	28.38	47,317,464	1832.15 €	71.00

* in local Currency

Currency rates from 03/08/2007 12:00pm EST

100 US Dollar

Currency Name	Currency Code	Exchange Rate to US \$	Exchange Amount
Australian Dollar	AUD	1.287830006	128.78
Baht	THB	32.7	3270.00
Bolivar	VEB	2144.6	214460.00

[Make a trade](#)
[Log out](#)

Formatting and Parsing

Key Design Decision

- Thread-safe and immutable
- Flexible and extendible
- Potentially ME-Compatible
- Supporting a fluent programming style
- JDK formats as preconfigured default formats still available

Examples

SPI

Demo:

■ Adding a Currency Provider

JavaMoney OSS Project

org.javamoney.*



JavaMoney OSS Project

General

- During the JSR additional features were discussed...
... and implemented!
- This ensured the design is working
- Helped to consolidate discussions

...but...

- exceeded the JSR's scope
- Some parts still experimental

So we created an OSS project!

Links

- Examples: <https://github.com/JavaMoney/javamoney-examples/tree/master/console/javamoney-console-simple>
- Umbrella Page: <http://javamoney.org>
- JSR 354: <http://jcp.org>
- Java.net Project: <http://java.net/projects/javamoney>
- GitHub Project (JSR and JavaMoney):
<https://github.com/JavaMoney/javamoney>
- Twitter: @jsr354

Q & A

???