

TamayaProposal

Apache Tamaya - Proposal

Abstract

Tamaya is a highly flexible configuration solution based on an modular, extensible and injectable key/value based design, which should provide a minimal but extendible modern and functional API leveraging SE, ME and EE environments.

Tamaya hereby translates into *in the middle*, which is exactly, what configuration should be. It should be in the middle between your code and your runtime.

NOTE: Alternative names could be *Mahkah=earth*, *Dakota=friend* or *Orenda=magic force*.

Proposal

Tamaya is a highly flexible configuration API based on an modular, extensible and injectable key/value based design. The basic building blocks hereby are:

- *property providers* implementing a small and easily implementable subset of a `Map<String,String>`.
- support for configuration injection
- a type-safe configuration template mechanism
- serializable and remote configuration support
- a JMX/Rest based management console
- Configuration will follow the GoF composite pattern and support several combination strategies.
- An extendible and adaptable environment model, so configuration can be provided dependent of the environment currently active.
- extension points and a powerful SPI to seamlessly add additional logic to the API, such as secured views, multi-valued validation schemes, encryption/decryption etc.
- Configuration (and property providers) are designed and implemented as indirectly mutable types, providing thread-safe and performant to configuration.
- Configuration changes can be observed by listening on `ConfigChange` events.

The API's focus is on simplicity and ease of use. Developers should only have to know a minimal set of artifacts to work with the solution. The API is built on latest Java 8 features and therefore fit perfectly with the functional features of Java 8.

Additionally Apache Tamaya will provide

- A Java SE based implementation with minimal features and dependencies.
- A Java EE extension module for integration with Java EE and Apache Deltaspike.
- Once Java ME supports Lambdas, default methods, method references and functional interfaces an implementation targeting Java ME should be provided as well.
- Extension modules for different features.
- Adapter/inter-operation modules for other configuration solutions including Apache commons-config

Background

There is a global initiative running now for about a year lead by Anatole Tresch (Credit Suisse) with the target of standardizing configuration in Java EE and SE. Due to several reasons it seems currently most sensible to start an OSS project on the topic to join forces that actively want to contribute to the project. It is highly probably that standardization will be restarted at a later point once we have a widely used Apache standard. For further information you may look at <http://javaeeconfig.blogspot.com>.

Rationale

Configuration is one of the most cross-cutting concerns, which still lacks of a standard. Java EE is currently (EE7) in most areas strictly only configurable during build time of the deployed artifacts. Especially dynamic provisioning of resources or runtime configuration is not supported and in many cases impossible to add without tweaking the underlying application server. On the other hand running two separate configuration solutions for Java EE and Java SE as well make no or little sense. So it would be important we have a unified configuration model at hand, that is flexible enough, so

- it can be used in Java SE, EE and ME
- it can support contextual behaviour (like in Java EE and multi-tenancy/SaaS scenarios)
- it provides a uniform API, regardless, if its used in SE or EE scenarios
- it supports existing APIs, e.g. `System.getProperties`, `java.util.preferences` in SE and CDI, `JNDI` in Java EE
- it supports service location pattern like access as well as *injection* of configured values.
- it is simple in use and easily extensible.
- it support integration with existing configuration solutions currently in use, both OSS as well as customized in-house proprietary solutions

Initial Goals

The initial goals of the Apache Tamaya project are to:

- Setup the governance structure of the project
- Receive code donations from <https://github.com/java-config>

- Ensure all donated code is appropriately licensed under the Apache License
- Merge and rename code to reflect new project name
- Define the project modules and structure (API, implementation modules, adapter modules, examples)
- Provide examples demonstrating feature usage
- Produce release/s based on a schedule created by the PMC
- Attract contributions from the greater Java community
- Setup collaboration with other projects and the JCP to bring in ideas and enhancement proposals, e.g. to Java EE 8

Current Status

There is an existing running code base implementing a significant part of the features mentioned already at <https://github.com/java-config> and licensed under Apache v2.0, which will be contributed into the incubator. The separation between API and implementation hereby should stay enforced, since

- it reflects the structure also required for later JSRs
- it helps focusing discussions on the core API artifacts before dive into implementation details.
- it helps to ensure the core API is simple and overall comprehensive.
- it enables to provide different implementations, especially also a Java ME compatible solution.

Meritocracy

We plan to do everything possible to encourage an environment that supports a meritocracy. We did the same as well with JSR 354, where people throughout the world helped us to get the RI/TCK at a very good level. Similarly, whenever possible, we encouraged people to join the expert group, so they also would be capable of contributing to the API directly. In all cases we discussed all questions and feedback transparently regardless if it was an EG member or just a member of Hackday, Hackergarten.

Community

The project initiative already is significantly supported by JUGs such as [SouJava](#), LJC, iJUG, Berlin Brandenburg JUG, JUG Zurich, as well as companies such as Credit Suisse and Walmart. It is expected that support will raise very quickly so the library will evolve and be widely used as well.

Core Developers

The core team will be a set of well known experts from the Java SE and EE area (in random order):

- **Anatole Tresch** (Lead) is employed at Credit Suisse. He leads JSR 354 (Money & Currency) and also was planned as cospec lead for Java EE configuration JSR together with Oracle. He also is a member of the CDI 2.0 expert group and is actively driving the configuration topic.
- **Gerhard Petracek** is Apache [MyFaces](#) und [DeltaSpike](#) PMC chair.
- **Andres Almiray**: Groovy aficionado, Griffon project lead, Java Champion.
- **Joe Pullen** is a known expert, especially for JPA and Batch and also former EC member of the corresponding JSRs.
- **Mark Struberg** acts as PMC and Committer for Apache [OpenWebBeans](#), BatchEE, [MyFaces](#), Maven, OpenJPA, BVal, [DeltaSpike](#) and other projects
- **Werner Keil** aka "Java Godfather" is individual JCP EC member and member of the Advisory Board of Developer Week contributing to several JSRs in the SE and EE area. He is spec lead of the Units and Measurements JSR. Werner is already a committer of ASF.
- **Otávio Gonçalves de Santana** Member of [SouJava](#) and OpenJDK committer. He contributes regularly to several JSRs and recently was awarded in 2014 as most valuable JCP member.
- **Marco Zurmühle**: Member of Credit Suisse working in the Core Frameworks Area, which is also responsible for application configuration management, and a regular member of the Zurich Hackergarten.
- **Oliver B. Fischer**: Leader of the JUG Berlin Brandenburg and conference speaker.
- **David Blevins**: Founder of the Apache TomEE, OpenEJB and Geronimo projects. JCP participant in JavaEE and EJB.
- **John D. Ament**: Author of Arquillian Testing Guide an Enterprise software architect,
- **Romain Manni-Bucau**: Developer convinced by Open Source, highly active on Apache TomEE, Sirona and other Apache projects (OpenEJB, Camel, CXF, Sirona, BatchEE, [OpenWebBeans](#)...).

It is expected that more people will join the incubator once it's running:

- We are already in contact with several companies from Europe and US, that are heavily interested in contributing to this initiative.
- **LJC (London Java Community)**, [SouJava](#), **JUG Chennai** will do Hackdays and provide feedback.
- **JUG Berlin Brandenburg** is one of the bigger JUGs in Germany and would probably also actively contribute to this project.
- **JUG Zurich** organizes regular (monthly) Hackergarten and will as well contribute to this project.

Alignment

Credit Suisse, which lead the initiative through Anatole Tresch during the last year, has a strong commitment to Open Source Software. As a consequence also their first JSR (354, Money & Currency) was released under Apache v2. The same is the case for all other core contributors and supporters.

Known Risks

Main Risk could be that main committers could cease the project before it is possible to build up a public community. Nevertheless the wide support of JUGs and companies involved already as well as the engagement of main drivers of the initiatives during the last year makes this not a very probable scenario.

Orphaned products

See main risks. Basically the engagement of all stakeholders (Credit Suisse, JUGs, other companies) should ensure this initiative will evolve hopefully rather quickly to a key component in the Java eco-system, both in SE, as well as ME and EE. Additionally all stakeholders involved (companies, as well as individuals/JUGs) have direct benefits of the functionality provided.

Inexperience with Open Source

Starting point will be the experimental repository at <https://github.com/java-config>. Additionally the talks given by Anatole (e.g. at Javaone 2014) and the blogs under <http://javaeeconfig.blogspot.com> help to give a good starting point on some of the concepts implemented/contributed. Nevertheless the idea is that the ideas are further evolved, basically similar to a JSR, to ensure all relevant views and aspects will be included.

All of the core committers have already experience working on open source projects or JSRs. Many of them even already are members of the ASF.

Homogenous Developers

The current list of committers includes developers from several different companies plus many independent volunteers. The committers are geographically distributed across the U.S., Brazil, Europe, and Asia. They are experienced with working in a distributed environment.

Reliance on Salaried Developers

Some of the developers are paid partially by their employer to contribute to this project, but given the anticipation from the Java community for a powerful Configuration implementation and the committers' sense of ownership for the code, the project would continue without issue if no salaried developers contributed to the project. Anatole, as the main committer and driver of the initiative currently, is paid only partially and basically drives the initiative as part of his community engagement in general already as of now.

Relationships with Other Apache Products

The project's core API will be independent of any other projects, because

- The API should be implementable by different third party providers, modules using defined SPIs.
- The API should if possible have minimal dependencies (or even be standalone), so it is highly portable to different environments.

Tamaya will provide a minimal standalone implementation as well. Nevertheless it will be possible that other solutions implement the API as well, e.g. *Apache Commons Configuration* (especially version 2). The API/SPI should be build in a way, so features from other solutions such as

- Apache Commons Configuration
- Spring Property Sources
- JFig
- Configuration Builder
- and more

can be used or even be leveraged (e.g. by adding environment dependent configuration instance management).

Tamaya will also provide adapter modules for other technologies/projects, so the solution can inter-operate with existing frameworks and solutions as a provider similarly. This explicitly also includes the possibility to use Tamaya as a configuration/property source for.

- Spring
- System Properties
- ...

Integration into Java EE has to be coordinated with Apache Deltaspike Configuration, to avoid two conflicting configuration standards for Java EE.

An Excessive Fascination with the Apache Brand

While we expect the Apache brand may help attract more contributors, our interests is in establishing a powerful and widely used standard for configuration. At a later stage, if successful, standardizing it within a JSR also may be an option. We believe this process starts with growing a strong and self-managed community that can someday lead the charge in any future standardization efforts. Furthermore, we have been enthusiastic users of Apache and feel honored at getting the opportunity to join and learn.

Documentation

References to further reading material.

[1] Java (EE) Configuration Blog:
<http://javaeeconfig.blogspot.com>

[2] Java Configuration Experimental Repo:
<https://github.com/java-config>

[3] The [JavaOne] Presentation Slideset:
<http://de.slideshare.net/AnatoleTresch/a-first-draft-to-java-configuration>

Links to some other existing solutions:

[4] Apache Commons Configuration: <http://commons.apache.org/proper/commons-configuration/>

[5] Apache Deltaspike Configuration: <https://deltaspike.apache.org/documentation/configuration.html>

[6] Spring Configuration: <http://projects.spring.io/spring-framework/>
<http://docs.spring.io/spring/docs/current/javadoc-api/org/springframework/context/annotation/PropertySource.html>

[7] Java Configuration Builder: <https://github.com/TNG/config-builder>

[8] JFig: <http://jfig.sourceforge.net/>

[9] Owner: <http://owner.aeonbits.org/>

Initial Source

Initial source will be from <https://github.com/java-config>. Most of the functionalities are already fully functional, documentation must be improved.

It is already licensed under Apache v2.

External Dependencies

The following external dependencies have been identified:

The core functionality will be dependent on/use

- Apache Maven - Java based build tool - Apache License 2.0, (non-runtime)
- Shrinkwrap - Java deployment packaging - Apache License 2.0 (non-runtime)

The parts requiring EE will probably make use of

- Arquillian - Java EE integration testing framework - Apache License 2.0, (non-runtime)
- various Java EE API packages - all Apache License 2.0 (non-runtime)

The API part of the current initial source is completely standalone (it does not have any further dependencies than the JDK). The SE 8 based part does mainly depend on Apache commons-logging for logging.

Cryptography

The framework will not bring along additional cryptographic algorithms.

Required Resources

- The project's build currently is based on Maven, it might be moved to gradle.
- Continuous build and integration is important. Depending on the integration and third party solutions/versions supported this may require several external solutions to be loaded. All of them must be available as OSS projects or freely accessible.
- Continuous quality control with [SonarSource](https://sonarcloud.io) would be important as well to guarantee very high quality. This is important to have a good adoption rate as well.

Mailing lists

We initially would like to start with the minimum required lists:

- `private@tamaya.incubator.apache.org` will be used for confidential PPMC discussions.
- `dev@tamaya.incubator.apache.org` is used for public discussions and support.
- Commits for Tamaya will be emailed to `commits@tamaya.incubator.apache.org`.

Git Repository

It is proposed that the source code for the Apache Tamaya project be hosted in the Apache Git repository:

- <https://git-wip-us.apache.org/repos/asf/incubator-tamaya.git>

Issue Tracking

The following JIRA project would be required to track issues for the Apache Tamaya project:

- TAMAYA

Other Resources

None.

Initial Committers

- Anatole Tresch (atsticks at gmail dot com, anatole dot tresch at credit dash suisse dot com)
- Mark Struberg (struberg at apache dot org)
- Gerhard Petracek (gpetracek at apache dot org)
- John D. Ament (johndament at apache dot org)
- Joe Pullen (joe dot pullen at espalier dot com)
- David Blevins (dblevins at apache dot org)
- Andres Almiray (aalmiray at gmail dot com)
- Werner Keil (werner dot keil at gmail dot com)
- Otávio Gonçalves de Santana (otaviopolianasantana at gmail dot com)
- Marco Zurmühle (marco dot zumuehle at gmail dot com)
- Oliver B. Fischer (o dot b dot fischer at swe dash blog dot net)
- Romain Manni-Bucau (rmannibucan at gmail dot com)

Affiliations

- Anatole Tresch - Credit Suisse
- Marco Zurmühle - Credit Suisse
- Joe Pullen - Espalier
- Andres Almiray - Canoo

Sponsors

Champion:

- David Blevins (dblevins at apache dot org)

Sponsors:

- David Blevins
- Mark Struberg
- Gerhard Petracek

Nominated Mentors

- John D. Ament (johndament at apache dot org)
- Mark Struberg (struberg at apache dot org)
- Gerhard Petracek (gpetracek at apache dot org)

Sponsoring Entity

We would like Apache Incubator to sponsor this project.