Original Java Specification Request (JSR)
Identification | Request | Contributions

Section 1. Identification
Submitting Member: Werner Keil
Name of Contact Person: Werner Keil
E-Mail Address: java-social@catmedia.us
Telephone Number: +49 176 36954273
Fax Number: -

Specification Lead: Werner Keil, Antoine Sabot-Durand

E-Mail Address: java-social@catmedia.us, antoine@sabot-durand.net
Telephone Number: +49 176 36954273
Fax Number: -

Initial Expert Group Membership:

- Werner Keil
- Antoine Sabot-Durand
- JUGChennai
- RedHat
- Twitter
- Johan Vos

Supporting this JSR:

- RedHat
- Twitter
- CoreMedia AG
- Oracle

Section 2: Request

2.1 Please describe the proposed Specification:

This specification proposes to provide an API for accessing social information networks, both Public (Facebook, Twitter, Google+, LinkedIn, Xing, Yammer,...) and Corporate, e.g. within the Enterprise or Institution (University, Hospital, etc.)
The primary API will build upon standards and formats specified by OpenSocial 2.0 or other relevant technologies defined by leading social networks.

2.2 What is the target Java platform? (i.e., desktop, server, personal, embedded, card, etc.)

Desktop, Server, Mobile

2.3 The Executive Committees would like to ensure JSR submitters think about how their proposed technology relates to all of the Java platform editions. Please provide details here for which platform editions are being targeted by this

JSR, and how this JSR has considered the relationship with the other platform editions.

While this is primarily targeted at Java SE, it is particularly useful in a Java EE environment, especially when related to other EE technologies such as CDI. Further, it helps enhance Java EE making it more cloud-friendly by providing standardized access to public social networks. Corporate social networks are commonly used along with Java EE, Portals, or any highly scalable configuration.

2.4 Should this JSR be voted on by both Executive Committees?

No. It should be voted on by the Java SE / EE Executive Committee only.
At least until there still are two separate Executive Committees.

2.5 What need of the Java community will be addressed by the proposed specification?

A social network is a social structure made up of individuals (or organizations) called "nodes", which are connected by one or more specific types of interdependency, such as friendship, kinship, common interest, financial exchange, dislike, cultural relationships, or relationships of beliefs, knowledge or prestige. This JSR provides a standard API for accessing and working with social networks.
Except one (probably the least succesful) case, every major social network's existence is a result of using Open Source technologies. Many of those are based on Java or other JVM-based languages like Scala, Clojure, etc.

2.6 Why isn't this need met by existing specifications?

The closest specification is probably Apache Shindig, or the recently released Spring Social. However, most of these built on top of quasi-standards or proprietary technologies adding the risk of a vendor lock-in even if the API itself may be provided Free or Open Source. Apache Shindig has been contributed by Google, but the project has not seen a lot of activity in recent years, unlike the most common social networks. It is said to be RI for Open Social 0.8.x and 0.9.x, but Google also released an Open Social Java Client on Google Code. Neither of them use Java (EE) standards like CDI or JSR 330, instead frameworks like Guice are used directly. While Shindig tends to separate model and implementation, and e.g. for Persistence even uses JPA, the Google version lacks both. It also puts a strong emphasis on MySpace which increasingly lost users to other services. Other examples like Ning API for Java also haven't exceeded OpenSocial 0.8.x as of now. Beside Shindig, 2 other projects, DeltaSpike and Rave were recently added to the Incubator.

2.7 Please give a short description of the underlying technology or technologies:

Storing data in-memory for fast, low-latency access helps achieve scalability, but in a clustered, distributed world, such in-memory stores need to be coherently maintained to prevent reading of stale data. Furthermore, to be truly scalable, a data distribution model is needed to ensure adequate copies are maintained in a cluster to provide resilience to server failure, yet not too many copies which would hamper scalability. Such data should also be accessible from any server, regardless of whether the data is locally stored or not.

2.8 Is there a proposed package name for the API Specification? (i.e., javapi.something, org.something, etc.)

javax.social

2.9 Does the proposed specification have any dependencies on specific operating systems, CPUs, or I/O devices that you know of?

No

2.10 Are there any security issues that cannot be addressed by the current security model?

No. It will build on Java Security APIs, especially those within Java SE/EE. Possible exceptions are security and authentication mechanisms used in Social Media, like OAuth or OpenID, which may in some cases have no standard Java API. Where applicable, the aim of the specification is to map these onto the Java security model. Also there seem to be quite some synergies with the new Java Identity API (JSR 351) given it aims at unified security and compliance for Social Networking and Information services.

2.11 Are there any internationalization or localization issues?

This specification uses the I18N support in Java

2.12 Are there any existing specifications that might be rendered obsolete, deprecated, or in need of revision as a result of this work?

No, while this specification should build on top of well-established Java APIs like CDI, RESTful Web Services, JSON (JSR 353) or other yet to be filed JSRs there is no Java specification for Social Media as such, thus nothing can be directly replaced.

2.13 Please describe the anticipated schedule for the development of this specification.

Q4 2011: finalise expert group
Q2 2012: Early Draft of spec and API
Q4 2012: Public Review
Q1 2013: Final Release

2.14 Please describe the anticipated working model for the Expert Group working on developing this specification.

The primary means of communication will be email (via mailing list) and conference calls, with meetings on IRC channels if deemed appropriate. Face-to-face meetings will be scheduled if needed.

2.15 Provide detailed answers to the transparency checklist, making sure to include URLs as appropriate:

* *The public can read the names of the people on the Expert Group.*
We intend to use the Java.net infrastructure, popular among JSRs and other open source projects. This will be publicly visible and accessible. The preferred code repository is GitHub, also frequently used by JSRs and many other open source projects.
* *The Expert Group business is regularly reported on a publicly readable alias.*
We intend to use the Java.net infrastructure, popular among JSRs and other open source projects. This will be publicly visible and accessible.
* *The schedule for the JSR is publicly available, it's current, and I update it regularly.*
We intend to use the Java.net infrastructure, popular among JSRs and other open source projects. This will be publicly visible and accessible.
* *The public can read/write to a wiki for my JSR.*
We intend to use the Java.net infrastructure, popular among JSRs and other open source projects. This infrastructure provides a publicly accessible wiki system.
* *I read and respond to posts on the discussion board for my JSR on jcp.org.*
Yes, once the JSR is approved these details will be made public.
* *There is an issue-tracker for my JSR that the public can read.*
We intend to use the Java.net infrastructure, popular among JSRs and other open source projects. This infrastructure provides a publicly accessible issue tracker.
* *I have spoken at conferences and events about my JSR recently.*
I am a respected speaker at numerous conferences and JCP EC Member since 2008. I wrote a system called eCampus for Donau Uni Krems between 1999 and 2000. A J2EE 1.0 based service allowing students to collaborate on their studies over the web, as well as rate courses and tutors. Somewhat similar to the CourseMatch, a system, Facebook founder Mark Zuckerberg wrote in his somophore year (2003) Also similar projects like the client facing design of Nokia Pilots, a Social Networking Portal for trials of devices, software or services by Nokia. (recently integrated with Ovi/NokiaBetaLabs) Followed by integration of Enterprise Social for other leading names among Mobile, Telco as well as Social and Cloud providers themselves.
* *I am using open-source processes for the development of the RI and/or TCK.*
Yes, this is the plan. While, the spec should use Java.net, RI and TCK will either use GitHub or a infrastructure used by Apache projects like Shindig or DeltaSpike, where either RI or TCK are created within the Apache ecosystem. Both spec leads have a track record in this regard.
* *The Update tab for my JSR has links to and information about all public communication mechanisms and sites for the development of my JSR.*
We intend to use the Java.net infrastructure for the spec, popular among JSRs and other open source projects. This will be publicly visible and accessible.

2.16 Please describe how the RI and TCK will de delivered, i.e. as part of a profile or platform edition, or stand-alone, or both. Include version information for the profile or platform in your answer.

Werner Keil and Antoine Sabot-Durand will deliver a Reference Implementation (RI) and Technology Compatibility Kit (TCK) compatible with Java (EE) 7 and where applicable Java ME 7. The license used will be the Apache Software License (ASL) 2.0, an open source license which is compatible with Java EE licensing.

2.17 Please state the rationale if previous versions are available stand-alone and you are now proposing in 2.13 to only deliver RI and TCK as part of a profile or platform edition (See sections 1.1.5 and 1.1.6 of the JCP 2 document).

N/A

2.18 Please provide a description of the business terms for the Specification, RI and TCK that will apply when this JSR is final.

Werner Keil and Antoine Sabot-Durand will deliver a Reference Implementation (RI) and Technology Compatibility Kit (TCK) compatible with Java EE 7. The license used will be the Apache Software License (ASL) 2.0, an open source license which is compatible with Java EE licensing. The Specification will use the standard license template.

Section 3: Contributions

3.1 Please list any existing documents, specifications, or implementations that describe the technology. Please include links to the documents if they are publicly available.

- Open Social
- Apache DeltaSpike Proposal / Seam Social
- Apache Shindig
- Twitter4J
- SocialSite
- DaliCore
- Open Social Java
- fb4java
- Google+ Java API
- Social Java
- Ning Java API
- foursquare APIv2

3.2 Explanation of how these items might be used as a starting point for the work.

With most authors of above frameworks discussions are in progress as to how they might get involved. Especially those already EG Members or supporting the JSR wish to contribute their solutions, and given some those make extensive use of Java standards, those items look especially promising.