RFP 178 - DS Field Reference Enhancements

Draft

7 Pages

Abstract

Declarative Services provide functionality to implement Dependency Injection programming in OSGi based applications. R6 introduced field injection by using a field @Reference annotation. This RFP gathers requirements for additional @Reference properties, supporting often occuring use cases.

0 Document Information

0.1 License

DISTRIBUTION AND FEEDBACK LICENSE, Version 2.0

The OSGi Alliance hereby grants you a limited copyright license to copy and display this document (the “Distribution”) in any medium without fee or royalty. This Distribution license is exclusively for the purpose of reviewing and providing feedback to the OSGi Alliance. You agree not to modify the Distribution in any way and further agree to not participate in any way in the making of derivative works thereof, other than as a necessary result of reviewing and providing feedback to the Distribution. You also agree to cause this notice, along with the accompanying consent, to be included on all copies (or portions thereof) of the Distribution. The OSGi Alliance also grants you a perpetual, non-exclusive, worldwide, fully paid-up, royalty free, limited license (without the right to sublicense) under any applicable copyrights, to create and/or distribute an implementation of the Distribution that: (i) fully implements the Distribution including all its required interfaces and functionality; (ii) does not modify, subset, superset or otherwise extend the OSGi Name Space, or include any public or protected packages, classes, Java interfaces, fields or methods within the OSGi Name Space other than those required and authorized by the Distribution. An implementation that does not satisfy limitations (i)-(ii) is not considered an implementation of the Distribution, does not receive the benefits of this license, and must not be described as an implementation of the Distribution. "OSGi Name Space" shall mean the public class or interface declarations whose names begin with "org.osgi" or any recognized successors or replacements thereof. The OSGi Alliance expressly reserves all rights not granted pursuant to these limited copyright licenses including termination of the license at will at any time.

EXCEPT FOR THE LIMITED COPYRIGHT LICENSES GRANTED ABOVE, THE OSGi ALLIANCE DOES NOT GRANT, EITHER EXPRESSLY OR IMPLIEDLY, A LICENSE TO ANY INTELLECTUAL PROPERTY IT, OR ANY THIRD PARTIES, OWN OR CONTROL. Title to the copyright in the Distribution will at all times remain with the OSGi Alliance. The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted therein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

THE DISTRIBUTION IS PROVIDED "AS IS," AND THE OSGi ALLIANCE (INCLUDING ANY THIRD PARTIES THAT HAVE CONTRIBUTED TO THE DISTRIBUTION) MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DISTRIBUTION ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.
NEITHER THE OSGi ALLIANCE NOR ANY THIRD PARTY WILL BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THE DISTRIBUTION.

Implementation of certain elements of this Distribution may be subject to third party intellectual property rights, including without limitation, patent rights (such a third party may or may not be a member of the OSGi Alliance). The OSGi Alliance is not responsible and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights.

The Distribution is a draft. As a result, the final product may change substantially by the time of final publication, and you are cautioned against relying on the content of this Distribution. You are encouraged to update any implementation of the Distribution if and when such Distribution becomes a final specification.

The OSGi Alliance is willing to receive input, suggestions and other feedback (“Feedback”) on the Distribution. By providing such Feedback to the OSGi Alliance, you grant to the OSGi Alliance and all its Members a non-exclusive, non-transferable,

worldwide, perpetual, irrevocable, royalty-free copyright license to copy, publish, license, modify, sublicense or otherwise distribute and exploit your Feedback for any purpose. Likewise, if incorporation of your Feedback would cause an implementation of the Distribution, including as it may be modified, amended, or published at any point in the future ("Future Specification"), to necessarily infringe a patent or patent application that you own or control, you hereby commit to grant to all implementers of such Distribution or Future Specification an irrevocable, worldwide, sublicenseable, royalty free license under such patent or patent application to make, have made, use, sell, offer for sale, import and export products or services that implement such Distribution or Future Specification. You warrant that (a) to the best of your knowledge you have the right to provide this Feedback, and if you are providing Feedback on behalf of a company, you have the rights to provide Feedback on behalf of your company; (b) the Feedback is not confidential to you and does not violate the copyright or trade secret interests of another; and (c) to the best of your knowledge, use of the Feedback would not cause an implementation of the Distribution or a Future Specification to necessarily infringe any third-party patent or patent application known to you. You also acknowledge that the OSGi Alliance is not required to incorporate your Feedback into any version of the Distribution or a Future Specification.

I HEREBY ACKNOWLEDGE AND AGREE TO THE TERMS AND CONDITIONS DELINEATED ABOVE.

0.2 Trademarks

OSGi™ is a trademark, registered trademark, or service mark of the OSGi Alliance in the US and other countries. Java is a trademark, registered trademark, or service mark of Oracle Corporation in the US and other countries. All other trademarks, registered trademarks, or service marks used in this document are the property of their respective owners and are hereby recognized.

0.3 Feedback

This document can be downloaded from the OSGi Alliance design repository at https://github.com/osgi/design The public can provide feedback about this document by opening a bug at https://www.osgi.org/bugzilla/.

0.4 Table of Contents

0.5 Terminology and Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in 5.1.

`Source code is shown in this typeface.`

0.6 Revision History

The last named individual in this history is currently responsible for this document.

Revision	Date	Comments
Initial	*November 12 2015*	*Initial contribution* *Tim Verbelen, iMinds – Ghent University, tim.verbelen@intec.ugent.be*
Update	*November 12 2015*	*Updates on review of initial contribution* • *unify concept around term "key property"* • *add requirement for multi-value properties* • *clarify single/List Map value ordering heuristics* *Raymond Augé, Liferay, Inc., raymond.auge@liferay.com*

1 Introduction

Since Compendium R6 Declarative Services has support for field injection, by using the @Reference annotation on fields. However, often the use case arises where one adds a custom (key) service property on a service instance with the intention of collecting service instances of a particular type into a map - keyed by this **key property**. This scenario is currently not supported by field injection, although it could be easily added.

2 Problem Description

When having multiple instances of the same service interface, often a key property is used to identify and differentiate these instances. When using these service instances from a DS component, it makes sense to collect these services and map them by their key property.

For example using the good old method injection:

```
Map<String, SomeService> services = new ConcurrentHashMap<>();

@Reference(
      cardinality=ReferenceCardinality.MULTIPLE,
      policy=ReferencePolicy.DYNAMIC)
void addService(SomeService s, Map<String, Object> properties){
    String key = (String)properties.get("keyProperty");
    services.put(key, s);
}

void removeService(SomeService s, Map<String, Object> properties){
    String key = (String)properties.get("keyProperty");
    services.remove(key);
}
```

However, this is currently not possible using field injection.

3 Use Cases

Given the example stated above, one could easily translate this to a field injection equivalent, i.e.:

```
@Reference(mapBy="keyProperty")
Map<K, S> services;
```

Where `keyProperty` is the key property that is looked up in the service instance's service properties, the value of this service property is then converted/casted to type K, and used as key in the map with as value the actual service instance of type S.

4 Requirements

- R0001: The solution must use the @Reference annotation on a Map field, mapping a key property value to a service instance
- R0002: The solution must allow to define the key property that is used to map the service instances
- R0003: The resulting Map must be either single value or a List (e.g. list of all service instances that have the key property with the same value defined)
- R0004: In the case of single value Map, the value must be determined either by a specified Comparator or by typical service ordering heuristics of which the “best match” is selected.
- R0004: In the case of a List value Map, the order of the list items must be determined, and maintained by a specified Comparator or when no comparator is specified by typical service ordering heuristics.
- R0005: In cases where the key property contains multiple values (e.g. string+) the service may be mapped by several keys in the map and in either case of single or List map would be subject to ordering as specified in R0004 & R0005

5 Document Support

5.1 References

[1]. Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, RFC2119, March 1997.

[2]. Software Requirements & Specifications. Michael Jackson. ISBN 0-201-87712-0

5.2 Author's Address

Name	Tim Verbelen
Company	iMinds, Ghent University
Address	Gaston Crommenlaan 8 box 201, 9050 Ghent, Belgium
Voice	
e-mail	tim.verbelen@intec.ugent.be

Name	Raymond Augé
Company	Liferay, Inc.
Address	24 Leonard St N Box 580, Dowling Ontario, P0M1R0, Canada
Voice	
e-mail	raymond.auge@liferay.com

5.3 End of Document