RFP 182: Service Decorations

Draft

8 Pages

Abstract

Service properties provide a powerful way to add metadata to a service. It they can be used for filtering, inspection and are posted to interested parties via a change notification mechanism. Up until now service properties are owned by the bundle that registers the service. Only that bundle has the ServiceRegistration that allows modification of these properties.

This RFP explores the possibility to allow other bundles to provide service properties or decorate services somehow.

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 6.1.

```
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	August 2016	David Bosschaert, Tim Ward, Peter Kriens - Initial Version
0.1	September 2016	David Bosschaert, Feedback from San Jose F2F: add section 2 and 3, remove whiteboard marker use-case.

1 Introduction

Service properties provide a powerful way to add metadata to a service. It they can be used for filtering, inspection and are posted to interested parties via a change notification mechanism. Up until now service properties are owned by the bundle that registers the service. Only that bundle has the Service Registration that allows modification of these properties.

This RFP explores the possibility to allow other bundles to provide service properties or decorate services somehow.

2 Application Domain

OSGi services are widely used for a large variety of applications. Services are used to represent runtime objects in Java but can also be used to represent live devices in the physical world or compute nodes in a cloud environment. OSGi services have properties associated with them to represent metadata. This metadata can be used to advertise additional capabilities of the service, for example a supported Mime-Type for a decoder service, or as a mechanism to convey information to other parts of the system. For example the Service PID is used to advertise under what persistent key the service can be configured.

Service properties can be modified dynamically. The bundle that registers the service is given a `ServiceRegistration` object which can be used to update the service properties. However, this Service Registration object is not retrievable by other bundles, so if other bundles wish to update the service properties there is generally no straightforward way to achieve this.

There exist a number of use-cases where other bundles want to augment the service properties of a given service. For example to geotag service that represent physical devices, or to tag services with additional information which can then be exposed in a distributed environment using Remote Services.

This RFP seeks to add a general mechanism to allow bundles to augment service properties of service registered by other bundles.

2.1 Terminology + Abbreviations

3 Problem Description

Service properties can only be updated via the `ServiceRegistration` object, so unless the service registering bundle explicitly shares this registration other bundles cannot directly update the service properties.

Normally bundles don't share out the Service Registration, given that this object contains an API to unregister the service they will most likely not do this in the future either as bundles generally want to be in control of the lifecycle of their own services.

3.1 DS Configured Components

Configuration-admin configured components in Declarative Services reflect their configuration via their service properties. So one way to update the properties of these services is to update the configuration associated with their PID to include any additional data that needs to appear as service properties.

Other non-DS bundles sometimes follow the same pattern of exposing their configuration as service properties.

While this is a common pattern, it's not required by the OSGi specifications. Additionally, it may be undesirable to update the configuration of a component to merely represent additional transient metadata.

Finally, this is not a mechanism that will work for all services. Only services that follow the pattern of exposing their configuration as properties will make this possible. There is a large class of OSGi services that are not configured at all and tying these to Configuration Admin just for service properties seems like an undesirable extra dependency.

3.2 Service Registry Hooks

Service Registry Hooks provide a means to hide services from certain bundles. This can be used to hide a certain service, create a new service with extra properties and expose this proxy service to service consumers instead.

While this will work in general, especially for services that implement interfaces, it is a heavyweight solution that can lead to a lot of surplus service registrations of hidden services. Additionally there are timing issues. The service registry hook needs to be present before the first service client appears, otherwise this client will see the original (non-proxy) service and will not be notified of any additions to the service properties made via the proxies.

4 Use Cases

4.1 NodeStatus services

RFC 183 defines a NodeStatus service which represents a node running in the cloud or some other compute platform. The RFC needs a way for nodes to mark themselves so that a provisioning agent or scheduler can know what type of work should be given to this node. For example, a node might have a very strong CPU and the application deployer wants to use the node for offloading computational jobs. Or a node might have a lot of disk space and should be used to host a database.

A NodeStatus service is a simple OSGi service which is exposed via Remote Services. Its metadata is made available via service properties. While RFC 183 could define a proprietary mechanism to allow other bundles to extend the NodeStatus service properties, a general solution on the Core OSGi level would be preferable.

4.2 Location for Services

A bundle keeps a configurable database for devices in the house. When it sees the service appear, it adds a 'location' property to the service so that other bundles can see where the service is located.

4.3 Service Wiring

Each DS component gets its references with a target that asserts its own identity.

```
@Reference(target="(identity=me)")
```

A bundle then provides a GUI to wire services together. In runtime, this bundle adds a property to wired services with a list of identities that it should wire to.

4.4 ~~Whiteboard Marking~~

~~As a whiteboard implementation I want to "mark" a service as having been processed. This way parties that are interested in the processed services can react to them by looking for the "processed" property.~~

~~This would still need to work in the case where the whiteboard service was registered using Declarative Services and potentially had its properties updated using Config Admin. This could get messy, with the service property being briefly deleted by SCR on a config change, and then having to be re-added by the whiteboard implementation, even if there was no period of unavailability for the processed service…~~

5 Requirements

SD0010 – Allow bundles to provide additional properties to an OSGi service, even if these bundles do not have access to the original ServiceRegistration of that service.

6 Document Support

6.1 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

6.2 Author's Address

Name	David Bosschaert
Company	Adobe
Address	
Voice	
e-mail	bosschae@adobe.com

Name	Tim Ward
Company	Paremus
Address	
Voice	
e-mail	

Name	Peter Kriens
Company	
Address	
Voice	
e-mail	

6.3 End of Document