Lyo OSLC4J, Oauth Library and OSLC Test Suite 1.1 Release and Graduation Review

Michael Fiedler (fiedler.mf@gmail.com)
Steve Speicher (sspeiche@us.ibm.com)
Eclipse Lyo Project Leads
23 January, 2013

About

The Eclipse Lyo project is focused on providing an SDK to enable adoption of OSLC specifications. OSLC (Open Services for Lifecycle Collaboration: http://open-services.net) is an open community dedicated to reducing barriers for lifecycle tool integration. The community authors specifications for exposing lifecycle artifacts through uniform (REST) interfaces and relying on Internet and Linked Data standards.

Eclipse Lyo project is a companion to the continuing specification efforts of the OSLC community. Its main purpose is to expand adoption of OSLC specifications and to enable the Eclipse community to easily build OSLC compliant tools.

The project was approved by the PMC in July 2011 and initial contributions were committed in September 2011.

The project had a 1.0 release and graduated from incubator status on 10 October, 2012

Introduction

Lyo is a Technology sub-project: http://eclipse.org/lyo

This is the second release of Lyo components as an Eclipse project. The goals are to continue to build an Eclipse community around Lyo by providing enhancements to the Java SDK for OSLC (OSLC4J), an update to the OSLC assessment test suite and code samples of OSLC provider and consumer implementations.

New in the 1.1 release are several enhancents:

1. An additional OSLC4J library which provide assistance with OSLC query syntax (org.eclipse.lyo.core.query package)
2. An additional OSLC4J library which provides a set of client/ consumer APIs to consumer OSLC from a provider. This is in the org.eclipse.lyo.clients project in the org.eclipse.lyo.client.git repository.
3. New libraries providing APIs for Oauth 1.0/1.0a provider development. OAuth is commonly used for OSLC provider implementations.

Milestones leading to Release

Lyo has a history of of quarterly Milestones in preparation for this graduation:

- M1 – December 2012
- M2 – March 2012
- M3 – June 2012
- 1.0 – October 2012
- 1.1 - (Proposed) January 2013

1.1 is an update to 1.0 with some additional enhancements. It has not undergone a formal milestone progression, but has been adopted by several Lyo consumers prior to proposing a release.

- Early adoption of 1.1
  - Snapshot/interim releases are being used to develop real OSLC integrations.
  - All new elements (Query library, Client library and OAuth library have beenb adopted by Lyo consumers

Committer Diversity

The following individual committers are actively involved:

- Michael Fiedler (IBM)
- Steve Speicher (IBM)
- Olivier Berger (Institut Telecom)
- Paul McMahan (IBM)

Additional active significant contributors

- Yuhong Yin
- Steven Pitschke
- Timothy Eck
- Stephanie Ouillon
- Lars Ohlsen
- Alberto Giammaria

- http://www.eclipse.org/projects/project.php?id=technology.lyo

Main Features to be released as 1.1

- OSLC4J – Java SDK for OSLC implementations (1.0 and 1.1)
  - Annotations to add OSLC attributes and meta-data to Java objects representing OSLC resources
  - Serialize/De-serialize Java objects as OSLC resources
  - (NEW) 1.1 Enhancements
    - OSLC Query library – API for OSLC Query syntax
    - OSLC Client library – API for OSLC consumers
- OSLC assessment Test Suite (1.0 and 1.1)
  - Tests cover OSLC Core spec + Change Management, Quality Management, Asset Management and Requirements Management specifications
    - (NEW) Coverage for Automation Specification in 1.1
- (NEW) OAuth provider library
  - Core, persistence and webapp libraries to ease OAuth provider implementation

API Maturity

- OSLC4J API
  - No breakage of 1.0 API – continued stability
  - Early 1.1 adopters have developed real integrations – intent is to not break backward compatibility
  - Future additions to API will be designed to be non-breaking
  - Currently, no deprecated API elements

Additional Lyo project content

- Additional Lyo components NOT being formally released in 1.1 – under consideration for the future releases
  - SDKs for other languages (Perl)
- Lyo also contains code samples which are for developers to use as aids and examples for their own implementations. Not formally released.
  - Reference implementations for specifications
  - Sample OSLC integrations for Bugzilla, Excel and Sharepoint.
  - Client library sample implementations
- Previously proposed .NET content has moved to Microsoft Codeplex due to IP challenges with 3rd party dependencies.

Non-code aspects

OSLC4J and TestSuite documentation:

http://wiki.eclipse.org/Lyo/LyoOSLC4J
http://wiki.eclipse.org/Lyo/BuildTestSuite

Contributor Guide:

http://wiki.eclipse.org/Lyo/ContributorsGettingStarted

Mailing List:

https://dev.eclipse.org/mailman/listinfo/lyo-dev

Forum:

http://www.eclipse.org/forums/eclipse.lyo

IRC Channel:

irc://irc.freenode.net/#eclipse-lyo

Continous Integration:

Hudson-based builds

Bugzilla

Bugzilla Statistics (on 2013-01-14):

Open: 24 bugs, 36 enhancement requests

- No Blocker/Critical open

Fixed since 1.0 : 55 bugs, 18 enhancement requests

Community

- Eclipse DemoCamp : Durham, NC, November 2011
- EclipseCon Europe: Ludwigsburg, Germany, November 2011
- EmbeddedWorld: Nuremberg, Germany, February 2012
- EclipseCon: Reston, VA March 2012 – Boston, MA, March 2013
- IBM Rational Innovate: Orlando, FL June 2012
- JavaOne: San Francisco, CA, October 2013
- OSLC Community Site: http://open-services.net
- CESAR Project: http://www.cesarproject.eu/ (Lyo adopter)

IP Log for Lyo

http://www.eclipse.org/projects/ip_log.php?projectid=technology.lyo

Proposed Schedule

- 1.0 – October 2012
- 1.1 – January 2013
  - Enhanced OSLC query library
  - OSLC Client library
  - OSLC4J Enhancements
  - Test suite improvements
- 1.2/2.0 (TBD) – June 2013
  - Enhancements based on community feedback

Project Plan

http://www.eclipse.org/projects/project-plan.php?projectid=technology.lyo

http://wiki.eclipse.org/Lyo/ProjectPlans

Themes for future releases:

- SDKs for technologies other than Java
- Test suite improvements (additional domain coverage, improved depth of test cases, improved reporting)
- API stabilization
- Improved documentation
- Usability
- Building a community

