

ClerezzaProposal

Clerezza Proposal

This proposal is being discussed on the [general at incubator.apache.org](mailto:general@incubator.apache.org) mailing list.

Abstract

Clerezza is an OSGi-based modular application and set of components (bundles) for building RESTful Semantic Web applications and services.

Proposal

Clerezza can be used as a platform providing all the compile and runtime requirement for building semantic applications, or used as individual bundles within an OSGi framework, e.g. Apache Sling, Apache [ServiceMix](#), or the Eclipse platform.

Clerezza provides:

- An API modeling the W3C RDF standard without any vendor specific additions.
- Adapter for various triple stores including Sesame, Jena TDB, and Mulgara.
- Front-End adaptors, currently to run applications written against the Jena API. Support for RDF2Go is planned.
- A JAX-RS implementation designed to work in an OSGi environment and allowing to provide Root-Resources as OSGi services.
- Web access to RDF graphs, including a SPARQL-Endpoint.
- Extensions to JAX-RS allowing to bind Root-Resource classes to specific RDF-Types rather than to URI-Paths.
- Templating mechanism (Renderlets) allowing to render RDF resources returned by JAX-RS resource methods to various formats
- Support for Scala for writing modules, [ScalaServerPages](#) to easily write renderlets, DSL for accessing graphs.
- Authentication and authorization based on JAAS and OSGi Conditional Permission Admin
- Support for user bundles: Users can have a permission to upload their own sandboxed bundles. The URI space these bundles are allowed to register their JAX-RS resources can be limited with a prefix.
- Scripting: Scripting based on javax.script (currently support for JRuby and Scala)
- Documentation: Bundles can provide their documentation in RDF. These are used for online documentation as well as for building Maven sites (with a Maven reporting plugin)

The RDF abstraction layer can be used independently of other aspects of Clerezza. It allows applications to be written regardless the used backend. In its purpose, it is similar to RDF2Go, but provides a significantly more modular interface allowing e.g. to independently switch the storage, querying, or serialization layer. Furthermore, it doesn't introduce concepts alien to the RDF model such as blank node labels, but is in its core strictly limited to RDF semantics.

The JAX-RS implementation can also be used independently of any other components. It allows OSGi services to provide a RESTful interface to their methods. By being based on wmiwyg WRHAPI, it can run both on the default OSGi Web Service as well as on a jetty instance listening on a different port.

Background

The current web trends focusing on information sharing, interoperability and collaboration. Therefore the behaviour of the end-user has changed over the last years: end-users not only consuming information they also producing content anytime anywhere - in contrast to non-interactive websites where users are limited to the passive viewing of information that is provided to them. Since the end-users are sensitized to the possibilities of the web the web application requirements increases. Examples of such applications are social-networking sites, wikis, blogs and mashups.

The REST paradigm and Semantic Web technologies support these trends and form the basis for the upcoming Web of Data (a.k.a. linked data, Web 3.0). They change the paradigms for developing complex Web applications. Clerezza allows to develop applications that integrate perfectly in the Semantic Web providing all accessible resources in machine understandable formats without imposing additional burdens on the developer. Additionally, thanks to the flexibility of the RDF model used as back-end, some tedious database related tasks required for traditional Web application development are no longer needed.

Rationale

Most Web application framework are not designed to leverage the full power of HTTP but often try to reproduce non Web design patterns for the Web environment. In general, application frameworks are oriented towards relational or hierarchical data structures. While attempts to overcome this such as Drupal have become very popular, they do not at their core base on the stack of Semantic Web standards. Clerezza will prove that the flexibility of the RDF doesn't result in increased complexity, but on the contrary allows for fast prototyping and development.

Initial Goals

The initial goals for Clerezza are:

- Donate the existing codebase and import it.
- Setup the incubation infrastructure (svn repository, build system, website), so we can run continuous builds with automated tests and publish all available documentation.
- Get people involved in advancing the code base in different directions, integrating it with other projects at Apache.
- Prepare for an initial release that demonstrates the systems core capabilities.

Current Status

The current codebase is developed and tested using Apache Felix. It has been developed intensively and reviewed at trialox since August 2008 using Scrum, with a development process emphasizing individual accountability and reviews. We have internally demonstrated that we can release codes as scheduled. Platform core functionalities are available, however the need for new features may arise, performance and robustness could be improved. Incomplete documentation for the project is available with the individual artifacts, both in the generated maven sites as in a version available exposed by Clerezza at runtime. We also have a wiki at <http://wiki.trialox.org> with some information mainly on the development process. We also use mailing lists for communication among developers and users.

Meritocracy

The core developers understand what it means to have a process based on meritocracy. We will provide continuous efforts to build an environment that supports this, encouraging community members to contribute.

Community

Trialox has been developing the current codebase since August 2008. Trialox was founded in partnership with the University of Zurich and could benefit from previous research work at the Department of Informatics. Part of the team from the beginning was Reto Bachmann Gmür who has been developing open source Semantic Web applications for many years, including working with the Jena team at HP Labs.

Trialox has contributed to the JAX-RS specification. Some of the code written by trialox is used by Open Source projects such Paxle and Gradino.

Clerezza is used by globally active non-profit organisations such as the WWF. These organisations have strong developer networks including motivated volunteers, which will contribute to Clerezza.

Core Developers

People from Trialox, the University of Zurich, as well as partner companies of Trialox have contributed to the project. Currently, the following persons are core developers of Clerezza:

- Manuel Innerhofer, Developer at Trialox since November 2008.
- Hasan Hasan, Developer and Senior Researcher at University of Zurich since 2006, developing Clerezza since August 2008. His current research interests cover P2P networking, Service Level management, and Internet security.
- Tsuyoshi Ito, Developer and Scrum Master at Trialox, developing Clerezza since August 2008. He has researched at the University of Zurich since 2005. His research interest was computer-supported Learning (Educational Engineering)
- Reto Bachmann-Gmür, Developer and Architect at Trialox, developing Clerezza since August 2008.

Alignment

We provide a launcher which runs Clerezza's bundles within Apache Felix. We also provide feedback about the usage of Apache Felix and its components including the framework security via the mailing list. For building Clerezza's bundles we use Apache Maven and various plugins. We also have developed a plugin to help managing projects which contain ontologies, so that Java classes representing those ontologies can be pre-compiled. Other projects which based on Web services and/or RDF can benefit from Clerezza or its specific bundles. We are open to collaborate with other Apache projects which can benefit from functionality provided by Clerezza. Clerezza has the advantage of being very modular and independent of application frameworks, thus can be easily integrated with other Apache projects. UIMA and Tika come to mind, as they would help extract semantic information from various data types and formats. An alignment to the JAX-RS implementation in Apache CXF could not only help removing the dependencies to the CDDL-licensed code taken from Jersey but also help providing a fully framework independent implementation with a larger group of developers and thus higher quality.

Known Risks

The current team of Clerezza core developers is small, but being an innovative project in the semantic "space", we are confident that Clerezza can attract new developers.

Clerezza has been started as an Open Source project providing mercurial repository for public access to source codes and also a publicly accessible JIRA instance for issues tracking. Clerezza is licensed since project begin under Apache License version 2.0. Some of the initial committers already have strong experiences with Open Source software development. Others, while not being totally inexperienced, are willing to learn.

The risk that Clerezza will be an orphaned product is considered small. Three main factors will avoid this to happen:

- Trialox and its founder Getunik have a vital interest in continuous development in this open source foundation
- Clerezza is used as foundation for research as well as student projects at the University of Zurich
- There is a strong commitment by Reto Bachmann-Gmür to maintain Clerezza
- WWF expressed their support to deploy Clerezza

Documentation

A small set of further documentation is available under the following links:

- 1 <http://trialox.org/projects/org.clerezza.rdf.core/documentation/overview.xhtml>
- 2 <http://trialox.org/projects/org.clerezza.triaxrs.parent/org.clerezza.triaxrs/documentation/>

Initial Source

Clerezza has been in development since mid 2008. Public access to the source is provided through <http://scm.trialox.org/>.

Source and Intellectual Property Submission Plan

The current codebase is owned by trialox, and will be donated together with its documentation. We will get the paperwork out of the way as soon as possible.

External Dependencies

There are quite a few open source libraries already used. They have Apache compatible licenses, with one issue to solve around Jersey which is CDDL.

The libraries, their sources and licenses are listed here:

Apache Felix, ASL:

- Framework
- Framework Security
- Configuration Admin
- maven-scr-plugin
- maven-bundle-plugin

OSGi Alliance, ASL:

- Core
- Compendium

Apache Maven, ASL:

- apache-maven

Eclipse, ASL:

- Jetty

OPS4J, ASL:

- Pax Exam
- Pax Logging
- Pax protocol mvn-uri

WYMIWYG, ASL:

- wrhapi
- wymiwyg-commons

jQuery, MIT license:

- jquery

Hewlett-Packard Development Company, BSD License:

- Jena (for the optional jena frontend adaptor, as well as for jena based serializer/parser)
- Jena TDB (for the optional tdb backend adaptor)

OpenRDF.org, BSD license (for the optional sesame backend adaptor):

- Sesame

Mulgara.org, Open Software License ("OSL") v. 3.0 (for the optional mulgara backend adaptor):

- mulgara

XSite (<http://xsite.codehaus.org/>), BSD license:

- xsite-maven-plugin

Jersey (<https://jersey.dev.java.net/>), CDDL license

The current code bases on code licensed under the CDDL, according to <http://apache.org/legal/resolved.html> we understand we have to get rid of these before making a release, or redistribute in binary form only. The following files are affected.

- [QualityFactor.java](#)
- [HttpDateFormat.java](#)
- [HttpHeaderReaderImpl.java](#)
- [UriPattern.java](#)
- [UriComponent.java](#)
- [HttpHeaderListAdapter.java](#)

- [MultivaluedMapImpl.java](#)
- [HttpHeaderReader.java](#)
- [ByteArrayProvider.java](#)
- [SourceProvider.java](#)
- [AbstractMessageReaderWriterProvider.java](#)
- [StreamingOutputProvider.java](#)
- [FormMultivaluedMapProvider.java](#)
- [FileProvider.java](#)
- [ReaderProvider.java](#)
- [InputStreamProvider.java](#)
- [MediaTypeProvider.java](#)
- [EntityTagProvider.java](#)
- [CacheControlProvider.java](#)
- [NewCookieProvider.java](#)
- [CookieProvider.java](#)

Required Resources

Mailing lists:

- clerezza-dev
- clerezza-commits
- clerezza-user (only after leaving the incubator)

Subversion:

- <https://svn.apache.org/repos/asf/incubator/clerezza>

Issue Tracking:

- JIRA: Apache Clerezza (Clerezza)

Initial Committers

These committers have either worked on the initial codebase (Reto, Immanuel, Tsuy, Hasan) or expressed an interest in extending the project:

- Reto Bachmann-Gmür (trialox)
- Manuel Innerhofen (trialox)
- Tsuyoshi Ito (trialox)
- Hasan Hasan (University of Zurich)
- Bertrand Delacretaz (ASF member, Day Software)
- Michael Marth (Day Software)
- Tommaso Teofili (Apache UIMA committer)

Affiliations

Manuel Innerhofen, Tsuyoshi Ito and Reto Bachmann-Gmür work at trialox and might get paid to work on Clerezza.

Hasan Hasan from University of Zurich is paid to work in a project that is based on Clerezza.

Michael Marth and Bertrand Delacretaz work for Day Software.

Sponsors

We have approached both the champion and an initial list of mentors that have agreed to mentor this project.

Champion:

- Bertrand Delacretaz

Mentors:

- Gianugo Rabellino
- Niclas Hedhman
- Ross Gardler
- Karl Pauls
- Reinhard Pötz

Sponsor:

- Apache Incubator