The ELisA facility - RESTful API and client libraries

Alina Corso-Radu, Luca Magnoni, Raul Murillo Garcia
University of California, Irvine, USA

Contact: alina.radu@cern.ch

1 Introduction

The ATLAS collaboration is composed of about 3000 scientific users from 38 countries who, together, run one of the largest high energy physics experiments in the world.

An electronic logbook is used to record and share messages about ATLAS data taking activities by system operators, experts and automated services. Since the log messages are of interest to the whole ATLAS collaboration, the logbook tool has to meet strict requirements:

- High-availability and scalability
- Fast and effective message browsing, editing and visualization
- Support for different programming languages via a standard API

2 REST-based solution

An HTTP-based REST API has been chosen as solution to provide logbook functionalities for programmable access. This approach decouples server-side logic from client implementation details. Moreover a set of client libraries offer a developer friendly and homogenous interface across different languages.

Following the REST architectural principles, a client makes an HTTP request using the standard verbs (GET/PUT/POST/DELETE) to a structured URL. Operations such as creating a new message, editing existing ones or getting a list of entries with specified criteria are supported in this way.

The client obtains (or sends) a representation of the requested resource in several formats, such as JSON and XML. This standard and lightweight approach is fully integrated within the ELisA MVC design and decouples the server implementation from the client technologies.

3 MVC and Spring

The Model-View-Controller approach introduces a well defined structure in the ELisA architecture. The Spring framework provides the means and the technology to implement it. Model objects, mapped to messages, are retrieved from the Oracle database via Spring Data and Hibernate tools. Client requests, coming both from the web front-end or the REST API, are handled by Controllers and resolved in dedicated Views. Controllers are standard Spring beans responsible for handling the mapping between user requests and model objects.

4 REST API

http://elisa.cern.ch/api/messages
http://elisa.cern.ch/api/messages/{ID}
http://elisa.cern.ch/api/messages/{ID}/attach
http://elisa.cern.ch/api/messages/{ID}/attach
http://elisa.cern.ch/api/messages/{ID}/attach

GET configuration info, i.e. message type

5 Client libraries

Client libraries offer a new layer of abstraction in order to hide the REST architecture and underlying HTTP communication. Methods in the libraries map to typical logbook functionality such as insert, reply, update and retrieve operations. Currently, two programming languages are supported: Java and Python. A third one (C++) will be added soon to the list of supported languages.

Depending on the source of the request, user credentials are verified either using LDAP or SSO. In the former case, the user specifies the username and password and the library uses base64 encoding to embed the credentials in the HTTP requests. In the latter case, the library embeds the credentials in the HTTP requests based on a SSO cookie and the authentication is done by the HTTP server and the REST server.

In addition, a set of command line utilities has been developed on top of the Python client API library to provide a programmatic-free facility for users to perform the most common operations:

- elisa_get: retrieves messages based either on an ID or a search criteria.
- elisa_insert: inserts a new message.
- elisa_reply: replies to an existing message.
- elisa_update: updates an existing message

6 Backend performance

The plot in this box shows the comparison of the front-end loading time for the old logbook implementation (ATLOG) and for ELisA. The improvement of ELisA compared with the old implementation is clearly visible in the plot. The new implementation scales well with the number of messages retrieved, and it is adding a minor time overhead on top of the Oracle query-response time. The database schema has been preserved for backward compatibility, but it can be subject to revision and optimization in the future to further improve the data retrieval performance.

7 Conclusion

ELisA logbook is used by ATLAS at LHC since the beginning of data taking in 2012 and has been positively received by many users. The adoption of a MVC-driven architecture has allowed to focus code development on specific features of the logbook, while profiting from the reliability of established third-party technologies, such as the Spring framework.

The implementation of a REST API provides fast access to the logbook features to clients and services beyond the provided web front-end.

References

1. Spring framework: http://www.springsource.org/
2. Spring Source Tool Suite: http://www.springsource.org/sts
3. DataTables: http://www.datatables.net