

Empire-dbProposal

Empire-db Proposal

This proposal is an application for the Empire-db Open Source component to become a new top-level project within the Apache Software Foundation. It is our desire to open Empire-db to a wider audience in order to build up a larger and more diverse community with a high level of collaboration.

Rationale

Empire-db is a lightweight relational database access layer that is significantly different from other ORM or Data Mapping solutions. At its core, Empire-db uses a Java Object Model to describe the physical data model rather than XML or annotations which leads to the following benefits:

*An intuitive command API allows dynamic generation of select, update, insert or delete SQL-statements of any complexity exactly as desired. The SQL statements are created purely by using Java methods which reference the model objects (like columns, tables, views, etc.). This provides type-safety and entirely eliminates the use of string literals for names or expressions in code. Additionally DBMS independence is achieved through a pluggable driver model.

*The object model also provides safe and easy access to meta-information of the data model such as field data type, maximum field length, whether a field is mandatory and a finite choice of options for a field's values. Metadata is user-extensible and not limited to DBMS related metadata. Availability of meta-information encourages more generic code and eliminates redundancies throughout application layers.

*Using references to table and column objects significantly improves compile-time safety and thus reduces the amount of testing. As a positive side effect the IDE's code completion can be used to browse the data model, increases productivity and eliminates the need for other external tools or IDE-plugins. Empire-db is not a traditional OR-Mapper as it does not use traditional Beans / POJO's representing full database entities. Instead database entities are usually handled by dynamic "Record"-objects with support for identity management and optimistic locking. This makes it even possible to change the data model at runtime. Optionally data retrieval with POJO's is also supported. With its approach we believe that Empire-db would complement the other relational database access solutions at Apache and be an enrichment for the platform and all its users.

Criteria

Meritocracy:

We are committed to actively encourage talented members of the community to participate and contribute to the project in order to support an Apache style meritocracy.

Community:

Currently there is only a small community based around the core developers. But even though there are many other solutions in the area of relational database access including several Apache projects, we believe that benefits from the alternative approach taken by Empire-db provides the potential to attract a large community, which we are hoping to develop during incubation.

Core Developers:

The community currently consists of the following 4 developers:

*Rainer Döbele (ESTEAM Software)

*Matthew Bond (ESTEAM Software)

*Jörg Reiher (ESTEAM Software)

*Manuel Gämderinger (T-Systems)

Alignment:

Empire-db's metadata management features make it possible to achieve a high level of integration with existing presentation layer frameworks, which leads to reduced redundancies as well as a better separation of model and view compared to other approaches. With our Empire-Struts2-Extensions subproject we offer such an implementation for the Apache Struts2 web application framework project that acts as the glue between presentation and business / persistence layer. We would like to see further integration efforts for other presentation layers evolve in further subprojects.

Known Risks

Orphaned products:

All current developers are committed to continue their work hence there is little risk of the project being orphaned.

Inexperience with Open Source:

Empire-db has been Open Source from its start in 2001, but it has only been publicly available since January 2008. All committers have long experience in using Open Source projects, but none has served as a committer on other Apache projects. We do, however, not expect any difficulty in adapting the Apache development process and follow the meritocracy rules.

Homogenous developers:

All core developers have initially worked for the same employer, with one now working for a different employer at a different location. It is one of our primary goals to become a more heterogeneous community.

Reliance on Salaried Developers:

The development of Empire-db was fuelled in the past by the requirements of professionally developed applications. None of the developers were paid solely for developing, maintaining or monitoring the project and a lot of work has been done on a voluntary basis.

Relationships with Other Apache Products:

Empire-db itself uses some Commons projects and Log4j.

The Empire-Struts2-Extensions subproject offers a special high level integration with the Apache Struts2 web application framework that delivers compile-time-safety and metadata access to the presentation layer.

In comparison to other relational data access projects at Apache Empire-db is probably somewhere between the OR-Mappers (Cayenne, OpenJPA, Torque) and iBATIS. Unlike the OR-Mappers Empire-db is not hiding away data access SQL statements from the developer but allows the developer to control every aspect of an SQL statement and its execution. And unlike iBATIS the SQL statements are not provided as string literals but are generated from a command object for the selected target DBMS. This leads to particular benefits when statements need to be generated conditionally with varying column selection and / or row filter criteria.

A Excessive Fascination with the Apache Brand:

We believe that one of the biggest advantages of the ASF is the provision of variety and choice. With Empire-db there is a remarkably different solution for a very common task that we think fits in very well with other existing database access layer projects allowing developers to choose the best solution for their needs. Even though Empire-db could exist on its own, as an Apache project it will certainly benefit from increased visibility as well as the friendly competition with other projects such as Cayenne, OpenJPA and iBATIS.

Documentation

Comprehensive information and documentation can be found on <http://www.empire-db.org>

1. Scope of the project

The project consists of a core component which contains the data access layer.

It is planned to create further subprojects with integration code for different presentation layer frameworks in order to fully benefit from empire-db's metadata management features. Currently there is one such project for the Apache Struts2 web framework called Empire-Struts2-Extensions.

2. Initial Source from which the project is to be populated

Current Empire-db sources (release 2.0.2 of July 6, 2008) can be found here: <http://heanet.dl.sourceforge.net/sourceforge/empire-db/empire-db-2.0.2.zip>

Sources for the Empire-Struts2-Extensions (release 1.0.2 of July 6, 2008) can be found here: <http://heanet.dl.sourceforge.net/sourceforge/empire-db/empire-struts2-ext-1.0.2.zip>

3. Identify the ASF resources to be created

3.1 Mailing Lists

- *empire-db-dev
- *empire-db-user
- *empire-db-scm
- *empire-db-ppmc

3.2 SVN Repositories

- */incubator/empire-db

3.3 Issue Tracking

- *Need a new Jira project called empire-db.

4. Identify the initial set of committers

- *Rainer Döbele
- *Matthew Bond
- *Jörg Reiher
- *Manuel Gernerding
- *Henning Schmiedehausen
- *Thomas Fischer
- *Martijn Dashorst

5. Identify Apache sponsoring individual

We kindly request the Apache Incubator PMC to be the sponsor for this project.

Champion and Mentor

Henning Schmiedehausen (henning *at* apache.org)

Mentors

*Thomas Fischer (tfischer *at* apache.org)

*Martijn Dashorst (martijn.dashorst *at* gmail.com)

*Henning Schmiedehausen (henning *at* apache.org)