EL4J

FeaturesOfEl4j

Generated from Wiki page:
http://wiki.elca.ch/twiki/el4j/bin/view/EL4J/FeaturesOfEl4j

Version	Date	Author(s)	Visa
1.13	04 Apr 2008 - 09:04	Main.PhilippHOser, Main.StefanWismer, StefanWismer	

©ELCA Informatique SA, Switzerland 2009

Table of Contents

1 Unique Features of EL4J.....1

1 Unique Features of EL4J

This document lists the distinctive features of EL4J. A frequent question about EL4J is what it provides additionally to the frameworks it includes. One benefit of EL4J is certainly the selection, integration, and pre-configuration of leading components. More benefit comes from the *new* features that EL4J provides.

The following list shows the distinctive features of EL4J (this list is not exhaustive, please check also the module documentation and the javadoc):

- Application templates to get quickly started: for GUIs and Web UIs. The goal is to have a running sample application within 10 minutes! In this running application you have a proven structure and sample solutions for typical development issues.
- Support for modules with code, default configuration and dependencies. This feature is based on the build system (Maven 2), the basic spring abstractions, some EL4J support and conventions.
    - ♦ More flexible and robust loading of configuration resources
        - ◊ Inclusion and exclusion list to include/ exclude configuration files
        - ◊ Store the list of configuration resources to load in the jar-file manifest
        - ◊ Merging of spring configuration: adding more parameters to an existing list of parameters
    - ♦ Each EL4J module packages functionality with samples, documentation and default configuration.
- Improved remoting
    - ♦ Easier switching between remoting protocols (unification of remoting protocols)
    - ♦ Remote POJOs via SOAP (simpler than with basic Spring), support for JAXB
    - ♦ Auto-generation of RMI-wrappers for POJOs (via Interface Enrichment)
    - ♦ Provide light load-balancing via the more flexible remoting layer
    - ♦ Implicit context passing over process boundaries
    - ♦ Automatically deploy POJOs as EJB 2.1 beans (currently frozen)
- EL4J cockpit
    - ♦ Auto-publication of the list of spring beans with their configuration values, interceptors and other useful info
    - ♦ Get a simple overview of the running threads
    - ♦ Change the log4j configuration dynamically
- Exception handling
    - ♦ Exception handling guidelines
    - ♦ Safety facade
    - ♦ More exception mappings for database accesses (additionally: duplicate values, out of bound values)
- Convenient Maven 2.0 setup
    - ♦ Well thought-through use of Maven. Hierarchical split of configurations. Use of fine-grained projects.
    - ♦ Bugfixes for maven and related tools (we have submitted about 20 patches, some of which are already included in maven)
    - ♦ Own plugin to extend maven: copy tool for combined report generation.
    - ♦ Presentation about how to migrate to mvn and many detailed information and hints
    - ♦ Maven cheat sheet
- GUI: Light Swing framework featuring: Binding of POJOs to Swing components, Event Bus, Docking and MDI support, Exception handling, i18n and resource management, user preference management, simple way to define Actions and selectively enable them, convenience code to simplify the design of forms, ...
- JSF framework: an integration based on Seam, Facelets, Ajax4Jsf, and Richfaces. It does not require EJB3 (is is based on Spring).
- Daemon manager
- License manager
- XML Merger
- Extended file support (fast file observation, directory size information, easier file search capabilities)

- Generic DAO implementation (reduce coding, improve homogenization)
- Easier support for annotation to interceptor mappings (no coding required for basic cases)
- Ajax demo
- TCP forwarder to automatically test TCP connection failures
- Tracking the invocation graph (potentially over process boundaries), measuring performance and generating a sequence diagram for it
- Auto-idempotency interceptor (makes your service calls idempotent)
- Better documentation
    - Architecture discussions
    - EL4J Datasheet
    - Annotation cheat sheets
    - FAQ & infos on how to solve common problems
    - Documentation of each feature
    - Tracing stack document: hints on how to get more information from the layers of your application

The following external components are integrated in EL4J (this list is not exhaustive, please check also the list of included jar-files):

- Spring 2.5.1 framework
- Maven 2.0, JUnit
- Commons logging, log4j
- Hibernate
- Ibatis
- Acegi security framework
- Swing application framework (from Sun)
- JWebUnit and HtmlUnit
- Eclipse BIRT
- CGLib
- XFire
- Axis
- Caucho remoting: Hessian & Burlap
- Seam
- JSF
- Struts
- JaMon
- Quartz