



JSP-Develop Newsletter

Ausgabe 4

01.07.2004

Impressum

In diesem Newsletter widmen wir uns Jakarta Projekten. Es hat sich im Open Source Bereich viel getan und es gibt eine Reihe von Komponenten und Frameworks, die das Leben eines Entwicklers einfacher gestalten können. Dem Nutzen eines Open Source Frameworks steht natürlich immer noch der Einarbeitungsaufwand gegenüber.

Open Source ist, obwohl sich auch dort viel getan hat, teilweise noch nicht sehr gut dokumentiert und die Einarbeitung kann sich deshalb langwierig gestalten. Auf der anderen Seite haben verschiedene Autoren diese Nische erkannt und bieten Bücher über die Anwendung von Open Source Frameworks und Komponenten an. Wir sind der Meinung, dass es sich unter Zuhilfenahme der entsprechenden Literatur sowie Foren auf jeden Fall lohnt Open Source Luft zu schnuppern und freuen uns auf weitere Projekte in diesem Bereich.

Marc & Jörg

News

LogOn Informationstage

LogOn hat im Juni 2004 eine Reihe von Informationstagen zum Thema SAP und XML/Web Services veranstaltet. Die Konferenz in Frankfurt findet am 22. Juni 2004 statt.

<http://www.ltt.de/sap.xml-days.2004/frankfurt/conference>

Java Forum in Stuttgart

Am 1. Juli 2004 findet in Stuttgart im Millennium Hotel das 7. Java Forum statt. Die Konferenz umfasst 35 Vorträge mit den Themen Technologie, Projekterfahrungsberichte sowie Produktpäsentationen.

<http://www.java-forum-stuttgart.de>

Jakarta Commons Logging

Das Jakarta Commons-Team hat das Logging Release 1.0.4 publiziert. Der Commons Logging Framework fungiert als Bindeglied zwischen den unterschiedlichen Java Logging Frameworks und abstrahiert deren Schnittstellen in einem gemeinsamen Logging Framework.

<http://jakarta.apache.org/commons/logging>



© bucur

In dieser Ausgabe:

News	1
Tech-Tipp	1
Buchbesprechung	2
FAQ	3

Tech-Tipp:

BeanUtils ist eine Komponente des Jakarta Commons Packages, welche sich sehr gut für die Verarbeitung von Java Beans / Value Objects eignet. BeanUtils fungieren dabei als Introspection Utility und beinhalten Methoden zum Setzen/Lesen der Attribute eines Java Beans.

Das BeanUtils Package enthält auch Funktionen für dynamische Beans, die beispielsweise in Struts 1.1 verwendet werden (dynamische ActionForm Klassen). Durch die Verwendung von dynamischen Beans wird die Anzahl der Java Klassen einer Anwendung verringert und es kann durch weniger Programmierarbeit auch Entwicklungszeit eingespart werden.

jakarta.apache.org/commons/

Buchbesprechung

Den **Jakarta Commons** begegnet man in Foren, der Literatur und auch in Java Projekten. Was sind die Commons und wofür setzt man diese ein?

Diese und weitere Fragen werden wir in diesem Textabschnitt klären und dazu gleich das entsprechende Buch vorstellen.

Commons sind Komponenten die unabhängig von einem Framework eingesetzt werden können. Aus Sicht des Entwicklers sind die Commons: Helper bzw. Tool Klassen, die sowohl server- als auch client-seitig eingesetzt werden können.

Commons sind in die Kategorien Commons Proper (Komponenten die bereits produktionsreif sind) und Sandbox (noch nicht produktionsreife Komponenten) eingeteilt.

Der folgende Auszug spiegelt nur einen Teil der Mächtigkeit dieser Komponentensammlung wieder.

Lang:

Die Commons Lang Komponenten können als Erweiterung der Core Java API angesehen werden. Lang beinhaltet ein reichhaltiges Set von String-, Object-, System- und Klassen-Komponenten sowie Utilities für die Verarbeitung von Zeit- und Datumswerten, mathematische Funktionen, Exception Handler und sogar einen Java Enumerationstyp.

Logging:

Log4J, JDK 1.4 Logging und Avalon Logkit, sind nur ein kleiner Auszug von Log-Komponenten. Welchen Framework verwendet man am besten und wie kann man den zunächst gewählten Framework schnell und ohne

große Source Code Anpassungen austauschen?

Genau diese Fragen klärt und löst der Commons Logging Framework. Commons Logging abstrahiert die gängigen Logging Frameworks und fungiert dadurch als übergeordneter Framework, der mit den gängigen Logging-Frameworks kombiniert werden kann.

BeanUtils:

Komponenten für die Verarbeitung von JavaBeans und Value Objects. Die BeanUtils sind insbesondere für Web-Entwickler und im Struts Kontext nützlich.

CLI:

Komponenten zum parsen von Kommandozeilenparametern. CLI ist besonders für Swing-Programmierer interessant, die Java Standalone Applikationen entwickeln wollen.

Collections:

Die Commons Collections sind eine sehr gute Ergänzung zu dem Java Collection Framework. Commons Collections beinhalten neben Utility-Klassen auch weiterführende Listen, Map, Buffer und eine Bag Umsetzung.

Validator:

Struts Entwicklern ist der Commons Validator bestens bekannt, weil dieser Teil der Struts 1.1 Distribution ist.

Der Validator Framework kann allerdings auch unabhängig von Web-Anwendungen zum Beispiel in Java Swing-Clients eingesetzt werden.

Pools:

Die Commons Pooling Komponenten sind insbesondere für clientseitige Java Entwickler interessant, die Ressourcen in Form von Pools verwalten möchten. Commons bietet verschiedene Pooltypen an, die je nach Anwendungskontext entsprechend eingesetzt werden können.

Digester:

Commons Digester ist ein Framework für die Verarbeitung von XML-Dokumenten. Digester ist relativ einfach anzuwenden und entfaltet sich besonders gut im Kontext der Verarbeitung von Value Objects und Java Beans.

HttpClient:

Der HttpClient ist eine Programmierschnittstelle (API) um über das HTTP Protokoll kommunizieren zu können.

FileUpload:

Dem Commons FileUpload ist man im JSP-Develop Forum schon über den Weg gelaufen. Die FileUpload Komponenten ermöglichen es auf einfache Weise Dateien auf einen Server zu laden um diese zentral zu verwalten.

Die in dieser Buchbesprechung vorgestellten Commons Komponenten (mit Ausnahme von CLI) werden in dem Buch „Pro Jakarta Commons“, welches vom Apress Verlag aufgelegt worden ist, besprochen. Darüber hinaus behandelt das Buch auch noch weitere nicht gelistete Komponenten aus der Commons Reihe, die den Leser erfreuen dürfen.



Fazit

Das englischsprachige Buch ist sehr verständlich geschrieben. Die reichhaltigen Java Source Code Beispiele untermauern die Anwendung der Commons, so dass erfahrene Java-Entwickler zügig Commons Funktionalitäten in der Programmierung nutzen können.

Die Commons können dabei die Entwicklungszeit von Software-Produkten reduzieren und auch die Robustheit der Software fördern.

In diesem Sinne ist das Buch zu empfehlen, weil es nützliche Tipps und Tricks rund um die Jakarta Commons vermittelt und eine sehr gute Ergänzung zu der teilweise doch recht dünnen Commons Dokumentation ist.

Jakarta Commons Webseite:

Jakarta.apache.org/commons/

FAQ - Was überhaupt ist Jakarta Avalon?!

Im Umfeld von Avalon liegt es nahe von dem - **Nebel von Avalon** - zu sprechen. Man hört von Avalon und man sieht in Projekten auch partiell die Anwendung des Frameworks bzw. es werden projektspezifisch entsprechende Skills gefordert. Dabei wird der Framework serverseitig eingesetzt, um Konfigurationsdateien zu verwalten und um Services im Applikationsserver-Kontext zu lokalisieren und zu realisieren. Es gibt kaum Dokumentation zu dem Framework, welche durchgängig die Anwendung der Software unterstreicht, sodass man als serverseitiger Entwickler gezwungen ist, sich das Wissen über die Anwendung von Avalon selbst zu erarbeiten.

Lohnt sich die Mühe der Einarbeitung und wie wird Avalon klassifiziert? Wir sind der Meinung, dass sich die Mühe der Einarbeitung lohnen kann, um Avalon in der Folge als Framework einzusetzen.

Besonders im EJB-Umfeld, dort wo Konfigurationen administrierbar hinterlegt werden, um beispielsweise Kontextinformationen (JNDI-Parameter, Applikationsparameter, etc.) aus dem Source Code herauszufiltern, kann sich Avalon gut entfalten.

Neben dem Verwalten von Konfigurationsdateien im XML-Format ist Avalon auch eine gute Ergänzung zur Entwicklung eines „Service Locators“, welcher das Auffinden und Ansprechen von Komponenten im Applikationsserver unterstützt. EJB-Anwendungen leben nun mal von dem Zugriff auf Ressourcen (z.B.: JCA-Adaptoren, Datenbankverbindungen, etc.) und den Komponenten, die im Applikationsserver verwaltet und zur Laufzeit angesprochen werden müssen.

Um die zweite Frage zu beantworten (wie wird Avalon klassifiziert): Im Rahmen von Avalon spricht man von einem Komponenten Framework, welcher die Entwicklung von serverseitigen Komponenten durch Java Interfaces und API-Methoden unterstützt.

Weitere Anregungen zu diesem Thema findet man dort: jakarta.apache.org/avalon/

Anregungen, Kritik oder Vorschläge an: webmaster@jsp-develop.de.

Wenn Sie unseren Newsletter nicht mehr erhalten möchten, können Sie ihn unter www.jsp-develop.de/newsletter/ abbestellen.

Newsletter (c) 2004 by www.jsp-develop.de Alle Rechte vorbehalten.

Java, JSP, JavaServer Pages, J2EE, EJB, JDBC, JNDI, JTA, Sun, Sun Microsystems are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and in other countries. Other trademarks and registered trademarks are the property of their respective owners.