

TomC@ - Die Kolumne: Aktuelle Ausgabe

by Peter Rossbach

1. Thema der aktuellen Ausgabe

1.1. 06.2005: Update - Apache 2 und Tomcat mit mod_jk 1.2.10 verbinden: Ein alter Baum in neuer Blüte...

Autor: Peter Roßbach



Die Wurzeln des Projekts Tomcat sind eng mit dem Apache Server HTTPD verbunden und mit der Zeit ordentlich gewachsen. Die Bande mit dem Apache-Server haben bereits 1997 mit dem legendären Projekt JServ begonnen. Heute ist der Tomcat ein vollständiger und in der Praxis erprobter eigenständiger Webcontainer, kann aber eben an andere Webserver angebunden werden. Das dafür häufig eingesetzte Modul mod_jk ist Ende letzten Jahres einer kräftigen Verjüngungskur unterzogen worden. In dieser Kolumne konzentrieren wir uns auf das aktuelle mod_jk-Version 1.2.10 und die Anbindung an den Apache 2.

[Beispiel aus dem Kolumnenbeitrag 06.2005 zum Thema: Update - Apache 2 und Tomcat mit mod_jk 1.2.10 verbinden](#)

mod_jk 1.2.10 Dokumentation

2. Thema des nächsten Monats

2.1. 07.2005: Flexible Apache Tomcat Ausgaben mit dem Java Logging

API (JSR47)

Autor: Peter Roßbach

Das richtige Logging-API zu nutzen, gehört zu einer fast schon religiösen Entscheidung im großen Java-Land. Jeder hat dazu eine andere Meinung und viele meinen es doch lieber gleich selber implementieren zu können. Die Anzahl der verschiedenen Logging-Frameworks, die in den letzten Jahren entstanden sind, ist schon erschreckend. Meist enden diese #Innovationen# aktuell mit einer Portierung auf das beliebte Log4j oder Logging-API aus dem JDK. Wer sich nicht entscheiden kann oder darf, nutzt meist das Jakarta Commons-API für die Kapselung. Der Apache Tomcat Webcontainer bildet da keine Ausnahme, aber erst der aktuelle Tomcat 5.5 setzt diese Entscheidung konsequent um. Im Tomcat 5.5.9 wird nun endlich auch das Java Logging-API für Webanwendungen wirkungsvoll unterstützt.

[Beispiel aus dem Kolumnenbeitrag 07.2005 zum Thema: Der Apache Tomcat Sommer beginnt mit JULI: Das Standard Java Logging API für Ihre Protokollausgaben verwenden.](#)

3. Thema des letzten Monats

3.1. 05.2005: JDBC-Zugriffe mit verteilten Transaktionen steuern: Der Steuermann

Autoren: Peter Roßbach und Frank Wegmann

Definition und Verwaltung von JNDI-Ressourcen sind ein unverzichtbarer Bestandteil moderner Anwendungen. Der Webcontainer stellt die konkreten Ressourcen den Anwendungen mit Namen zur Verfügung. Vorbei die Zeit, als Anwendungen noch ihre eigenen Datenbanktreiber und JDBC Connection Pools implementiert und verwendet haben. Allzu oft erlebt man in der Realität dann aber, dass in einem J2EE-Container nur in eingeschränktem Umfang und zudem fehlerbehaftete Ressourcenfabriken realisiert sind. In dieser Kolumne beschreiben wir, wie man eigene JNDI-Ressourcen im Tomcat definiert. Als Beispiel haben wir die Integration des verteilten Transaktionsmanagers JOTM aus dem ObjectWeb- Projekt ausgewählt und erprobt.

[Beispiel aus dem Kolumnenbeitrag 05.2005 zum Thema: JDBC-Zugriffe mit verteilten Transaktionen steuern](#)