



## J2ME und J2EE

Integration von J2ME Clients in J2EE basierte Serveranwendungen

) Schulung )

### AUTOR



**Steffen Schluff**  
Orientation in Objects GmbH

) Beratung )

Veröffentlicht am: 15.6.2002

### ABSTRACT

) Entwicklung )

Java hat sich in den letzten Jahren als eine der bedeutendsten Technologien für serverseitige Anwendungen etabliert. Basierend auf der Java 2 Plattform, Enterprise Edition (J2EE) sind mittlerweile in vielen Firmen und Institutionen Intranet-Anwendungen, wie zum Beispiel Arbeitszeiterfassung oder Raumbuchungssysteme, zu finden. Oftmals sind solche Anwendungen durch die Verwendung von Java Servlets und JavaServer Pages JSP auf Webbrowser als Clients ausgelegt, welche sich durch generelle Verfügbarkeit und einfache Nutzbarkeit auszeichnen.

Der vorliegende Artikel soll anhand einer Fallstudie zeigen, wie darüberhinaus die ebenfalls weit verbreiteten Personal Digital Assitents (PDA) als auch Mobiltelefone auf einfache Weise in eine bereits vorhandene, Java basierte Serverlandschaft integriert werden können. Durch die Java 2 Plattform, Micro Edition (J2ME) können diese Arten von Geräten ohne großen Mehraufwand und ohne Nutzung proprietärer Technologien zur idealen Ergänzung bestehender Anwendungen werden.

) Artikel )

#### Orientation in Objects GmbH

Weinheimer Str. 68  
D-68309 Mannheim

Tel. +49 (0) 6 21 - 7 18 39 - 0  
Fax +49 (0) 6 21 - 7 18 39 - 50

www.oio.de info@oio.de

## MOTIVATION

---

Zu internen Verwaltungs- und Abrechnungszwecken existiert bereits eine J2EE basierte Zeiterfassungslösung, welche sich jedoch bei Außendienstesätzen wie Schulung oder Beratung oftmals aufgrund externer Gegebenheiten nicht nutzen läßt. Das umständliche und fehlerträchtige Notieren der Arbeitszeiten auf Notizzetteln soll durch eine einfache Anwendung ersetzt werden, die auf den Palm Pilot PDAs der Mitarbeiter lauffähig ist. Diese Anwendung soll im ersten Schritt die Zeiterfassung im Stile einer Stechuhr ermöglichen und weiterhin die Möglichkeit bieten, die erfassten Zeiten in die Standardzeiterfassung zu importieren, wo sie bei Bedarf noch einmal bearbeitet werden können.

## VERWENDETE TECHNOLOGIEN

---

Nachfolgend werden kurz die einzelnen Technologien beschrieben, die während der Machbarkeitsstudie zum Einsatz kamen.

### J2ME WIRELESS TOOLKIT

---

Das bei Sun Microsystems erhältliche J2ME Wireless Toolkit ermöglicht die Entwicklung von Java Applikationen, die auf jedem Gerät lauffähig sind, welches das Mobile Information Device Profile (MIDP) implementiert. Bei MIDP handelt es sich um einen von Sun Microsystems entwickelten Standard für Java Laufzeitumgebungen, der auf Kleingeräte wie etwa Mobiltelefone und PDAs ausgerichtet ist. Sun bietet ebenfalls eine lauffähige Implementierung dieses Standards für Palm OS Geräte an, so daß die entwickelten Anwendungen auch sofort auf den Zielgeräten verwendet werden können. Ergänzend wurde zur Vereinfachung der Entwicklung der Palm OS Emulator (POSE) verwendet, der das Emulieren eines Palm OS Gerätes auf einem herkömmlichen PC ermöglicht.

### JAKARTA TOMCAT

---

Tomcat ist ein Open-Source Servletcontainer des Jakarta Projekts der Apache Software Foundation und wird in der offiziellen Referenzimplementierung für Servlet und JSP Technologien verwendet.

### APACHE ANT

---

Ant ist ein auf Java basiertes Build Tool für Java Softwareprojekte und läßt sich entfernt mit Make vergleichen, ist jedoch vor allem durch seine XML Builddateien ungleich einfacher in der Handhabung. Darüberhinaus läßt sich Ant in fast alle populäre integrierten Entwicklungsumgebungen integrieren und ist bei Bedarf beliebig um eigene Tasks erweiterbar.

## REALISIERUNG

---

### CLIENT

---

Da die Anforderungen an den Client in der ersten Stufe wie zuvor beschrieben vergleichsweise gering waren, wurden auf dem Handheldgerät nur die folgenden vier Klassen benötigt:

- [ClockingEntry.java](#)

Diese Klasse repräsentiert Stechuhreinträge, wobei die jeweiligen Objekte unveränderbar (immutable) sind. Neben Factory und 'getter' Methoden besitzt die Klasse ebenfalls zwei Methoden zur einfacheren Anbindung an die J2ME MIDP Persistenzmechanismen. Diese Methoden erlauben es, Objekte in Byte-Arrays umzuwandeln und bei Bedarf wieder aus entsprechenden Byte-Arrays Objekte zu erzeugen.

- [ClockingEntryConnectionHelper.java](#)

Hierbei handelt es sich um eine Hilfsklasse, die die komplette Kommunikation basierend auf HTTP mit der Server Komponente der Zeiterfassung übernimmt. Da sowohl auf Client- als auch auf Serverseite die Java Plattform verwendet wird, wird hier auf ein textbasiertes Austauschformat wie XML verzichtet. Stattdessen wird direkt mit einem Java `DataOutputStream` gearbeitet, wodurch sich der Client kleiner und performanter realisieren läßt. Insbesondere bei Handheldgeräten ist dies naturgemäß von großer Bedeutung.

- [ClockingEntryRecordStore.java](#)

Eine weitere Hilfsklasse, die die Speicherung der erzeugten Stechuhreinträge regelt und hierfür intern einen sogenannten `RecordStore` verwaltet. Hierbei handelt es sich um einen auf ByteArrays basierenden Mechanismus zur persistenten Datenhaltung, der von MIDP zur Verfügung gestellt wird.

- [MidpClient.java](#)

Bei dieser Klasse handelt es sich um ein sogenanntes `MIDlet`, das heißt um die eigentliche MIDP Anwendung. Die Klasse stellt Methoden zur Verfügung, die der Laufzeitumgebung auf dem Handheldgerät das Erzeugen, das Starten, das Unterbrechen und das Beenden der Anwendung ermöglichen. Zusätzlich sind in dieser Klasse die Komponenten der Benutzerschnittstelle sowie die dazugehörige Ereignisverarbeitung realisiert.

## SERVER

---

Für die Anbindung auf der Serverseite wurde nur die folgende Servletklasse benötigt:

- [MidpServlet.java](#)

Dieses Servlet stellt das Gegenstück zu der zuvor erwähnten `ClockingEntryConnectionHelper` Klasse des Clients dar. Auf eine weiterführende Verarbeitung wurde hier verzichtet, die Ausgaben auf der Konsole sollen lediglich demonstrieren, daß alle vom Handheldgerät gesendeten Daten vollständig und korrekt empfangen wurden. Bei Bedarf könnte hier wieder die `ClockingEntry` Klasse des Clients verwendet, oder aber auch die Daten direkt an Businesslogikobjekte des Servers übergeben werden.

## DEPLOYMENT

---

Das tatsächliche Deployment auf den Palm OS basierten Endgeräten, die mit der von Sun Microsystems verfügbaren Palm OS MIDP Implementierung ausgestattet waren, verlief problemlos. Lediglich für die HTTP Verbindung der Palms zu dem zugehörigen Servlet wurde separate Software verwendet. Diese ermöglicht es, dem Handheldgerät mittels seiner Docking-Station, über die serielle Schnittstelle des PCs eine TCP/IP Verbindung zu öffnen, was die Verwendung eines Handys oder Modems überflüssig macht.

# SCREENSHOTS

Die folgenden Screenshots sollen zur besseren Veranschaulichung dienen und außerdem demonstrieren, wie unterschiedliche Endgeräte die MIDP Benutzerschnittstellen realisieren. Es handelt sich hierbei einerseits um Screenshots des sogenannten "DefaultColorPhone" des J2ME Wireless Toolkit und andererseits um Aufnahmen des Palm OS Emulators (POSE).



Abbildung 1: Java Startbildschirm



Abbildung 2: Hauptbildschirm der Stechuhrenanwendung



Abbildung 3: Geräteabhängiger Menüaufbau



Abbildung 4: Anzeige nach erfolgter Datenübertragung

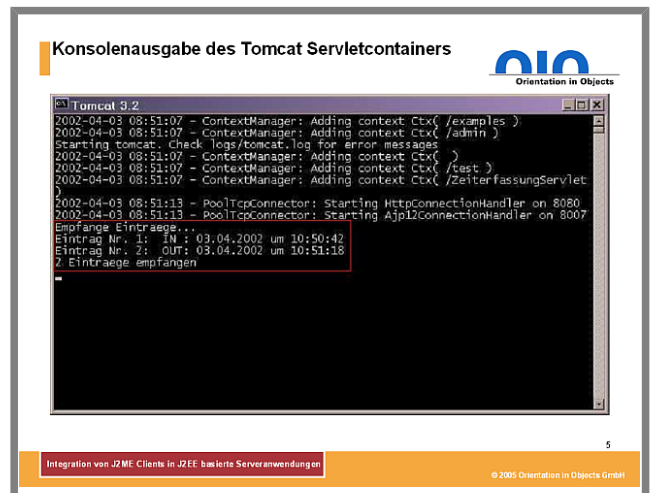


Abbildung 5: Konsolenausgabe des Tomcat Servletcontainers

## FAZIT

Zusammenfassend lässt sich sagen, daß sich mittels J2ME MIDP binnen sehr kurzer Zeit vielseitige kleine Anwendungen realisieren lassen, die problemlos an serverseitige Lösungen anbindbar sind, insbesondere wenn diese durch die J2EE ebenfalls in Java realisiert sind. Der hohe Verbreitungsgrad der Palm OS basierten Handheldgeräte und eine verfügbare lauffähige MIDP Implementierung können dafür sorgen, daß die erstellten Anwendungen auch sofort praktisch genutzt werden können. Da von vielen Handyherstellern bereits eine MIDP Laufzeitumgebung auf ihren Geräten realisiert oder zumindest angekündigt wurde, ist zudem sichergestellt, daß die entsprechenden Applikationen im Gegensatz zu proprietären Lösungen auch in Zukunft nutzbar sind.

## REFERENZEN

---

- Java(TM) 2 Platform, Enterprise Edition  
<http://java.sun.com/j2ee>
- Java(TM) Servlet Technology  
<http://java.sun.com/products/servlet>
- Java(TM) 2 Platform, Micro Edition  
<http://java.sun.com/j2me>
- Java(TM) 2 Platform Micro Edition, Wireless Toolkit  
<http://java.sun.com/products/sjwtoolkit/>
- Mobile Information Device Profile (MIDP)  
<http://java.sun.com/products/midp>
- MIDP for Palm OS  
<http://java.sun.com/products/midp/>
- Palm OS: Palm OS® Emulator  
<http://www.palmos.com/dev/tools/emulator>
- The Jakarta Site - Jakarta Tomcat  
<http://jakarta.apache.org/tomcat>
- Apache Ant - Ant  
<http://jakarta.apache.org/ant>
- MochaSoft, Home of Mocha Telnet and Mocha TN3270 / TN5250 / LPD  
<http://www.mochasoft.dk>