

Oracle Servlet Engine OSE

Karl Seematter, Akadia AG

Einführung

Für Internetapplikationen mit dynamischen Inhalten werden immer häufiger J2EE Komponenten (Servlets, JSP's, EJB) eingesetzt. Bei bisherigen auf Servlet basierenden Applikationen werden die Servlets im Filesystem abgelegt und von einem WEB-Server mit einer Servlet Engine (z.Bsp. Apache/Jserv) ausgeführt. Datenbankzugriffe erfolgen z.Bsp. mit JDBC oder SQLJ. Oracle8i Release 3 (8.1.7) bringt im Bereich serverside Java Programmierung einige Erweiterungen. In diesem Artikel wird die Konfiguration einer Webapplikation betrachtet bei der:

- Java Servlets **mit** DB-Zugriff in die DB geladen und von der Oracle Servlet Engine ausgeführt werden.
- Java Servlets **ohne** DB-Zugriff im Filesystem abgelegt und vom Oracle HTTP Server (Apache/Jserv) ausgeführt werden.

Referenzen

#	Bezeichnung	Quelle
[1]	Oracle 8i Oracle Servlet Engine User's Guide	http://technet.oracle.com/docs/products/oracle8i/doc_index.htm
[2]	Oracle 8i Java Developers Guide	
[3]	Oracle Java Tools Reference	
[4]	Java Naming and Directory Interface JNDI	http://java.sun.com/products/jndi/
[5]	Oracle 8i Database Access with Apache / Jserv	http://www.akadia.com

Komponenten

Alle benötigten Komponenten (Oracle HTTP Server von Apache, JRE, JDK, JSDK) sind auf der Installations-CD Oracle 8i Release 3 enthalten.

Begriffe

Oracle Servlet Engine OSE

Siehe [1]

OSE ist ein in Oracle8i integrierter Webserver und Servlet-Runner mit welchem Java Server Pages, Servlets und Java Stored Procedures in der Datenbank ausgeführt werden können.

Java Naming and Directory Interface JNDI

Siehe [3] [4]

JNDI definiert ein hierarchisches Directorymodell zur Verwaltung von Objekten (Servlets, Initialparameter, JSP's etc.) einer Webapplikation.

Der Zugriff (kreieren, ändern, löschen, navigieren und anzeigen) auf JNDI Strukturen und Objekte erfolgt über eine Shell (Session Shell Tool „sess_sh“) mit einer Unix-ähnlichen Kommandosprache. Analog einem Unix Filesystem unterstützt JNDI drei verschiedene Zugriffsberechtigungen: READ, WRITE, EXECUTE.

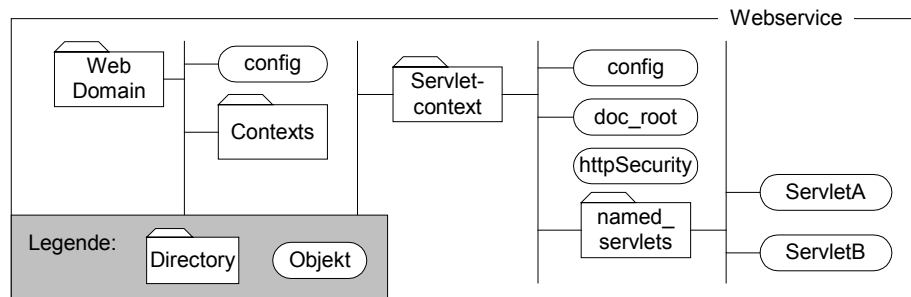


Fig 1 Beispiel JNDI Struktur für einen Single-Domain Webservice

Ein **Webservice** definiert ein Rootdirectory im JNDI mit der darunterliegenden - teilweise vorgegebenen - hierarchischen Directorystruktur und den darin gespeicherten Objekten für eine oder mehrere **Webdomains**.

OSE unterstützt zwei verschiedene Webservicekonfigurationen: Single-Domain und Multi-Domain.

Beispiel (Kommando im Session Shell Tool):

```
$ createwebservice -root /SougDemoRoot SougDemo
                        |                               |
                        |                               |→Name des Web Service
                        |→Name des Rootdirectories
```

In einem Single-Domain Webservice entspricht die **Webdomain** dem Rootdirectory des Webservice.

Im **Servletcontext** (eine Art Teilapplikation) sind zusammengehörende Servlets, Konfigurationsparameter und Pointer auf statische Webinhalte im Filesystem zusammengefaßt.

Die hierarchische Darstellung ist nur eine Präsentation. Effektiv werden die Directories und Objekte in Datenbanktabellen abgelegt.

Endpoint

Ein Endpoint entspricht einem dynamischen Port im Listener für einen Webservice. Der Listener nimmt Request's von einem HTTP-Client oder externen HTTP-Server entgegen und leitet diese an den Webservice in der OSE weiter.

Beispiel:

```
$ addendpoint -port 7778 -register SougDemo SougEndpt01
                        |                               |
                        |                               |→Name des Endpoint
                        |→Webservice
```

MOD_OSE

Das Modul MOD_OSE wird zum Oracle HTTP Server (Apache) konfiguriert (mit der Directive LoadModule“). MOD_OSE stellt die Verbindung vom Apache Server via Net8 zur OSE her.

Zugriff auf OSE Servlets

Oracle8i OSE ermöglicht zwei verschiedene Zugriffsarten:

1.) Direkter Zugriff von einem HTTP-Client auf die OSE

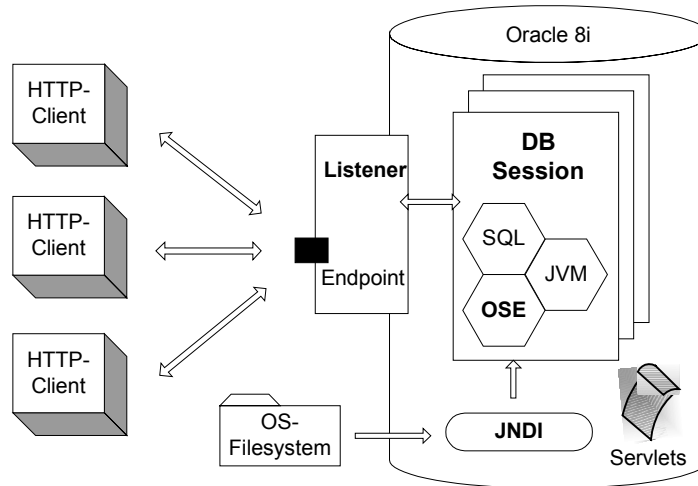


Fig2 Direkter Client Zugriff

Ablauf

Der HTTP-Client sendet mittels einer URL in der Form:

`http://<host>:<port>/<webservice>/<servlet>`

einen Request an einen Listenerport (Endpoint). Dieser leitet den Request an die OSE weiter. Die OSE liest die notwendigen Informationen aus dem JNDI und instanziert das Servlet.

Für eine URL in der Form:

`http://<host>:<port>/<webservice>/<html_page>`

wird die HTML-Page aus dem definierten OS-Directory gelesen.

Database Session

Ein Zugriff von einem HTTP-Client öffnet eine neue DB Session. Die Session wird im Kontext des Webservice-Eigentümers gestartet. Jede DB-Session startet eine eigene Java Virtual Maschine (JVM). Die JVM läuft im selben Adressraum wie die SQL- und PLSQL-Engine und ermöglicht somit einen performanten Zugriff auf DB-Objekte ohne zusätzliches Login.

Das Beenden einer DB-Session wird mit einem Timeoutparameter gesteuert.

2.) Zugriff über den Oracle HTTP-Server (Apache)

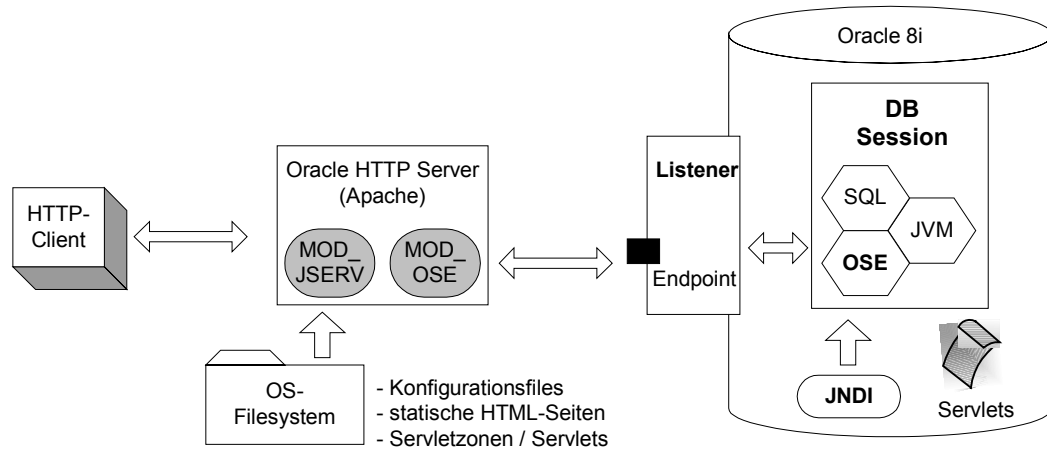


Fig3 Indirekter Zugriff auf die OSE über den Oracle HTTP Server

In dieser Konfiguration übernimmt die OSE die Funktion einer zusätzlichen Servlet Engine für den Oracle HTTP-Server. Servlets **ohne DB-Zugriffe** können im einem Repository bzw. einer Servletzone des Apache Servers abgelegt werden (siehe [5]), solche **mit DB-Zugriffen** werden in der DB installiert. Diese Architektur ermöglicht eine flexible Skalierung für Webapplikationen.

Ablauf

Der Oracle HTTP-Server analysiert den Client-URL:

- Ist darin eine statische Seite aus dem mit der Direktive DOC_ROOT definierten Verzeichnis enthalten, wird diese Seite zurück gegeben.
- Ist darin eine Servletzone enthalten, wird der Request an MOD_JSERV weitergeleitet.
- Ist darin ein Servletkontext eines OSE-Handlers enthalten, wird „http://<host>:<port>“ vom URL entfernt und an MOD_OSE weitergeleitet.

Hinweis: Ein Servletkontext „SougDemo“ wird mit folgenden Direktiven definiert:
(Konfigurationsfile mod_ose.conf)
`<Location /SougDemo/* >`
`SetHandler aurora-server`
`</Location>`

„Stateful“ und „stateless“ Servlets in der OSE

In einem Servletkontext zusammengefasste Servlets (Applikation) können ‚stateful‘ oder ‚stateless‘ konfiguriert werden. Eine ‚stateful‘ Applikation unterhält Zustandsinformationen über alle Request-Response Sequenzen einer HTTP-Session. Cookies oder URL-rewrite sind die Mechanismen dazu. Details dazu in [1].

Fazit

Mit der OSE ergeben sich für 3-tier Applikationen neue Möglichkeiten der Skalierung. Java Servlets können je nach Aufgabe im middle-tier oder im DB-Server installiert werden.

Die Verwaltung eines Webservice im JNDI ist eher umständlich und aufwändig. Ein GUI-Tool oder zumindest ein Browser würde die Konfigurationsarbeit erleichtern. Das Session Shell Tool „sess_sh“ zur Verwaltung des JNDI hat noch Mängel (Version 1.0 !). Z. Bsp. funktioniert das Kommando „rm- r“ nicht korrekt, d.h. die Directoryhierarchie muss in einzelnen Schritten gelöscht werden.

Beispielkonfiguration

Auf unserer Homepage www.akadia.com unter der Rubrik „Current Publications“ finden Sie Beispiele mit den verschiedenen Zugriffsarten und eine Step by Step Anleitung für die Konfiguration der Oracle Servlet Engine, des Oracle HTTP Servers (Apache/Jserv) und den einzelnen Beispielen.

Mit www.akadia.com:7777 kommen Sie direkt zu dieser Beispielkonfiguration.

Karl Seematter, Akadia AG

Mail: karl.seematter@akadia.com

Internet: www.akadia.com