

HANSER

**Pragmatisch
Programmieren
Projekt-Automatisierung**

von Michael Clark

ISBN 3-446-40008-7

Leseprobe Kapitel 6.8

Weitere Informationen oder Bestellung
unter <http://www.hanser.de/3-446-40008-8>
sowie im Buchhandel

6.8 Überwachen der Gesundheit mit einem Debug-Befehl

Wenn eine entfernte Anwendung Anzeichen von Unwohlsein zeigt, wäre es doch hilfreich, sie nach dem Ort der Schmerzen fragen zu können. Dazu müssen Sie einen internen Debug-Befehl implementieren, der den Gesundheitszustand der Anwendung bei Bedarf ausgibt. Die Ausgaben können dabei Folgendes beinhalten:

- JVM-Statistiken
- Die zuletzt aufgezeichnete Fehlermeldung
- Anzahl der gleichzeitigen Anwender im System
- Offene Datenbankverbindungen und die Größe des Verbindungspools
- Durchschnittliche Antwortzeit einer wichtigen Webseite

Eine Möglichkeit, an eine Debug-Ausgabe zu kommen ist, ein Programm auf der Kommandozeile auszuführen, das sich mit einem entfernten Programm verbindet und nach einem Bericht fragt. Eine Webschnittstelle bietet jedoch eine mächtige Alternative.

Einen Webserver fragen

HTTP ist ein allgegenwärtiges und bequemes Schnittstellenprotokoll, das selbst über die meisten Firewalls hinweg funktioniert. Wenn eine Anwendung in einem Webserver ausgeführt wird, können Sie von ihr Informationen mittels HTTP-Anfragen abrufen. Um zum Beispiel eine Debug-Ausgabe der laufenden Anwendung *DMS* zu bekommen, senden Sie eine HTTP-Anfrage wie diese:

```
http://xyz.com:8080/dms/debugdump
```

Diese URL verweist auf eine Debug-Übersicht mit einer beliebigen Menge an Diagnoseinformation der Anwendung, die vielleicht

erst im Moment der Anfrage zusammengestellt werden. Um einen Diagnosebericht zu bekommen, gehen Sie entweder mit einer Browser zu dieser URL oder verwenden einen Screen-Scraper, der die Anfrage absendet und die Antwort nach den verräterischen Zeichen von Problemen absucht.

Diese Überwachungstechnik ist leistungsfähig genug, dass Sie selbst bei entfernten Anwendungen, die noch nicht innerhalb eines Webservers laufen, über die Integration eines Webservers nachdenken sollten. Bevor Sie also selbst eine Diagnoseschnittstelle bauen, ergänzen Sie die Anwendung um HTTP-Fähigkeiten.

Das Einbetten eines Webservers kann bedeuten, die Minimalversion eines Webservers zu schreiben oder einfach eine Bibliothek wie Jetty zu benutzen:¹⁰ ein kleiner, schneller, kostenloser und leicht integrierbarer HTTP-Server und Servlet-Container. So einfach kann die Integration von Jetty sein:

```
import org.mortbay.http.HttpServer;
import org.mortbay.jetty.servlet.ServletHandler;

public class WebServer {
    public static void main(String[] args) throws Exception {
        HttpServer server = new HttpServer();
        server.addListener("8080");
        ServletHandler handler = new ServletHandler();
        handler.addServlet("DebugDumpServlet", "/debugdump/*",
            "com.pragprog.DebugDumpServlet");
        server.getContext("/").addHandler(handler);
        server.start();
    }
}
```

monitors/webserver/WebServer.java

Wenn Sie diesen minimalen Code ausführen, startet ein HTTP-Server auf Port 8080 und bindet das `DebugDumpServlet` an den Pfad `/debugdump/`. Noch einfacher ist es, statische Inhalte aus einem Verzeichnis bereitzustellen.

6.9 Erstellen eines Absturzberichtes

Manchmal stürzen Anwendungen ab. Wenn das passiert, möchten wir eine postmortale Analyse durchführen, um das Problem zu finden. Wir hoffen, so ein erneutes Auftreten verhindern zu können.

¹⁰ <http://jetty.mortbay.org>