

HANSER

**Pragmatisch  
Programmieren  
Projekt-Automatisierung**

von Michael Clark

ISBN 3-446-40008-7

Leseprobe Kapitel 6.9

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

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:<sup>10</sup> 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.

---

<sup>10</sup> <http://jetty.mortbay.org>

Protokolldateien können einige Hinweise enthalten, aber es können auch zu viele Informationen zum Durchforsten sein. Und normalerweise sind sie auch nur ein Teil des Puzzles.

Gehen Sie einen Schritt weiter und schreiben ein einfaches Skript (oder eine Batch-Datei), das genau weiß, wie es an die richtige Menge von Beweisen aus all den unterschiedlichen Quellen kommt. Das könnten beispielsweise folgende Daten sein:

- Die letzten  $x$  Meldungen in der Protokolldatei
- Die Version der Anwendung
- Das Betriebssystem und die JVM-Version
- Wichtige Umgebungsvariablen und Systemeigenschaften
- Namen anderer laufender Prozesse

Wenn ein Anwender einen Absturz erlebt, startet er das Skript und schickt Ihnen den resultierenden Absturzbericht. Noch besser wäre, wenn die Anwendung den Absturzbericht automatisch erzeugen könnte und ihn in einem Dialogfenster oder HTML-Formular anzeigen würde. Der Anwender hätte so die Möglichkeit, den Absturzbericht noch einmal gegenzulesen, bevor er ihn per Knopfdruck an den Server des technischen Supports übermittelt. Wie wir in der vorangegangenen Geschichte über die automatischen Absturzberichte gesehen haben, könnte dieser Server den Absturz als neuen Fehler automatisch in Ihre Problemdatenbank eintragen.