

S.mon Installation

1 Einführung

In diesem Dokument wird beschrieben, wie man S.mon installiert und in Betrieb nimmt. Es wird vorausgesetzt, dass sich der Benutzer mit Tomcat auskennt und weiß, wie man ein neues Servlet bei Tomcat aufsetzt.

2 Anforderungen

Folgendes wird für den Betrieb von S.mon notwendig:

- Java Runtime Environment 1.3 oder JDK1.3 (Siehe <http://java.sun.com/>)
- Tomcat 4.0.3 oder höher (Siehe <http://jakarta.apache.org/tomcat/>)
- S.mon Distribution
- S.mon License-File
- Zugriff auf einen Datenbank-Server (MS SQL, Oracle, MySQL oder PostgreSQL)

3 Umgebung

Folgende Umgebungsvariablen sollten gesetzt sein:

- JAVA_HOME (zeigt auf das Verzeichnis, wo Java installiert ist)
- CATALINA_HOME (zeigt auf das Verzeichnis, wo Tomcat installiert ist)

Windows	Unix (bash)
set JAVA_HOME=c:\jdk1.3 set CATALINA_HOME=c:\jakarta-tomcat-4.0.6	export JAVA_HOME /usr/local/jdk1.3 export CATALINA_HOME /usr/local/jakarta-tomcat-4.0.6

Wir nehmen auch an, dass sich das ausführbare Programm „java“ auch schon im Pfad befindet.

Wir nehmen weiter an, dass sich die S.mon Distributionsdateien unter **Windows** in dem Verzeichnis „c:\smon“ und unter **Unix** in dem Verzeichnis „/opt/smon“ befinden.

Wir nehmen an, dass der zu verwendete Datenbank-Server ein MySQL Server ist. Dass der DB-Server auf der lokalen Maschine läuft. Dass eine Datenbank-Instanz mit dem Namen „smon“ von dem DB-Administrator angelegt wurde. Und dass dem Benutzer mit dem Namen „smon“ und dem Passwort „smonpwd“ entsprechende Rechte für diese Datenbank eingeräumt wurden.

Daraus ergeben sich folgende Einstellungen.

Für die Umgebungsvariable „CLASSPATH“

Unter Windows:

```
set CLASSPATH=%CLASSPATH%;c:\smon\lib\mm.mysql.jar
set CLASSPATH=%CLASSPATH%;c:\smon\lib\smon.jar
```

und unter Unix (bash):

```
export CLASSPATH=$CLASSPATH:/opt/smon/lib/mm.mysql.jar
export CLASSPATH=$CLASSPATH:/opt/smon/lib/smon.jar
```

Benutzt man eine andere Datenbank, so ist natürlich ein anderer JDBC-Treiber in das CLASSPATH aufzunehmen. Die Datei „smon.jar“ beinhaltet die S.mon-Klassen.

Weiterhin wird für die Aufrufe von SQL-Skripts von der Command-Line eine Konfigurationsdatei in dem Benutzer-Homeverzeichnis benötigt. Diese Datei soll „**sqlquery.conf**“ heißen und muss folgende Einträge enthalten:

- db.platform (Der Name der zu verwendeten Plattform. In unserem Fall: mysql)
- db.driver (JDBC-Treiber. In unserem Fall: org.gjt.mm.mysql.Driver)
- db.url (URL für die S.mon Datenbak. In unserem Fall: jdbc:mysql://localhost:3306/smon)
- db.user (Name des DB-Benutzers. In unserem Fall: smon)
- db.password (Passwort für den Benutzer. In unserem Fall: smonpwd)

4 Datenbank-Initialisierung

Um die Datenbank-Tabellen in der S.mon-Datenbank anzulegen und diese zu initialisieren, müssen folgende Schritte durchgeführt werden:

1. Gehen Sie in das Unterverzeichnis „sql“ von S.mon
cd c:\smon\sql (Windows) bzw. cd /opt/smon/sql (Unix)
2. java gs.db.tools.CreateTables tables.xml mysql mysql_create.sql
Erzeugt SQL-Skript um die DB-Tabellen anzulegen.
3. java gs.db.tools.CreateSQL tables.xml mysql mysql_init.sql
Erzeugt SQL-Skript um die S.mon DB-Tabellen zu initialisieren
4. java gs.db.Sql mysql_create.sql
Führt das SQL-Skript aus. Es werden die Tabellen angelegt.
5. java gs.db.Sql mysql_init.sql
Führt das SQL-Skript aus. Die S.mon Datenbank wird initialisiert.

5 Anlegen eines Administrators

6 Konfiguration des S.mon Servlets

In dem Konfigurationsverzeichnis (config) von S.mon befindet sich die Datei „web.xml“. Es ist das Deployment-Deskriptor für die Servlet-Engine.

Sie müssen diese Datei mit einem Text- oder einem XML-Editor an Ihre Bedürfnisse anpassen. Die untere Tabelle erklärt die möglichen Einträge (Init-Parameters):

Name	O/R ¹	Beschreibung	Default-Wert
gs.uploadDir	R	Das Verzeichnis, wo das Servlet die Upload-Dateien temporär speichert. Das Verzeichnis muss existieren und Schreibrechte sollten entsprechend gesetzt werden. Das Verzeichnis sollte aus Sicherheitsgründen nicht unter dem Web-Root liegen!	
gs.xslDir	R	Das Verzeichnis, wo die XSL-Dateien zu finden sind. Unter Windows bitte auf „c:\smon\xsl“ und unter Unix auf „./opt/smon/xsl“ setzen.	

¹ O: optional, R: required

gs.commands	R	Verweist auf die XML-Datei mit den Kommandos, die von dem Servlet ausführbar sind: Unter Windows bitte auf „c:\smon\config\commands.xml“ und unter Unix auf „./opt/smon/config/commands.xml“ setzen.	
gs.log4jConfig	O	Verweist auf eine Log4Java Konfigurationsdatei. Wenn nicht gesetzt wird BasicConfigurator zur Initialisierung von Log4j verwendet.	BasicConfigurator
licenseFile	R	Verweist auf die Lizenz-Datei. Haben Sie eine Lizenz-Datei für S.mon erhalten, so kopieren Sie diese am besten in das license Verzeichnis und setzen hier entsprechend den Wert.	
smtpHost	O	Name des Rechners, wo ein SMTP-Daemon läuft. Dies wird benötigt, um E-Mails zu verschicken	localhost
mailFrom	R	Das ist die Sender-Adresse bei E-Mail-Benachrichtigung.	
mailGatewayDomain	O	Ist die MailExchange-Funktionalität freigeschalten, dann sollte man hier diese eintragen.	
mailDomain	O	Angabe der lokalen Mail-Domain. Hat man für die Bearbeiter die Mail-Adressen nicht konfiguriert, so wird es bei einer Benachrichtigung versucht, eine Mail-Adresse aus dem Login-Namen und dieser Domain zu bilden.	leer
uiLink	R	Die vollständige URL für S.mon Servlet. Wird benötigt beim Versenden von E-Mails	
staticUILink	R	Die vollständige URL für den Zugriff auf die statischen S.mon Seiten.	
errorXSL	R	Name der XSL-Datei, welche die Fehlermeldungen anzeigen soll.	
aclErrorXSL	R		
db.log	O	Sollen die SQL-Abfragen in der Log-Datei gespeichert werden? Man kann hier den Wert true oder 1 setzen, wenn die Speicherung gewünscht ist, oder false bzw. 0, wenn die Speicherung unerwünscht ist.	false
dbPlatform	R	Name der Datenbank-Plattform, die für S.mon benutzt wird. Folgende Werte sind zulässig: <ul style="list-style-type: none"> • MySQL • MSSQL • Oracle • PostgreSQL Die Schreibweise ist egal.	
jdbcDriver	R	Klassenname für den JDBC-Treiber	

jdbcURL	R	Die URL für die S.mon Datenbank	
jdbcUser	R	Name des DB-Benutzers, der auf die S.mon Datenbank zugreifen darf.	
jdbcUserPassword	R	Das Passwort für den DB-Benutzer	
db.connectionPool.min	O	Der Zugriff auf die Datenbank erfolgt mittels eines Pools von DB-Connections. Statt bei jeder DB-Abfrage eine neue Verbindung aufzubauen, greift man auf eine Connection aus dem Pool zurück. Das Pool wird am Anfang initialisiert. Hier wird angegeben wie viele Connections am Anfang erzeugt werden sollen.	5
db.connectionPool.max	O	Hier kann man die maximale Anzahl von DB-Connections in dem oben beschriebenen Pool angeben.	20
searchIndex	O	Ist die Fulltext-Search-Funktionalität in dem License-File freigeschalten, so spezifiziert man hier das Verzeichnis, wo die Index-Dateien für die Volltextsuche durch die Search-Engine angelegt werden sollen.	
defaultLanguage	O	Diese Sprache wird als Default in dem Web-Frontend benutzt. Man sollte hier den ISO-Code eingeben. Wichtig ist, dass die Lokalisierung für die Sprache auch tatsächlich erledigt wurde. In S.mon werden derzeit folgende Sprachen unterstützt: <ul style="list-style-type: none"> • Englisch (en) • Deutsch (de) • Polnisch (pl) • Tschechisch (cs) 	en
supportedLanguages	O	Liste mit der unterstützten Sprachen, welche in dem GUI ausgewählt werden können. Diese Liste beinhaltet Sprach-Codes, die durch Kommas oder Leerezeichen voneinander getrennt sind. Ist die Liste leer, so wird die Default-Sprache benutzt. Ist die Liste nicht leer, so bitte zu beachten, dass diese Liste auch die Default-Sprache enthält.	
defaultCountry	O	Default für das Land, wo dies Service läuft. Dies wird für die automatische Berechnung von Feiertagen benötigt. Man setzt hier den ISO Country-Code. Derzeit werden folgende Werte unterstützt: <ul style="list-style-type: none"> • Austria (at) • Germany (de) • Czech (cz) • Poland (pl) • Hungary (hu) • Slovakia (sk) 	at

attachment.limit	O	Maximale Größe einer Beilage in KBytes. Wenn auf 0 gesetzt, dann gibt es kein Limit.	0 -> unlimited
state.new	O	Man kann hier den Namen des Zustandes „New“ festlegen. Es stehen bis zu drei Zeichen zur Verfügung. Der Name des Zustandes wird in jedem Defect mitgespeichert.	NA
state.postponed	O	Hier legt man fest, wie der Zustand „Postponed“ heißen soll.	PP
state.assigned	O	Hier legt man fest, wie der Zustand „Assigned“ heißen soll.	AS
state.opened	O	Hier legt man fest, wie der Zustand „Opened“ heißen soll.	OP
state.forwarded	O	Hier legt man fest, wie der Zustand „Forwarded“ heißen soll.	FW
state.waiting	O	Hier legt man fest, wie der Zustand „Waiting“ heißen soll.	WT
state.resolved	O	Hier legt man fest, wie der Zustand „Resolved“ heißen soll.	SO
state.verified	O	Hier legt man fest, wie der Zustand „Verified“ heißen soll.	VF
state.closed	O	Hier legt man fest, wie der Zustand „Closed“ heißen soll.	CL
cron.job1	O	S.mon bietet eine Plugin-Architektur an, um Kunden spezifische Cron-Jobs auszuführen. Es ist möglich bis zu fünf solche Cron-Jobs zu konfigurieren. Dieses Feature wird nur dann aktiv, wenn es in dem License-File freigeschalten wurde. Hier wird der erste Job konfiguriert.	
cron.job2	O	Klassenname für den zweiten Cron-Job.	
cron.job3	O	Klassenname für den dritten Cron-Job.	
cron.job4	O	Klassenname für den vierten Cron-Job.	
cron.job5	O	Klassenname für den fünften Cron-Job.	
event.listener1	O	S.mon bietet eine Plugin-Architektur an, um an die Kunden spezifischen Event-Handlers die S.mon Events zu feuern. Man kann bis zu fünf solche Handler-Klassen konfigurieren. Hier konfiguriert man den ersten Handler.	
event.listener2	O	Klassenname für den zweiten Event-Handler.	
event.listener3	O	Klassenname für den dritten Event-Handler.	
event.listener4	O	Klassenname für den vierten Event-Handler.	
event.listener5	O	Klassenname für den fünften Event-Handler.	

7 Deployment-Datei

Nachdem, die Konfiguration für **S.mon** abgeschlossen wurde, kann man die Deployment-Datei für den Tomcat-Server erzeugen.

Bitte kopieren Sie zuerst die Konfigurationsdatei web.xml in das richtige Deployment-Verzeichnis.

Unter **Windows**:

```
copy c:\smon\config\web.xml c:\smon\deploy\WEB-INF
```

und unter **Unix**:

```
cp /opt/smon/config/web.xml /opt/smon/deploy/WEB-INF
```

Gehen Sie bitte in das Deployment-Verzeichnis.

Unter **Windows**:

```
cd c:\smon\deploy
```

und unter **Unix**:

```
cp /opt/smon/deploy
```

Nun geben Sie bitte den folgenden Befehl:

```
jar cvf smon.war *
```

Es sollte nun die Deployment-Datei für Tomcat erstellt werden. Kopieren Sie diese Datei in das WebApps-Verzeichnis von Tomcat:

Unter **Windows**:

```
copy c:\smon\deploy\smon.war %CATALINA_HOME%\webapps
```

und unter **Unix**:

```
cp /opt/smon/deploy/smon.war ${CATALINA_HOME}/webapps
```

8 Starten von Tomcat

Alles ist nun vorbereitet und Tomcat kann gestartet werden. Bitte lesen Sie die Tomcat-Dokumentation für weitere Details.

Unter **Windows**:

```
%CATALINA_HOME%\bin\catalina start
```

und unter **Unix**:

```
${CATALINA_HOME}/bin/catalina start
```

Nach einem erfolgreichen Start sollten Sie auf **S.mon** zugreifen können.

9 Hinweise

Sollten Sie unter Unix Probleme mit den Zeichensätzen entdecken, so sollten Sie die Umgebungsvariable „LANG“ entsprechend setzen.

Csh	Bash
setenv LANG de	export LANG de

Mittels der Umgebungsvariable „CATALINA_OPTS“ kann man den Speicher für Java erweitern

Windows	Unix (bash)
set CATALINA_OPTS=-Xmx256m	export CATALINA_OPTS=-Xmx256m