

Webseitenautomat

LARS KOTTHOFF

Das Veröffentlichen von Inhalten für das Internet ist heutzutage allgegenwärtig. Ob es nun um ein privates Blog, die Internet-Präsenz der Firma, oder die Dokumentation für eine entwickelte Software geht – überall werden Webseiten publiziert. Die Redaktionsprozesse lassen sich mit Jenkins sehr einfach automatisieren.



Eine Webseite im Internet bereit-zustellen kann einen einzigen Mausklick dauern. Es gibt beispielsweise eine Unmenge von Anbietern von Software für das Veröffentlichen eines Blogs. Das ist spezielle Software, die den Prozeß so vereinfacht, daß auch Anwender mit minimalen technischen Kenntnissen nicht überfordert werden. Die Anpassungsmöglichkeiten sind bei solchen Systemen aber meist gering. Reichen sie nicht aus, müssen Speziallösungen entwickelt werden. Diese verlangen meist aber mehr Fachkenntnisse vom Benutzer. Weniger technisch versierte Anwender haben in diesem Fall schnell Schwierigkeiten im Umgang mit dem System, was die Anforderungen an die Systemadministratoren erhöht.

Jenkins

Ein einfacher Weg, die technischen Anforderungen an den Benutzer zu beschränken, ist, das Publizieren der Inhalte zu automatisieren. Dafür eignen sich speziell automatische Build-Systeme wie Jenkins (vorgestellt in der freeX 5'2011). Sie wurden ursprünglich dazu konzipiert, eine Software bei Quelltextänderungen neu zu kompilieren und gleich noch Tests zu unterziehen, um sicherzustellen, daß nicht aus Versehen Fehler eingebracht werden. Das bereitgestellte Gerüst ist jedoch allgemein genug, um

auch andere Aufgaben zu meistern. Jenkins ist in Java geschrieben und benötigt einen Servlet-Container. Die Distribution bringt jedoch alle Abhängigkeiten mit, lediglich die Java-Installation muß bereitgestellt werden. Nach dem Starten von der Kommandozeile steht eine Web-Oberfläche für die Steuerung zur Verfügung.

Konfiguration

Neben der eigentlichen Bereitstellung einer Webseite können im Rahmen einer Automatisierung auch gleich noch weitere Aufgaben erfüllt werden. Jeder hat sicherlich schon Seiten mit fehlerhaftem HTML und darin begründeten Darstellungsproblemen gesehen. Solch grundlegende Sachverhalte lassen sich vollautomatisch

überprüfen. Das gilt genauso für ausgehende Links – ist beispielsweise die Seite, auf die verwiesen wird, überhaupt noch vorhanden?

Schließlich hat die Automatisierung auch einen Sicherheitsaspekt: Speziell bei größeren Seiten ist nicht gewünscht, daß jeder Autor der Inhalte vollen Zugriff auf die Seite hat. Ohne die Benutzerkonten und die Zugriffskontrollen eines Content-Management-Systems (das dann wieder seine eigenen Probleme mitbringt) ist diese Anforderung kaum zu realisieren. Auch hier kann Jenkins helfen, denn anstelle die Inhalte unmittelbar selbst zu veröffentlichen, müssen die Benutzer lediglich die Änderungen bereitstellen.

Die Konfiguration eines solchen Projektes in Jenkins ist auch für Anwender, die keine Erfahrung mit automatischen

```
# Inhaltsverzeichnis aktualisieren
./baue_inhaltsverzeichnis html/*.html > html/inhalt.html

# HTML überprüfen
tidy -q -e html/*.html

# Links überprüfen
for html in html/*.html; do
  out=`checklink -q $html`
  if [X$out != X ]; then
    exit 1
  fi
done

# Inhalte kopieren
scp -i /ssh/schlüssel html/*.html webserver:public_html/
```

Listing 1:
Build-Schritte

Build-Systemen haben, kein Problem. Alle Einstellungen werden auf der Web-Oberfläche vorgenommen. Dort können auch die Ergebnisse der Durchläufe eingesehen werden. Die Inhalte der Webseite werden in einem Git-Versionsverwaltungssystem gehalten, das aller 15 Minuten auf Änderungen überprüft wird. Wird eine Änderung festgestellt, werden die neuen Inhalte geholt. Dies geschieht automatisch, es ist keine spezielle Konfiguration erforderlich. Danach wird das Inhaltsverzeichnis neu generiert. Dieser Vorgang ist hier nur symbolisch dargestellt, indem ein Skript aufgerufen wird, das die eigentlichen Schritte vornimmt. Danach wird die Gültigkeit des HTML-Quelltextes mit HTML-Tidy und die Gültigkeit der Links mit *checklinks* überprüft. Schließlich werden die neuen Inhalte auf den Webserver hochgeladen. Dabei ist der SSH-Schlüssel für die Person, die die Änderung vorgenommen hat, nicht sichtbar. Die einzelnen Schritte sind in Listing 1 aufgeführt und die relevanten Teile der Konfiguration sind in Bild 1 gezeigt.

Versionsverwaltung für Webseiten

Falls einer der Schritte fehlschlagen sollte, werden die nachfolgenden Schritte nicht ausgeführt. Sein Erfolg oder Mißerfolg wird über den Rückgabewert des zuletzt ausgeführten Befehls festgestellt. Sollte HTML Tidy zum Beispiel ein Problem feststellen, bricht Jenkins den Vorgang ab und die neuen Inhalte werden nicht auf den Server hochgeladen. Es besteht die Möglichkeit, den Autor der Änderung per E-Mail auf das Problem hinzuweisen. Dann kann er den Fehler beheben, ohne daß der Weltöffentlichkeit eine fehlerhafte Webseite präsentiert wird. Der Benutzer selbst benötigt keinen Zugang zu Jenkins, sondern lediglich zum Versionsverwaltungssystem, in dem die Inhalte der Webseite gespeichert werden. Somit hat der Systemadministrator die volle Kontrolle über den Prozeß und braucht Paßwörter oder Schlüssel, die mißbraucht wer-



Bild 1: Relevante Teile der Konfiguration

den könnten, nicht an die Anwender weiterzugeben. Jenkins verfügt über eine interne Benutzerverwaltung, die auch in eine vorhandene Infrastruktur wie zum Beispiel LDAP integriert werden kann. Damit kann leicht verschiedenen Administratoren Vollzugriff oder privilegierten Benutzern Teilzugriff auf die Funktionalität gegeben werden.

Die obige Konfiguration ist nur ein Beispiel, das nahezu beliebig angepaßt und erweitert werden kann. So verfügt Jenkins über ein Plugin, das ein Verzeichnis auf Änderungen überwacht und feststellt, wann etwas getan werden muß. Meldungen über Fehler oder das erfolgreiche Publizieren der neuen Inhalte können als RSS-Feed, über verschiedene Instant-Messaging-Systeme oder durch Twitter versandt werden. Nach dem erfolgreichen Abschluß können automatisch weitere Vorgänge wie zum Beispiel das Verteilen der Inhalte auf mehrere Server, ausgelöst werden.

Dieser Beitrag soll nur einen Geschmack davon geben, was mit Jenkins möglich ist. Nicht nur Webseiten, auch andere Inhalte können damit verwaltet werden – Plugins für virtuelle Maschinen machen es zum Beispiel möglich, Server-Konfigurationen vor dem Produktivbetrieb zu testen. ♦

Links

- [1] Jenkins -Homepage:
<http://jenkins-ci.org/>
- [2] Jenkins-Plugins:
<https://wiki.jenkins-ci.org/display/JENKINS/Plugins>
- [3] HTML-Tidy:
<http://tidy.sourceforge.net/>
- [4] W3C checklink:
<http://validator.w3.org/docs/checklink.html>
- [5] Die jeweils aktuelle Jenkins Version:
<http://mirrors.jenkins-ci.org/war/latest/jenkins.war>