

Java-basiertes CMS – PHP-basierte Seitenlogik

# FirstSpirit und PHP

Das Java-basierte Enterprise-Content-Management-System FirstSpirit wird häufig als Basis für unterschiedlichste Webprojekte, von Corporate-Internetseiten über Intra- und Extranets und Webanwendungen, genutzt. Auf Grund von eventuell vorhandener Infrastruktur und Kompetenzen kann es sich auch hier anbieten, interaktive Seitenfunktionen mittels PHP umzusetzen. Im Folgenden wird eine Einführung in die grundlegenden FirstSpirit-Konzepte und die Verwendung in Kombination mit PHP gegeben.

► von Jan Eickmann

**F**irstSpirit überzeugt als Enterprise-Content-Management-System (ECMS) durch seine umfangreichen Funktionen, Skalierbarkeit und Unterstützung verschiedener Ausgabekanäle und -sprachen. Aus diesem Grund kommt es immer häufiger zur Umsetzung von Webprojekten zum Einsatz. Auch wenn das System auf einer Java-Plattform basiert, ist die Entwicklung von interaktiven Funktionen innerhalb der produzierten Seiten nicht zwangsweise auf J2EE-Technologien beschränkt. Insbesondere wenn die vorhandene Infrastruktur, Kompetenzen und Bibliotheken bereits auf PHP basieren, sollte die Möglichkeit, diese innerhalb von FirstSpirit-Templates zu nutzen, nicht missachtet werden. Der vorliegen-

de Artikel gibt einen Überblick über die dafür notwendige Konfiguration einer FirstSpirit-Installation und ein kleines Beispiel zur Umsetzung interaktiver Funktionen mittels PHP innerhalb von FirstSpirit-Templates.

## Das First-Spirit-Seiten-/Publishing-Konzept

Zur sinnvollen Verwendung des FirstSpirit-Template-Systems in Verbindung mit PHP ist ein grundlegendes Verständnis des Publishing-Konzepts von FirstSpirit erforderlich. Generell unterscheidet das ECMS zwischen Seiten- und Absatzvorlagen. Zur Erstellung einzelner Seiten wird jeweils eine Seitenvorlage ausgewählt und kann vom Redakteur mit verschiedenen Absatzvorlagen gefüllt

werden. Eine Vorlage besteht dabei aus einer Definition der für den Redakteur zur Konfiguration der Seite/des Absatzes verfügbaren Felder (wie etwa Seitentitel, Absatztext etc.). Weiterhin enthält die Vorlage eine oder mehrere Ausgabe-Templates für verschiedene Ausgabekanäle. Diese bereiten die Feldinhalte über eine FirstSpirit-eigene Template-Sprache auf. Innerhalb von FirstSpirit erfolgt eine klare Trennung zwischen Redaktionssystem und erstellter Webseite. Durch den Vorgang der Generierung werden die angelegten Seiten unter Verwendung der gewählten Templates einmalig generiert und auf einen separaten Webserver veröffentlicht. Auf diese Weise werden statische Seiteninhalte erzeugt, die sich sehr performant ausliefern lassen und keine

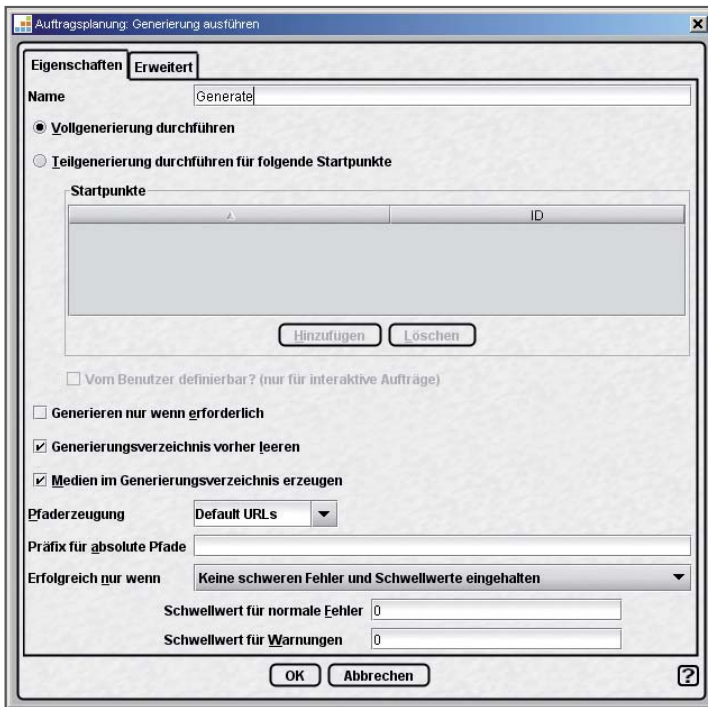


Abb. 1:  
Konfiguration  
der Generier-  
ungsaktivität

fehle direkt in die in FirstSpirit definierten Templates, die anschließend durch die Generierung automatisch auch in die auf dem Webserver veröffentlichten Seiten eingefügt werden.

## System bereitstellen

Hierzu wird neben der Installation des FirstSpirit-Redaktionssystems ein PHP unterstützender Webserver benötigt. Dieser wird im Folgenden zur Vereinfachung der Architektur auf dem gleichen System wie das Redaktionssystem installiert und liefert aus einem regulären Webroot PHP-Seiten aus. Im vorliegenden Fall wurde ein Apache mit *mod\_php* gewählt. Die Generierung der Seiten erfolgt dann aus FirstSpirit heraus in dieses Webroot.

Selbstverständlich kann der Webserver in einem Livebetrieb auch auf einem separaten System unabhängig vom Redaktionsserver betrieben werden. In diesem Fall werden die generierten Seiten entweder während der Generierung von FirstSpirit automatisch mittels eines FTP-Zugangs auf dem Webserver veröffentlicht, oder es erfolgt eine regelmäßige Synchronisation des lokalen Verzeichnisses mittels eines Tools wie rsync. Die Erfahrung zeigt hier eine deutlich schnellere Veröffentlichung bei Verwendung von rsync im Gegensatz zur FTP-basierten Veröffentlichung.

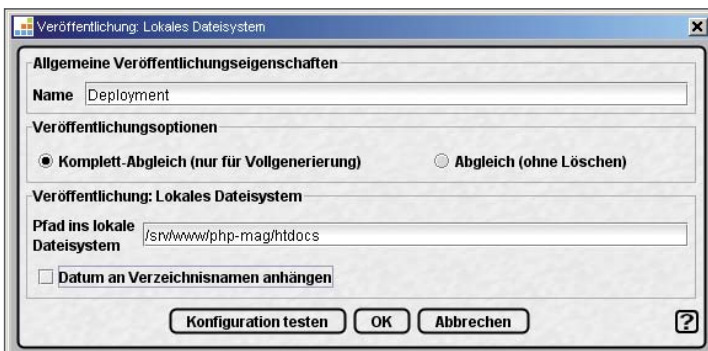


Abb. 2:  
Konfiguration  
der Veröffentli-  
chungsaktivität

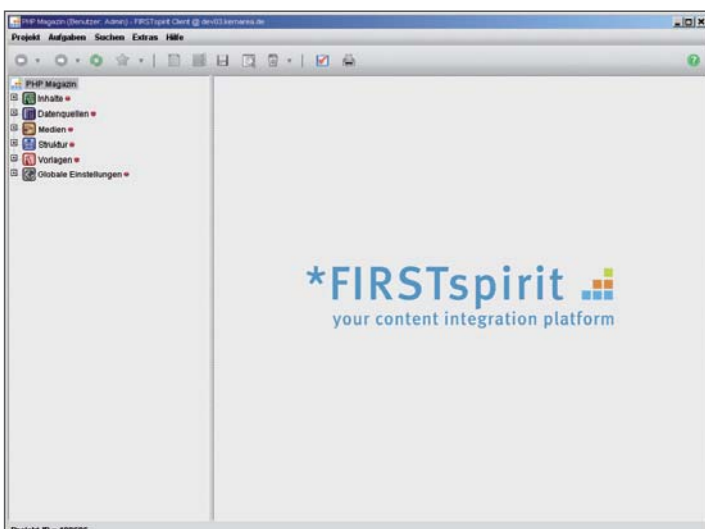


Abb. 3:  
Die FirstSpirit-  
Oberfläche im  
Redaktionsclient

## Anlegen und Konfigurieren des Beispielprojekts in FirstSpirit

Über die FirstSpirit-Projektverwaltung wird ein Beispielprojekt für die PHP-basierten Templates angelegt. Dieses muss anschließend entsprechend konfiguriert werden. Nachdem das Projekt unter Angabe eines Projektnamens angelegt wurde, wird zuerst unter dem Punkt *Auftragsverwaltung* die Generierung und Veröffentlichung konfiguriert. Hierzu wird ein neuer Auftrag namens *Deployment* angelegt, der zwei Aktivitäten beinhaltet. Für die erste Aktivität wird der Typ *Generierung ausführen* gewählt. Im folgenden Dialogfenster können die Standardeinstellungen größtenteils übernommen werden. Wichtig ist hier nur, auf der Reiterkarte *Erweitert* die automatisch angelegte Sprache *Deutsch (de)* auszuwählen, da sonst auf Grund

weitere Verfügbarkeit des FirstSpirit-Servers erfordern. Zur Umsetzung dynamischer Funktionen mittels PHP müssen die entsprechenden PHP-Befehle in den

generierten Seiteninhalt eingefügt und zur Auslieferungszeit durch den Webserver interpretiert werden. Dies geschieht durch Einfügen der entsprechenden Be-

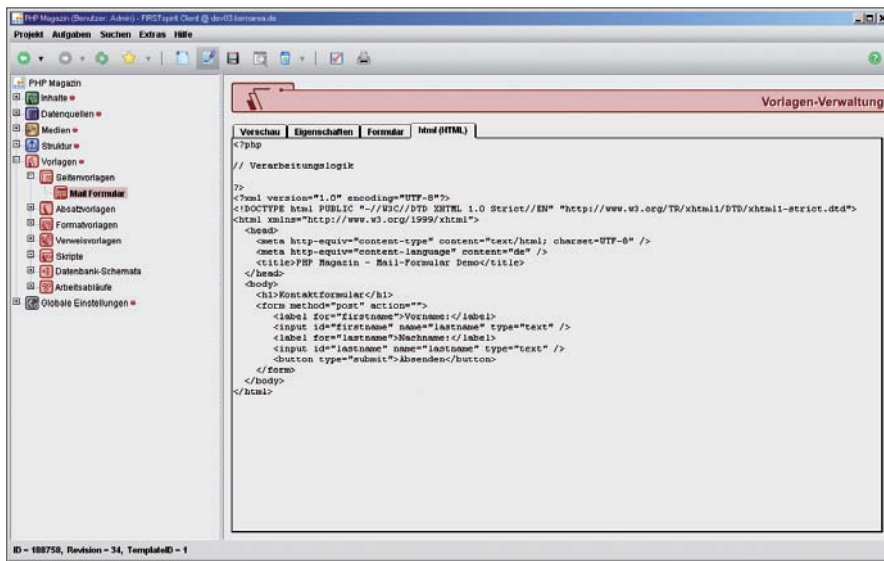


Abb. 4: Bearbeiten der Seitenvorlage

mangelnder Sprachversionen keinerlei Generierung erfolgt.

Die zweite Aktivität erhält den Typ *Veröffentlichung ausführen*. Als Art der Veröffentlichung wird hier *lokale Veröffentlichung* gewählt (an dieser Stelle könnte, wie weiter oben besprochen, auch eine FTP-basierte Veröffentlichung auf einen separaten Server erfolgen). Zur Konfiguration dieser Aktivität muss der Pfad im lokalen Dateisystem angegeben werden, in dem die generierten Seiten abgelegt werden sollen. Dieser entspricht üblicherweise dem Webroot oder einem darin befindlichen Projektverzeichnis.

Nachdem das Projekt auf diese Weise konfiguriert wurde, kann es mittels des Redaktionsclients geöffnet werden. In ihm werden sowohl die eigentlichen Seiten durch Redakteure als auch die Seiten- und Absatzvorlagen durch Entwickler verwaltet.

Die gewählte Art der Generierung erstellt bei jedem Generierungsvorgang die vollständige Webseite. Für größere Projekte kann auch eine Teilgenerierung konfiguriert werden. Hierauf wurde zu Gunsten der einfacheren Konfiguration allerdings in diesem Fall verzichtet.

### Eine Seitenvorlage anlegen

Zur Einführung von PHP-Funktionalität innerhalb von Templates soll eine einfache Seitenvorlage mit interaktiver Funktionalität erstellt werden. Diese erlaubt es dem Redakteur, ein konfigurierbares

Formular zu erstellen, das nach dem Absenden eine E-Mail mit dem Formularinhalt an eine vom Redakteur angegebene Adresse versendet. Ziel des Beispiels ist hier die Verdeutlichung der Integration mit FirstSpirit, sodass auf komplexe Eingabvalidierung und Layouting auf PHP-Seite verzichtet wird. Die Seitenvorlage wird innerhalb des FirstSpirit-Clients im Bereich *Vorlagen* durch einen Rechtsklick auf *Seitenvorlagen* erstellt. Es öffnet sich ein Fenster zur Angabe eines Anzeige- und Referenznamens. In unserem Beispiel werden hier die beiden Werte *Mail-Formular* und *mailform* angegeben. Danach kann die Seitenvorlage bearbeitet werden. Als Erstes wird im Reiter *Eigenschaften* die Zielendung der generierten Dateien von *html* in *php* geändert, damit diese vom Webserver korrekt als PHP-Code interpretiert werden. Alternativ kann selbstverständlich auch der Webserver umkonfiguriert werden, um auch HTML-Dateien als PHP-Code zu interpretieren. Anschließend wird im Reiter *html* der eigentliche Template-Code eingefügt. Für einen ersten Test wird hier lediglich ein HTML-Gerüst des Formulars erstellt. Später wird dieser Code um eingebettetes PHP ergänzt.

### Seite erzeugen und veröffentlichen

Nachdem die Seitenvorlage mit einem grundlegenden HTML-Gerüst erstellt wurde, kann eine Seite basierend auf

# Beste Bücher für besten Code!



Heiko Sippel, Jens Bendisposto,  
Michael Jastram

## Die Eclipse Rich Client Platform

Entwicklung von erweiterbaren  
Anwendungen mit RCP

369 Seiten, Hardcover

ISBN: 978-3-939084-91-4

34,90 €

Plug-in- und Rich-Client-Entwicklung ist auch für erfahrene Entwickler eine große Herausforderung. Sie müssen sich an neue Standards anpassen. Das macht ein Umdenken notwendig. Das Buch zeigt, warum dieses Umdenken sich lohnt. Es stellt die Plug-in-Entwicklung in den Mittelpunkt. Eine komplexe RCP-Beispielanwendung wird in kleinen, gut verständlichen Schritten erarbeitet. Ein leichter Einstieg für alle Entwickler, die zum ersten Mal eine RCP-Anwendung schreiben wollen.

Weitere Informationen und Bestellmöglichkeiten finden Sie unter [www.entwickler.press.de](http://www.entwickler.press.de). Unsere Bücher erhalten Sie auch in jeder gut sortierten Buchhandlung.



Abb. 5: Die erste Generierung war erfolgreich

## Kontaktformular

Vorname:  Nachname:

Abb. 6: Die vom Webserver ausgelieferte, generierte Seite im Browser

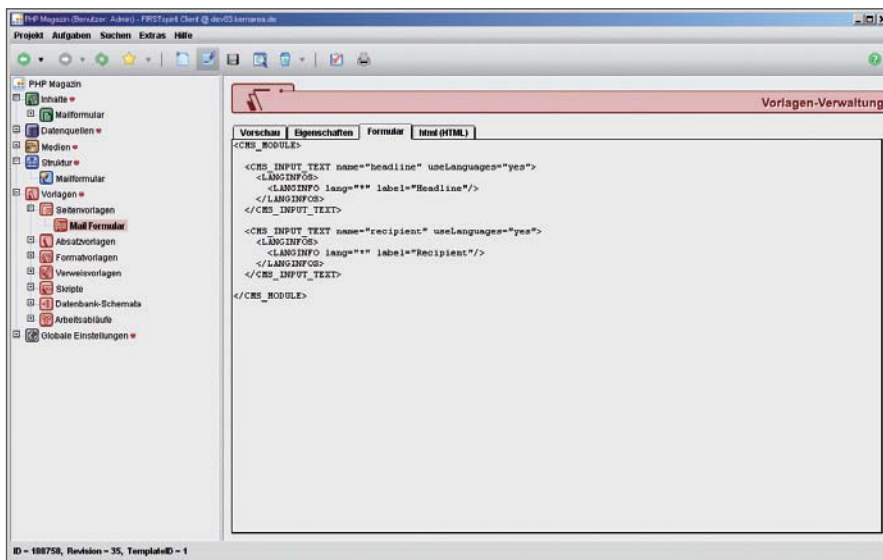


Abb. 7: Die Konfiguration des Formulars

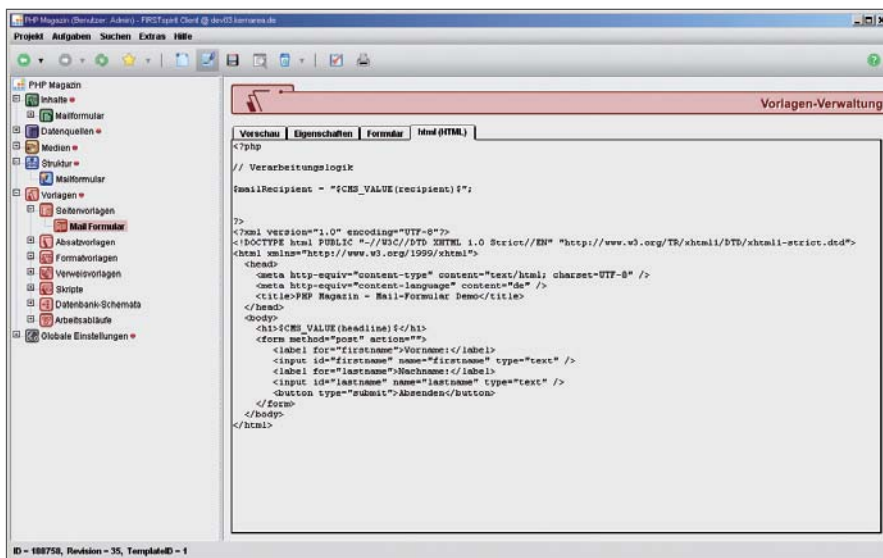


Abb. 8: Ausgabe von vom Redakteur gefüllten Werten

dieser Vorlage angelegt und die Veröffentlichung auf dem Webserver getestet werden. Hierzu wird im Bereich *Seiten* eine neue Seite angelegt und die soeben erstellte Seitenvorlage gewählt. Da bisher keinerlei vom Redakteur zu füllende Felder definiert wurden, ist keine weitere Angabe erforderlich. Ebenso können die Vorgaben für den Namen der Seite übernommen werden. Um die gerade angelegte Seite mit einem konkreten URL und einem Ablagepunkt zu versehen, muss im Bereich *Struktur* noch eine Seitenreferenz erzeugt werden. Sie verbindet die Seite mit einem konkreten Ablageort im generierten Projekt. Bei der Erstellung der Referenz wird die angelegte Seite gewählt. Abschließend kann ein Dateiname für die zu generierende (PHP-)Datei angegeben werden; vorerst kann hier ebenfalls der Vorgabewert, der dem Seitennamen entspricht, übernommen werden. Sind Seite und Seitenreferenz angelegt, kann die Generierung des Projekts gestartet werden. Hierzu wird aus dem Menü *Projekt* der Punkt *Auftrag ausführen* ausgewählt und der zu Beginn definierte Auftrag *Deployment* gewählt. Nach einer kurzen Weile erfolgt die Bestätigung, dass die Generierung abgeschlossen ist.

Im Webroot des Webservers findet sich nun die generierte Datei. Über einen Zugriff auf den Webserver kann auf diese zugegriffen und die generierte Seite betrachtet werden. Zu diesem Zeitpunkt sollte ebenfalls überprüft werden, ob der innerhalb des FirstSpirit-Templates angegeben leere PHP-Block vom Server interpretiert wurde und somit in der ausgelieferten Seite im Browser nicht mehr vorhanden ist. Sofern hier noch Probleme auftreten, liegt dies üblicherweise daran, dass entweder die Endung der generierten Datei nicht korrekt konfiguriert oder versehentlich nicht über den Webserver, sondern direkt über das Dateisystem auf die Datei zugegriffen wurde.

### Erweiterung der Seitenvorlage um vom Redakteur zu füllende Felder

Sobald die Seite auf diese Weise korrekt ausgeliefert wird, kann die Seitenvorlage in unserem ECMS erweitert werden. Dabei sollen nun zum einen dem Re-

dakteur Konfigurationsmöglichkeiten geboten, zum anderen die interaktiven Funktionen (im vorliegenden Fall das Versenden des Formularinhalts via E-Mail) realisiert werden. Zuerst werden die vom Redakteur zu befüllenden Felder im Reiter *Formular* der Seitenvorlage angegeben. Dies erfolgt über eine XML-basierte, FirstSpirit-eigene Beschreibungssprache. Eine detaillierte Beschreibung der möglichen Felder und Optionen findet sich direkt innerhalb des FirstSpirit-Clients unter dem Menüpunkt HILFE. Für die vorliegende Seitenvorlage werden zwei Felder angelegt. Zum einen kann vom Redakteur die Überschrift des Formulars angepasst werden, zum anderen die Empfängeradresse, an die die Formularinhalte gesendet werden sollen. Beide werden als einfache Textfelder umgesetzt. Zusätzlich sind in unserem ECMS selbstverständlich auch komplexere Eingabefelder wie Selectboxen, WYSIWYG-Editoren etc. verfügbar.

Um die vom Redakteur vergebenen konkreten Werte für die beiden Felder auch im Template zu verwenden, müssen diese innerhalb der generierten Seite ausgegeben werden. Dies erfolgt über die FirstSpirit-eigene Template-Sprache. Auch hier findet sich eine ausgiebige Dokumentation unter dem Menüpunkt HILFE direkt innerhalb des Clients. Für den vorliegenden Fall genügt der Befehl `$CMS_VALUE()`. Dieser erlaubt es, den Inhalt von konfigurierten Feldern direkt auszugeben. Da FirstSpirit bei der Seitengenerierung die gesamte Seite als Text behandelt, kann der Befehl sowohl zur Ausgabe innerhalb von eingebettetem PHP als auch zur Ausgabe innerhalb des HTMLs verwendet werden. Im Beispiel-Template wird etwa der Empfänger in eine PHP-Variable ausgegeben, während die Überschrift direkt als HTML-Inhalt für den Besucher sichtbar ausgegeben wird.

Bei der Verwendung von Feldinhalten innerhalb von PHP-Code ist zu beachten, dass diese behandelt werden sollen wie alle Benutzereingaben innerhalb von Webanwendungen, d. h. eine Überprüfung auf korrekt gefüllte Werte und eventuelle Steuerzeichen ist in einer Produktivumgebung unerlässlich. Im

vorliegenden Beispiel wurde zugunsten der Klarheit des Codes auf solche Überprüfungen verzichtet.

Öffnet man die weiter oben bereits angelegte Seite innerhalb des Redaktionssystems, können nun die für die Seitenvorlage konfigurierten Formularfelder gefüllt werden. Nachdem erneut der Deployment-Auftrag ausgeführt wurde, wird durch Zugriff auf den Webserver überprüft, ob die Felder korrekt innerhalb der generierten Seite dargestellt werden. Ist dies der Fall, kann nun die interaktive Funktionalität ergänzt werden, indem das Template um den notwendigen PHP-Code ergänzt wird.

### Umsetzung der interaktiven Funktionen

Wie bereits erläutert, wird PHP-Code einfach vom Entwickler innerhalb des FirstSpirit-Templates eingetragen und damit automatisch in die vom System generierten Dateien eingebettet. Zur Illustration dieses Vorgehens wird die Seitenvorlage des Beispiels um einen Versand der Formularinhalte ergänzt:

```
<?php
$mailRecipient = "$CMS_VALUE(recipient)$";
if($_POST['lastname'])
{
    $mailText = 'Das Formular wurde abgesendet von: '
    . $_POST['firstname'] . '
    . $_POST['lastname'];
    mail($mailRecipient, "Neuer Formular-Submit",
    $mailText);
}
?>
```

Prinzipiell stehen auf diese Weise sämtliche innerhalb des Webservers verfügbaren PHP-Funktionen zur Verfügung. Wie bereits weiter oben beschrieben, wurde zugunsten der Anschaulichkeit auf eine Überprüfung der Parameter und eine sinnvolle Fehlerbehandlung weitestgehend verzichtet. In einem Produktivbetrieb sollten allerdings die gleichen Grundsätze zur Entwicklung robuster Webanwendungen gelten, die auch ohne eine Integration mit FirstSpirit zum Einsatz kämen. Nachdem das Template angepasst wurde, kann erneut der Deployment-Auftrag ausgeführt und die Mailfunktion des Formulars im Browser getestet werden.

## Beste Bücher für besten Code!



Alexander Schulze

### RIA Best Practices vom Core bis zum Desktop

583 Seiten, Hardcover

49,90 €

ISBN 978-3-86802-000-7

Rich-Internet-Applikationen (RIA) werden als die nächste Generation von Softwareanwendungen angesehen. Sie vereinigen die clientseitigen Vorteile verbesserter Usability und höherer „Intelligenz“ ohne jeglichen Installationsaufwand, mit geringerem Kommunikationsaufwand und niedrigeren Kosten für Hardware, Wartung und Infrastrukturen auf der Serverseite. Das Buch stellt Best Practices für alle Ebenen einer RIA vor. Lernen Sie, wie Sie sichere und performante Crossbrowser-kompatible RIAs im Web 2.0 erstellen. Erfahren Sie, wie ein komplettes RIA-Framework aufgebaut ist. Sie werden am Ende in der Lage sein, eine vollständige Anwendung auf einem Live-Server herzustellen.

Weitere Informationen und Bestellmöglichkeiten finden Sie unter [www.entwickler.press.de](http://www.entwickler.press.de). Unsere Bücher erhalten Sie auch in jeder gut sortierten Buchhandlung.

## Hinweise für die Entwicklung von Templates mit PHP-Code

In der vorgestellten Seitenvorlage wurden die grundlegenden Möglichkeiten zur Einbettung von PHP-Code in FirstSpirit-Templates dargestellt. Auf diese Weise lassen sich theoretisch beliebige komplexe PHP-Skripte realisieren. Bei einem konkreten Projekteinsatz sollten allerdings einige Besonderheiten beachtet werden, die sowohl den Komfort bei der Entwicklung als auch die Robustheit der resultierenden Anwendung beeinflussen.

Im Verlauf des Beispiels wurde bereits deutlich, dass zum Testen einer Änderung des im Template eingebetteten Skriptcodes ein erneutes Deployment des Projekts erforderlich wird. Gerade bei größeren Projekten kann dies allerdings einen erheblichen Zeitaufwand darstellen. Weiterhin erfolgt die Bearbeitung des im Template-Code vorhandenen Skripts immer über die Oberfläche des FirstSpirit-Clients, der

nicht zwangsweise auf die Bearbeitung von PHP-Code ausgelegt ist. Um diese beiden Punkte zu umgehen, sollte der innerhalb des Templates vorhandene PHP-Code möglichst reduziert und die eigentliche Funktionalität in Bibliotheken ausgelagert werden, die dann mittels *include*-Statements in das Template eingebunden werden können.

Zum Testen von kleinen Änderungen mag es auf den ersten Blick ebenfalls sinnvoll erscheinen, direkt die generierten Dateien zu bearbeiten. Dabei ist allerdings zu beachten, dass diese spätestens bei der nächsten Generierung aus dem ECMS heraus überschrieben werden. Gerade bei von mehreren Entwicklern genutzten Redaktionssystemen werden auf diese Weise schnell Änderungen überschrieben, deren Fehlen eventuell erst später deutlich wird. Im Zuge des Beispiels wurde die gesamte Funktionalität innerhalb eines einzigen Seiten-Templates umgesetzt. Häufig ist es allerdings sinnvoller, einzelne Funk-

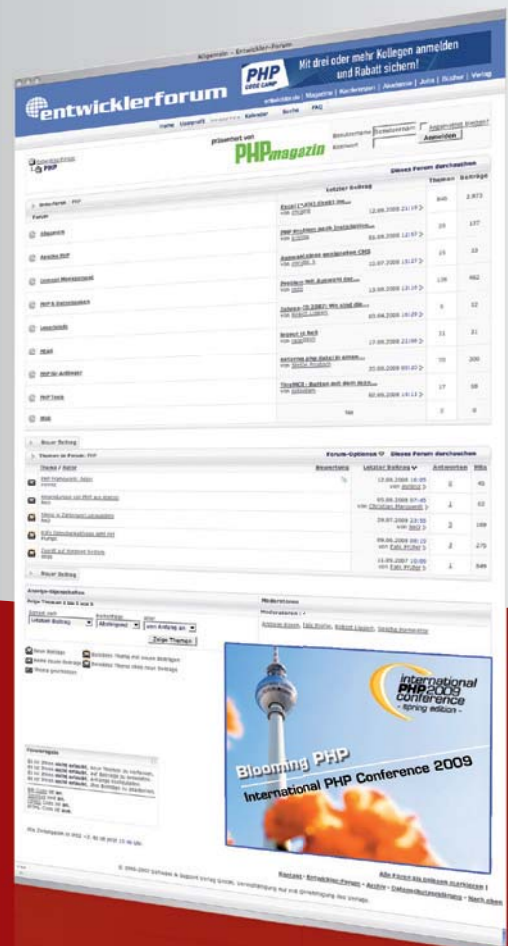
tionen innerhalb von Absatz-Templates und Absätzen, die vom Redakteur innerhalb einer Seite platziert werden können, umzusetzen. Auch hier ist es problemlos möglich, innerhalb des Absatz-Templates PHP-Code einzufügen. Zu beachten ist dabei allerdings, dass dieser nicht mehr isoliert betrachtet werden, sondern durch die Verwendung globaler Variablen zu Konflikten mit anderen Absätzen führen kann. Dies liegt darin begründet, dass FirstSpirit den für die Absätze generierten Code lediglich in die Seite einfügt und ihn nicht voneinander abgrenzen kann.

Auch wenn durch den Redakteur mehrere Absätze mit dem gleichen Absatztyp innerhalb einer Seite verwendet werden, ist Vorsicht geboten. Werden innerhalb des im Absatz-Templates eingebetteten Codes Funktionen oder Klassen definiert, so führt dies zu einer doppelten Definition und daraus resultierend zu einer PHP-Fehlermeldung. Insofern sollten derartige Definitionen,

Anzeige

# Darf's ein bisschen mehr sein?

# PHP-Forum



wie oben bereits angesprochen, in Bibliotheken ausgelagert und dann mittels `include_once` hinzugefügt werden.

Eine weitere Schwierigkeit bei der Verwendung von Absatzvorlagen besteht darin, dass üblicherweise innerhalb der umgebenden Seitenvorlage bereits HTML-Output generiert wird. Dies bedeutet, dass zu dem Zeitpunkt, in dem der Code der Absatzvorlage ausgeführt wird, bereits Output gesendet wurde und somit eine Modifikation von Headern nicht mehr möglich ist. In einem solchen Fall sollte in der Seitenvorlage im PHP-Code Output-Buffering aktiviert werden, um für den Code der Absatzvorlage die Möglichkeit offen zu halten, Header setzen zu können.

Die Generierung und Veröffentlichung erfolgt im vorliegenden Beispiel lediglich auf einen separaten Webserver. Dies hat zur Folge, dass die in FirstSpirit integrierte Vorschaufunktion nicht zur Verfügung steht. Für diese Vorschau werden die generierten Seiten innerhalb eines mitgelieferten Jetty-Application-Servers dargestellt. Damit dieser auch den in die Seiten integrierten PHP-Code interpretiert, muss dessen Konfiguration angepasst werden. Hierbei wird über ein spezielles Servlet PHP im CGI-Modus ausgeführt. Ein guter Ausgangspunkt für die notwendige Konfiguration findet sich unter [2].

## Alternative Architekturen

Die Einbettung von PHP-Code innerhalb der Seiten- und Absatzvorlagen bietet eine einfache Möglichkeit, interaktive Funktionalitäten unter Verwendung des FirstSpirit CMS umzusetzen. Durch sinnvolles Auslagern in Bibliotheken können auch umfangreichere Anwendungen umgesetzt und ein Großteil des Quellcodes kann auf traditionelle Weise inkl. Verwendung von Sourcecontrol-Systemen wie SVN oder Tools wie Unit-Tests entwickelt werden. Nichtsdestotrotz verbleibt ein Teil des ausführbaren Codes innerhalb des Content-Management-Systems und entzieht sich diesen Werkzeugen der Entwicklung. Ist auch hier ein größeres Maß an Kontrolle erforderlich bzw. erscheint das Vorhalten von ausführbarem Code innerhalb des CMS aufgrund der Rahmenbedingun-

gen des Projekts nicht sinnvoll, so kann eine andere Art der Architektur gewählt werden. Diese lässt sich folgendermaßen skizzieren:

Über das Template-System von FirstSpirit werden lediglich statische Inhalte generiert und es wird keinerlei PHP-Code ausgegeben. Stattdessen wird über die Konfiguration des Webservers (beispielsweise durch die Verwendung von URL-Rewriting) bei Aufruf einer der statischen Seiten ein Pre- oder Post-processing durch ein unabhängig vom CMS verwaltetes PHP-Skript durchgeführt. Dieses kann dann die notwendigen interaktiven Funktionalitäten bereitstellen. Es ist dabei allerdings der zusätzliche Entwicklungsaufwand und die durch die zusätzliche Abstraktionsebene höhere Komplexität gegenüber einer direkten Einbettung des Codes zu beachten.

Je umfangreicher und komplexer jedoch die Interaktion mit dem Besucher auf der Webseite wird, desto eher bietet sich ein derartiges Architekturkonzept an.

## Links & Literatur

- [1] Webseite des CMS FirstSpirit:  
[www.firstspirit.de/](http://www.firstspirit.de/)
- [2] Konfiguration des Jetty-Application-Servers für PHP: <http://docs.codehaus.org/display/JETTY/jetty+and+PHP>



Jan Eickmann

Jan Eickmann arbeitet als Berater und Konzepter bei der kernpunkt GmbH, einer in Köln ansässigen Internetagentur. Im Bereich der Entwicklung ist er insbesondere verantwortlich für die Planung von Softwarearchitekturen und Systemkonzepten. Dabei kommen vor allem objektorientiertes PHP und XML-basierte View-Technologien zum Einsatz.

## Verlag:

Software & Support Verlag GmbH

## Anschrift der Redaktion:

PHP Magazin  
Software & Support Verlag GmbH  
Geleitsstr. 14  
60599 Frankfurt am Main  
Tel.: +49(0) 69 630089-0  
Fax: +49(0) 69 630089-89  
E-Mail: [redaktion@phpmagazin.de](mailto:redaktion@phpmagazin.de)  
Web: [www.phpmagazin.de](http://www.phpmagazin.de)

**Chefredaktion:** Björn Schotte  
**E-Mail:** [bjoern@phpmagazin.de](mailto:bjoern@phpmagazin.de)

**Redaktion:** Robert Lippert  
**E-Mail:** [rlippert@phpmagazin.de](mailto:rlippert@phpmagazin.de)

**Chefin vom Dienst:** Nicole Bechtel  
**Schlussredaktion:** Nicole Bechtel, Katharina Klassen, Frauke Pesch

**CD/DVD-Erstellung:** Daniel Zuzek  
**E-Mail:** [cd@phpmagazin.de](mailto:cd@phpmagazin.de)

**Leitung Grafik/Produktion:** Jens Mainz  
**Layout, Titel:** Daniela Albert, Kristin Brockmann, Pobjorn Fischer, Melanie Hahn, Karolina Kresic, Katharina Ochsenhirt, Lena Reichwein, Maria Rudi, Patricia Schwesinger

## Autoren dieser Ausgabe:

Andreas Bohne-Lang, Christian Effenberger, Ralf Eggert, Jan Eickmann, Timo Haberkern, Markus Hasenbein, Daniel Koch, Nils Langner, Alexander Merz, Daniel Nitz, Kore Nordmann, David Soria Parra, Tobias Schlitt

## Anzeigen

Verlagsbüro Ohm-Schmidt  
Herr Osmund Schmidt  
Schneckenburgerstr. 22  
30177 Hannover  
Tel.: +49-(0)511-2354164  
Fax: +49-(0)1805-060-336-956-69  
E-Mail: [Osmund@Ohm-Schmidt.de](mailto:Osmund@Ohm-Schmidt.de)  
Es gilt die Anzeigenpreisliste Nr. 7

## Pressevertrieb:

DPV Network  
Tel.: +49 (0) 40 378456261  
Web: [dpv-network.de](http://dpv-network.de)

## Aboservice

Software & Support Verlag GmbH  
Tel.: +49 (0) 69 630089-0  
Fax: +49 (0) 69 630089-89  
E-Mail: [abo@phpmagazin.de](mailto:abo@phpmagazin.de)

## Abonnementpreise

Inland	6 Ausgaben	52,00 €
Ausland	6 Ausgaben	62,00 €
Student Inland	6 Ausgaben	41,60 €
Student Ausland	6 Ausgaben	51,60 €

## Einzelverkaufspreis

Deutschland	9,80 €
Österreich	10,80 €
Luxemburg	11,25 €
Schweiz	19,20 sFr

**Erscheinungsweise:** zweimonatlich

**Druck:** PVA, Landau  
**ISSN:** 1619-7976

© 2009 für alle Beiträge.

Alle Rechte, auch für Übersetzungen, sind vorbehalten. Reproduktionen jeglicher Art (Fotokopie, Nachdruck, Mikrofilm oder Erfassung auf elektronischen Datenträgern) nur mit schriftlicher Genehmigung des Verlags. Jegliche Software auf der CD-ROM unterliegt den Bestimmungen des Herstellers oder der zuständigen Organisation. Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden. Honorierte Artikel gehen in das Verfügungsrecht des Verlags über. Mit der Übergabe der Manuskripte, Abbildungen eventueller Quellcodes an den Verlag erteilt der Verfasser dem Herausgeber das Exklusivitätsrecht zur Veröffentlichung. Für unverlangt eingesandte Manuskripte, Abbildungen oder Quellcodes keine Gewähr.

Alle Markennamen sind in der Regel eingetragene Warenzeichen der entsprechenden Hersteller oder Organisationen.