



Alex Kempkens

Das Joomla! Entwicklerhandbuch

Joomla!-Komponenten und -Templates
programmieren mit dem Joomla!-Framework



3 Die Arbeitsumgebung für Joomla! einrichten

Nach der Einführung in das Thema Open-Source-Software im Allgemeinen und in das Projekt Joomla! im Speziellen ist es an der Zeit, mit der praktischen Arbeit anzufangen. Hierzu ist es zunächst einmal erforderlich, dass wir eine Arbeitsumgebung einrichten, die unseren Anforderungen gerecht wird.

Wir haben den Joomla!-spezifischen Teil der Architektur bereits besprochen und ergänzen dieses Szenario nun mit den für die Nutzung von Joomla! erforderlichen Servern. Für den Einstieg kümmern wir uns aber nur um den Aufbau einer einfachen Entwicklungsumgebung und den dazugehörigen Entwicklungswerkzeugen. Detaillierte Informationen zum hier gegebenen Überblick bekommen Sie im Kapitel 8, *Weitere Informationen zu den Technikgrundlagen*. Im Kapitel 8.2, *Konfiguration eines Produktivservers*, gehen wir auch auf die Besonderheiten der Serversysteme im Produktivbetrieb ein.

3.1 Überblick über die Arbeitsumgebung

Bevor wir in die Details der neuen Arbeitsumgebung einsteigen, möchte ich Ihnen diese in einem kurzen Überblick vorstellen. Hierbei ist wichtig, dass wir für die Arbeitsumgebung fast alle Elemente benötigen, die Sie zukünftig auch in einer produktiven Umgebung einsetzen. So werden wir unter anderem die folgenden Dienste, Werkzeuge und Programme brauchen:

- Webbrowser mit Entwicklungsfunktionen für Debugging und CSS-Bearbeitung
- Grafikprogramm
- Integrierte Entwicklungsumgebung (IDE)
- Programm zur Versionsverwaltung
- Programm zur Administration von Datenbanken
- Serversysteme/-dienste
 - Webserver
 - Datenbankserver
 - PHP-Erweiterung für den Webserver

Da wir nicht vorhaben, unseren Arbeitsplatz als Webserver für andere bereitzustellen, sind Dienste wie z.B. FTP oder SSH nicht erforderlich. Auch werden wir uns bei

den Serversystemen auf das absolute Minimum konzentrieren und keine besonderen Konfigurationen oder speziellen Dienste nutzen, aber dazu später mehr.

Das folgende Schema zeigt, wie die einzelnen Dienste und Werkzeuge miteinander zusammenhängen bzw. zusammenarbeiten.

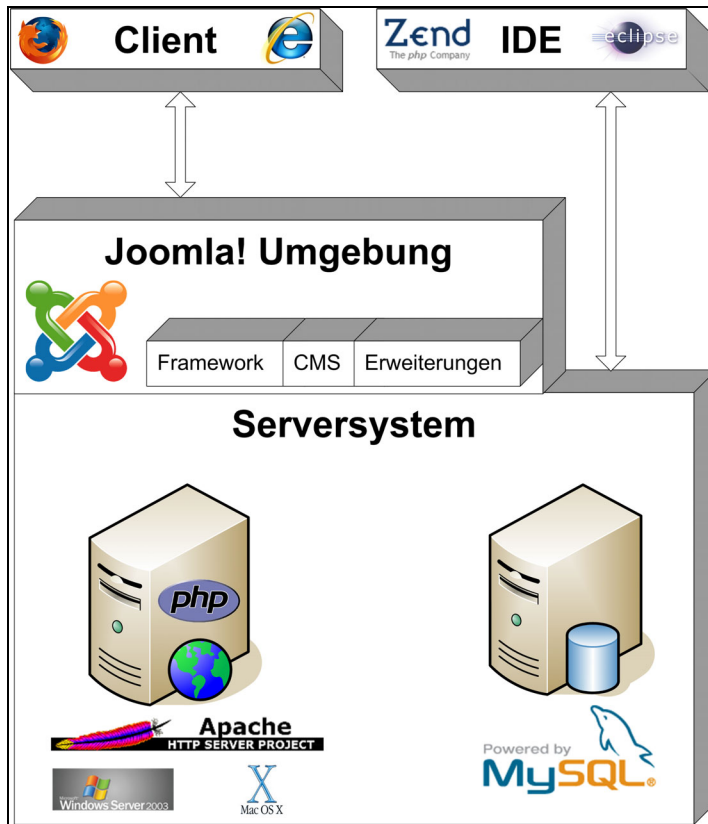


Abbildung 3.1: Dienste und Werkzeuge der Arbeitsumgebung

Beginnen wir auf der Seite der Anwendungen und Clients. Die Browser bzw. die Entwicklungsumgebung stellen dort eigenständige Anwendungen dar, welche wir je nach den persönlichen Wünschen auswählen und installieren können. Ich selber entwickle auf einem Mac OS X-System und verwende eine erweiterte Version der Eclipse IDE. Das Manuskript zu diesem Buch entstand z. T. auf Windows Vista unter der Verwendung von Internet Information Server, IE7 und der Zend Studio Entwicklungsumgebung. Da das Ergebnis zählt, ist die eigentliche Entwicklungsumgebung eher sekundär – probieren Sie einfach aus, womit Sie zu den besten Ergebnissen kommen. Jedes Produkt hat hier sicher seine Vor- und Nachteile. In den nachfolgenden Beispielen stelle ich verschiedene Produkte und deren Installation vor (setzen Sie ruhig aktuellere Versionen ein, wenn diese in der Zwischenzeit erschienen sein sollten). Für ein Linux-basiertes System (z. B. openSuSE, centos oder Ubuntu) können Sie die Entwick-

lungsumgebung mit den meist auf den Distributions-CDs/-DVDs mitgelieferten Paketen direkt installieren. Eine spezielle Installation bzw. der Einsatz zusätzlicher Pakete ist dort in der Regel nicht erforderlich.



Exkurs: Im Buch verwendete Systemkonfiguration

Alle Beispiele im Buch basieren auf den folgenden Systemkonfigurationen:

Microsoft Windows	Apple
Windows Vista Business	Mac OS X 10.5 (Leopard)
Internet Information Server (IIS) 7.0	Apache 2.0
PHP 5.2.1	PHP 5.2.1
MySQL 5.0	MySQL 5.0
Zend Studio 5.5	Zend Studio for Eclipse

Tabelle 3.1: Übersicht der Systemkonfiguration aus dem Buch

Andere Konfigurationen als diese führen meist nur zu unterschiedlichen Vorgehensweisen, z. B. beim Zugriff auf bestimmte Dateien oder bei der Aktivierung bestimmter Dienste. Auch Konfigurationsdateien können unterschiedlich aussehen.

Für die Zugriffe von der IDE auf unsere Serverdaten sind in der Arbeitsumgebung keine besonderen zusätzlichen Pakete erforderlich. Wir können direkt über das Dateisystem auf die Programmquellen zugreifen und diese direkt nutzen. Die Joomla!-Umgebung installieren wir im Rahmen unserer Serversysteme, da sie einen Teil dieser Umgebung darstellt. Die darin enthaltenen Softwarekomponenten werden wir meist nur über die Webbrowser nutzen und ansprechen.

Bei den Servern ist die Aufteilung eher organisatorischer als physikalischer Art. Wir werden auf einem Computer sowohl einen Webserver als auch einen Datenbankserver installieren. Als Referenz zeigen wir die Installation und Integration wieder auf den beiden Plattformen – diesmal mit einem Apache Server und einem IIS. Für beide Webserver wird zusätzlich PHP 5 installiert und integriert. Die jeweiligen Konfigurationen lassen sich bei Bedarf auch auf andere Betriebssysteme übertragen. Die Datenbank ist bei allen Plattformen MySQL 5.

3.2 Werkzeuge und Produkte für die Arbeitsumgebung

Für den Einsatzzweck im Rahmen eines Entwicklungsarbeitsplatzes gibt es eine Vielzahl verschiedener Lösungen. Um Ihnen ein Gefühl für die Möglichkeiten zu geben, habe ich in der nachfolgenden Übersicht einige Werkzeuge und Produkte ausgewählt und werde die Besonderheiten kurz ansprechen. Sicherlich ist diese Liste nicht vollständig, und der einzelne mag ein anderes Produkt bevorzugen. Dies ist grundsätz-

lich immer möglich, und die individuelle Erfahrung sollte auch stets den Ausschlag für eine entsprechende Auswahl geben. Wir werden nach der Übersicht sowohl die Grundinstallation für Windows als auch für Apple Mac OS X besprechen.

Ein weiterer Aspekt ist die Effektivität der einzelnen Produkte. Ich habe versucht, in der Tabelle die Produkte etwas zu bewerten. Sicher sind diese Bewertung subjektiv, vielleicht geben sie Ihnen aber Anhaltspunkte, um zu den für Sie richtigen Schlüssen zu gelangen.

Softwarepaket	Mögliche Produkte / Bemerkung
Webbrowser	<ul style="list-style-type: none"> ■ Firefox (http://www.mozilla-europe.org) ■ IE 7 (http://www.microsoft.de) ■ Safari (http://www.apple.com/de/) ■ Opera (http://www.opera.com/) <p>Grundsätzlich sind alle Browser für die Entwicklung geeignet, und es ist sogar empfehlenswert, einmal mit allen Browsern die erstellte Site anzusehen. Für die Entwicklung eignet sich derzeit Firefox am besten, da die Extensions für Firefox die Entwicklung und Feinabstimmungen der Sites sehr vereinfachen. Sie können die einzelnen Addons finden, indem Sie die Nummer in Klammern am Ende der URL <a href="https://addons.mozilla.org/de/firefox/addon/<Nummer>">https://addons.mozilla.org/de/firefox/addon/<Nummer> eintragen. Zu den empfohlenen Erweiterungen gehören:</p> <ul style="list-style-type: none"> ■ HTML Validator (887) Der HTML Validator zeigt auf einfache Weise an, ob der erstellte HTML-Code valide ist. Ggf. werden Erklärungen und Vorschläge zur Korrektur dargestellt. ■ Firebug (1843) Firebug ist eine geniale Erweiterung, um alle Arten von CSS- oder JavaScript-Code innerhalb einer Website zu überwachen und on the fly zu ändern. Mit Firebug ist es auch möglich, komplexe AJAX-Skripte leicht zu debuggen und somit Fehler schnell zu beheben. ■ Web Developer (60) Diese Erweiterung ist eines der sehr umfangreichen und effektiven Tools für die Entwicklung von HTML-Inhalten. Mithilfe von Web Developer können Sie nicht nur automatisch CSS-Inhalte ausblenden oder die Anpassung des Browsers an Bildschirmgrößen optimieren, sondern eigentlich jeden Teil der Site kontrollieren.
Grafikprogramm	<p>Die Grafikprogramme können Sie nach eigenen Wünschen und Präferenzen auswählen. Aus Sicht von Joomla! gibt es hier keine Empfehlungen. Für einige Grafikprogramme sind heute bereits Vorlagen bzw. Unterstützungswerkzeuge zum Design und der Erstellung von Templates verfügbar. Hilfreicher sind da aber meist die verfügbaren Tutorials (z.B. auf http://help.joomla.org, http://dev.joomla.org/downloads/Joomla15TemplateTutorial.zip)</p>

Tabelle 3.2: Übersicht der Pakete für die Entwicklungsumgebung

Softwarepaket	Mögliche Produkte / Bemerkung
Integrierte Entwicklungs-umgebung	<ul style="list-style-type: none"> ■ EasyEclipse for PHP (http://www.easyeclipse.org/site/distributions/php.html) ■ Eclipse mit PDT (http://www.eclipse.org/pdt/) ■ Zend Studio bzw. Zend Studio für Eclipse (http://www.zend.com/de/products/zend_studio) <p>Die Integrierten Entwicklungsumgebungen haben den Vorteil, dass sie Funktionen wie z. B. die des Debuggers, die Syntax-Hervorhebung und PHP-Hilfestellungen mit integrieren. Auch werden die Versionsverwaltung und andere Elemente der Entwicklung in eine Oberfläche integriert, was die Arbeit erleichtert. Es gibt noch weitere IDE-Lösungen, und letztlich ist die Entscheidung von Ihren persönlichen Erfahrungen und Wünschen abhängig. Ich empfehle, zu Anfang die Testversionen auszuprobieren und dann ggf. eine Entscheidung zu treffen.</p> <p>Zend Studio ist eine kommerzielle Lösung, welche aber durch die sehr gute Integration und den günstigen Preis (die Standard Edition gibt es ab ca. 99,- EUR) für ambitionierte Entwickler empfehlenswert ist. Die Variante Zend Studio für Eclipse basiert auf der sehr weit verbreiteten Java-Entwicklungsumgebung Eclipse, die unter anderem von IBM und TogetherSoft mitentwickelt wird. Zend hat sich an dem PHP-Developer-Tools (PDT)-Projekt beteiligt und im Rahmen der <i>Zend Studio für Eclipse</i>-Version diese Variante noch weiter verbessert. Das Projekt PDT ist als Erweiterung für Eclipse kostenlos verfügbar. Alle von Zend direkt gelieferten Versionen sind kostenpflichtig und mit speziellem Support ausgestattet.</p>
PHP-Editoren	<ul style="list-style-type: none"> ■ PHPEd (http://www.soysal.com/PHPEd) ■ phpEdit (http://www.phpedit.com) ■ Ultra Edit (http://www.ultraedit.com) ■ skEdit (http://www.skti.org) – nur für Mac OS X <p>Die Editoren sind eine Alternative zu den integrierten Entwicklungsumgebungen. Wenn Sie nur seltener in den Quellcode schauen müssen, sind diese Editoren wesentlich kleiner und einfacher im Umgang. PHPEd z. B. ist ein Freeware-Produkt, welches alle wichtigen Funktionen für die Bearbeitung der Dateien enthält und für unsere Zwecke grundsätzlich durchaus geeignet ist.</p>
Clients für die Versionsverwaltung	<ul style="list-style-type: none"> ■ TortoiseSVN (http://tortoisesvn.tigris.org/) ■ SCPlugin (http://scplugin.tigris.org/) ■ Syncro SVN (http://www.syncrosvnclient.com/) <p>Innerhalb der meisten Open-Source-Projekte wird heute die Versionsverwaltung Subversion (http://subversion.open.collab.net/) eingesetzt. Dieses Serverprogramm erfordert die Installation eines Clients auf Ihrem lokalen System. Die beiden ersten Projekte sind freie Lösungen für die Integration der Subversion-Funktionalität in den Explorer bzw. Finder. Syncro SVN ist eine Anwendung, welche die Arbeit mit vielen Versionsverzeichnissen weiter unterstützt.</p>

Tabelle 3.2: Übersicht der Pakete für die Entwicklungsumgebung (Forts.)

Softwarepaket	Mögliche Produkte / Bemerkung
Datenbank Administration	<ul style="list-style-type: none"> ■ MySQL (http://www.mysql.com) ■ phpMyAdmin (http://www.phpmyadmin.net) <p>Für die Pflege der Datenbankstruktur und Inhalte ist es empfehlenswert, einen flexiblen Administrationsclient zu installieren. phpMyAdmin ist bei den nachfolgend genannten Komplettpaketen enthalten und sehr einfach über das Web zu bedienen. Die Client-Werkzeuge von MySQL haben den Vorteil, dass weitere Funktionen und Informationen über den Server dargestellt und Optimierungen so manchmal einfacher gefunden werden können.</p>
Serverumgebungen	<ul style="list-style-type: none"> ■ XAMPP (http://www.apachefriends.org/de/xampp.html) ■ MAMP (http://www.mamp.info) ■ Zend Core (http://www.zend.com/de/products/zend_core) ■ MySQL (http://www.mysql.com) ■ PHP (http://www.php.net) <p>Die genannten Umgebungen sind Lösungen, die alle erforderlichen Komponenten (Webserver, Datenbank und PHP-Erweiterung) in einem Paket integrieren. Jeweils für Windows (XAMPP) oder Mac OS X (MAMP) können Sie so über eine einfache Installation alle für den lokalen Betrieb erforderlichen Dienste installieren. Die genannten Pakete enthalten einen Apache Webserver mit PHP 4- bzw. 5-Erweiterung, die MySQL-Datenbank und zusätzliche kleine Tools wie phpMyAdmin. Für die einfache Installation sind diese Umgebungen ausreichend, und wir können sie einfach für verschiedene Bereiche anpassen. Der Zend Core ist eine kleinere Version der Zend-Plattform, die auch den Apache Server und eine entsprechende PHP-Konfiguration enthält. Die Datenbank muss hier zusätzlich installiert werden. Gerade im Zusammenspiel mit dem Zend Studio ist Zend Core aber eine sehr effektive Lösung.</p> <p>In der Windows-Umgebung werden wir den eingebauten IIS in Verbindung mit Zend Core für PHP mit einer eigenen installierten MySQL-Datenbank nutzen. Grundsätzlich sind diese Varianten für Joomla! aber transparent, d.h. es spielt aus Sicht von Joomla! keine Rolle, ob Sie die eine oder die andere Variante wählen.</p>
Debugger	<ul style="list-style-type: none"> ■ dbg (http://dd.cron.ru/dbg/) ■ Xdebug (http://www.xdebug.org/) <p>Die Nutzung effektiver Debugger ist bei einer Skriptsprache wie PHP sehr wichtig. Leider ist dies auch ein sehr komplexes Thema und erfordert gerade bei der Installation und Konfiguration einiges Geschick. Für Xdebug wurde kürzlich mit der Version 2.0 ein stabiler Code veröffentlicht. Leider stehen Binärdateien nur für Windows-Systeme zur Verfügung, so dass eine Installation auf Linux- oder Mac OS X-Systemen ein individuelles Kompilieren erfordert. Dies ist nur dann wirklich möglich, wenn sowohl der Apache Server als auch PHP manuell kompiliert werden. Um dies zu vermeiden, bleibt auf allen Plattformen derzeit eigentlich nur die Zend Core-Umgebung. Diese lässt sich auch nachträglich ohne manuelle Neuerstellung installieren. Wir werden diese daher auf beiden Betriebssystemen einsetzen.</p>

Tabelle 3.2: Übersicht der Pakete für die Entwicklungsumgebung (Forts.)

3.3 Installation und Konfiguration der Serverdienste

Für den nun folgenden Aufbau der eigenen Entwicklungsumgebung können Sie meist die jeweiligen Standardinstallationen durchführen. Ich konzentriere mich bei meiner Beschreibung insbesondere auf die abweichenden Details. Die Installation der Pakete ist eigentlich unkritisch und wird auf einem Arbeitsplatz keine besonderen Probleme verursachen, dennoch ist ein Backup immer ratsam.

Für unsere ersten Entwicklungsschritte werden wir die nachfolgend konfigurierten Standardverzeichnisse für Webdokumente nutzen. Die im Laufe der Beschreibung angesprochene erweiterte Konfiguration verschiedener virtueller Webserver erläutern wir ausführlich in Abschnitt 8.2.1, *Webserver und virtuelle Hosts*.

3.3.1 Grundinstallation unter Windows

Die nachfolgend beschriebene Installation auf Windows wurde unter Windows Vista Business und unter Verwendung des integrierten Internet Information Servers (IIS) durchgeführt. Als PHP-Umgebung kommen Zend Core und das Zend Studio zum Einsatz.

1. Installation und Einrichtung des Internet Information Servers

Der Webserver von Microsoft wird bei jedem Betriebssystem grundsätzlich in einer kleineren Version mitgeliefert. Für die Entwicklungsumgebung müssen wir sicherstellen, dass der Server eingerichtet und aktiviert ist.

Für die Installation des IIS benutzen Sie die Standard-Softwareinstallation **START → SYSTEMSTEUERUNG → PROGRAMME UND FUNKTIONEN**. In dem Dialog wählen Sie bitte **WINDOWS-FUNKTIONEN EIN- ODER AUSBLENDEN**, um die in Abbildung 3.2 abgebildete Darstellung zu erhalten. Für einen reibungslosen Betrieb des IIS als Entwicklungssystem aktivieren Sie bitte unter **INTERNETINFORMATIONSDIENSTE** die **WEBVERWALTUNGSTOOLS, WWW-DIENSTE**. Damit haben sie die zentralen Features bereits freigeschaltet. Zusätzlich aktivieren Sie bitte unter **ANWENDUNGSENTWICKLUNGSFEATURES CGI, ISAPI-Erweiterungen, ISAPI-FILTER**. Die letzten ermöglichen die Integration von PHP als Fast-CGI oder Modul je nach späterem Bedarf. Mit einem Klick auf **OK** wird die Installation durchgeführt. Nach der abgeschlossenen Installation läuft der Webserver bereits. Wenn Sie jetzt in Ihrem Browser *http://localhost* aufrufen, sollten Sie die in Abbildung 3.3 gezeigte Website sehen.

Der IE zeigt unterhalb der Tabs einen Warnhinweis für die Nutzung von Intranetinhalten an. Aktivieren Sie die Nutzung der Intranetinhalte, um zukünftig einfacher mit diesen Seiten arbeiten zu können.

2. Einrichtung der PHP-Umgebung

Die PHP-Umgebung kann einfach durch die Installation des kostenlos verfügbaren Zend Core¹ eingerichtet werden. Zend Core ist die Basisversion der Zend Platform bzw. weiterer Zend-Produkte. Diese zusätzlichen Produkte sind kostenpflichtig.

¹ http://www.zend.com/de/products/zend_core

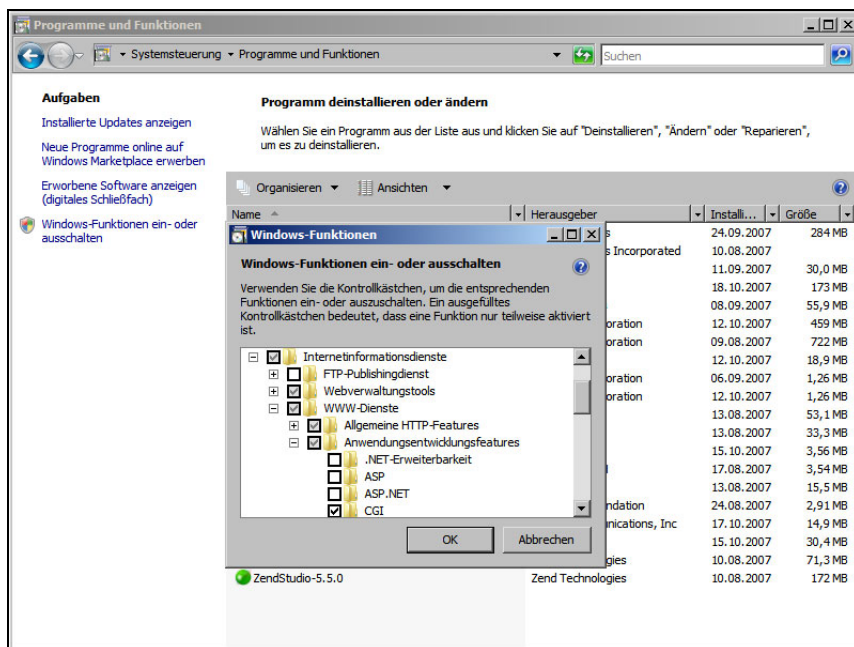


Abbildung 3.2: Installation des IIS auf Windows Vista

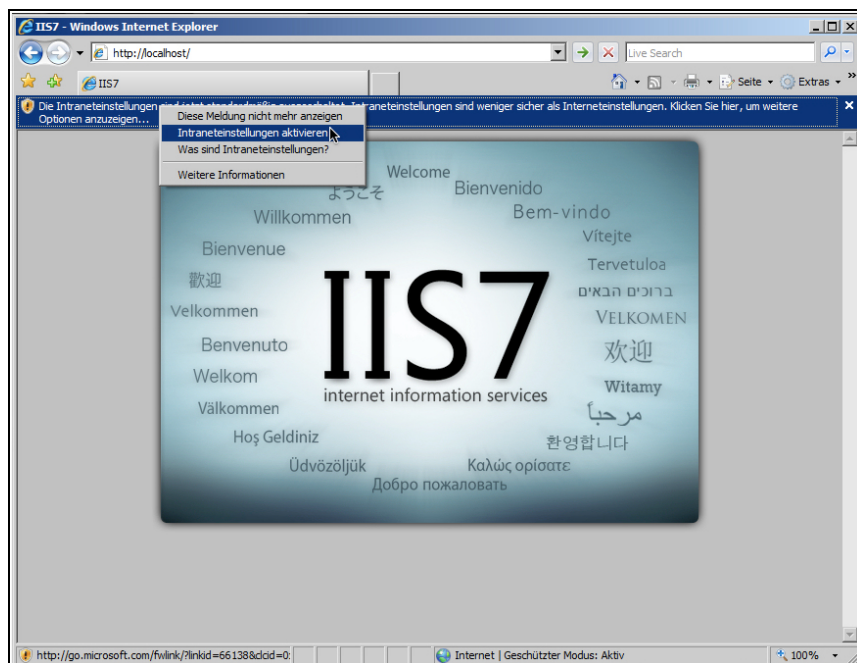


Abbildung 3.3: Begrüßungsseite des IIS nach der erfolgreichen Installation

Die Installation ist am besten auf Basis der individuellen Installation (*Custom*) durchführbar. Belassen Sie die Features, wie sie vorgeschlagen werden (ZEND CORE PHP AND EXTENSIONS, ZEND CORE ADMINISTRATION), und installieren Sie keinen weiteren Server. Dies machen wir nachträglich manuell. Bei der Auswahl des Web-servers wählen Sie bitte den IIS aus. Alle weiteren Einstellungen können vom Standardwert übernommen werden.

Nach der erfolgreichen Installation können Sie sich an der Zend Core-Administration unter der URL *http://localhost/ZendCore* anmelden. Nach der Anmeldung sollten Sie den in Abbildung 3.4 zu sehenden Bildschirm sehen. Sie können dort die Einstellungen für Zend überprüfen und entsprechend optimieren.

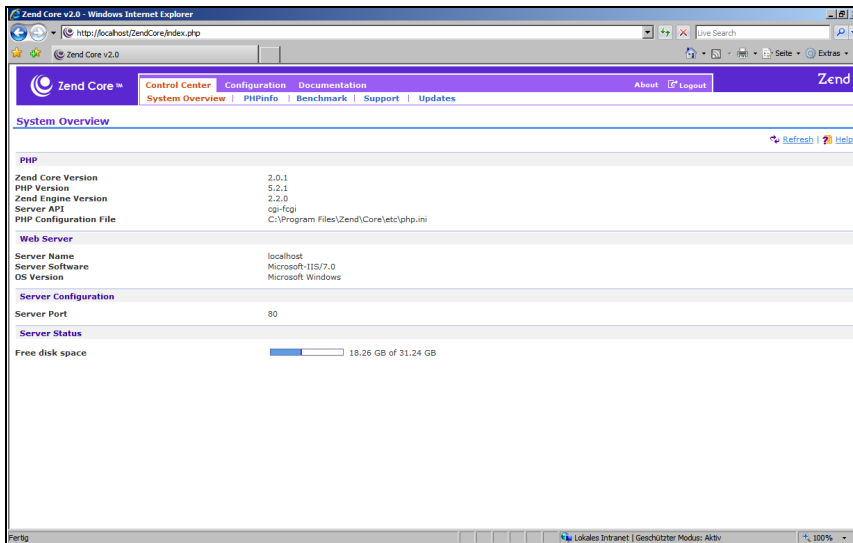


Abbildung 3.4: Administrationsdialog von Zend Core nach der Anmeldung

Damit Joomla! auf dem System optimiert lauffähig ist, werden wir hier ein paar weitere Erweiterungen aktivieren. Nutzen Sie dazu bitte die Konfiguration (CONFIGURATION) → Erweiterungen (EXTENSIONS). In der nachfolgenden Übersicht der für PHP aktivierten Erweiterungen schalten Sie bitte die folgenden Erweiterungen am Schalter (mittleres Icon) ein.

- GD – GD (IMAGE MANIPULATION)
- MBSTRING – MULTIBYTE CHARACTER PROCESSING
- XMLRPC – XML-RPC

Nachdem Sie die Einstellungen gespeichert haben (SAVE SETTINGS), ist ein Neustart des IIS erforderlich. Dies können Sie am einfachsten über den *Dienste Manager* machen. Klicken Sie dazu mit der rechten Maustaste auf COMPUTER (unter XP ARBEITSPLATZ) und wählen Sie dann VERWALTEN.

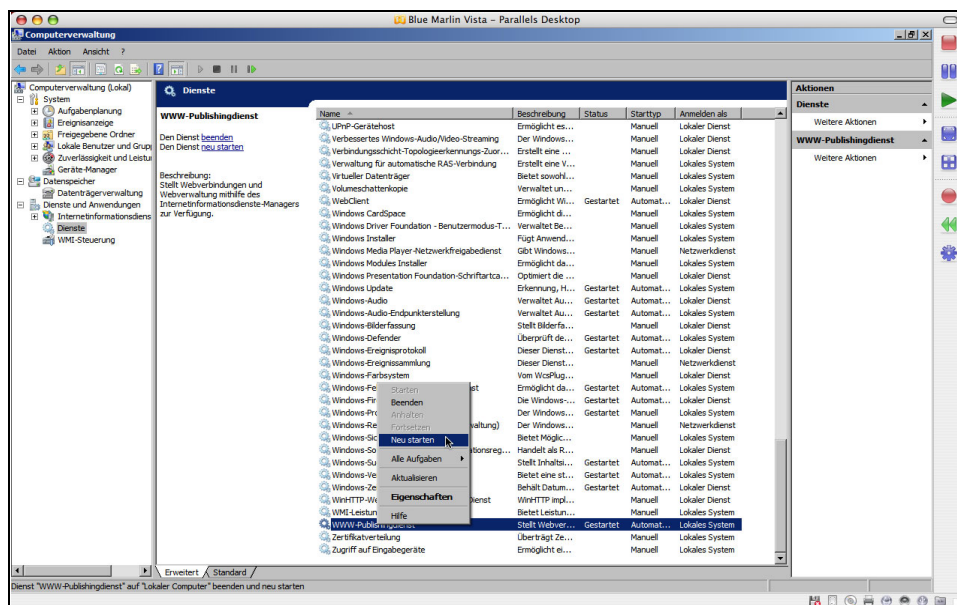


Abbildung 3.5: Dienste in der Computerverwaltung für Neustart des IIS

In der Computerverwaltung (SYSTEMSTEUERUNG\VERWALTUNG) können Sie unter DIENSTE den WWW-PUBLISHINGDIENST finden. Den Neustart dieses Dienstes können Sie mit dem grünen Pfeil in der Toolbar oder über einen Klick mit der rechten Maustaste und Anklicken von NEU STARTEN veranlassen.

3. Installation des Datenbankservers MySQL

Abschließend installieren wir den Datenbankserver und dessen Administrationswerkzeuge auf dem System. Dazu brauchen Sie den Community Server 5.0.², der in dieser Version kostenlos verfügbar ist. Zusätzlich können Sie sich gleich auch die GUI Tools 5.0.³ aus dem Netz laden, diese enthalten eine Benutzeroberfläche für die Administration und Tabellenverwaltung, die wir im Weiteren verwenden werden.

Die Installation des Datenbankservers und der Administrationswerkzeuge erfolgt durch das Ausführen der Setup-Dateien und kann nach den Standardvorgaben erfolgen (TYPICAL INSTALLATION). Am Ende der Serverinstallation erfolgt automatisch die Konfiguration, welche Sie unter Vista wegen der Sicherheitsbeschränkungen erneut aus dem Menü aufrufen müssen. Wählen Sie unter START → MYSQL → MYSQL SERVER 5.0 → MYSQL SERVER INSTANCE CONFIG WIZARD. Wählen Sie bitte die individuelle Konfiguration (DETAILED CONFIGURATION), um entsprechende Einstellungen für Joomla! vorzunehmen. So weit können Sie die Standardwerte übernehmen. Auf der sechsten Seite können Sie die Standardzeichensätze festlegen. Hier empfiehlt es sich, BEST SUPPORT FOR MULTILINGUALISM mit UTF-8-Unterstützung als Standard auszuwählen. Bei den Sicherheitseinstellungen müssen Sie für den Administrator (*root*) ein

² <http://dev.mysql.com/downloads/mysql/5.0.html>

³ <http://dev.mysql.com/downloads/gui-tools/5.0.html>

3.3 Installation und Konfiguration der Serverdienste

Passwort festlegen, welches Sie später wieder brauchen werden. Zum Abschluss wird die Konfiguration gespeichert und der Service gestartet.

Die Installation der GUI Tools (Administration) erfolgt auf die gleiche einfache Art und Weise. Hierbei können Sie die komplette Installation aller Werkzeuge durchführen. Nach der Installation finden Sie im Startmenü unter MySQL zwei neue Programme; MySQL Administrator und MySQL Query Browser. Diese beiden Werkzeuge dienen für die weitere Konfiguration der MySQL-Datenbank.

Rufen Sie bitte zum Test den MySQL Administrator auf und tragen Sie als SERVER HOST: *localhost*, NUTZERNAME: *root* und als PASSWORT das von Ihnen vergebene Passwort ein. Wenn die Anmeldung erfolgreich ist, dann sollten Sie danach die Informationsseite über den gerade eingerichteten Server sehen (siehe Abbildung 3.6).

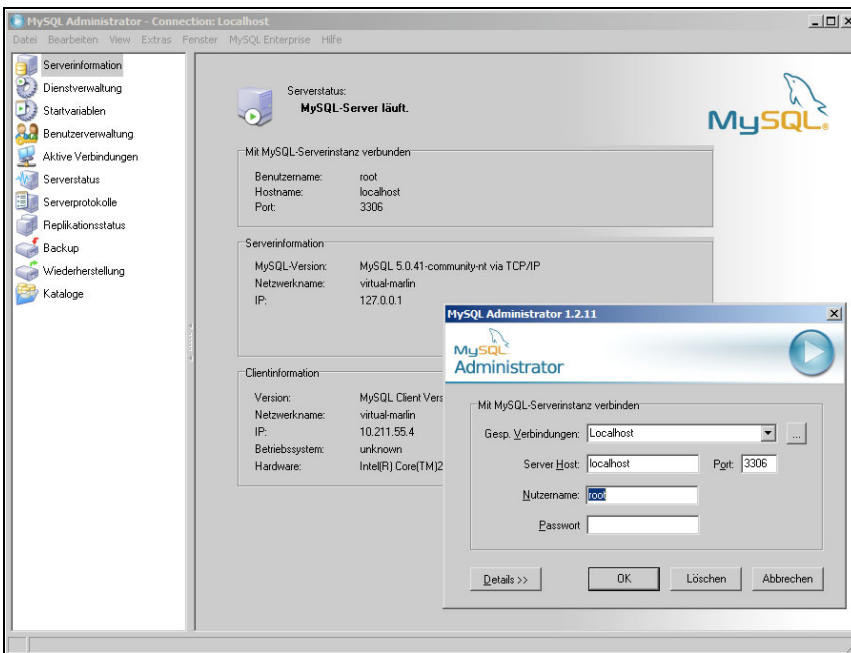


Abbildung 3.6: Informationsseite und Login-Dialog von MySQL Administrator

4. Installation von Joomla!

Grundsätzlich sind jetzt alle erforderlichen Vorarbeiten durchgeführt. Als nächsten Schritt installieren wir zum Test eine lokale Joomla!-Version. Der IIS erstellt während der Installation ein Standarddokumentenverzeichnis unter `C:\inetpub\wwwroot`. Bevor wir dieses Verzeichnis ohne Probleme benutzen können, müssen wir die Rechte auf das Unterverzeichnis etwas vereinfachen. Geben Sie für unsere Entwicklungsumgebung insbesondere dem Benutzer IIS-User vollen Zugriff auf das Verzeichnis und alle Unterverzeichnisse.



Achtung

Es versteht sich von selbst, dass diese Konfiguration ausschließlich für den Entwicklungsbetrieb geeignet ist. Auf einem produktiven System müssen spezielle Rechte vergeben und Sicherheitsvorkehrungen getroffen werden. Der IIS verwendet den oben erwähnten Benutzer als Basis für sein Rechte-System, und über die normalen Dateirechte wird auch der Zugriff auf die Skripte gesteuert.

Löschen Sie nun alle Dateien in dem Web-Root-Verzeichnis und kopieren Sie eine vollständige Joomla!-Distribution in das Verzeichnis. Nach dem Kopiervorgang sollten Sie beim Zugriff auf <http://localhost> die normale Joomla!-Installation sehen und durchführen können (siehe Abbildung 3.7).

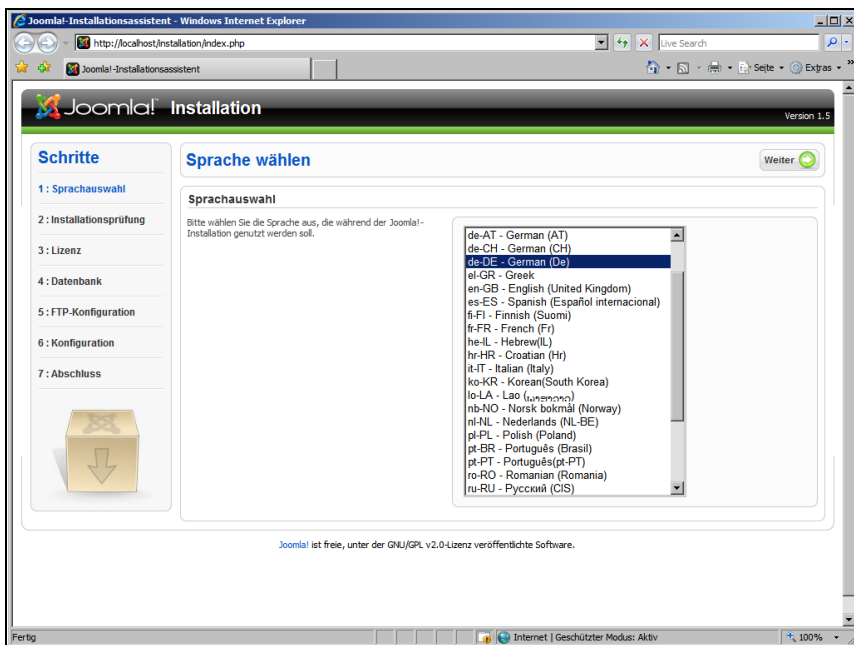


Abbildung 3.7: Installationsbildschirm von Joomla!

3.3.2 Grundinstallation unter Mac OS X

Die Installation für Mac OS X werden wir auf dem bereits erwähnten Paket MAMP aufbauen. Dieses Paket enthält sowohl die erforderlichen Apache- und MySQL-Server als auch zusätzliche Tools für die weitere Konfiguration. Es gibt mittlerweile auch eine Pro-Version dieses Pakets, die weitere Oberflächen und Vereinfachungen erlaubt. Für unsere Zwecke ist die kostenlose Version aber vollkommen ausreichend.

Damit müssen wir zwar für die weitere Entwicklung einige Konfigurationen manuell durchführen, dies ist aber nicht kompliziert.



Achtung

Leider ist die Version des MAMP-Paketes seit seiner Veröffentlichung bis zum Zeitpunkt der Drucklegung dieses Buches nicht weiter aktualisiert worden. Die Pakete stehen noch weiterhin zur Verfügung und können auch genutzt werden. Für einen produktiven Einsatz auf einem Mac-Server sollte aber unbedingt auf die aktuellen Versionen des Webservers und der anderen Komponenten zurückgegriffen werden.



Hinweis

Es sei kurz angemerkt, dass auf allen Mac OS X-Systemen bereits in der Standardkonfiguration ein Apache Webserver mitgeliefert wird. Dieser kann durch das PERSONAL WEB SHARING in den SYSTEMEINSTELLUNGEN aktiviert werden. Der Server ist dann in den Grundverzeichnissen des Systems installiert und kann von dort genutzt werden.

Es empfiehlt sich, diesen Webserver nicht zu nutzen, da er in der Standardkonfiguration erst seit Mac OS X 10.5 »Leopard« die Version PHP 5 unterstützt und auch andere, für Joomla! wichtige Module nicht integriert sind. Die Nutzung eines zweiten Servers hat nur zur Folge, dass dieser entweder auf einem anderen Netzwerk-Port betrieben werden muss oder beim Starten des Servers der Administrator sich authentifizieren muss. Keine der beiden Varianten hat auf die Entwicklung bzw. Nutzung des Servers aber einen direkten Einfluss.

Zusätzlich werden wir den Zend Debugger über die Zend Core-Erweiterung installieren und die Konfiguration von MAMP entsprechend an diese Änderungen anpassen.

1. Installation und Einrichtung von MAMP

Die Installation von MAMP erfolgt nach der für Mac OS X üblichen Vorgehensweise. Wenn Sie die Installation in das allgemeine Programmverzeichnis (PROGRAMME) durchgeführt haben, finden Sie in diesem Verzeichnis ein Unterverzeichnis MAMP mit den installierten Programmen. Von diesen Anwendungen ist MAMP das Programm zur Steuerung und Konfiguration der Serverumgebung. Es gibt auch eine Erweiterung für das Dashboard (MAMP CONTROL.WDGT), die das Starten und Stoppen des Servers sehr einfach über das Dashboard ermöglicht.

Starten Sie bitte MAMP über den Finder. Nach dem Start öffnet sich das Kontrollfenster, und kurz darauf sollte sich Ihr Webbrowser mit der Startseite des Servers öffnen (siehe Abbildung 3.8).



Abbildung 3.8: Startseite von MAMP mit Kontrollfenster

Wie Sie an der URL erkennen können, wurde der Server gestartet und so konfiguriert, dass er auf den Hostnamen `localhost` reagiert. Die Startseite von MAMP gibt Ihnen außerdem einige Informationen über Benutzer und Kennwort für die Nutzung des Datenbankservers.

2. PHP- und Datenbank-Umgebung

Aufgrund der Sicherheitseinstellungen unter Mac OS X ist es erforderlich, entweder in der lokalen Firewall den von MAMP für die MySQL-Datenbank verwendeten Port freizuschalten oder den Port innerhalb von MAMP zu ändern. Ich empfehle, in den EINSTELLUNGEN des Kontrollfensters den Port des MySQL-Datenbankservers auf den MySQL-Standardport 3306 zu setzen. Dies hat den Vorteil, dass keine weiteren Anpassungen an dem System durchgeführt werden müssen und weitere Werkzeuge auch direkt auf die Datenbank zugreifen können. Grundsätzlich ist die Nutzung eines anderen Ports dann wichtig, wenn Sie mehrere MySQL-Datenbankserver auf dem System betreiben.

In der von uns besprochenen Konfiguration werden wir annehmen, dass WEB SHARING nicht aktiviert ist und damit der in MAMP integrierte Apache Server der einzige Webserver auf Ihrem System ist. Aus diesem Grund können wir seinen Port auf den Standardport 80 setzen. Bitte beachten Sie, dass Sie sich mit dieser Konfiguration beim Start des Webservers als Administrator authentifizieren müssen.

Durch das Paket MAMP ist die PHP-Konfiguration bereits durchgeführt. Es werden sowohl PHP 4 als auch PHP 5 als Erweiterungen angeboten. Alle für Joomla!

wichtigen Erweiterungen sind bereits aktiviert, so dass Sie keine weiteren Änderungen vornehmen müssen. Grundsätzlich befinden sich alle Konfigurationsdateien im MAMP-Programmverzeichnis `/Applications/MAMP`.

3. Installation von Zend Core



Achtung

Leider hat die Firma Zend während der Entwicklung dieses Buches den Support für Zend Core auf dem Mac eingestellt. Auf der CD befindet sich noch eine alte Version, die grundsätzlich funktioniert. Alternativ ist es auch möglich, lediglich den frei verfügbaren Debugger zu installieren und PHP selbst entsprechend zu konfigurieren.

Die Installation des Zend Core erfolgt über ein im Paket enthaltenes Shellskript. Laden Sie das Archiv für die Zend Core⁴-Installation von der Zend-Website⁵. Nun unpacken Sie das Archiv in einem lokalen Verzeichnis und öffnen dieses Verzeichnis in einem Terminal-Fenster (PROGRAMME → DIENSTPROGRAMME → TERMINAL). Zend Core erfordert Administratorenrechte auf Ihrem System. Da der *root*-Benutzer unter Mac OS X aber gesperrt ist, verwenden wir hierzu die Authentifizierung über einen Administrator-Account.

Führen Sie folgende Befehle aus, um die Installation zu starten:

```
sudo /bin/bash
cd <das_verzeichnis_mit_dem_entpackten_archiv>
./install.sh
```

Nun startet das Installationsprogramm von Zend Core.

Als Installationsverzeichnis von Zend Core können Sie das vorgeschlagene `/usr/local/Zend/Core` verwenden. Danach werden Sie gefragt, ob Sie Support möchten und welches Passwort Ihre Installation haben soll.

Bei der Frage nach der Installation des integrierten Apache Servers bitte mit NEIN (NO) antworten (wir nutzen hier den schon installierten Apache Server von MAMP). Im nächsten Dialog werden Sie auch schon nach dessen Installationsverzeichnis gefragt. Angeboten wird der Standardserver (`/etc/httpd/httpd.conf`) und die Auswahl 2, welche die individuelle Festlegung erlaubt – bitte treffen Sie diese Auswahl. Als Pfad für den Webserver geben Sie dann an:

```
/Applications/MAMP/bin/apache2
```

Wenn der Pfad richtig eingetragen ist, erhalten Sie eine kurze Information darüber, wo sich die Konfigurationsdatei befindet (`/Applications/MAMP/conf/apache/httpd.conf`). Dann wird gefragt, ob Sie mit dieser Auswahl fortfahren möchten: JA.

4 Bei uns aktuell `ZendCore-v2.0.1-MAC-Uni.tar.gz`.

5 http://www.zend.com/de/products/zend_core

Abschließend werden Sie noch gefragt, ob Zend Core als Modul installiert werden soll, was Sie auch mit OK bestätigen, genauso wie die Auswahl keines speziellen virtuellen Servers.

Zum Abschluss der Installation können Sie noch weitere Komponenten wie das Zend Framework installieren. Dies ist für Joomla! aber nicht erforderlich, und Sie können dies daher entsprechend übergehen.



Tipp

Während unserer Tests hatte ich anfangs das Problem, dass die Installation von Zend Core auf dem Arbeitsplatz nicht einwandfrei durchgeführt wurde. Zum Teil wurden die Dateien nicht richtig kopiert oder auch die Installation nicht vollständig abgeschlossen. Nach dem ich das Archiv `ZendCore.tar` manuell ausgepackt und in das Zielverzeichnis kopiert hatte, lief die Installation problemlos durch. Wichtiges Indiz für eine fehlerhafte Installation ist, wenn im Zend Core-Verzeichnis die Datei `share/.registry` nicht geschrieben wurde.

Bis hierher hat die Installation verschiedene Dateien geschrieben und entsprechend für Zend Core konfiguriert. Wir müssen abschließend diese Inhalte lediglich in die MAMP-Konfiguration übertragen und sie entsprechend anpassen.

Listing 3.1: Auszug aus der Datei `Applications/MAMP/conf/php5/php.ini` mit Zend Core-Erweiterung

```
[Zend]
zend_optimizer.optimization_level=15
zend_extension_manager.optimizer=/Applications/MAMP/bin/php5/zend/lib/Optimizer-
3.2.6
zend_optimizer.version=3.2.6
#zend_extension=/Applications/MAMP/bin/php5/zend/lib/ZendExtensionManager.so

# New Zend Core settings
zend_core_dir="/usr/local/Zend/Core"
zend_gui_password=<md5(kennwort) // wert wie aus dem Original>
zend_core.default_gui_language=en
zend_gui.ini_modifier_path="/usr/local/Zend/Core"
zend_extension="/usr/local/Zend/Core/lib/zend/ZendExtensionManager.so"
zend_extension_manager.optimizer="/usr/local/Zend/Core/lib/zend/optimizer"
zend_extension_manager.debug_server="/usr/local/Zend/Core/lib/zend/debugger"
zend_ini_file="/usr/local/Zend/Core/etc/zend.ini
date.timezone=CEST
```

Die letzte Zeile im Original, welche den Zend Extension Manager lädt, wird mit einem Hash auskommentiert. Die neuen Zend Core-Informationen können Sie direkt aus der Datei `/etc/php.ini` übernehmen. Insbesondere das Kennwort wurde dort während der Installation entsprechend gesetzt. Die hier angegebenen Pfade beziehen sich auf die Standardinstallation im Zend Core-Verzeichnis.

Abschließend stoppen und starten Sie bitte den MAMP-Webserver erneut. Dieser sollte ohne Probleme starten und dann in der `phpinfo`-Seite die in Abbildung 3.9 gezeigten Hinweise auf den Debugger zeigen (letzte Zeile neben dem Zend Engine 2-Logo).

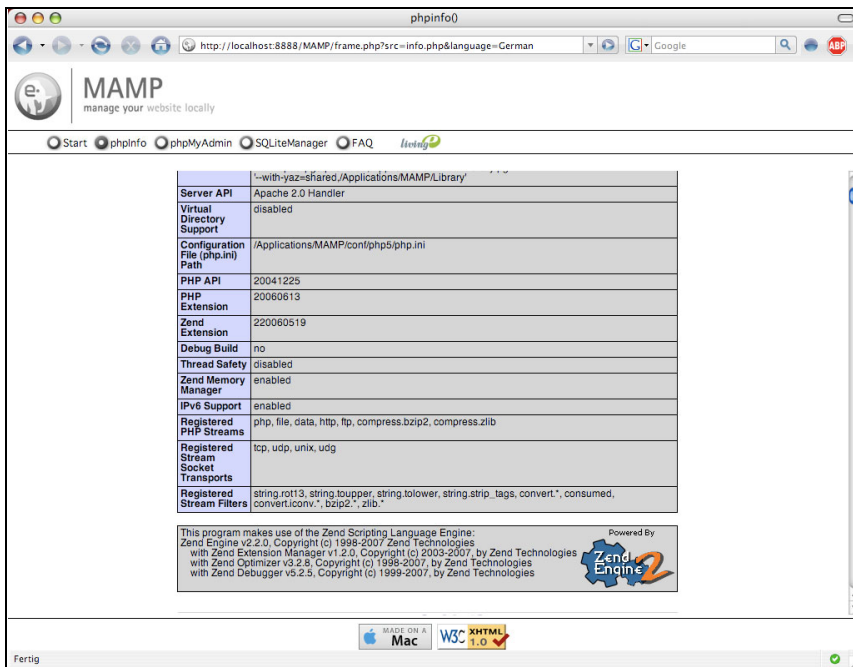


Abbildung 3.9: Ausgabe von `phpinfo` innerhalb von MAMP, enthält Zend Debugger

4. Installation von Joomla!

In diesem Verzeichnis befindet sich auch das Webserver-Verzeichnis `htdocs`, in welches Sie Ihre Joomla!-Dateien für die Testinstallation kopieren müssen. Sie können nun mit dem Zugriff auf `http://localhost` (bzw. `http://localhost:8888`, wenn Sie den Standard-MAMP-Port nutzen) die Installation von Joomla! durchführen.

Wie Sie sehen, ist die Installation auf Basis eines der bestehenden so genannten AMP (Apache, MySQL, PHP)-Pakete wesentlich einfacher. Es gibt aber immer wieder Fälle, in denen auch eine manuelle Konfiguration des Apache- oder MySQL-Servers erforderlich ist. An diesem Punkt liegt es sicher mehr an den Präferenzen bzw. dem individuellen Wissen über diese Serversysteme, was die empfohlene Vorgehensweise ist.

3.3.3 Serverinstallation in einer virtuellen Umgebung

Heute wird immer häufiger von der Virtualisierung von Systemen gesprochen. Damit ist die Technik gemeint, die es ermöglicht, auf einem PC mehrere vollständige Betriebssysteme gleichzeitig zu betreiben. Um z. B. verschiedene Programme zu tes-

ten und nicht Ihre Hauptinstallation zu gefährden, können Sie in einer virtuellen Umgebung diese Tests durchführen. Wenn ein Fehler auftritt, dann betrifft dies nur die virtuelle Umgebung, nicht aber Ihr Hauptsystem.

Die Installation der Server in einer virtuellen Umgebung ist aus meiner Sicht für ein Entwicklungssystem nicht unbedingt sinnvoll. Die Unterschiede zwischen der IIS- oder Apache-basierten Implementierung von PHP sind sehr gering, und es ist für den Entwickler nicht erforderlich, dass er beide Umgebungen testet. Die Nutzung einer virtuellen Umgebung wie z.B. Parallels⁶ oder VMware⁷ ist grundsätzlich möglich und mag z.B. auf einem Server auch interessant sein. Für eine auf einem Arbeitsplatz installierte Variante kostet diese Lösung aber viele Ressourcen und damit auch Performance.

Die Vorgehensweise ist bei der Installation wie oben bereits beschrieben. Eines der Themen, welches eine größere Rolle spielen wird, ist sicherlich die Konfiguration geeigneter DNS- bzw. Host-Einträge, wie wir sie im nächsten Kapitel besprechen werden.

3.4 Einrichtung der lokalen Entwicklungswerkzeuge

Nachdem wir nun die Serverumgebung eingerichtet haben und eine lauffähige Joomla!-Installation für Testzwecke nutzen können, ist es nun an der Zeit, die erforderlichen Entwicklerwerkzeuge zu installieren. Auch hier werden wir die beiden Betriebssysteme und verschiedene Tools betrachten, wobei wir den Schwerpunkt auf die angesprochenen Integrierten Entwicklungsumgebungen legen (siehe Abschnitt 3.2, *Werkzeuge und Produkte für die Arbeitsumgebung*).

3.4.1 Vorschlag für die lokale Umgebung

Bevor wir mit der Installation der Werkzeuge beginnen, ein kurzer Überblick, was die lokale Umgebung für uns leisten soll und wie dies umgesetzt werden kann. Ziel ist es, eine Umgebung zu schaffen, die die Entwicklung von verschiedenen Projekten auf einfache Art und Weise unterstützt. Auch hier gibt es verschiedene Möglichkeiten des Setups und auch unterschiedliche Nutzungsprofile. Die von mir beschriebene Lösung ist daher mehr eine »Best Practice«-Variante, die sich für mich in den letzten Jahren als effektiv herausgestellt hat. Die nachfolgende Tabelle zeigt einige der Festlegungen und Vorgaben, die ich für die Entwicklungsumgebung nutze.

6 <http://www.parallels.com/>

7 <http://www.vmware.com/>

Vorgabe	Bemerkung
Verzeichnisstrukturen	<p>Im Rahmen der Entwicklung passiert es häufig, dass an mehreren Projekten gearbeitet wird. Für mich hat es sich als praktisch erwiesen, ein zentrales Arbeitsverzeichnis, z. B. <code>work</code>, zu haben und unterhalb von diesem pro Projekt ein Verzeichnis auf Basis des Domainnamens.</p> <p>z.B. <code>/work/joomla.org</code></p> <p>Unterhalb jedes dieser Projektverzeichnisse findet sich dann für Erweiterungen eine Struktur ähnlich zu dieser:</p> <pre> /development /development/package /development/releases /development/tags /development/test /development/trunk /www </pre> <p>Bei Webprojekten eher:</p> <pre> /server /tags /www </pre> <p>Zusätzliche Verzeichnisse für Dokumentation, Backup usw. je nach Bedarf.</p>
Hostnamen und Webverzeichnisse	<p>Für die Zugriffe auf die einzelnen Projekte gibt es grundsätzlich die Möglichkeit, Unterverzeichnisse des <code>DocumentRoot</code> aus der Webserver-Konfiguration zu nutzen. Auch können entsprechende Aliase eingeführt werden, welche dann auf andere Verzeichnisse zeigen. Beide Varianten haben den Nachteil, dass immer eine URL wie <code>http://localhost/<verzeichnis></code> genutzt wird. Dies ist für den Umgang mit den Webseiten etwas kompliziert, da die Projekte schlecht auseinandergehalten werden können.</p> <p>Der etwas aufwändigere, dafür aber saubere Weg ist das Konfigurieren eines eigenen Host-Eintrags und die Konfiguration eines entsprechenden virtuellen Servers. Damit lassen sich dann URLs wie z. B. <code>http://home.meinprojekt.de</code> realisieren. Dies werden wir näher in Kapitel 8.2.1, <i>Webserver und virtuelle Hosts</i> besprechen.</p>
Datenbanken	<p>Sie sollten für jedes Projekt eine eigene Datenbank einrichten. Dies hilft insbesondere Probleme mit unterschiedlichen Versionen der Datenbankstruktur zu vermeiden. Wenn Sie innerhalb eines Projektes mehrere Datenbanken z. B. für Testversionen benötigen, dann empfiehlt sich die Nomenklatur <code><Basisname>.<Suffix></code>. Diese hat den Vorteil, dass z. B. phpMyAdmin dann kürzere Namen in den Auswahl-listen anzeigt.</p>

Tabelle 3.3: Überblick der Vorgaben für die Entwicklungsumgebung

Innerhalb der Entwicklung haben Sie meist mit verschiedenen Anforderungen und Wünschen zu tun. Oft arbeiten Sie lokal an einer Weiterentwicklung, während z.B. eine Website mit individuellen Erweiterungen bereits produktiv im Einsatz ist. Die Verzeichnisse wie `/server` oder auch `/dokumentation` bieten sich für die Speicherung der unterschiedlichen Konfigurationen oder Backups von Datenbanken an.

3.4.2 Entwicklungswerkzeuge auf Windows einrichten

Beginnen wir zunächst mit der Installation von Zend Studio als Integrierte Entwicklungsumgebung (IDE). Der Ablauf der Installation ist sehr einfach, und unter Windows können die Standardvorgaben übernommen werden. Die vorgeschlagenen Erweiterungen wie z.B. die Werkzeugleiste für den Internetexplorer sind für die Arbeit sehr hilfreich. Weitere Produkte von Zend etwa z.B. der Zend Guard sind für die Entwicklung innerhalb von Joomla! nicht erforderlich. Auch die Zend-Plattform ist nicht erforderlich, da wir die erforderlichen Elemente bereits durch den Zend Core bereitgestellt haben. Auch wenn Joomla! immer noch mit PHP 4 funktionsfähig ist, ist es sinnvoll, zukünftige Erweiterungen auf Basis von PHP 5 aufzubauen. Die Entwicklung von PHP 4 wurde Ende eingestellt⁸. Seit diesem Zeitpunkt erhalten Sie zwar noch Sicherheitsupdates, aber keine neuen Versionen und Entwicklungen mehr. Aus diesem Grund empfiehlt sich während des Setups gleich die Auswahl der PHP-Sprachunterstützung.

Als weiteres Element installieren wir nun den TortoiseSVN Client für die Versionsverwaltung von Subversion. Die Versionsverwaltung dient den Entwicklern von Joomla! und anderen Programmen dazu, dass die verschiedenen Änderungen am Quellcode zu jedem Zeitpunkt nachvollzogen werden können. Das Ziel ist es, ggf. auftretende Konflikte, z.B. wenn zwei Entwickler an der gleichen Datei Änderungen vorgenommen haben, leicht auflösen zu können. Für Sie ist dieses Programm hilfreich, um ständig auf dem aktuellsten Stand der Entwicklung zu bleiben. Auch diese Installation kann ohne besondere Einstellungen vorgenommen werden. Wie TortoiseSVN genau genutzt wird, werden wir im später folgenden Abschnitt 4.3.3, *Download von Releases bzw. Quellcodes*, erfahren.

Nachdem wir nun alle Werkzeuge (Zend IDE und TortoiseSVN Client) für die Entwicklungsumgebung installiert haben, richten wir uns noch das erste Projekt in der IDE ein. Dazu starten Sie einfach Zend Studio und richten mit `[Strg]+[Umschalt]+[N]` oder (PROJEKT → NEUES PROJEKT) ein neues Projekt ein. Wählen Sie als Name z.B. »Joomla! Test« und fügen Sie als Pfad das Dokumentenverzeichnis des Webservers hinzu (`c:\inetpub\wwwroot`). Weitere Einstellungen wie Debugger und Java Runtime sind nicht erforderlich. Sie sollten nun einen Bildschirm haben, der mit dem nachfolgenden Dialog vergleichbar ist.

⁸ <http://www.php.net/archive/2007.php>

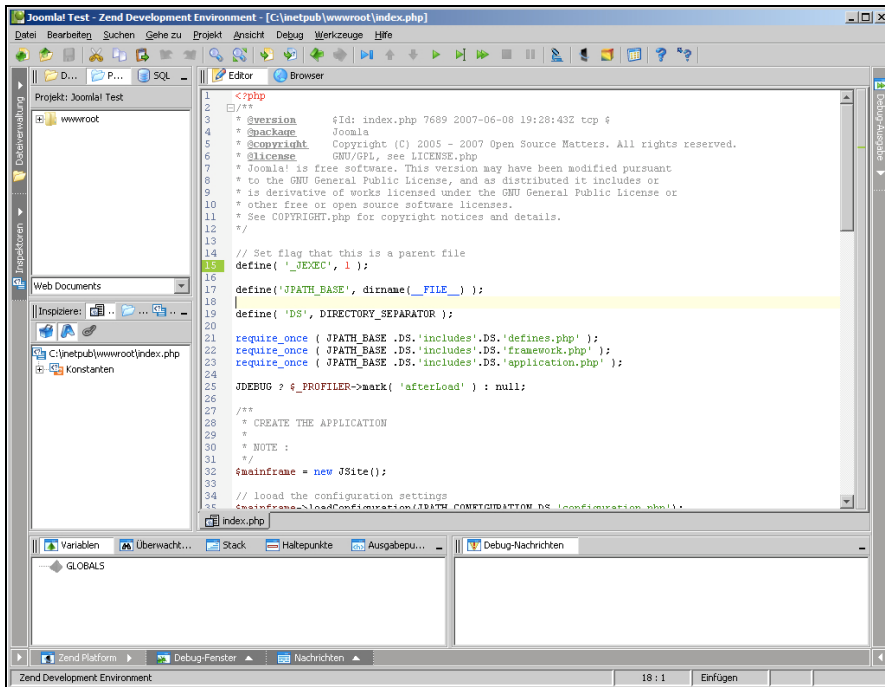


Abbildung 3.10: Installiertes Zend Studio mit eingerichtetem Joomla!-Projekt

3.4.3 Entwicklungsumgebung für Mac OS X

Wie bereits angesprochen, nutzen wir Eclipse für die Entwicklungsumgebung auf dem Mac. Wegen der einfacheren Integration in den Debugger nutzen wir die Eclipse PDT-Variante. Diese Version wird von Zend, IBM und weiteren Unternehmen entwickelt, um eine gute kostenfreie IDE für PHP zu liefern. Die Eclipse-Umgebung ist in vielen Punkten sehr unterschiedlich zu der Zend Studio die, gerade wenn es um die Handhabung verschiedener Themen geht. Wir werden auf ein paar Besonderheiten dieser IDE eingehen, im weiteren Verlauf des Buches dann aber Zend Studio primär nutzen.

Es empfiehlt sich, die aktuellste Release-Version, z. B. »1.0 RELEASE BUILD«, zu nutzen. Für die unterschiedlichen Betriebssysteme stehen dann verschiedene Pakete zur Auswahl. Die größten »PDT ALL-IN-ONE«-Pakete enthalten alle erforderlichen Dateien und zusätzlichen Plugins. Nach dem Download entpacken Sie bitte das Archiv und kopieren Sie dann das entpackte »ECLIPSE«-Verzeichnis in Ihren PROGRAMME-Ordner. Für diesen Vorgang müssen Sie sich, wie bei OS-X üblich, als Administrator identifizieren. Sollten Sie nicht über die erforderlichen Rechte/Kennwörter verfügen, können Sie Eclipse auch von jedem anderen Verzeichnis aus starten.

Zur besseren Verwaltung mehrerer Projekte bietet Eclipse eine Besonderheit, die sich hier »Arbeitsbereiche« (»Workspaces«) nennt. Diese Arbeitsbereiche gruppieren ähnliche oder zusammengehörende Projekte, um sie einfacher in der IDE zu nutzen. Je nach eigener Organisation könnten Sie z. B. einen Arbeitsbereich »Eigenentwicklun-

gen«, »Webprojekte« oder »Kundenprojekte« erstellen. Die einzelnen Projekte innerhalb dieser Arbeitsbereiche können Sie nach den aktuellen Erfordernissen aktivieren. Wenn Sie die Standardvorgaben von Eclipse verwenden, dann werden die Quellcodes von Projekten innerhalb des Arbeitsbereiches gespeichert. Dabei wird als Verzeichnisname der Projektname verwendet. Sie können aber auch andere Verzeichnisse in einer zweiten Struktur nutzen. Dies ist insbesondere dann sinnvoll, wenn die Projektnamen eher sprechend sein sollen. Da Namen mit Leerzeichen oder Sonderzeichen in den Konfigurationspfaden etwas kompliziert zu nutzen sind, kann hier ein Verweis auf einen einfacheren Systemnamen helfen.

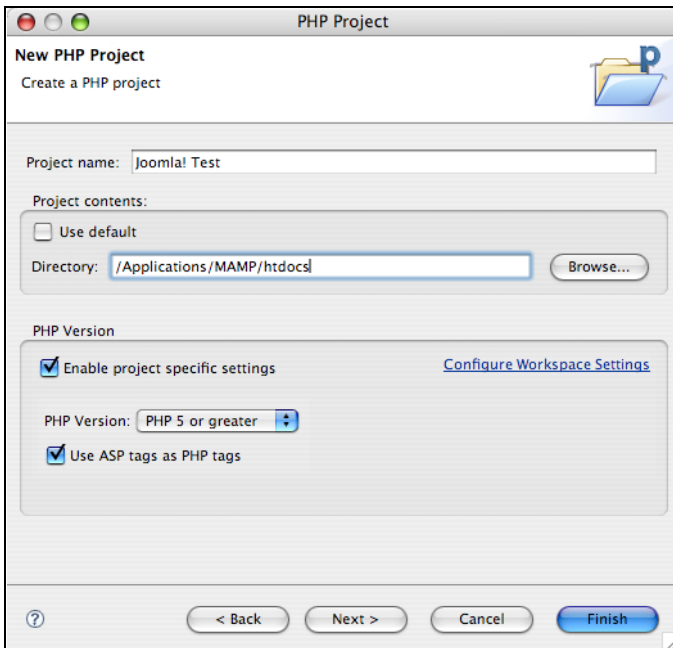


Abbildung 3.11: Einrichten des Projektes innerhalb von Eclipse

Nachdem Sie den Arbeitsbereich eingerichtet und Eclipse gestartet haben, erhalten Sie ein leeres Fenster. In diesem können Sie über FILE → NEW PROJECT das Projekt anlegen. Wählen Sie PHP und konfigurieren Sie das Projekt wie in Abbildung 3.11 dargestellt. Nach diesem Schritt haben Sie das Projekt erfolgreich eingerichtet. Eine weitere Besonderheit sind die so genannten Perspektiven. Sie werden gefragt, ob Sie zur PHP-Perspektive wechseln wollen, was nicht mehr ist als ein Wechsel zu einer Ansicht, die für PHP-Projekte optimiert ist. Nach dem Wechsel sollte Ihr Bildschirm ungefähr wie in der nächsten Abbildung aussehen.

Für die Zugriffe auf Subversion-Verzeichnisse werden wir vorläufig den in Eclipse integrierten SVN Client verwenden. Dieser SVN Client ist einfach zu verwenden und ermöglicht alle wichtigen Funktionen. Für speziellere Aufgaben wie etwa die Suche von Unterschieden innerhalb von Versionen bietet sich z. B. Synchro SVN an. Die Nutzung dieses Programms ist kostenpflichtig, aber für komplexere Aufgaben zu empfehlen.

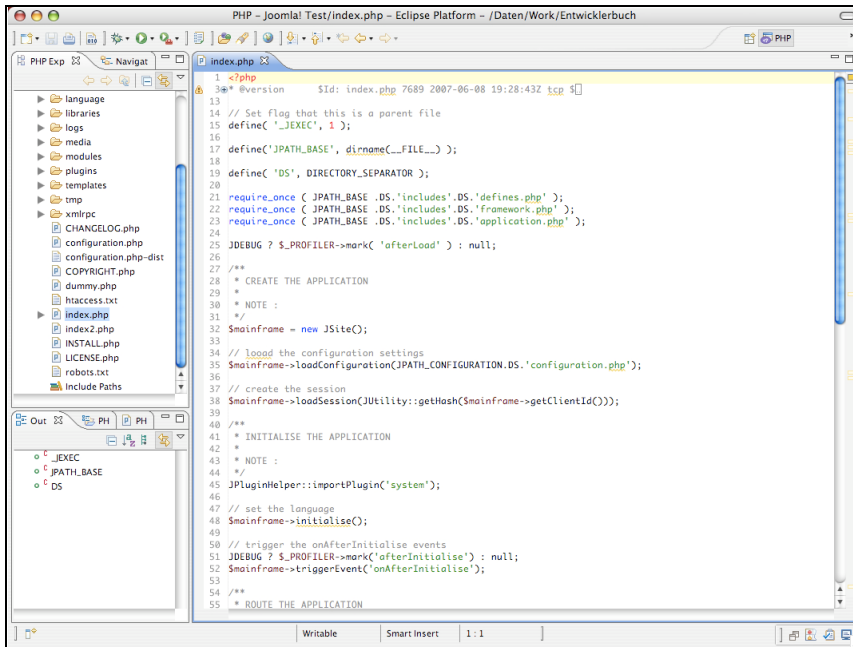


Abbildung 3.12: Darstellung der Eclipse-Oberfläche mit index.php-Datei

3.4.4 Einrichtung eines Subversion-Clients

Der letzte Teil der Entwicklungsumgebung ist der Subversion-Client, um auf die verschiedenen Projekte auch über diesen Weg zugreifen zu können.



Hinweis

Sollten Sie in Ihrem Unternehmen den Einsatz von Subversion als interne Versionsverwaltung erwägen, dann ist auch die Installation des Subversion-Servers erforderlich. Dieser Server basiert auf den Standardpaketen von Subversion für Ihre Umgebung (z. B. Windows oder Linux). In der Regel wird entweder über HTTP & WebDAV oder über SSH eine Verbindung zu dem Server aufgebaut. Die Authentifizierung der Entwickler erfolgt über spezielle Konfigurationsdateien oder zentrale Dienste wie einen LDAP/ADS Server. Da die Einrichtungen sehr von Ihren Wünschen und auch der entsprechenden Umgebung abhängig sind, empfehle ich hier, die Dokumentation zu konsultieren. Es gibt verschiedene Bücher und Online-Dokumentationen, die genau die jeweiligen Installationen beschreiben.

Die Installation eines Subversion-Clients ist normalerweise über die Installationsroutinen leicht durchführbar. Bei den Projekten wie Tortoise (Windows) und SCPlugin (Mac OS X) wird der Client in den Explorer bzw. Finder integriert und kann dort über

die rechte Maustaste genutzt werden. Grundsätzlich ist die Struktur der Versionsverwaltung wie folgt aufgebaut:

- Serversystem

Auf dem Server sind die so genannten Speicher oder Verwaltungsorte (engl. *repositories*) abgelegt. Unter der CVS-Versionverwaltung wurden die Daten im Dateisystem abgelegt. Unter Subversion wird nun aus Effektivitätsgründen eine Datenbank verwendet. In beiden Fällen ist der Zugriff für den Entwickler transparent, und es ergibt sich eine normale Verzeichnisstruktur, welche die aktuelle Version des Projektes enthält. Auf dem Server werden sowohl die Zugriffe als auch die Veränderungen zwischen den einzelnen Versionen gespeichert bzw. protokolliert. Das Serversystem enthält für jedes Projekt ein eigenständiges Repository⁹, welches alle Verzeichnisse und Dateien enthält.

- Arbeitsversion

Auf Ihrem System wird in einem beliebigen Verzeichnis eine so genannte Arbeitsversion oder Arbeitskopie angelegt. Diese Version entspricht genau einem Stand der auf dem Server befindlichen Daten. Durch eine initiale Verbindung (Checkout) werden neben der Struktur der Programmdateien auch alle erforderlichen Referenzdaten für den Subversion-Client erzeugt. Nach diesem ersten Schritt können Sie jederzeit Ihre Arbeitsversion aktualisieren, und die neuen Dateien bzw. Änderungen werden Ihnen damit vom Server übertragen.

Je nach Betriebssystem wählen Sie bitte eines der beiden Projekte und führen eine Standardinstallation des Clients durch. Nach der Installation können Sie bereits auf entfernte Subversion-Server zugreifen. Ein Beispiel wird im folgendem Kapitel in Abschnitt 4.3.3, *Download von Releases bzw. des Quellcodes*, genau durchgesprochen.

⁹ Der Aufbau eines Repositories ist üblicherweise so wie in Abschnitt 3.4.1, *Vorschlag für die lokale Umgebung*, besprochen.