

Redaxo-Dokumentation

Version 2.7.4



Verfasser: Peter Wolfrum
Dagmar Uttich

Stand: 7. Oktober 2005

Inhaltsverzeichnis

Dokumentation REDAXO 2.7.....	4
0. Einleitung.....	4
A. Erste Schritte.....	5
A1. Installation.....	5
A.1.01 Download.....	5
A.1.02 Systemvoraussetzungen	6
A.1.03 Upload	7
A.1.04 Setup	8
A.2 Demoversionen	12
A.3 Webseiten erstellen mit Redaxo.....	13
B. Redaktion	16
B1. Strukturverwaltung	17
B1.01 Kategorien	18
B1.02 Artikel	19
B1.03 Block/Modul	20
B2. Medienpool	22
B2.01 Medien verwalten	22
B3. Statistiken	25
B4. Community	26
B4.01 Userverwaltung	26
B4.02 Artikelkommentarverwaltung	26
B4.03 Newsletter	27
C. Entwickler & Admin	28
C1. Templates	29
C1.01 Template erstellen	29
C1.02 Navigation	30
C1.02.01 Navigation für bis zu drei Navigationsebenen.....	31
C1.02.02 Navigation mit Unternavigations-Links.....	32
C1.03 Mehrere Artikel einbinden	34
C1.04 Seitenspezifische Metatags	34
C1.05 Includes	35
C2. Module & Aktionen	36
C2.01 Modul erstellen und bereitstellen.....	37
C2.02 REDAXO Variablen.....	38
C2.03 Actions.....	41
C3. Benutzer	42
C3.01 Navigation / Programmteile.....	42
C3.02 Struktur.....	43
C3.03 Modul / Block	44
C4. Addons	45
C4.01 Struktur.....	45

- C4.02 Entwicklung..... 47
- C5. Specials 48
- C6. Object Oriented Framework 49
 - C6.01 OOCategory..... 49
 - OOCategory-Klasse..... 50
 - getCategoryById(2)..... 50
 - searchCategoriesByName()..... 51
 - getRootCategories()..... 52
 - getChildren()..... 54
 - getParent()..... 55
 - getArticles()..... 56
 - getStartArticle()..... 57
 - getId()..... 58
 - getName()..... 59
 - getPriority ()..... 59
 - getUrl()..... 60
 - isOnline ()..... 60
 - toString()..... 61
 - C6.02 OOArticle..... 62
 - OOArticle-Klasse..... 62
 - getArticleById()..... 62
 - searchArticlesByName()..... 64
 - getArticlesByType(id)..... 66
 - getCategory()..... 68
 - getFirstSlice()..... 69
 - getSlicesOfType()..... 70
 - getId()..... 71
 - getName()..... 71
 - getDescription()..... 72
 - getUrl()..... 73
 - isOnline()..... 74
 - toString()..... 74
 - C6.03 OOArticleSlice..... 75
 - OOArticleSlice-Klasse..... 75
 - getArticleSliceById()..... 76
 - getFirstSliceForArticle()..... 77
 - getSlicesForArticleOfType()..... 78
 - getNextSlice()..... 79
 - getPreviousSlice()..... 80
 - getArticle()..... 81
 - getId()..... 82
 - getValue()..... 82
 - getLink()..... 83
 - getLinkUrl()..... 83
 - getFile()..... 84
 - getFileUrl()..... 84
 - getHtml()..... 85
 - getPhp()..... 85
- C7. Import & Export 86

Dokumentation REDAXO 2.7

0. Einleitung

Die Dokumentation bezieht sich in der aktuellen Fassung auf die **REDAXO Version 2.7.4**. Mit dem Release der aktuellen Fassung von REDAXO 3.0 werden wir die Veränderungen in den jeweiligen Bereichen kennzeichnen und beschreiben.

Die Struktur der Dokumentation.

Wir haben die Dokumentation in fünf Teile gegliedert um so den Redakteur, dem Einsteiger und dem Profi die geeigneten Anlaufstellen zu bieten.

A. Installation

Hier werden die Installation und die ersten Schritte mit den Demoversion beschrieben. Dabei werden auch aktuelle Fragen und Lösungen aus dem Forum einbezogen. Für den Einstieg in der Programmierung einer Webseite mit REDAXO ist dieser Teil geeignet.

B. Redaktion

In diesem Bereich wird REDAXO aus der Sicht des Redakteurs vorgestellt. Struktur, Bedienung und die Verwaltung der Inhalte.

C. Entwickler & Admin

Die Bereiche C und D helfen bei der Entwicklung einer Webseite. Hier werden die Backend-Bereiche vorgestellt.

D. Erweiterungen

REDAXO kann in viele Richtungen erweitert werden. Hier finden auch Artikel und Beiträge aus dem Forum ihren Platz.

E. FAQ & Technotes

Die Dokumentation wird in den nächsten Wochen und Monaten mit neuen Themen erweitert. Es konnten im ersten Schritt nicht alle Themen, Funktionen oder Szenarios beschrieben werden, die sich mit dem Aufbau einer Seite mit REDAXO ergeben.

Für eine kontinuierliche Erweiterung die Dokumentation benötigen wir auch Ihre Hilfe. Für Hinweise, Tipps, Anleitungen, Artikel, Fehlerbeseitigungen sind wir immer dankbar.

A. Erste Schritte

Auf den folgenden Seiten dieses Kapitels erfahren Sie mehr über die Installation von Redaxo auf dem Webserver Ihrer Wahl.

Überprüfen Sie vor Beginn die [Systemvoraussetzungen](#) für Ihr Webpaket und lernen Sie mehr über die ersten Schritte mit einem flexiblen Content-Management-System.

Links zu Redaxo:

Redaxo.de | <http://www.redaxo.de>

Aktuelle Informationen zu den aktuellen Versionen, die Basis-Installation und Updates erhalten Sie auf der offiziellen Redaxo-Webseite.

Forum.Redaxo.de | <http://forum.redaxo.de>

Eine der ersten Anlaufstellen für Support, Fragen, Tipps und "Insider-Wissen" erhalten Sie im Forum zu Redaxo.

A1. Installation

In den folgenden Kapiteln werden die ersten Schritte sowie die Installation von REDAXO beschrieben. Nach dem erfolgreichen Setup können Sie anhand der mitgelieferten Demos und diesem Manual das Konzept, Abläufe und Funktionen von REDAXO kennenlernen.

Für weitere aktuelle Tipps lesen Sie auch Hinweise in [E. FAQ & Technotes](#)

A.1.01 Download

Auf der offiziellen Redaxo-Webseite erhalten Sie die aktuelle Basisversion sowie Updates oder ältere Versionen:

[Redaxo-Download-Bereich](#)

Laden Sie die gewünschte Version und entpacken Sie das Zip-Archiv. Die entpackte Ordernstruktur sollte nun folgende Struktur enthalten:

REDAXO 2.7.4

```
/css  
/files  
/js  
/pics  
/redaxo  
_getstarted.txt  
_lizenz.txt  
_readme.txt  
index.php
```

Die komplette Ordnerstruktur sollte für die Installation nicht verändert werden. Einen Überblick über die komplette Ordnerstruktur von Redaxo, wird im Kapitel [Installation – Upload](#) beschrieben.

Kurze Beschreibung:

Im Ordner **files** werden alle Dateien gespeichert, die in Redaxo über den Medienpool auf den Server geladen werden. Die Ordner **css**, **js**, **pics** dienen zur Strukturierung Ihrer Webseite.??

Die Datei **Index.php** dient zur Darstellung aller Artikel/Seiten der Webseite.

CVS – Hinweis

CVS-Version downloaden

Unter Linux

Redaxo 2.7

```
cvs -z3 -d:pserver:anonymous@cvs.redaxo.berlios.de:/cvsroot/redaxo export redaxo2_7
```

Redaxo 3.0

```
cvs -z3 -d:pserver:anonymous@cvs.redaxo.berlios.de:/cvsroot/redaxo export redaxo3_0
```

Unter Windows mit Tortoise

CVSROOT

pserver

anonymous@cvs.redaxo.berlios.de:/cvsroot/redaxo

MODUL

redaxo2_7

redaxo3_0

A.1.02 Systemvoraussetzungen

PHP 4/5

mySQL-Datenbank

Daten Ihres Providers:

FTP-Zugang (Upload von Redaxo auf Ihren Server)

Host, Login, Passwort

mySQL-Datenbank

Host, Datenbankname, Benutzername, Passwort

A.1.03 Upload

Laden Sie nun den entpackten Ordnerinhalt der Basis-Version auf Ihren Webservice. Im allgemeinen empfehlen wir dafür das Root-Verzeichnis Ihres Webspaces. Für die notwendigen Einstellungen zum Upload auf Ihren Webserver, schlagen Sie in der Hilfe Ihres FTP-Programms nach.

Anpassungen um Redaxo in einem Unterverzeichnis zu betreiben, erfahren Sie Bereich D. FAQ.

Wenn Sie die komplette Struktur auf Webservice geladen haben, können Sie nun die Installation starten.

Einer der ersten Überprüfungen gilt den Rechten mehrerer Ordner und Dateien.

Sie können die Installation starten und auf die mögliche Fehlerliste warten, oder gleich die notwendigen Einstellungen über Ihr FTP-Programm überprüfen und anpassen.

Folgende Ordner und Dateien benötigen die genannten Rechte:

Version 2.7.x

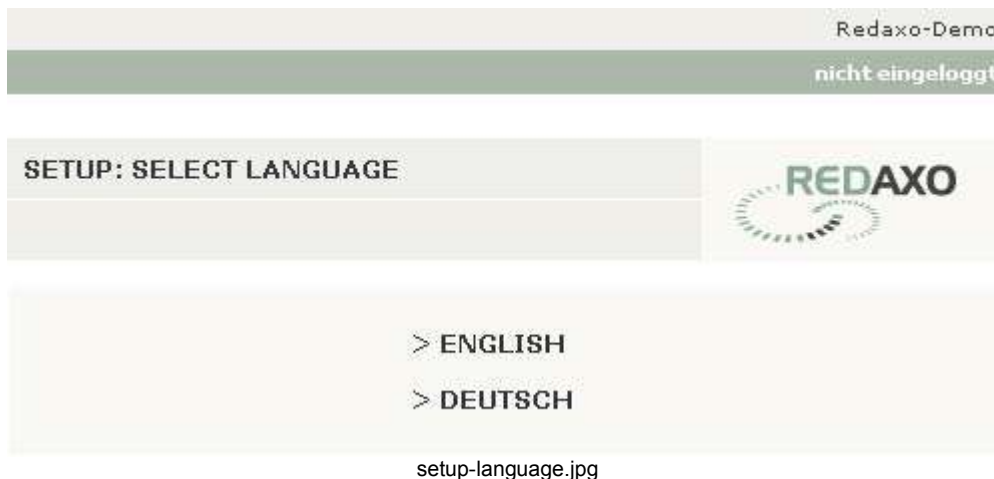
Rechte	Ordner
775	/files
775	/redaxo/include/addons/import_export/ files
775	/redaxo/include/ generated
775	/redaxo/include/generated/ articles
775	/redaxo/include/generated/ cache
775	/redaxo/include/generated/cache/cache.php
775	/redaxo/include/generated/ categories
775	/redaxo/include/generated/ logs
775	/redaxo/include/generated/ templates
775	/redaxo/include/ install
775	/redaxo/include/addons.inc.php
775	/redaxo/include/functions.inc.php
775	/redaxo/include/master.inc.php

A.1.04 Setup

Start

Um die Installation von Redaxo zu starten, geben Sie die URL Ihres Webspace ein und fügen an den Pfad

/redaxo oder
/redaxo/index.php an.



Auf manchen Server kann schon bei diesem ersten Schritt eine Fehlermeldung erscheinen.

PHP Warning: Variable passed to each() is not an array or object in F:\website edaxo edaxo\include\functions\function_rex_mquotes.inc.php on line 24

PHP Warning: Variable passed to each() is not an array or object in F:\website edaxo edaxo\include\functions\function_rex_mquotes.inc.php on line 31

Redaxo benötigt die Einstellung **Magicquotes on**.

Wählen Sie danach die gewünschte Sprachversion aus und bestätigen Sie, dass Sie die Lizenzbestimmungen gelesen haben und starten danach das eigentliche Setup.

Sprachversion

Wählen Sie danach die gewünschte Sprachversion aus und bestätigen Sie, dass Sie die Lizenzbestimmungen gelesen haben und starten danach das eigentliche Setup.

Schritt 1

Auf der folgenden Seite wird die PHP-Version und Rechteüberprüfung alle Ordner und Dateien überprüft. Passen Sie mit Ihren FTP-Programm ggfs die Rechte wie angegeben an.



Mögliche Fehlermeldungen

Fehler:

Die Datei / Das Verzeichniss ../redaxo/include/addons/import_export/files existiert nicht, bitte erstellen Sie es.

Lösung:

Der Ordner kann bei der Installation/Upload nicht auf Server kopiert worden sein. Erstellen Sie im Ordner "redaxo/include/addons/import_export/" einen Ordner "files" (Rechte: 777)

Und klicken Sie auf den Link ""Weiter mit Schritt 2""

Erscheint bei beiden Überprüfungen ok gehen Sie weiter zu Schritt 2.

Schritt 2

Im nächsten Schritt werden wichtige Einstellungen in die 'include/master.inc.php' geschrieben. Halten Sie hierfür die Angaben Ihres Providers bereit.

"Allgemeine Redaxo Einstellungen"

Serverdomain [optional]

Serverbezeichnung [optional]

Fehler E-Mailadresse [optional]

"Datenbankinformationen"

Name der Datenbank

MySQL Host

Login

Passwort

Redaxo-Demo
nicht eingeloggt

SETUP: SCHRITT 2 von 5 

Schreiben der 'include/master.inc.php'

// ---- Allgemein Redaxo Einstellungen

Serverdomain [optional]	<input type="text" value="redaxo.com"/>
Serverbezeichnung [optional]	<input type="text" value="Redaxo-Demo"/>
Fehler E-Mailadresse [optional]	<input type="text" value="jan.kristinus@pergopa.de"/>

// ---- Datenbankinformationen

Name der Datenbank	<input type="text"/>
MySQL Host	<input type="text"/>
Login	<input type="text"/>
Passwort	<input type="text"/>

Sollten alle Einstellungen korrekt und akzeptiert sein, geht es weiter zu Schritt 3.

Schritt 3

Mit diesem Schritt werden die Tabellen in der Datenbank neu angelegt und je nach Auswahl der Option **überschrieben!**



Redaxo-Demo
nicht eingeloggt

SETUP: SCHRITT 3 von 5

REDAXO

Datenbank anlegen

- Datenbank einrichten
- Datenbank einrichten und alte überschreiben falls vorhanden
[Vorsicht - Alte Seite wird komplett gelöscht]
- Datenbank existiert schon [Weiter ohne Datenbankimport]

Weiter zu Schritt 4

Für den ersten Start von Redaxo wählen Sie die Option **Datenbank einrichten**. Damit werden alle relevanten Tabellen die Redaxo benötigt erstellt.

Sollten Sie phpMyAdmin zur Verfügung haben, können Sie die erstellten Tabellen von Redaxo überprüfen.

Wollen Sie Ihre Datenbank nicht überschreiben wollen, dann wählen Sie Option **Datenbank existiert schon [Weiter ohne Datenbankimport]**.

Folgendes Tabellen sollten nach der Installation in den Datenbank erstellt worden sein:

```
rex__article_comment
rex__board
rex__session
rex__user
rex__user_comment
rex__user_mail
rex_action
rex_article
rex_article_slice
rex_article_type
rex_category
rex_email
rex_file
rex_file_category
rex_help
```

rex_module_action
 rex_modultyp
 rex_template
 rex_user

Schritt 4

Hier legen Sie den ersten Benutzer für Ihre Redaxo-Version fest. Mit diesen Login-Daten können Sie sich in die Redaxo -Version einloggen. Sie können als Administrator zu einem späteren Zeitpunkt die Angaben zu diesem Account ändern, löschen und neu anlegen.

The screenshot shows the 'Redaxo-Demo' interface with a 'nicht eingeloggt' status. The main heading is 'SETUP: SCHRITT 4 von 5' next to the REDAXO logo. The section is titled 'Administrator anlegen'. It contains two input fields for 'Login:' and 'Passwort:'. Below these is a checkbox labeled 'Keinen User anlegen'. At the bottom of the form is a button labeled 'Weiter zu Schritt 5'.

Schritt 5

Wenn Sie auf dieser Seite angekommen sind, haben Sie es geschafft. Redaxo ist installiert und bereit für den ersten Login. Folgen Sie dem Link oder geben Sie wie bei jedem Login die **Url/redaxo** in ihren Browser ein.

The screenshot shows the 'Redaxo-Demo' interface with a 'nicht eingeloggt' status. The main heading is 'SETUP: SCHRITT 5 von 5' next to the REDAXO logo. The content includes a congratulatory message: 'Herzlichen Glückwunsch zu Ihrem REDAXO!'. Below this is the instruction: 'Bitte noch dieses beachten:'. A numbered list follows: '1. Mit dem eingerichteten Zugang einloggen login' and '2. Importieren einer Demo über [wenn eingeloggt]: Import/Export und dort jeweils die Datenbank und Files installieren.'. At the bottom, it says 'Viel Spass und Erfolg' and 'Das REDAXO Team'.

A.2 DemoverSIONen

Ein erste Weg sich mir Redaxo vertraut zu machen, ist die Möglichkeit die mitgelieferten Demos zu installieren. In den Templates und Modulen finden Sie einen vollständige Webseite, die Ihnen die Arbeitsweise und die Möglichkeiten von Redaxo aufzeigt.

Um die mitgelieferten Demos zu installieren, wechseln Sie über die Navigation in den Bereich

Import/Export.

Ein Import besteht in der Regel aus dem Import der Datenbank (Tabellen und Inhalte) und den Dateien, die beim Export erstellt wurden. Dabei werden in der Regel die Dateien aus den Ordnern files, css, js und pics gesichert. (Mehr unter Export)

Import

Auf den linken Seite können Sie unter **Datenbankimport** die Tabellen und deren Inhalte importieren. Dazu können Sie entweder die Redaxo-Import Dateien auswählen und importieren, oder die darunter angegebenen Importe importieren. Dabei werden die bisherigen Inhalte der Datenbank gelöscht und überschrieben!

Danach können Sie unter **Dateienimport** die benötigten Dateien importieren. Nach erfolgreichem Import werden alle Artikel sowie der Cache neu generiert. Wechseln Sie danach in die Strukturverwaltung oder auf die Webseite und sehen das Ergebnis.

Export

Auf der rechten Seite finden Sie Einstellungen und Optionen für den Export der Tabellen und Dateien ihrer Webseite.

Für Sicherung der Datenbank, wählen Sie erste Option und entscheiden unten, ob Sie die Export-Datei auf dem Server speichern oder auf Ihrer Festplatte sichern wollen.

Wählen Sie die Option **Auf dem Server speichern** wird die Datei auf dem Server gespeichert und auf der linken Seite, unter *Import *angezeigt. Sie finden die auf dem Server gespeichert Datei im Ordner "redaxoincludeaddonsimport_exportfiles".

A.3 Webseiten erstellen mit Redaxo

Redaxo basiert auf PHP und Mysql. Kenntnisse in dieser Sprache und im Umgang mit der Datenbank sind zwar zu empfehlen, aber nicht unbedingt erforderlich. Anhand der Demo-Versionen kann man bereits eigene Webseiten erstellen und dabei lernen, das System zu nutzen.

Weitergehende Anpassungen an spezielle Anforderungen können mit PHP/HTML-Kenntnissen problemlos vorgenommen werden.

Viele Fragen und Probleme können mit Hilfe des Forums, des Wikis oder dieser Dokumentation bearbeitet werden. Man kann hier auf einen umfangreichen Fundus an Problemlösungen zurückgreifen.

Wenn man Redaxo ohne Demo-Versionen auf seinem Server installiert hat und nun eine Webpräsentation erstellen möchte, sind die folgenden Schritte durchzuführen. Diese Beschreibung ist nur eine Kurzfassung. Die Details sind in den Folgekapiteln beschrieben.

Schritte zur Erstellung einer Webseite mit Redaxo

1. Definition der Kategorien und Artikel
2. Erstellen eines Templates
3. Erstellen eines Stylesheets
4. Erstellen der seitenspezifischen Inhalte der Artikel (Blöcke und Module)
5. Festlegung eines Startartikels

0. Redaxo-Begriffe

Bei der Arbeit mit Redaxo werden einem die folgenden Begriffe immer wieder begegnen::

Kategorie

Artikel

Template

Module

Block/Slice

1. Definition der Kategorien und Artikel

Die einzelnen Seiten einer Präsentation werden bei Redaxo „Artikel“ genannt. Diese müssen unter der Rubrik „Struktur“ erzeugt und benannt werden. Zur besseren Gliederung und um die Navigation zu steuern, können die Artikel unterschiedlichen Kategorien zugewiesen werden. Dazu definiert man zunächst die Kategorie und legt dann in den Kategorien die zugehörigen Artikel an.

Strukturverwaltung				
Pfad : Homepage : Leistungen				
Sprachen: deutsch english				
	Kategorie	Prio	Kategorie editieren	
	..			
	CMS	1	Kategorie editieren/löschen	
	Template Design	2	Kategorie editieren/löschen	
	Artikelname	Prio	Template	Erstellt am
	Leistungen	1	Default	23. Jun. 2005

Abb.: strukturverwaltung.gif

Erstellen eines Templates

Damit ein Artikel angezeigt werden kann, braucht er ein Template, das ihm zugewiesen wird. Ein Template ist eine html-Datei ohne seitenspezifischen Inhalt. Es legt das Layout des Artikels fest. Templates werden in der eigenen Rubrik „Templates“ definiert und gespeichert. Die Zuweisung zu einem Artikel erfolgt in der Rubrik „Struktur“.

Templates			
	ID	Templatebezeichnung	Aktiv
	1	Default	ja

Abb.: templates.gif

Erstellen eines Stylesheets

Üblicherweise wird die komplette Formatierung einer Webseite in einem Stylesheet ausgelagert. Das wird auch für die Arbeit mit Redaxo empfohlen. Das Stylesheet wird wie sonst auch in html-Seiten in den head-Bereich des Templates eingebunden.

Erstellen der seitenpezifischen Inhalte für einen Artikel (Blöcke und Module)

Die Seiteninhalte werden in einzelnen Blöcken (auch Slices genannt) eingegeben. Die Anzahl der Blöcke ist beliebig. Bei dem Anlegen eines Blocks muß man ein Modul wählen, mit dem der Block editiert werden soll. Module sind „Minitemplates“, die den jeweiligen Inhaltsbereich formatieren. Die Eingabe von Inhalten mittels Modulen erleichtert die Einhaltung eines durchgängigen Layouts. Bei der „Blanko-Version“ von Redaxo sind noch keine Module enthalten. Sie können aber aus den Demoversionen oder von der Redaxo-Seite übernommen werden.

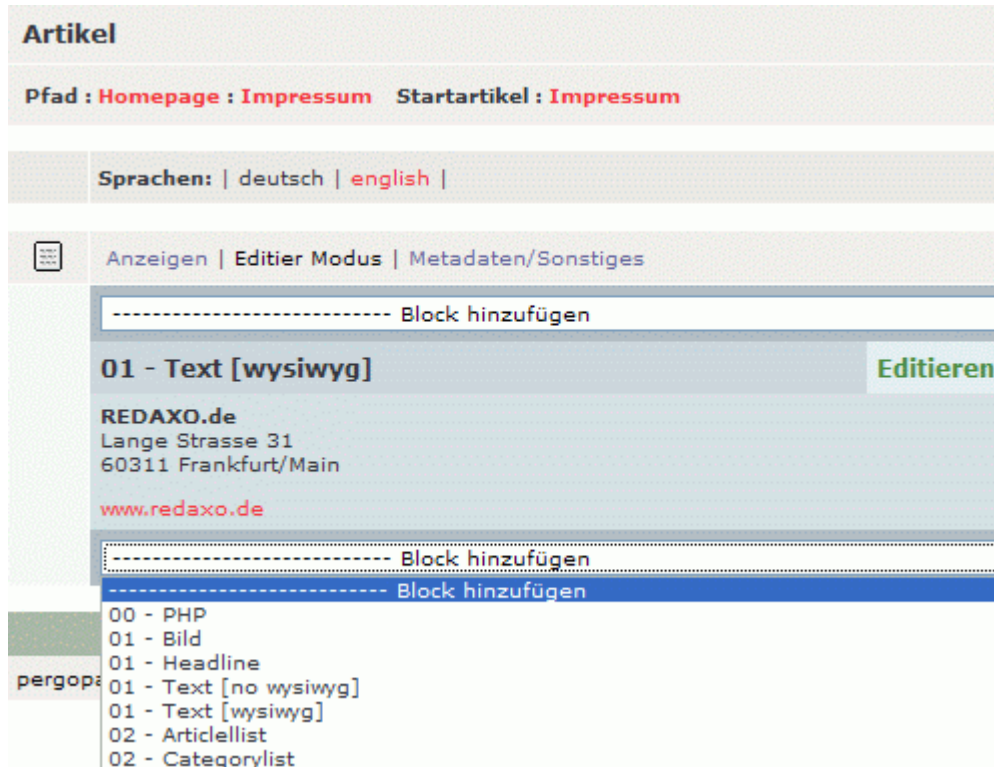


Abb.: artikel-editieren.gif

Festlegung eines Startartikels

Wenn die genannten Schritte erledigt sind, d. h., die Struktur ist festgelegt, Templates und Stylesheet sind erstellt und zugewiesen und die Blöcke der Artikel sind mit Inhalten gefüllt, dann braucht man nur noch einen Startartikel zu benennen. Dies geschieht unter der Rubrik „Specials“. Standardmäßig ist der Artikel mit der Artikel-ID 1 als Startartikel vorgesehen. Hier wird die ID des gewünschten Startartikels eingetragen. Wenn man nun die URL aufruft, sollte dieser Artikel angezeigt werden.



Abb.: startartikel.gif

B. Redaktion

Mit der Redaktion einer Webseite konzentrieren Sie sich auf die inhaltliche Verwaltung der Webseite.

Der Redakteur erhält in den meisten Fällen die Programmteile in REDAXO, die er für die Aufgabe benötigt. Alle anderen Teile werden dabei ausgeblendet. Die Einschränkung der Benutzerrechte können individuell vom Administrator oder Entwickler der Webseite vergeben werden.

In den folgenden Abschnitten finden Sie Informationen zur Redaktion einer Webseite in REDAXO.

B1. Strukturverwaltung

Nach dem erfolgreichen Login in REDAXO ist die Strukturverwaltung die erste Seite die Sie sehen. Die Strukturverwaltung ist der zentrale Bereich.

B2. Medienpool

Über den Medienpool können Dateien auf den Server geladen, verwaltet und gelöscht werden. Im allgemeinen werden darüber Grafiken und Textdateien (z. B. Stylesheets) verwaltet.

B3. Statistiken

Redaxo bietet umfangreiche Statistikfunktionen, mit denen man das Besucherverhalten für einen Webauftritt analysieren kann.

B4. Community

REDAXO bietet eine integrierte Verwaltung einer Community auf der verwalteten Webseite an.

Die in REDAXO weiterführenden, integrierte Programmteile wie

[Templates](#)

[Module](#)

[Import & Export](#)

[Addons](#)

werden im Bereich [C. Entwickler & Admin](#) beschrieben

B1. Strukturverwaltung

Artikel und Kategorie

Nach dem erfolgreichen Login in REDAXO ist die Strukturverwaltung die erste Seite die Sie sehen. Die Strukturverwaltung ist der zentrale Bereich.

Das seitenbildende Element ist der **Artikel**. Er steht in der Regel für eine Inhaltsseite ihrer Webseite. Für jeden Artikel kann ein **Template** ausgewählt werden, der den Rahmen oder das Erscheinungsbild der Webseite bestimmt. Über die Funktion **Status** kann soweit programmiert, die Seite **online und offline** geschaltet werden.

Strukturverwaltung				REDAXO			
Pfad : Homepage : HOME							
+	Kategorie	Prio	Kategorie editieren	Status/Funktion			
	..						
+	Artikelname	Prio	Template		Status/Funktion		
	Home	1	01 Standard	Startartikel	ändern	Löschen	online
	Nutzungsbeding	2	01 Standard	Artikel	ändern	Löschen	online
	Registrierung	3	01 Standard	Artikel	ändern	Löschen	online

Abb.: artikel.gif

Die Verwaltung der Artikel erfolgt über die **Kategorien**. In REDAXO wird eine hierarchische Struktur eingesetzt. D.h. Verschiedene Kategorien (Ordner) enthalten verschiedene Artikel (Inhalte). Jede Kategorie hat einen **Startartikel** (Startartikel sind Einstiegsseiten einer Kategorie- im Normalfall Übersichtsseiten).

Normale Artikel können in beliebiger Anzahl erstellt werden. Einem Artikel kann man ein Template zuweisen welches die Darstellungsform bestimmt. Ein Artikel besteht aus mehreren **Blöcken** und repräsentiert den eigentlichen Inhalt. Die Erstellung dieser Blöcke basiert auf **Modulen**. Mittels der Module werden Eingabemasken für Textbausteine, Bilder u. a. definiert und die Anzeige der dort eingegebenen Inhalte formatiert.

Weiterhin hat der Artikel Metadaten die ihn allgemein beschreiben (z.B. Kurzbeschreibung, Suchbegriff und Grafik).

Kategorien können auch in der Form genutzt werden, um Navigationsstrukturen abzubilden. Für die meisten Fälle gilt: Was in der Struktur zu sehen ist, sieht man auch auf der Sitemap und in der Navigation. Die Priorität organisiert die Reihenfolge.

B1.01 Kategorien

Kategorien werden, wie in einem Explorer oder Finder, zur Strukturierung und Verwaltung der Artikel erstellt.

Erstellen (1)

Zum Erstellen einer neuen Kategorie, klicken Sie auf (+)-Symbol und geben Sie den Namen der Kategorie ein und speichern Sie Ihre Eingabe über die Schaltfläche "edit_category" ab. Die so ersellte Kategorie ist in den **Funktion offline** gestellt. Für die Verwaltung hat die Einstellung keinerlei Funktion.

Funktion/Status (2)

Diese Einstellung kann vom Entwickler ausgelesen und genutzt werden. Neben der Verwaltung können Kategorien die Seitennavigation Ihrer Webseite bilden. Über die oben beschriebene Funktion "online/offline" kann der Name sichtbar geschaltet werden. Diese Funktion wird in der Regel bei der Programmierung eines Templates genutzt.

Weitere Information hierzu finden Sie im Bereich [C. Entwickler & Admin/C1.02 Navigation](#)

Jede Kategorie enthält nach dem Erstellen immer einen Startartikel, der nicht gelöscht oder in der **Funktion offline/online** verändert werden kann.

Wechseln Sie in die gewünschte Kategorie, indem Sie auf den Namen der Kategorie klicken. Die Kategorie wird danach mit den enthaltenen Artikeln angezeigt.

Prio (3)

Über die Einstellung Prio kann die Reihenfolge der Kategorien in der Strukturverwaltung verändert werden. Die Reihenfolge kann auch bei der Ausgabe der Kategorien auf der Webseite genutzt werden.



Strukturverwaltung		REDAXO		
Pfad : Homepage				
 Kategorie	Prio	Kategorie editieren	Status/Funktion	
 HOME	1	Kategorie editieren/löschen	online	
 NEWS	2	Kategorie editieren/löschen	online	
 BOARDS	3	Kategorie editieren/löschen	online	
 OPEN BOARD	4	Kategorie editieren/löschen	online	
 USERLISTE	5	Kategorie editieren/löschen	online	
	6	Kategorie editieren/löschen	online	
 MEIN PROFIL	7	Kategorie editieren/löschen	online	

Abb.: struktur.gif

B1.02 Artikel

Über einen Artikel können Sie Inhalte einbinden und auf der Webseite erscheinen lassen. Wie werden nun Artikel in REDAXO erstellt und verwaltet?

Erstellen

Über das (+)Symbol können Sie einen Artikel in der gewünschten Kategorie erstellen. Der Startartikel ist in jeder Kategorie grundsätzlich vorhanden und kann nicht gelöscht werden.

Darstellung – Template

Jeder Artikel und dessen Inhalte wird in der Regel in ein Template, dem "Rahmen" der Webseite, geladen. Dazu wird über Status/Funktion die Schaltfläche **ändern** aktiviert und danach das gewünschte Template für diesen Bereich oder je nach Funktion und Anzeige, aus der Templateliste ausgewählt. Daneben kann die Reihenfolge in Verwaltung über die Einstellung **Prio** sowie der Artikelname editiert werden. Über die Schaltfläche **Editieren** werden der Artikel mit dem ausgewählten Template gespeichert und angezeigt.

Funktion/Status – online/offline

Wie auch bei der Verwaltung der Kategorien, kann die Funktion in der Programmierung genutzt werden. Artikel können hier z.B. erst angezeigt werden, wenn die Funktion auf "online" gesetzt wird. Für weitere Information zur Verwendung dieser Funktion mit der Programmierung der Templates, erfahren Sie Bereich [C. Entwickler & Admin](#)

Artikel – Inhalt und Metainformationen

Über das Artikel-Icon (Abbildung) oder den Artikelnamen können Sie zu der Verwaltung der Inhalte eines Artikel wechseln. Hier wird der Artikel im **Editiermodus** geöffnet, d.h. hier werden alle Inhalte in **Blöcken** dargestellt und verwaltet. Wechseln Sie zur **Voransicht** und die Inhalte werden ohne die Darstellung in Blocks angezeigt.

Sie haben bei der Erstellung eines Artikels, die Möglichkeit, die Inhalte in einem oder mehreren Blöcken zu verwalten. Mit einer Auswahl an Blöcken wie Headline, Text, Bild oder Links stehen in eine flexible und freie Redaktion der Inhalte zur Verfügung. Sie können vor, zwischen und hinter jedem Block einen weiteren Block aus der Blockliste auswählen und einfügen.

Mehr zu Blöcken und Modulen erfahren Sie im nächsten Kapitel [B1.03 Block/Modul](#)

Metadaten

Die Informationen, Eingaben und Einstellungen dieser Seite können in der Programmierung genutzt und für Anzeige und Verwendung auf der Webseite verwendet werden.

Metadaten/Sonstige Funktionen

Ist das Recht **advancedMode[]** beim bearbeitenden User vergeben, werden erweiterte Funktionen unter Metadaten angezeigt. Hier kann dann ein Artikel kopiert oder verschoben werden.

Mehr zu der Verwendung die Einstellungen in der Programmierung erfahren Sie im Bereich [C. Entwickler & Admin](#)

B1.03 Block/Modul

Die eigentlichen Inhalte werden aus Modulen zusammengebaut. Man könnte sie auch Minitemplates nennen, wobei eine unendliche Anzahl von diesen in einem Artikel möglich sind. Diese Module können sehr unterschiedlich sein. Mögliche Formen eines Moduls sind z.B. Headline, Fliesstext, Grafiken hochladen und darstellen oder dynamische Listen aus Datenbanken, dynamische Grafiken, Unternavigationen etc.

Wie wird ein Modul/Block (Inhalt) einem Artikel zugefügt?

Öffnen Sie den Artikel, in dem Sie ihre inhaltlichen Änderungen hinzufügen wollen, über das Artikelicon oder den Artikelnamen. Wählen Sie aus der Modulliste ein Modul aus und editieren Sie nun nach den Angaben ihren Inhalt, z.b geben Sie einen Text in ein Feld ein.



Abb.: article_select_modul.jpg

Über **Block hinzufügen** wird das Modul zu einem **Block** in dem Artikel. Der Artikel wird nun mit dem hinzugefügten Block erneut geladen. Über **Editieren** können Sie den Inhalt editieren und erneut speichern. Über **Löschen** können Sie den Block aus dem Artikel entfernen.



Abb.: modul_addcontent.jpg

Wie wird die Modulliste erstellt?

Die Modulliste wird in der Regel vom Entwickler im Bereich **Module** erstellt. Im Bereich Module wird die Eingabe (Ansicht in der Redaktion) sowie die Ausgabe (Darstellung Webseite) vom Entwickler definiert und gestaltet.


Module				
Modules Actions				
+	Modulbezeichnung	Funktionen	PHP	HTML
☐	00.01 - php	Modul löschen		
☐	01.01 - Board	Modul löschen		
☐	01.02 - Boardteaser	Modul löschen		
☐	02.01 - Fliesstext [NO WYSIWYG]	Modul löschen		
☐	02.02 - Fliesstext [WYSIWYG]	Modul löschen		
☐	02.03 - Headline	Modul löschen		
☐	03.01 - Artikelliste	Modul löschen		
☐	03.02 - Artikeliste als Kalender	Modul löschen		
☐	04.01 - Artikelkommentarbox	Modul löschen		
☐	05.01 - Bild	Modul löschen		

Abb.: modul_list.jpg

B2. Medienpool

Über den Medienpool können Dateien auf den Server geladen, verwaltet und gelöscht werden. Im allgemeinen werden darüber Grafiken und Textdateien (z. B. Stylesheets) verwaltet. Es können aber auch beliebige andere Dateitypen bearbeitet werden.

Der Medienpool wird in einem Popup-Fenster aufgerufen.

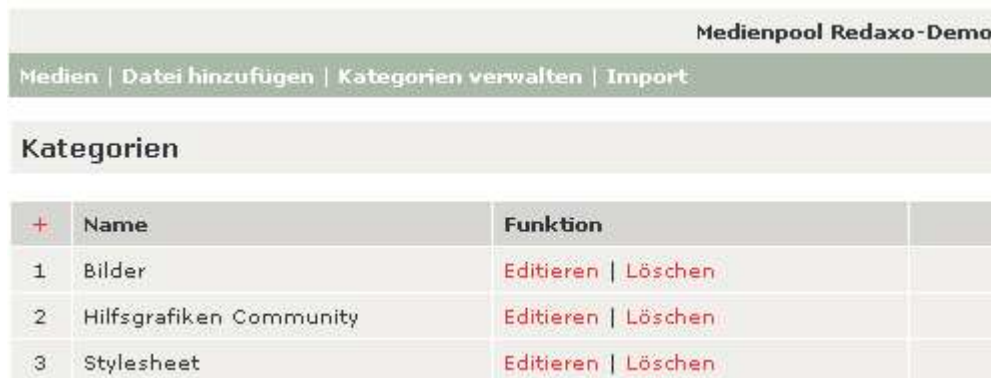
B2.01 Medien verwalten

Alle Dateien, die mit dem Medienpool verwaltet werden, befinden sich in dem Ordner "files". In dem Medienpool werden nur die Dateien angezeigt, die über den Medienpool in das Verzeichnis geladen wurden.

Um die Sortierung übersichtlicher zu gestalten, kann man Kategorien definieren und die hochgeladenen Dateien diesen Kategorien zuweisen. In der Demo-Version von Redaxo wird z. B. zwischen Bildern, Hilfsgrafiken und Stylesheets unterschieden. Es können beliebig viele Kategorien definiert werden.

Kategorien verwalten

Unter der Überschrift "Kategorien verwalten" können neue Kategorien angelegt und vorhandene editiert oder gelöscht werden. Zum Anlegen einer neuen Kategorie klickt man auf das Kreuzsymbol neben der Bezeichnung "Name" und trägt den Kategorienamen in das entsprechende Textfeld ein. Das Editieren beschränkt sich auf eine Umbenennung der Kategorie. Kategorien können nur gelöscht werden, wenn ihnen keine Dateien mehr zugewiesen sind. Ansonsten erscheint ein Warnhinweis.



Medienpool Redaxo-Demo			
Medien Datei hinzufügen Kategorien verwalten Import			
Kategorien			
+	Name	Funktion	
1	Bilder	Editieren Löschen	
2	Hilfsgrafiken Community	Editieren Löschen	
3	Stylesheet	Editieren Löschen	

Abb.: mp_kat.jpg

Dateien hinzufügen

Zum Hochladen von Dateien in den Medienpool bzw. das Verzeichnis "files", wählt man den Menüpunkt "Datei hinzufügen". Hier können dann zu der Datei ergänzende Informationen angegeben werden, wie z. B. Titel, Beschreibung, Copyright.

Abb.: mp_addfile.jpg

Medienpool Redaxo-Demo

Medien | Datei hinzufügen | Kategorien verwalten | Import

Datei hinzufügen

Titel :
 Kategorie :
 Beschreibung :
 Copyright :
 Datei:

Medien: Übersicht

Zu jeder Kategorie kann man sich die entsprechenden Dateien anzeigen lassen. Durch Klick auf den Bildtitel erhält man die Detailinformationen.

Medienpool Redaxo-Demo

Medien | Datei hinzufügen | Kategorien verwalten | Import

Medien

Kategorien

	Thumbnail	Dateiinfo	Beschreibung	Funktionen
<input type="checkbox"/>		[Kein Titel eingegeben] logo.gif 1.29 KBytes 21-Jul-2004 15:55h	[Keine Beschreibung eingegeben]	-
<input type="checkbox"/>		[Kein Titel eingegeben] hl_login.gif 289 Bytes 21-Jul-2004 15:54h	[Keine Beschreibung eingegeben]	-

Abb.: mp_menuue.jpg

Medien – Detailansicht

Klicken Sie in der Übersicht einer Kategorie auf eine Datei und man gelangt zu Detailansicht. Über die Detailansicht können die Kategorie wechseln, die Datei ändern, Beschreibungs- und Copyrighttexte einfügen oder die Datei löschen. Über die Schaltfläche "Aktualisieren" werden die Änderungen übernommen.

Datei können aus dem Medienpool nur gelöscht werden, wenn die Datei nicht in einem Artikel/Block eingefügt ist.

B3. Statistiken

Redaxo bietet umfangreiche Statistikfunktionen, mit denen man das Besucherverhalten für einen Webauftritt analysieren kann.

Im Einzelnen werden ausgewertet und angezeigt:

Zeitliche Verteilung der Visits

- Anzahl der Besuche gelistet nach Tagen oder Monaten,
- die Anzahl der Pageviews und
- die durchschnittliche Anzahl der Pageviews pro Besuch

Besuchte Artikel

- Verteilung der Pageviews pro Artikel,
- die am häufigsten und
- die selten aufgerufenen Artikel

Aussagen über die Besucher

- Herkunft der Besucher sortiert nach Land,
- die Häufigkeit von Suchmaschinenzugriffen,
- die Suchbegriffe, die zu der Seite geführt haben und die Referer-URLs der Besucher

Aussagen über die Browsertypen

- Verteilung der Browser, mit denen auf die Seite aufgerufen wurde und
- die Verteilung der entsprechenden Betriebssysteme

Auswertung erstellen

Um die Statistik zu erhalten, müssen die Daten zunächst ausgewertet werden. Dazu wählt man unter dem Punkt "Auswertung" den gewünschten Monat und bestätigt durch Anklicken des Buttons "Auswerten". Anschließend kann die neu erstellte oder aktualisierte Statistik unter dem Punkt "Statistik (rot)" angezeigt werden.

B4. Community

REDAXO bietet eine integrierte Verwaltung einer Community auf der verwalteten Webseite an. Sie können User für geschlossene Bereich anlegen und verwalten, eingetragene Newsletter-Abonnten einsehen, Newsletter im Plain-Text-Format versenden, Foren administrieren sowie Artikelkommentare verwalten.

Diese Funktionen und Möglichkeiten liegen in REDAXO vor, müssen aber über einen Entwickler der Webseite bereitgestellt und angepasst werden. REDAXO bietet die Basis für die einfache und flexible Verwaltung.

Aus der mitgelieferten Demo-Community können Sie Möglichkeiten und Anforderungen kennenlernen.

B4.01 Userverwaltung

Über die Userverwaltung können Sie eingetragene User in Ihrer Community, in REDAXO verwalten.

Wie unter [B4. Newsletter](#) beschrieben, können z.B. Newsletter-Abos hier verwaltet und editiert werden.

Sie können auf dieser Seite, manuell User anlegen und nach bestehenden User suchen und die Ergebnisse anzeigen.

Darüber hinaus lässt sich auch eine Community, mit Foren oder Geschlossenen Bereichen verwalten und erweitern.

Die Tabelle `rex__user`, in der alle User eingetragen werden, bietet mit einer Reihe von Attributen, eine detaillierte Beschreibung der Userdaten und Detailinformationen. Diese Attribute können von einen Entwickler verändert und über Formular gefüllt werden.

Weiterführende Informationen zu der Erstellung einer Webseite mit Community, können Sie aus der mitgelieferten Demos ersehen.

B4.02 Artikelkommentarverwaltung

Über diese Verwaltung können z.B. Kommentare zu einem Artikel in einem Blog verwaltet werden. Die Kommentare können nach Namen und Status des Kommentare durchsucht werden.

Weitere Informationen zur Nutzung und Programmierung der Artikelkommentare werden an dieser Stelle veröffentlicht.

B4.03 Newsletter

Auf der Seite Newsletter kann an die User, die über die [Userverwaltung](#) erstellt wurden, ein Newsletter im Plain-Text versendet werden. Dazu muss noch bei jedem User, ein Haken bei der Einstellung Newsletter gesetzt werden.

Mehr Information zu der Verwaltung von Userdaten lesen im Kapitel [Userverwaltung](#).

User über Newsletter-Formular erstellen

Alle User, die Sie in über die Userverwaltung einsehen können, werden mit den relevanten Werten in der Tabelle rex__user gespeichert.

Über ein Formular auf ihrer Webseite können Sie User in diese Tabelle hinzufügen und danach über Userverwaltung einsehen und verwalten.

Erstellen Sie ein Formular und verwerten Sie danach die gesendete Informationen.

```
<?
$news = new sql;
$news ->setTable("rex__user");
$news ->setValue("id",'');
$news ->setValue("name",$FORM[name]);
$news ->setValue("email",$FORM[email]);
$news ->setValue("stamp",time());
$news ->insert();
?>
```

C. Entwickler & Admin

In diesem Bereich finden Sie als Entwickler oder Administrator weiterführende Informationen zur Entwicklung und Programmierung einer Webseite.

Lernen Sie das Konzept von REDAXO kennen und erweitern so die Funktionen im Backend und auf Ihrer Webseite.

C1. Templates erstellen

Mit Hilfe eines Templates wird das Layout einer Seite (eines Artikels) festgelegt. Hier wird die generelle Struktur des Artikels einschließlich der Navigation ...

C2. Module & Aktionen

Mehr zur Entwicklung und Bereitstellung von Modulen und Aktionen ...

C3. Benutzer

Über die Erstellung verschiedenen Benutzer-Account, regeln Sie auch die Rechtevergabe in REDAXO.

C4. Addons

Addons erweitern den Funktionsumfang von REDAXO. Das Addon-Konzept erlaubt es REDAXO nach Ihren Wünschen zu erweitern ...

C5. Specials

Auf dieser Seite stehen Informationen zur Verfügung, die im sich auf die Einstellungen in der master.inc.php beziehen. Daneben können Sie drei weitere Funktionen aktivieren ...

C6. Object Oriented Framework

Sie können auf die Informationen in den Tabellen die REDAXO verwendet über SQL-Befehl zugreifen oder wie auf den folgenden Seite beschrieben, über ein Objekt-Orientiertes Framework.

C7. Import & Export

Ein Import besteht in der Regel aus dem Import der Datenbank (Tabellen und Inhalte) und den Dateien, die beim Export erstellt wurden.

C8. CVS

Im CVS können Sie aktuelle Entwicklungsversionen laden.

C1. Templates

Mit Hilfe eines Templates wird das Layout einer Seite (eines Artikels) festgelegt. Hier wird die generelle Struktur des Artikels einschließlich der Navigation definiert. Metatags, Stylesheets und Javascript können eingebunden werden.

Alle Artikel, denen ein bestimmtes Template zugewiesen wird, erhalten dadurch ein einheitliches Layout. Der artikelspezifische Inhalt wird jeweils über die Funktion `$this->getArticle()` eingefügt.

Es können Templates erstellt werden, die in den Artikeln selektiert werden sollen, oder welche, die in andere Templates eingebunden werden sollen.

C1.01 Template erstellen

Im einfachsten Fall besteht ein Template aus einer leeren html-Seite und einer Referenz auf den aktuellen Artikel:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<meta name="keywords" content="wort1, wort2, ...">
<link rel="stylesheet" type="text/css" href="css/style.css">
</head>

<body>

<?
/* hier wird der artikelspezifische Inhalt eingebunden;
$this verweist dabei auf den jeweils aktuellen Artikel */

echo $this->getArticle();

?>

</body>
</html>
```

Dieses Beispiel kann beliebig variiert werden. Das Layout wird in einem Template üblicherweise detaillierter definiert. So können hier unterschiedliche Seitenbereiche, z. B. Kopf- und Fußzeilen oder mehrere Spalten formatiert und die Navigation eingebunden werden. Der artikelspezifische Inhalt wird in einem beliebigen Bereich/Container eingefügt. Es ist auch möglich, die Inhalte mehrerer Artikel in einem Template anzuzeigen (s. Kap. C1.03).

C1.02 Navigation

Es gibt unterschiedliche Möglichkeiten, eine Navigation festzulegen:

Externe Links

Template

Links auf externe Seiten werden in einem Template in der üblichen Form festgelegt:

```
<a href=http://www.redaxo.de> Redaxo-Hompage </a>
```

Modul

In den Modulen hängt die Schreibweise für einen externen Link davon ab, wie die Ein- und Ausgabe definiert sind, welche Redaxo-Variablen verwendet werden und ob eine automatische Umwandlung bestimmter Signalwörter vorgesehen ist. Hier muß berücksichtigt werden, mit welchem Modul man aktuell arbeitet.

Interne Links, manuell definiert

Template

Soll in einem Template auf interne Seiten verlinkt werden, wird die index-Seite als Adresse angegeben und die gewünschte Artikel-ID als Variable übergeben.

```
<a href=index.php?article_id=22> Interner Link Artikel 22</a>
```

Eine andere Möglichkeit stellt folgende Schreibweise dar:

```
<a href="< ?= getURLbyID(22)"> Interner Link Artikel 22</a>
```

Dann funktionieren die Links auch wenn RexRewrite aktiv ist.

Siehe Wiki: TextUrlsOhneModRewrite

In den folgenden Beispielen stellen wir zwei Navigations-Templates vor:

Navigation-Beispiel 1

Navigation-Beispiel 2

Module

Bei Verwendung eines Moduls werden der Name und die ID eines Artikels als Formularwerte übergeben. Dem Formular wird der Name Link[1] zugewiesen.

Die Ausgabe erfolgt in dem Modul mittels der Redaxo-Variablen REX_VALUE[1]

Interne Links, dynamisch erzeugt

Statt einer manuell erzeugten Navigation kann man diese auch dynamisch erstellen lassen. Dazu bieten sich mehrere Möglichkeiten. Man kann sowohl das OO-Konzept von Redaxo nutzen als auch die Struktur mit Hilfe von SQL-Befehlen ermitteln und darstellen. Siehe dazu die Beispiele in den Folgekapiteln.

Sitemap, dynamisch generieren

Nach dem gleichen Prinzip kann man eine Sitemap dynamisch generieren lassen. Das hat den Vorteil, dass die Sitemap bei Änderungen an der Kategorie- oder Artikelstruktur automatisch aktualisiert wird. Siehe dazu das Beispiel in dem Kapitel "Sitemap".

C1.02.01 Navigation für bis zu drei Navigationsebenen

Mit diesem Quellcode werden bis zu drei Ebenen von Kategorien aufgelistet. Es werden alle Kategorien und Unterkategorien angezeigt. Eine Unterscheidung je nach Status "online" oder "offline" wird hier nicht gemacht.

Dies ist nur ein Beispiel, das den eigenen Anforderungen angepasst werden kann. Zur Erzeugung der Navigation werden Elemente des Objekt-Orientierten Frameworks genutzt:

```

OOCategory::getRootCategories() as $cat
//Es werden die Kategorien der ersten Ebene ausgelesen.
$this->getUrl()
//Für die Verlinkung wird die URL des Startartikels, der zu dieser Kategorie gehört, wird ausgelesen.
$this->getName()
//Es wird der Name der Kategorie ausgelesen.
$this->getChildren()
//Für eine bestimmte Kategorie werden die Unterkategorien ausgelesen.

<ul>
<!-- top level categories -->
<?foreach (OOCategory::getRootCategories() as $cat):?>
<li><a href='<?=$cat->getUrl()?>'><?=$cat->getName()?></a><ul>

<!-- 1st level categories -->
<?foreach ($cat->getChildren() as $sub1):?>
<li><a href='<?=$sub1->getUrl()?>'><?=$sub1->getName()?></a><ul>

<!-- 2nd level categories -->
<?foreach ($sub1->getChildren() as $sub2):?>
<li><a href='<?=$sub2->getUrl()?>'><?=$sub2->getName()?></a></li>
<?endforeach;?>

</ul></li>
<?endforeach;?>
</ul></li>
<?endforeach;?>
</ul>

```

C1.02.02 Navigation mit Unternavigations-Links

Dieses Listing generiert automatisch eine Navigation mit den entsprechenden Unternavigations-Links.

```
menupunkt1
    unterpunkt11
    unterpunkt12
    unterpunkt13

menupunkt2
    unterpunkt21
    unterpunkt22
```

Es werden nur die Kategorien angezeigt, deren Status "online" ist.

Dies wird mit der Bedingung "if (\$lev1->isOnline())" abgefragt.

```
<?php
// EXPLODE PATH
$PATH = explode("-", $this->getValue("path"));

// GET CURRENTS
$path1 = $PATH[1];
$path2 = $PATH[2];
$path3 = $PATH[3];

echo '<ul class="nav1st">';

/* START 1st level categories */
foreach (OOCategory::getRootCategories() as $lev1):

if ($lev1->isOnline()):

/* 1st level - active link */
if ($lev1->getId() == $path1) {
echo '<li class="active">'. $lev1->getName();
}

/* 1st level - no active link */
else {
echo
'<li>'. $lev1->getName();
}

/* 1st level had categories? -> go on */
$lev1Size = sizeof($lev1->getChildren());

if($lev1Size != "0"):
echo '<ul class="nav2nd">';

/* START 2nd level categories */
foreach ($lev1->getChildren() as $lev2):
```

```
if ($lev2->isOnline()):
/* 2nd level - active link */
if ($lev2->getId() == $path2) {
echo
'<li class="active"><a class="current" href="'. $lev2->getUrl()."'>' . $lev2->getName(). '</a>';
}

/* 2nd level - no active link */
else {
echo
'<li><a href="'. $lev2->getUrl()."'>' . $lev2->getName(). '</a>';
}

/* 2nd level had categories? -> go on */
$lev2Size = sizeof($lev2->getChildren());

if($lev2Size != "0"):
echo '<ul class="nav3rd">';

/* START 3rd level categories */
foreach ($lev2->getChildren() as $lev3):

if ($lev3->isOnline()):

/* 3rd level - active link */
if ($lev3->getId() == $path3) {
echo '<li class="active"><a class="current" href="'. $lev3->getUrl()."'>' . $lev3->getName(). '</a></li>';
}

/* 3rd level - no active link */
else {
echo '<li><a href="'. $lev3->getUrl()."'>' . $lev3->getName(). '</a></li>';
}

endif;
endforeach;
/* END 3rd level categories */

echo '</ul>';
endif;

echo '</li>';
endif;

endforeach;
/* END 2nd level categories */

echo '</ul>';
endif;

echo '</li>';
endif;

endforeach; /* END 1st level categories */
echo '</ul>'; ?>
```

C1.03 Mehrere Artikel einbinden

Der Inhalt des aktuell aufgerufenen Artikels wird durch Aufruf der Funktion `$this->getArticle()` eingefügt.

Es können darüber hinaus Inhalte beliebiger weiterer Artikel eingefügt werden. Dies geschieht, indem ein neuer Artikel deklariert und diesem eine bestimmte Artikel-ID zugewiesen wird. Die ID gehört zu dem Artikel, dessen Inhalt hier angezeigt werden soll. Er muß in der Struktur definiert sein. In diesem Artikel können Inhalte gespeichert werden, die auf mehreren Seiten angezeigt werden sollen, z. B. die Navigation oder eine News-Liste.

Der Inhalt wird ebenfalls durch Aufruf der Funktion `$artikel_2->getArticle` eingefügt.

Es ist möglich, einen zusätzlichen Artikel mehrfach in eine Seite einzubinden oder mehrere unterschiedliche Artikel innerhalb einer Seite einzubinden.

Das folgende Listing soll das das Prinzip erläutern. Das Einbinden in Tabellenfelder ist nur ein Beispiel.

```
<?
/* hier wird der Inhalt eingefügt, der zu dem aktuellen Artikel gehört */

echo "<table><tr><td>".$this->getArticle()."</td>";

/* hier wird ein weiterer Artikel eingebunden */

$artikel_2 = new article;
$artikel_2->setArticleID(123);

echo "<td>".$artikel_2->getArticle()."</td></tr></table>";
?>
```

C1.04 Seitenspezifische Metatags

Redaxo bietet die Möglichkeit, für jeden Artikel spezifische Metaangaben zu definieren und auszugeben. Dadurch können die Seiten von den Suchmaschinen gezielter gefunden werden.

Die Festlegung dieser Meta-Informationen erfolgt in der Kategorie "Struktur". Der gewünschte Artikel wird ausgewählt und die Bearbeitungsmaske "Metadaten/Sonstiges" aufgerufen. Hier können u.a. Angaben für die Beschreibung und artikelspezifische Suchbegriffe festgelegt werden.

Zum Aufrufen dieser Informationen werden in dem Template folgende Befehle eingefügt:

```
// Einbinden von Suchbegriffen
<meta name="keywords" content="
<?=$this->getValue("suchbegriffe")?> ">

//Einbinden der Beschreibung
<meta name="description" content="
<?=$this->getValue("beschreibung")?> ">
```

C1.05 Includes

Es besteht die Möglichkeit, Programmcode in einer separaten Text-Datei auszulagern und diesen dann mittels include()-Befehl an einer bestimmten Stelle in ein Template einzufügen.

Die include-Datei kann beliebigen php-Code enthalten. Sinnvollerweise sollte sie in dem für includes vorgesehen Ordner redaxo/include gespeichert werden. Wenn sie in einem anderen Ordner abgelegt wird, muß bei dem include-Befehl die Pfadangabe entsprechen angepasst werden. Nach einer allgemeinen Übereinkunft werden include-Dateien mit der Endung .inc.php gespeichert.

Ein Beispiel für den Einsatz von includes ist das Auslagern einer Navigation in einem Template. Durch die separate Ablage des Codes wird das Template übersichtlicher. Das Layout kann dadurch leichter bearbeitet werden.

Ein einfaches Beispiel:

```
<div>
<?
// hier wird der ausgelagerte Code eingelesen\
include($REX[INCLUDE_PATH].'/navigation.inc.php');
?>
</div>
```

```
<div>
<?
// hier wird der Seiten-Inhalt eingelesen
$this->getArticle();
?>
</div>
```

Je nach Erfordernis kann nach diesem Prinzip Programmcode ausgelagert und wieder eingelesen werden.

Wie in den Demos gezeigt, ist es auch möglich, andere Templates, die Sie in Redaxo erstellt haben, einzulesen. In diesem Beispiel wird ein bereits generiertes Template eingelesen. Infos zu generiertes Dateien in REDAXO ...

```
<?
//Einlesen eines weiteren Templates
include $REX[INCLUDE_PATH].'/generated/templates/2.template";
?>
```

C2. Module & Aktionen

Bevor Sie sich mit der Entwicklung und Bereitstellung von Modulen und Aktionen beschäftigen, sollten Sie sich mit dem Konzept, der Verwendung und sowie Verarbeitung der Inhalte über Module und Blöcke vertraut machen. (B1.03 Block/Modul)

Module

Im Bereich Module öffnet sich eine Liste mit allen Modulen die z.B. in der installierten Demo vorhanden sind. Über (+)-Symbol können Sie eigene Module erstellen und so für die Redaktion der Inhalte, in den Artikeln zur Verfügung stellen.

Eingabe und Ausgabe

Öffnen oder erstellen Sie ein neues Modul und Sie erkennen das einfach Prinzip eines Moduls. Die Eingabe definiert dabei die Eingabemaske, die Sie als Redakteur sehen, wenn Sie einen Block hinzufügen oder über Editieren bearbeiten wollen. In der Ausgabe können Sie Ausgabe oder Darstellung auf der Webseite, der eingegebenen Inhalte definieren.

In beiden Felder können Sie die Anzeigen über HTML und PHP erstellen.

Für die Speicherung der verschiedenen Inhalte stehen Ihnen in REDAXO eigene Variablen zur Verfügung.

Aktionen

Wechseln Sie über die Subnavigation zu Actions und Sie erhalten eine Auflistung aller zur Verfügung stehenden Aktionen.

Aktionen werden in Verbindung mit Modulen genutzt. Wollen Sie z.B. beim Editieren eines Blocks, die Eingabe eines Textfeldes überprüfen, dann können Sie dies mit Hilfe einer Aktion tun. Die Eingabe kann, je nach Definition und Programmierung der Aktion, überprüft, korrigiert und verändert werden.

Die Definition und Funktionen einer Aktion erfolgt in einer einfachen Eingabemaske. Stehen Aktionen zur Auswahl, so wird bei der Erstellung oder Bearbeitung eines Moduls, diese Auswahl angezeigt kann dem Modul hinzugefügt werden.

Backend: Blöcke – Slices

Die über ein Modul eingegeben Inhalte werden in der Tabelle "rex_article_slice" gespeichert. Die Zuordnung zu einem Artikel, wird über das Feld "article_id" autom. eingetragen. Die Reihenfolge wird über das Feld "re_article_slice_id" geregelt. Dabei wird in diesem Feld, beim Hinzufügen eines neuen Blocks im Artikel, der darüber stehende Block mit dem Wert aus dem Feld "ID" eingetragen. Die Slices bilden so eine Kette. Bei der Ausgabe wird die Kette aufgelöst und als Seite dargestellt.

Über das Feld "article_id" können somit z.B. einzelnen Slices ausgelesen und dargestellt werden.

C2.01 Modul erstellen und bereitstellen

Bei der Erstellung eines Moduls, haben Sie Möglichkeit, zwei Einteilungen zu verwenden.

Eingabe

Einerseits das Eingabeformular eines Moduls. Mit diesem kann die Eingabemaske erstellt werden, die beim Editieren oder Hinzufügen einen Blocks im Artikel erscheint.

Ausgabe

Andererseits das Ausgabeformular eines Moduls. Mit diesem kann die Ausgabe der eingegebenen Informationen angezeigt werden.

Eine einfache Form eines Modules:

Eingabe:

```
<input type=text size=20 name=VALUE[1] value="REX_VALUE[1]">
```

Ausgabe:

```
<font size=3>REX_VALUE[1]</font>
```

Dieses Modul erfasst einen Text, welcher dann formatiert ausgegeben wird. Allgemein werden bestimmte Platzhalter verwendet um unterschiedliche Texte, Grafiken etc. dann dafür wieder einsetzen zu können. Hier wird im der Platzhalter REX_VALUE-[1]- verwendet.

Ein einfaches Modul zum Bildupload mit Bildbeschreibung:

Eingabe:

```
<?
echo "Bild: <br>";
if ("FILE[1]" != "")
{
echo "<img src=../files/FILE[1] width=100><br>";
echo "<input type=checkbox name=FILEDEL1> delete";
}else
{
echo "<INPUT NAME=FILE1 TYPE=file size=2>";
}
?><br><br>Bildunterschrift:<br>
<input type=text size=50 name=VALUE[1] value="REX_VALUE[1]" class=inp100>
<br><br>
```

Ausgabe:

```
<?
if ("FILE[1]" != "") echo "<img src=/files/FILE[1]>";
if ("REX_VALUE[1]" != "") echo "<br>REX_VALUE[1]";
echo "<br><br>";
?>
```

In diesem Modul hat der Redakteur die Möglichkeit ein Bild hochzuladen und die Bildunterschrift einzugeben. Dabei werden folgende Platzhalter verwendet. FILE-[1]-, REX_VALUE-[1]- sowie feste Formularwerte wie VALUE-[1]-, FILE1?, FILEDEL1??. Sie können innerhalb der Modul HTML oder PHP verwenden.

Eine Auflistung der REDAXO Variablen oder Platzhalten finden Sie hier

C2.02 REDAXO Variablen

Wie in Abschnitt C2.01. Module erstellen beschrieben folgt hier eine Auflistung und Beschreibung von REDAXO-Variablen.

REX_VALUE[..] | VALUE[..]

Enthält Text-Eingaben, bis zu 10 Felder (Wie in Abschnitt C2.01. Module erstellen beschrieben folgt hier eine Auflistung und Beschreibung von REDAXO-Variablen.

REX_VALUE[..] | VALUE[..]

Enthält Text-Eingaben, bis zu 10 Felder (REX_VALUE[1]..) möglich...) möglich.

Beispiel

Modul-Eingabe:

```
<input type=text size=20 name=VALUE[1] value="REX_VALUE[1]">
```

Modul-Ausgabe:

```
<font size=3>REX_VALUE[1]</font>
```

REX_HTML_VALUE[..]

...

Beispiel

Modul-Eingabe:

```
<input type=text size=20 name=VALUE[1] value="REX_VALUE[1]">
```

Modul-Ausgabe:

```
<font size=3>REX_HTML_VALUE[1]</font>
```

REX_LINK[..] | LINK[..]

Bis zu 10 Felder (REX_LINK[1]..REX_LINK[10]) möglich.

Enthält im folgenden Beispiel, im Feld "link1" des jeweiligen Slices die Artikel-Id. In der Ausgabe wird die ID als "index.php?article_id=ID" ausgegeben.

Modul-Eingabe:

```
<input type=text size=30 name=VALUE[1] value="REX_VALUE[1]"><br>
<input type=text size=30 name=LINK[1] value="REX_LINK[1]">
```

Modul-Ausgabe:

```
<a href=REX_LINK[1]>REX_VALUE[1]</a>
```

REX_FILE[..] | FILE[..]

Bis zu 10 Felder möglich(..)

Enthält Dateinamen, die wie in diesem Beispiel, über den Medienpool geladen werden. Alle Dateien die über den Medienpool geladen werden, werden im Ordner "/files" gespeichert.

Modul-Eingabe:

```
REX_MEDIA_BUTTON[1]
```

Modul-Ausgabe:

```
<?
if ("FILE[1]" != "")
{
echo "<img src=$REX[HTDOCS_PATH]/files/FILE[1] width=100>";
}
?>
```

Weitere Variablen:

\$REX[version]

Verfügbar: **ab 2.7**

Rückgaben: **Versionnummer**

Beispiel:

```
<?
echo $REX[version];
?>
```

Ausgabe:

2.7

\$REX[SERVER]

Verfügbar: **ab 2.7**

Rückgaben: **Server (Specials/Install)**

Ort: master.inc.php

\$REX[subversion]

Verfügbar: **ab 2.7**

Rückgaben: **Server (Specials/Install)**

Ort: master.inc.php

\$REX[SERVERNAME]

Verfügbar: **ab 2.7**

Rückgaben: **Servername(Specials/Install)**

Ort: master.inc.php

\$REX[HTDOCS_PATH]

Verfügbar: **ab 2.7**

Rückgaben:

Ort: master.inc.php

\$REX[WWW_PATH]

Verfügbar: **ab 2.7**

Rückgaben:

Ort: master.inc.php

\$REX[error_emailaddress]

Verfügbar: **ab 2.7**

Rückgaben:

Ort: master.inc.php

\$REX[STARTARTIKEL_ID]

Verfügbar: **ab 2.7**

Rückgaben:

Ort: master.inc.php

\$REX[LANG]Verfügbar: **ab 2.7**

Rückgaben:

Ort: master.inc.php

\$REX[CACHING]Verfügbar: **ab 2.7**

Rückgaben:

Ort: master.inc.php

\$REX[CACHING_DEBUG]Verfügbar: **ab 2.7**

Rückgaben:

Ort: master.inc.php

\$REX[MOD_REWRITE]Verfügbar: **ab 2.7**

Rückgaben:

Ort: master.inc.php

\$REX[INCLUDE_PATH]Verfügbar: **ab 2.7**

Rückgaben:

Ort: master.inc.php

\$REX[MEDIAFOLDER]**\$REX[MOD_REWRITE]****\$REX[GG]****REX_IS_VALUE[1] ... REX_IS_VALUE[10]****REX_HTML_BR_VALUE[1] REX_HTML_BR_VALUE[10]**

Der eingegeben Text wird genauso ausgegeben. Jeder Umbruch wird durch ersetzt. Um zu sichern das kein php-code enthalten ist wird < ? Und ? > gelöscht.

REX_PHP_VALUE[1] REX_PHP_VALUE[10]

Der eingegeben Text wird genauso ausgegeben. Sofern php code eingegeben wurde wird dieser auch ausgeführt.

REX_ARTICLE_ID

Hier wird die aktuelle Artikel ID eingesetzt.

REX_CATEGORY_ID

Hier wird die aktuelle Kategorie ID eingesetzt.

REX_PHP

Sofern PHP als Modul verwendet werden soll. Ist es am sinnvollsten dieses Platzhalter zu verwenden.

C2.03 Actions

Aktionen können in Verbindung mit Modulen angewendet werden. Im Bereich Module/Aktionen können Aktionen erstellt, editiert und verwaltet werden. Die Aktionen können dann im jeweiligen Modul ausgewählt und dem Modul hinzugefügt werden.

1. Einfache Filterung einer Eingabe in einem Block.

Beim Editieren eines Blocks soll die Eingabe gefiltert und manipuliert werden.

Modul-Eingabe:

```
<input type=field class=inp100 name=VALUE[1] value="REX_VALUE[1]">
```

Modul-Ausgabe:

```
REX_VALUE[1]
```

Über ein Aktion kann der Wert des Feldes VALUE-[1]- gelesen und verändert werden. Soll dies beim Editieren des Block in einem Artikel passieren, dann wird die Aktion mit PRE/POST: PRE und STATUS: Edit erstellt.

Eingabe-Aktion:

```
<?
if ($REX_ACTION[VALUE][1] == "Redaxo") $REX_ACTION[VALUE][1] = "CMS";
?>
```

Über \$REX_ACTION können alle Felder (z.B. [VALUE][1], [FILE][1], [LINK][1]) gelesen und neue gesetzt werden.

Im Modul muss die Aktion nun hinzugefügt werden. Wird das Modul als Block einem Artikel hinzugefügt, wird die Eingabe beim Editieren des Blocks überprüft und ggfs. verändert gespeichert. Soll die Überprüfung auch beim Hinzufügen des Moduls erfolgen, dann muss eine zweite Aktion mit gleichem Code erstellt und mit STATUS: Add dem Modul hinzugefügt werden.

2. Eingabe in verschiedenen Felder oder Tabelle speichern

Die Eingabe in Textfeld soll z.B in der Beschreibung eines Artikels (Metadaten) zusätzlich gespeichert werden.

Modul-Eingabe:

```
<textarea cols="80" rows="10" name="VALUE[4]" class="inp100"></textarea>
```

Mit der folgenden Aktion wird der Wert aus VALUE-[4]- in das Feld "Beschreibung" des Artikels geschrieben. Der Wert dieses Feldes wird mit jedem Artikel in der Tabelle "rex_article" gespeichert.

```
<?
$ins = new sql;
$ins->query("update rex_article set
beschreibung='". $REX_ACTION[VALUE][4]."' where id='REX_ARTICLE_ID'")
?>
```

Der STATUS muss in zwei Aktionen auf Edit und Add dem Modul hinzugefügt werden.

C3. Benutzer

Über die Erstellung verschiedenen Benutzer-Account, regeln Sie auch die Rechtevergabe in REDAXO. Mit der Konfiguration des Account geben Sie Programmteile und Funktionen in REDAXO frei und können so auch sensible Bereiche verstecken und sperren.

REDAXO-Navigation

So können z.B. die Bereiche Templates und Module für einen Grafiker gesperrt und der Bereich Medienpool frei geschaltet werden.

Kategorien und Artikel

Sie können innerhalb der Strukturverwaltung die gewünschten Kategorien und Artikel über jeweilige ID freigeben und so anderen Seiten sperren.

Module

Über die Vergabe der beiden Einstellungen HTML und PHP eines Modul sowie eigenen Rechten können die Modullisten und Blöcke in den Artikel frei oder gesperrt geschaltet werden.

C3.01 Navigation / Programmteile

Über die folgenden Eingaben können die gewünschten Programmteile/-seiten in der Navigation von REDAXO ausgewählt und angezeigt werden:

Permission	Beschreibung
mediapool[]	ohne dieses Recht kann der Medienpool nicht verwendet werden
template[]	bestimmt ob der User Templates erstellen und bearbeiten darf oder nicht. Fehlt dieses Recht, dann kann die Seite nicht über die Navigation geöffnet werden.
module[]	ob man Module erstellen und bearbeiten darf. Fehlt das Recht sieht man die Module Seite gar nicht.
user[]	bestimmt ob man User anlegen und verändern darf. Fehlt dieses Recht, fehlt auch die Userrechte Seite.
addon[]	bestimmt ob man Addons installieren und aktivieren darf. Fehlt dieses Recht, fehlt auch die Addon- Seite.
specials[]	fehlt dieses Recht, dann fehlt auch die Specials Seite.
import[]	fehlt dieses Recht, dann fehlt auch die Importe Seite
export[]	fehlt dieses Recht, dann ist zwar die Importe Seite da, der EXPORT Button ist ohne Funktion
stats[]	fehlt dieses Recht, wird die Statistik Seite nicht angezeigt
community[]	ohne dieses Recht gibts keine Community-Admin Seite.

C3.02 Struktur

Über die folgenden Rechte kann die Strukturverwaltung in REDAXO eingeschränkt werden, dabei werden Kategorien und Artikel in der Funktion freigegeben oder gesperrt.

Permission	Beschreibung
structure[]	definiert den Zugriff auf Kategorien, z.B. [all] erlaubt Zugriff auf alle Kategorien, [2] hingegen nur auf Kategorie Nummer 2.
article[]	definiert den Zugriff auf Artikel, z.B. [all] erlaubt Zugriff auf alle Artikel, [2] hingegen nur auf Artikel Nummer 2.
startArticle [none]	Damit werden bei diesem Benutzer keine Startartikel angezeigt. startArticle[Category ID]. Mit dieser Permission kann man die [none] Permissions lockern. Der User kann den Startartikel der angegebenen Kategorie sehen.

Erweiterte Anzeigen Struktur

Permission	Beschreibung
advancedMode[]	über Metadaten kann der Artikel kopiert oder verschoben werden.
editPrio[]	die Reihenfolge kann über Pfeilbuttons, ohne die Eingabe von Priowerten, verändert werden
expertMode[]	mit diesem Recht, werden Kategorien und Artikel mit Angabe der ID-Werte angezeigt

Beispiele Kombinationen:

structur[] entfernen, **article[all]** setzen

Damit ist das Editieren/Löschen alle Kategorien nicht mehr möglich. Das Löschen, Ändern des templates und der Prio sowie die Funktion online/offline wird gesperrt.

C3.03 Modul / Block

Über die Seite **Module** können Sie bei jedem geöffneten Modul, diese beiden "Häckchen" setzen und so die oben beschriebenen Rechte für das Modul aktivieren.

Permission	Beschreibung
module [php]	dieses Recht bestimmt ob ein User in einem Artikel ein Modul bearbeiten oder einfügen darf, welches als php Modul gekennzeichnet wurde.
module [html]	dieses Recht bestimmt ob ein User in einem Artikel ein Modul bearbeiten oder einfügen darf, welches als html Modul gekennzeichnet wurde.

Über die möglichen Kombinationen können Sie auf einfache Weise, drei unterschiedliche Rechtegruppen für die Benutzung der Module bilden lassen.

Eigene Rechte definieren

Einschränkung innerhalb eines Moduls. Sie können eigene Rechte definieren und diese bei den jeweiligen Benutzer eintragen. Im folgenden Beispiel wird innerhalb eines Moduls, das Rechte zur Eingabe eines Textfeldes abgefragt und freigegeben oder gesperrt.

Definition Permission unter **Benutzer**:

eingabeHtml[]

```
<?php
if ($REX_USER->isValueOf("rights", "eingabeHtml[]")) {
?>
MEDIA_HTMLAREA(1, "REX_VALUE[1]");
<?php
}
?>
```

C4. Addons

Addons erweitern den Funktionsumfang von REDAXO. Das Addon-Konzept erlaubt

- das Installieren von weiteren Navigationspunkten im Backend
- das Installieren von weiteren mySQL-Tabellen
- das Anlegen von Sprachversionen für das aktiviert Addon
- das Anlegen von Rechten für das Addon (über Benutzer)
-

Addon: Import/Export

Mit der Installation von REDAXO erhalten Sie das ersten Addon. Es ist **installiert** und **aktiviert**. Die beiden Funktionen machen ein Addon im Backend verfügbar.

Weitere Addons sind bereits geschrieben und können neben den eigenen Addons in REDAXO geladen werden.

Addon-Installation

Für das Setup laden Sie die erforderlichen Dateien in das Verzeichnis "redaxo/include/addon/". Danach sollte im Bereich Addon das neue Addon verfügbar sein. Klicken Sie danach **Installieren** und danach **aktivieren**. Nach der Meldung "Addon aktiviert" sollte das Addon in der Navigation von REDAXO erscheinen. Für weitere Hinweise steht für jedes Addon ein Hilfedatei (?) zur Verfügung und sollte vom Entwickler genutzt werden.

C4.01 Struktur

Die Ordnerstruktur für ein Addon sieht in der Regel so aus:

```
/lang  
/pages  
config.inc.php  
help.inc.php  
install.inc.php
```

config.inc.php

Konfigurieren Sie Ihr Addon in dieser Datei. Hier werden Einstellungen für die Sprachdateien, ID und Name, sowie die verwendeten Rechte geschrieben.

```
<?php  
  
$mypage = "import_export"; // only for this file  
  
// CREATE LANG OBJ FOR THIS ADDON  
$I18N_ADDON = new i18n($REX[LANG], $REX[INCLUDE_PATH].  
"/addons/$mypage/lang/");
```

```

$REX[ADDON][rxid][$mypage] = "REX_001"; // unique id /
$REX[ADDON][page][$mypage] = "$mypage"; // pagename/foldername
$REX[ADDON][name][$mypage] = "Import/Export"; // name
$REX[ADDON][perm][$mypage] = "import[]"; // permission
?>

```

Die Instanz für die Sprachversion Ihres Addons wird bei der Programmierung der Seiten verwendet.
Hier: \$!18N_ADDON

help.inc.php

In diese Datei können Sie eine Beschreibung des Addons, Hilfen und Tipps, sowie Version und Autor in HTML beschreiben. Diese Seite erscheint, wenn auf der Seite "Addon" das (?) eines Addons angeklickt wird.

```

<b>Import Export</b>

<br><br>

```

install.inc.php

Über diese Datei können Sie alle nötigen Schritte durchführen, die für die Installation des Addon benötigt werden. Dazu können Module installiert, Ordner angelegt, mySQL-Tabellen erstellt oder Seiten regeneriert werden (.

```

<?php

// CREATE/UPDATE DATABASE

// CREATE/UPDATE MODULES

// CREATE/UPDATE PAGES

// CREATE/UPDATE FILES

// REGENERATE SITE

$REX[ADDON][install]["import_export"] = 1;

?>

```

Sprachdateien (/lang)

Über die Sprachdateien können Sie Ihr Addon für die verschiedenen Sprachversionen im Backend anbieten. (Specials)

In dieser Datei können auf den Seite(n) des Addon verwendeten Elementen (Titel, Funktion, Bezeichnungen) definiert werden.

```

# addon:import/export en_GB
import = Import
importexport = Im-/Export
file_deleted = Datei wurde gelöscht

```

So kann auf einer Seite die Sprachdefinition genutzt werden:

```
<th align=left width=50%><?php echo $I18N_ADDON->msg('import'); ?></th>
```

Dabei wird die in der Datei `install.inc.php` definierte Sprachinstanz (`$I18N_ADDON`) genutzt und ausgegeben.

Seiten (/pages)

Um im Layout der REDAXO Seiten nun die Seiten des Addons zu programmieren stehen Ihnen für die Navigation, sowie die Abschlussleiste folgender Syntax zur Verfügung

Erstellen der Navigation und grauen Titelleiste einer Seite:

```
<?
include $REX[INCLUDE_PATH]."/layout_redaxo/top.php";
title($I18N_MESSAGES->msg("title"), "");
?>
```

Abschlussleiste:

```
<?
include $REX[INCLUDE_PATH]."/layout_redaxo/bottom.php";
?>
```

C4.02 Entwicklung

Wie ein Addons entsteht, wir ein Artikel zu Erstellung eines Gästebuchs zeigen

C5. Specials

Auf dieser Seite Informationen zur Verfügung, die im sich auf die Einstellungen in der master.inc.php beziehen. Daneben können Sie drei Funktionen aktivieren.

Features

Regeneriere Artikel & Cache

Über die Funktionen werden Informationen zu Kategorien, Artikel und Templates neu erstellt. Dazu werden auch die gecachten Dateien gelöscht.

Wie im Manual oder in den Demos beschrieben, werden Informationen zu Kategorien und Artikeln, z.B. in der Navigation einer Webseite nicht über SQL-Anfragen gelöst, sondern aus generierten Dateien gelesen. Die Informationen werden beim Arbeiten in REDAXO automatisch angelegt und für eine schnelle Ausgabe bereitgestellt.

Die Dateien werden in den Ordnern articles, categories und templates im Ordner "redaxo/include/generated/" erstellt.

REDAXO nutzt bei der Ausgabe von Webseiten diese Informationen für einen schnellen und Datenbank unabhängigen Darstellung.

Mehr zu Generierten Datei in REDAXO:

Kategorien

Artikel

Templates

Einstellungen

Auf der rechten Seite können Einstellungen zur Webseite editiert und eingesehen werden.

`$REX[STARTARTIKEL_ID]` verweist hier auf den Artikel, der bei der Eingaben der URL als erster Artikel angezeigt wird.

Artikeltyp

Auf dieser Seite werden notwendigen Artikeltypen erstellt und verwaltet. Mit Eintrag eines ersten Artikeltyps, wird unter Metadaten eines Artikel die Einstellung angezeigt und kann so ausgewählt werden.

Mit Hilfe der Artikeltyps können Artikel gekennzeichnet und unterschieden werden. Dies kann z.B. hilfreich bei Suchanfrage sein oder wie in Communiy-Demo genutzt, zur Kennzeichnung von Seiten, die im geschlossenen Bereich liegen.

C6. Object Oriented Framework

Sie können auf die Informationen in den Tabellen, die REDAXO verwendet, über SQL-Befehle zugreifen oder wie auf den folgenden Seite beschrieben, über ein Objekt-Orientiertes Framework. In REDAXO 3.x greifen diese Funktionen auf die generierten Dateien zurück und verbessern so die Performance. (REDAXO 2.7.x greift dabei über SQL-Befehl auf die Inhalte der Tabelle zurück).

Praktische Beispiele zur Verwendung findet man in den Demos, zb. in der Abbildung von Navigationen. Oder hier in Bereich [C.C1. Template/Navigationen](#). Alle verwendete Funktionen finden sich in den folgenden Klassen.

Die Klasse und enthaltenen Funktionen zu diesem Konzept finden Sie in den folgenden Dateien:

- class.oocategory.inc
- class.ooarticle.inc
- class.ooarticleslice.inc

Im Folgenden werden Funktionen und Beispiele gezeigt:

[OOCategory](#)

[OOArticle](#)

[OOArticleSlice](#)

C6.01 OOCategory

Die Beispiele wurden mit der Community Demo erstellt. Einige Befehle (#) sind noch nicht beschrieben.

Übersicht:

- [OO Category-Klasse](#)
- [getCategoryById\(\)](#)
- [searchCategoriesByName\(\)](#)
- [getRootCategories\(\)](#)
- [getChildren\(\)](#)
- [getParent\(\)](#)
- [# isParent\(\)](#)
- [getArticles\(\)](#)
- [# getArticlesByDate\(\)](#)
- [getStartArticle\(\)](#)
- [getId\(\)](#)
- [getName\(\)](#)
- [getPriority\(\)](#)
- [getUrl\(\)](#)
- [isOnline\(\)](#)
- [toString\(\)](#)

OOCategory-Klasse

Verfügbarkeit

ab 2.7

Beschreibung

Mit der Kategorie-Klasse können Sie Informationen zu den einzelnen Kategorie erhalten, wie ID, Name, Description, Func, Prior, Path, Status.

getCategoryById(2)

Verfügbarkeit

ab 2.7

Verwendung

```
$cat = OOCategory::getCategoryById('id');
```

Parameter

id, Kategorie-ID

Rückgaben

OOCategory (_id, _name, _description, _func, _re_category_id, _prior, _path, _status)

Beschreibung

Methode; gibt Informationen zu der über den Parameter id angegebenen Kategorie zurück.

Beispiel

```
<?
$cat = OOCategory::getCategoryById(2);
print_r($cat);
?>
```

Ausgabe:

```
oocategory Object
(
  [_id] => 2
  [_name] => BOARDS
  [_description] =>
  [_func] =>
  [_re_category_id] => 0
  [_prior] => 7
  [_path] =>
  [_status] => 1
)
```

Auf die einzelnen Attribute können Sie z.B. mit weiteren Methoden zugreifen:

```
<?
echo $cat -> getName();
?>
```

Ausgabe:\

```
BOARDS
```

searchCategoriesByName()

Verfügbarkeit

ab 2.7

Verwendung

```
$cat = OOCategory::searchCategoriesByName(name);  
$cat = OOCategory::searchCategoriesByName("%Name%");
```

Sucht nach Namen die offline gesetzt sind.
OOCategory::searchCategoriesByName("%Name%", 1);

Parameter

name, Kategorie-Name; optional \$ignore_offlines;

Rückgaben

Array mit ein oder mehreren, gefunden OOCategory?.

Beschreibung

Methode; gibt die Suchergebnisse nach prior gelistet zurück.

Beispiel

In diesen Beispiel wird nach der Kategorie Redaxo gesucht und eine Kategorie gefunden. Sie können dabei den vollen Namen oder wie in einee SQL-Anfrage "LIKE '%...&'" die Kategorie suchen.

```
$cat = OOCategory::searchCategoriesByName("%Mein%");  
print_r($cat);
```

Ausgabe:

```
Array  
(  
[0] => oocategory Object  
(  
[_id] => 13  
[_name] => ALLGEMEIN  
[_description] =>  
[_func] =>  
[_re_category_id] => 2  
[_prior] => 2  
[_path] => -2  
[_status] => 1  
)  
  
[1] => oocategory Object  
(  
[_id] => 6  
[_name] => MEIN GÄSTEBUCH  
[_description] =>  
[_func] =>  
[_re_category_id] => 0  
[_prior] => 12  
[_path] =>  
[_status] => 1  
)  
  
[2] => oocategory Object  
(  
[_id] => 4  
[_name] => MEIN PROFIL  
[_description] =>  
[_func] =>
```

```
[_re_category_id] => 0
[_prior] => 11
[_path] =>
[_status] => 1
)

[3] => oocategory Object
(
  [_id] => 7
  [_name] => MEINE NACHRICHTEN
  [_description] =>
  [_func] =>
  [_re_category_id] => 0
  [_prior] => 13
  [_path] =>
  [_status] => 1
)
)
```

getRootCategories()

Verfügbarkeit ab 2.7

Verwendung

```
$cat = OOCategory:: getRootCategories();
```

Rückgabe aller Root-Kategorien online

```
$cat = OOCategory:: getRootCategories(1);
```

Parameter

optional, 1/0 | online/offline

Rückgaben

Array mit alle Root-Kategorien

Beschreibung

Methode; gibt alle Kategorien im Root nach prior sortiert zurück. Ist der Parameter auf true gesetzt, werden alle Kategorien die Status online haben, ausgegeben.

Beispiel

```
<?
$cat = OOCategory:: getRootCategories();
print_r($cat);
?>
```

```
Array
(
  [0] => oocategory Object
  (
    [_id] => 1
    [_name] => HOME
    [_description] =>
    [_func] =>
    [_re_category_id] => 0
    [_prior] => 5
    [_path] =>
    [_status] => 1
  )

  [1] => oocategory Object
  (
    [_id] => 3
    [_name] => NEWS
    [_description] =>
    [_func] =>
    [_re_category_id] => 0
    [_prior] => 6
    [_path] =>
    [_status] => 1
  )

  [2] => oocategory Object
  (
    [_id] => 2
    [_name] => BOARDS
    [_description] =>
    [_func] =>
    [_re_category_id] => 0
    [_prior] => 7
    [_path] =>
    [_status] => 1
  )

  ...

  [11] => oocategory Object
  (
    [_id] => 11
    [_name] => IMPRESSUM
    [_description] =>
    [_func] =>
    [_re_category_id] => 0
    [_prior] => 16
    [_path] =>
    [_status] => 1
  )
)
```

getChildren()

Verfügbarkeit

ab 2.7

Verwendung

```
$cat->getChildren();
```

Parameter

optional \$ignore_offlines;

Rückgaben

Array mit oocategory Objects

Beschreibung

Methode; gibt ein Array mit Subkategorien zurück, sortiert nach prior

Beispiel

```
$cat = OOCategory::getCategoryById(2);  
$cat2 = $cat->getChildren();  
print_r($cat2);
```

Ausgabe:

```
Array  
(  
[0] => oocategory Object  
(  
[_id] => 13  
[_name] => ALLGEMEIN  
[_description] =>  
[_func] =>  
[_re_category_id] => 2  
[_prior] => 2  
[_path] => -2  
[_status] => 1  
)  
  
[1] => oocategory Object  
(  
[_id] => 14  
[_name] => LERN MICH KENNEN  
[_description] =>  
[_func] =>  
[_re_category_id] => 2  
[_prior] => 3  
[_path] => -2  
[_status] => 1  
)  
)
```

getParent()

Verfügbarkeit

ab 2.7

Verwendung

```
oocategory->getParent();
```

Parameter

keine

Rückgaben

oocategory Object

Beschreibung

Methode; gibt Parent Kategorie zurück

Beispiel

```
<?  
$cat = OOCategory::getCategoryById(13);  
  
$cat2 = $cat->getParent();  
print_r($cat2);  
  
?>
```

Ausgabe:

```
oocategory Object  
(  
  [_id] => 2  
  [_name] => BOARDS  
  [_description] =>  
  [_func] =>  
  [_re_category_id] => 0  
  [_prior] => 7  
  [_path] =>  
  [_status] => 1  
)
```

getArticles()**Verfügbarkeit**

ab 2.7

Verwendung`oocategory->getArticles()`**Parameter**` $ignore_offlines: offline/online, 0/1`**Rückgaben**

Array mit Artikeln einer Kategorie

BeschreibungMethode; gibt Array mit Artikel einer Kategorie zurück, mit Parameter `$ignore_offlines` auf 1, werden alle Artikel mit Status 0 aus dem Array entfernt**Beispiel**

```
$cat = OOCategory::getCategoryById(3);
$newArt = $cat->getArticles();
print_r($newArt);
```

Ausgabe:

Array

```
(
[0] => ooarticle Object
(
[_id] => 3
[_name] => NEWS
[_beschreibung] =>
[_attribute] =>
[_file] =>
[_category_id] => 3
[_type_id] => 1
[_startpage] => 1
[_prior] => 1
[_path] => -3
[_status] => 1
[_online_von] => 20040713
[_online_bis] => 20100101
[_erstelldatum] => 20040713
[_suchbegriffe] =>
[_template_id] => 1
[_checkbox01] => 0
[_checkbox02] => 0
[_checkbox03] => 0
[_checkbox04] => 0
)
```

```
[1] => ooarticle Object
(
[_id] => 20
[_name] => AKtuelle Infos 1
[_beschreibung] =>
[_attribute] =>
[_file] =>
[_category_id] => 3
```

```

[_type_id] => 1
[_startpage] => 0
[_prior] => 2
[_path] => -3
[_status] => 1
[_online_von] => 20040715
[_online_bis] => 20100101
[_erstelldatum] => 20040715
[_suchbegriffe] =>
[_template_id] => 1
[_checkbox01] => 0
[_checkbox02] => 0
[_checkbox03] => 0
[_checkbox04] => 0
)
)

```

getStartArticle()

Verfügbarkeit

ab 2.7

Verwendung

```
oocategory->getStartArticle()
```

Parameter

keine

Rückgaben

gibt ooarticle objekt zurück

Beschreibung

Methode; gibt das Artikel-Objekt zurück, mit den Eigenschaften der Artikels (id, name, beschreibung, attribute, file, category_id, type_id, startpage, prior, path, online_von, online_bis, erstelldatum, suchbegriffe, template_id, checkbox01-04

Beispiel

```

<?
$cat = OOCategory::getCategoryById(1);
print_r($cat->getStartArticle());
?>

```

Ausgabe:

```

ooarticle Object
(
  [_id] => 1
  [_name] => Home
  [_beschreibung] =>
  [_attribute] =>
  [_file] =>

```

```
[_category_id] => 1
[_type_id] => 1
[_startpage] => 1
[_prior] => 2
[_path] => -1
[_status] => 1
[_online_von] => 20040713
[_online_bis] => 20100101
[_erstelldatum] => 20040713
[_suchbegriffe] =>
[_template_id] => 1
[_checkbox01] => 0
[_checkbox02] => 0
[_checkbox03] => 0
[_checkbox04] => 0
)
```

getId()

Verfügbarkeit

ab 2.7

Verwendung

```
oocategory->getId()
```

Parameter

keine

Rückgaben

ID der Kategorie

Beschreibung

Methode; gibt die ID der Kategorie zurück

Beispiel

```
<?
$cat = OOCategory::getCategoryById(1);
echo $cat->getId();
?>
```

Ausgabe:

1

getName ()**Verfügbarkeit**

ab 2.7

Verwendung`oocategory->getName ()`**Parameter**

keine

Rückgaben

Name der Kategorie, string

Beschreibung

Methode; gibt die Namen der Kategorie zurück.

Beispiel

```
<?
$cat = OOCategory::getCategoryById(1);
echo $cat->getName();
?>
```

Ausgabe:

HOME

getPriority ()**Verfügbarkeit**

ab 2.7

Verwendung`oocategory->getPriority ()`**Parameter**

keine

Rückgaben

Prior

Beschreibung

Methode; gibt die Priorität einer Kategorie zurück, kann sich von der angezeigten Prio in REDAXO unterscheiden.

Beispiel

```
<?
$cat = OOCategory::getCategoryById(1);
print_r($cat);
echo $cat->getPriority ();
?>
```

Ausgabe:

5

getUrl()**Verfügbarkeit**

ab 2.7

Verwendung`oocategory->getUrl()`**Parameter**

keine

Rückgaben

Url Startartikel Kategorie

BeschreibungMethode; gibt die Url zu dem Startartikel einer Kategorie an z.B. `index.php?article_id=25`**Beispiel**

```
<?
$cat = OOCategory::getCategoryById(3);
echo $cat->getUrl();
?>
```

Ausgabe:

`index.php?article_id=3`***isOnline()*****Verfügbarkeit**

ab 2.7

Verwendung`oocategory->isOnline()`**Parameter**

keine

Rückgaben

true (1) – online, false (0) – offline

Beschreibung

Methode; gibt den Status der Kategorie zurück, Rückgabewert 1 oder 0

Beispiel

```
<?
$cat = OOCategory::getCategoryById(1);
echo $cat->isOnline();
?>
```

Ausgabe:

1

toString()**Verfügbarkeit**

ab 2.7

Verwendung

```
oocategory->toString();
```

Parameter

keine

Rückgaben

String, Information Kategorie (Debugging)

Beschreibung*

Methode; gibt einen String mit Information zu einem Artikel zurück. (Kategorie-Id, Kategorie-Name, Status)

Beispiel

```
<?
$cat = OOCategory::getCategoryById(3);
echo $cat->toString();
?>
```

Ausgabe:

Category: 3, NEWS, online

C6.02 OOArticle

Die Beispiele wurden mit der Community Demo erstellt.

Übersicht:

- [OOArticle - Klasse](#)
- [getArticleById\(\)](#)
- [searchArticlesByName\(\)](#)
- [getArticlesByType\(\)](#)
- [getCategory\(\)](#)
- [getFirstSlice\(\)](#)
- [getSlicesOfType\(\)](#)
- [getId\(\)](#)
- [getName\(\)](#)
- [getDescription\(\)](#)
- [getUrl\(\)](#)
- [isOnline\(\)](#)
- [toString\(\)](#)

OOArticle-Klasse

Verfügbarkeit

ab 2.7

Beschreibung

Klasse; Mit der Article-Klasse können Sie Informationen zu den einzelnen Artikeln erhalten, wie ID, Name, Beschreibung, Attribute, Typ, Prior, Path, Status, Startpage, Kategorie ...

getArticleById()

Verfügbarkeit

ab 2.7

Verwendung

```
$cat = OOArticle::getArticleById('id');
```

Parameter

id, Artikel-ID

Rückgaben

oarticle Object, (id, name, beschreibung, attribute, file, category_id, type_id, startpage, prior, path, status, online_von, online_bis, erstelldatum, suchbegriffe, template_id, checkbox01, checkbox02, checkbox03, checkbox04

Beschreibung

Methode; gibt Informationen zu dem, über den Parameter id angebenen, Artikel zurück.

Beispiel

```
<?
$art = OOArticle::getArticleById(21);
print_r($art);
?>
```

Ausgabe:

```
ooarticle Object
(
  [_id] => 21
  [_name] => test
  [_beschreibung] =>
  [_attribute] =>
  [_file] =>
  [_category_id] => 0
  [_type_id] => 1
  [_startpage] => 0
  [_prior] => 3
  [_path] =>
  [_status] => 0
  [_online_von] => 20050629
  [_online_bis] => 20100101
  [_erstelldatum] => 20050629
  [_suchbegriffe] =>
  [_template_id] => 1
  [_checkbox01] => 0
  [_checkbox02] => 0
  [_checkbox03] => 0
  [_checkbox04] => 0
)
```

Auf die einzelnen Attribute können Sie z.B. mit weiteren Methoden zugreifen:

```
%block bgcolor=#eee border='1px solid black' padding=6px% [=
<?
echo $art -> getName();
?>
```

Ausgabe:

MEIN PROFIL

searchArticlesByName ()**Verfügbarkeit**

ab 2.7

Verwendung

```
OOArticle::searchArticlesByName ("%Name%");
```

Parameter

name, Artikel-Name;

Rückgaben

Array, gefundene Artikel der definierten Suche

Beschreibung

Methode; gibt die Suchergebnisse in einem Array gelistet zurück.

Beispiel

In diesen Beispiel wird nach einem Artikel mit dem Begriffe "Mein" gesucht. Sie können dabei den vollen Namen oder wie in eine SQL-Anfrage "LIKE '%...&'" suchen.

```
<?
$article_search = OOArticle::searchArticlesByName ("%Mein%");
print_r ($article_search);
```

```
?>
```

Ausgabe:

Array

```
(
[0] => oarticle Object
(
  [_id] => 4
  [_name] => MEIN PROFIL
  [_beschreibung] =>
  [_attribute] =>
  [_file] =>
  [_category_id] => 4
  [_type_id] => 3
  [_startpage] => 1
  [_prior] => 1
  [_path] => -4
  [_status] => 1
  [_online_von] => 20040713
  [_online_bis] => 20100101
  [_erstelldatum] => 20040713
  [_suchbegriffe] =>
  [_template_id] => 1
  [_checkbox01] => 0
  [_checkbox02] => 0
  [_checkbox03] => 0
  [_checkbox04] => 0
)
[1] => oarticle Object
(
  [_id] => 6
  [_name] => MEIN GÄSTEBUCH
```

```
[_beschreibung] =>
[_attribute] =>
[_file] =>
[_category_id] => 6
[_type_id] => 3
[_startpage] => 1
[_prior] => 1
[_path] => -6
[_status] => 1
[_online_von] => 20040713
[_online_bis] => 20100101
[_erstelldatum] => 20040713
[_suchbegriffe] =>
[_template_id] => 1
[_checkbox01] => 0
[_checkbox02] => 0
[_checkbox03] => 0
[_checkbox04] => 0
)

[2] => oarticle Object
(
  [_id] => 7
  [_name] => MEINE NACHRICHTEN
  [_beschreibung] =>
  [_attribute] =>
  [_file] =>
  [_category_id] => 7
  [_type_id] => 3
  [_startpage] => 1
  [_prior] => 1
  [_path] => -7
  [_status] => 1
  [_online_von] => 20040713
  [_online_bis] => 20100101
  [_erstelldatum] => 20040713
  [_suchbegriffe] =>
  [_template_id] => 1
  [_checkbox01] => 0
  [_checkbox02] => 0
  [_checkbox03] => 0
  [_checkbox04] => 0
)

[3] => oarticle Object
(
  [_id] => 17
  [_name] => ALLGEMEIN
  [_beschreibung] =>
  [_attribute] =>
  [_file] =>
  [_category_id] => 13
  [_type_id] => 1
  [_startpage] => 1
  [_prior] => 1
  [_path] => -2-13
  [_status] => 1
  [_online_von] => 20040713
  [_online_bis] => 20100101
)
```

```
[_erstelldatum] => 20040713
[_suchbegriffe] =>
[_template_id] => 1
[_checkbox01] => 0
[_checkbox02] => 0
[_checkbox03] => 0
[_checkbox04] => 0
)
)
```

getArticlesByType(id)

Verfügbarkeit
ab 2.7

Verwendung

```
OOArticle::getArticlesByType(id)
```

Parameter

Typ, Artikel-Typ (Specials)

Rückgaben

Array mit oarticle Objects der gesuchten Artikel

Beschreibung

Methode; gibt ein Array mit oarticle Objects zurück, die den gesuchten Typ besitzen.

Beispiel

Suche nach Artikeln mit dem Typ 3 (User).

```
<?
$article_search = OOArticle::getArticlesByType(3);
print_r($article_search);
?>
```

Ausgabe:

```
Array
(
  [0] => oarticle Object
  (
    [_id] => 4
    [_name] => MEIN PROFIL
    [_beschreibung] =>
    [_attribute] =>
    [_file] =>
    [_category_id] => 4
    [_type_id] => 3
    [_startpage] => 1
    [_prior] => 1
    [_path] => -4
    [_status] => 1
```

```
[_online_von] => 20040713
[_online_bis] => 20100101
[_erstelldatum] => 20040713
[_suchbegriffe] =>
[_template_id] => 1
[_checkbox01] => 0
[_checkbox02] => 0
[_checkbox03] => 0
[_checkbox04] => 0
)
```

```
[1] => ooarticle Object
(
  [_id] => 6
  [_name] => MEIN GÄSTEBUCH
  [_beschreibung] =>
  [_attribute] =>
  [_file] =>
  [_category_id] => 6
  [_type_id] => 3
  [_startpage] => 1
  [_prior] => 1
  [_path] => -6
  [_status] => 1
  [_online_von] => 20040713
  [_online_bis] => 20100101
  [_erstelldatum] => 20040713
  [_suchbegriffe] =>
  [_template_id] => 1
  [_checkbox01] => 0
  [_checkbox02] => 0
  [_checkbox03] => 0
  [_checkbox04] => 0
)
```

```
[2] => ooarticle Object
(
  [_id] => 7
  [_name] => MEINE NACHRICHTEN
  [_beschreibung] =>
  [_attribute] =>
  [_file] =>
  [_category_id] => 7
  [_type_id] => 3
  [_startpage] => 1
  [_prior] => 1
  [_path] => -7
  [_status] => 1
  [_online_von] => 20040713
  [_online_bis] => 20100101
  [_erstelldatum] => 20040713
  [_suchbegriffe] =>
  [_template_id] => 1
  [_checkbox01] => 0
  [_checkbox02] => 0
  [_checkbox03] => 0
  [_checkbox04] => 0
)
```

```
[3] => ooarticle Object
(
  [_id] => 9
  [_name] =>
  [_beschreibung] =>
  [_attribute] =>
  [_file] =>
  [_category_id] => 9
  [_type_id] => 3
  [_startpage] => 1
  [_prior] => 1
  [_path] => -9
  [_status] => 1
  [_online_von] => 20040713
  [_online_bis] => 20100101
  [_erstelldatum] => 20040713
  [_suchbegriffe] =>
  [_template_id] => 0
  [_checkbox01] => 0
  [_checkbox02] => 0
  [_checkbox03] => 0
  [_checkbox04] => 0
)
)
```

getCategory()

Verfügbarkeit

ab 2.7

Verwendung

```
ooarticle->getCategory()
```

Parameter

keine

Rückgaben

oocategory Object (_id, _name, _description, _func, _re_category_id, _prior, _path, _status)

Beschreibung

Methode; gibt oocategory Objekt zurück mit den jeweiligen Eigenschaften. Weitere Informationen zu den Funktionen eines oocategory Objekts finden sie unter [C6.01 OOCategory](#) .

Beispiel

```
<?
$article = OOArticle::getArticleById(20);
$cat = $article->getCategory();
print_r($cat);
?>
```

Ausgabe:

```
oocategory Object
(
  [_id] => 3
```

```
[_name] => NEWS
[_description] =>
[_func] =>
[_re_category_id] => 0
[_prior] => 6
[_path] =>
[_status] => 1
)
```

getFirstSlice()

Verfügbarkeit

ab 2.7

Verwendung

```
ooartice->getFirstSlice()
```

Parameter

keine

Rückgaben

ooArticleSlice Objekt

Beschreibung

Methode; gibt ein ooArticleSlice Objekt zurück.

Beispiel

```
<?
$article = OOArticle::getArticleById(20);
$article_slice = $article->getFirstSlice();
print_r($article_slice);
?>
```

Ausgabe:

```
ooarticleslice Object
(
  [_id] => 48
  [_re_article_slice_id] => 0
  [_value] => Array
  (
    [1] => Aktuelle Infos 1
    [2] =>
    [3] =>
    [4] =>
    [5] =>
    [6] =>
    [7] =>
    [8] =>
    [9] =>
    [10] =>
  )
  [_file] => Array
  (
```

```
[1] =>
[2] =>
[3] =>
[4] =>
[5] =>
[6] =>
[7] =>
[8] =>
[9] =>
[10] =>
)

[_link] => Array
(
[1] => 0
[2] => 0
[3] => 0
[4] => 0
[5] => 0
[6] => 0
[7] => 0
[8] => 0
[9] => 0
[10] => 0
)

[_php] =>
[_html] =>
[_article_id] => 20
[_modultyp_id] => 6
)
```

getSlicesOfType()

Verfügbarkeit

ab 2.7

Verwendung

```
ooArticle->getSlicesOfType($type_id)
```

Parameter

type_id

Rückgaben

Array ooArticleSlice

Beschreibung

Methode; gibt ein Array von ooArticleSlice-Objekten, in der Reihenfolge der Verwendung, eines Artikel zurück.

getId()**Verfügbarkeit**

ab 2.7

Verwendung`oarticle->getId();`***Parameter******Rückgaben***

Artikel-Id

Beschreibung

Methode; gibt die Artikel-Id des oarticle Objekts zurück

Beispiel

```
<?
$article_search = OoArticle::searchArticlesByName ("%Mein%");

foreach($article_search as $article){
echo "<a href=index.php?article_id=" . $article->getId() .">". $article-
>getName() . "</a><br>" ;
}
?>
```

Ausgabe:

```
<a href=index.php?article_id=4>MEIN PROFIL</a><br>
<a href=index.php?article_id=6>MEIN GÄSTEBUCH</a><br>
<a href=index.php?article_id=7>MEINE NACHRICHTEN</a><br>
<a href=index.php?article_id=17>ALLGEMEIN</a><br>
```

getName()**Verfügbarkeit**

ab 2.7

Verwendung`oarticle->getName();`**Parameter**

keine

Rückgaben

Artikelname

Beschreibung

Methode; gibt den Artikelnamen eines oarticle Objekts zurück.

Beispiel

Suche nach Artikel mit dem Namen "Mein", danach Ausgabe der Artikelnamen der gefundenen Artikel.

```
<?
```

```
$article_search = OOArticle::searchArticlesByName ("%Mein%");

for ($i=0; $i<count($article_search); $i++)
{
$name = $article_search[$i]->getName() ."<br>";
echo $name ;
}
?>
```

Ausgabe:
MEIN PROFIL
MEIN GÄSTEBUCH
MEINE NACHRICHTEN
ALLGEMEIN

getDescription()

Verfügbarkeit

ab 2.7

Verwendung

```
ooarticle->getDescription();
```

Parameter

keine

Rückgaben

Beschreibung des Artikels, Metadaten

Beschreibung

Methode; gibt den Wert der Beschreibung (Metadaten) zurück.

Beispiel

```
<?
$article = OOArticle::getArticleById(1);
echo $article->getDescription();
?>
```

Ausgabe:
Homepage Demo Community

getUrl()

Verfügbarkeit

ab 2.7

Verwendung

```
ooarticle->getUrl()
```

Parameter

Rückgaben

Url, index.php?article_id=

Beschreibung

Methode; gibt die url der Artikels zurück. Die Rückgabe unterstützt auch mod_rewrite. Parameter als assoziativer Array übergeben, werden an die Url angefügt.

```
$param = array("order" => "123", "name" => "horst");  
$article->getUrl($param);
```

```
index.php?article_id=1&order=123&name=horst
```

bei aktivierten mod_rewrite:

```
/1-The_Article_Name?order=123&name=horst
```

Beispiel

```
<ul>  
<?  
$article_search = OOArticle::searchArticlesByName ("%Mein%");  
  
foreach ($article_search as $article) {  
echo "<li><a href=" . $article->getUrl() . ">" . $article->getName() .  
"</a></li>";  
}  
  
>  
</ul>
```

Ausgabe:

```
<ul>  
<li><a href=index.php?article_id=4>MEIN PROFIL</a></li>  
<li><a href=index.php?article_id=6>MEIN GÄSTEBUCH</a></li>  
<li><a href=index.php?article_id=7>MEINE NACHRICHTEN</a></li>  
<li><a href=index.php?article_id=17>ALLGEMEIN</a></li>  
</ul>
```

isOnline()**Verfügbarkeit**

ab 2.7

Verwendung

```
ooarticle->isOnline();
```

Parameter

keine

Rückgaben

true, 1 wenn online

Beschreibung

Methode; gibt Wert true/1 zurück, wenn Artikel den Status online hat.

Beispiel

```
<?
$article = OOArticle::getArticleById(20);
echo $article->isOnline();
?>
```

Ausgabe:

1

toString()**Verfügbarkeit**

ab 2.7

Verwendung

```
ooArticle->toString()
```

Parameter

keine

Rückgaben

String, Artikel Informationen (Debugging)

Beschreibung

Methode; gibt einen String mit Information zu einem Artikel zurück. (Artikel-Id, Artikel-Name, Status)

Beispiel

```
<?
$article = OOArticle::getArticleById(20);
$debugString = $article->toString();
```

```
echo $debugString;
```

```
?>
```

Ausgabe:

Article: 20, Aktuelle Infos 1, offline

C6.03 OOArticleSlice

Die Beispiele wurden mit der Community Demo erstellt.

Übersicht:

- [OOArticleSlice-Klasse](#)
- [getArticleSliceById\(\)](#)
- [getFirstSliceForArticle\(\)](#)
- [getSlicesForArticleOfType\(\)](#)
- [getNextSlice\(\)](#)
- [getPreviousSlice\(\)](#)
- [fullTextSearch\(\)](#)
- [getArticle\(\)](#)
- [getId\(\)](#)
- [getValue\(\)](#)
- [getLink\(\)](#)
- [getLinkUrl\(\)](#)
- [getFile\(\)](#)
- [getFileUrl\(\)](#)
- [getHtml\(\)](#)
- [getPhp\(\)](#)

OOArticleSlice-Klasse

Verfügbarkeit
ab 2.7

Beschreibung

Klasse; Mit der ArticleSlice-Klasse können Sie Informationen zu den Variablen der einzelnen Blöcke, die zu einem Artikel gehören, erhalten. Diese Variablen sind id, re_article_slice_id, value1 – value10, link1 – link10, file1 – file10, php, html, article_id, modultypid.

Ein „Slice“ entspricht einem Block, der zu einem Artikel gehört.

Hinweis:

Die OOArticleSlice Klasse läuft direkt über die Datenbank läuft und ist daher im Moment noch nicht so performant ist. Die Anderen OO Klassen laufen über gecachte Dateien und haben daher eine ordentlich Performance.

getArticleSliceById()**Verfügbarkeit**

ab 2.7

Verwendung

```
$slice = OOArticleSlice::getArticleSliceById('id');
```

Parameter

id, Slice-ID

Rückgaben

OOArticleSlice-Objekt (id, re_article_slice_id, value1 – value10, link1 – link10, file1 – file10, php, html, article_id, modultypid)

Beschreibung

Methode; gibt Informationen über den Article-Slice mit einer bestimmten ID zurück. Jeder Slice hat eine eindeutige ID.

Beispiel

```
<?
$slice = OOArticleSlice::getArticleSliceById(31);
print_r($slice);
?>
```

Ausgabe

```
ooarticleslice Object (
  [_id] => 31
  [_re_article_slice_id] => 22
  [_value] => Array ( [1] => Ersparen Sie sich und ....
  [2] =>
  ...
  [10] =>
  )
  [_file] => Array ( [1] =>
  [2] =>
  [3] =>
  ...
  [10] =>
  )
  [_link] => Array ( [1] => 0
  [2] => 0
  [3] => 0
  ...
  [10] => 0 )
  [_php] =>
  [_html] =>
  [_article_id] => 1
  [_modultyp_id] => 2
)
```

Auf die einzelnen Attribute kann mit weiteren Methoden zugegriffen werden:

```
echo $slice -> getValue(1);
```

Ausgabe: Ersparen Sie sich und

getFirstSliceForArticle()

Verfügbarkeit

ab 2.7

Verwendung

```
$slice = OOArticleSlice::getFirstSliceForArticle('id');
```

Parameter

id, Article-ID

Rückgaben

OOArticleSlice-Objekt (id, re_article_slice_id, value1 – value10, link1 – link10, file1 – file10, php, html, article_id, modultypid)

Beschreibung

Methode; gibt Informationen über den ersten Article-Slice eines vorgegebenen Artikels aus. Der erste Slice eines Artikels hat die re_article_slice_id 0.

Beispiel

```
<?
$slice = OOArticleSlice::getFirstSliceForArticle(1);
print_r($slice);
?>
```

Ausgabe

```
ooarticleslice Object (
  [_id] => 22
  [_re_article_slice_id] => 0
  [_value] => Array ( [1] => HOME
  [2] =>
  ...
  [10] =>
  )
  [_file] => Array ( [1] =>
  [2] =>
  ...
  [10] =>
  )
  [_link] => Array ( [1] => 0
  [2] => 0
  ...
  [10] => 0
  )
  [_php] =>
  [_html] =>
  [_article_id] => 1
  [_modultyp_id] => 6 )
```

getSlicesForArticleOfType()

Verfügbarkeit

ab 2.7

Verwendung

```
$slice = OOArticleSlice::getSlicesForArticleOfType(article_id, type_id);
```

Parameter

article_id = Article-ID, type_id = Modultyp_ID

Rückgaben

OOArticleSlice-Objekt (id, re_article_slice_id, value1 – value10, link1 – link10, file1 – file10, php, html, article_id, modultypid)

Beschreibung

Methode; gibt Informationen aus über Article-Slices, die zu einem bestimmten Artikel gehören und die einem vorgegebenen Modultyp entsprechen. Hier können auch mehrere Slices selektiert werden, wenn bei einem Artikel der gesuchte Modultyp mehrfach verwendet wurde.

Beispiel

```
<?
$slice = OOArticleSlice::getSlicesForArticleOfType(1, 6);
print_r($slice);
?>
```

Ausgabe

```
Array ( [0] =>
ooarticleslice Object (
  [_id] => 22
  [_re_article_slice_id] => 0
  [_value] => Array ( [1] => HOME
[2] =>
[3] =>
...
[10] =>
)
  [_file] => Array ( [1] =>
[2] =>
[3] =>
...
[10] =>
)
  [_link] => Array ( [1] => 0
[2] => 0
[3] => 0
...
[10] => 0
)
  [_php] =>
  [_html] =>
  [_article_id] => 1
  [_modultyp_id] => 6
)
)
```

getNextSlice()**Verfügbarkeit**

ab 2.7

Verwendung`$slice = OOArticleSlice::getNextSlice();`**Parameter**

keine

Rückgaben

OOArticleSlice-Objekt (id, re_article_slice_id, value1 – value10, link1 – link10, file1 – file10, php, html, article_id, modultypid)

Beschreibung

Methode; gibt Informationen aus über den Article-Slice, der auf einen bereits selektierten Slice folgt.

Beispiel

```

<?
$slice1 = OOArticleSlice::getFirstSliceForArticle(27);
$slice2 = $slice1->getNextSlice();
$slice3 = $slice2->getNextSlice();
print_r($slice2);
?>
Ausgabe
oarticleslice Object (
  [_id] => 55
  [_re_article_slice_id] => 54
  [_value] => Array ( [1] => Streng dem definierten Wesen des Blindtextes folgend,
  [2] =>
  [3] =>
  ...
  [10] =>
  )
  [_file] => Array ( [1] =>
  [2] =>
  [3] =>
  ...
  [10] =>
  )
  [_link] => Array ( [1] => 0
  [2] => 0
  [3] => 0
  ...
  [10] => 0
  )
  [_php] =>
  [_html] =>
  [_article_id] => 27
  [_modultyp_id] => 2
  )

```

getPreviousSlice()**Verfügbarkeit**

ab 2.7

Verwendung

```
$slice = OOArticleSlice::getPreviousSlice();
```

Parameter

keine

Rückgaben

OOArticleSlice-Objekt (id, re_article_slice_id, value1 – value10, link1 – link10, file1 – file10, php, html, article_id, modultypid)

Beschreibung

Methode; gibt Informationen aus über den Article-Slice, der einem bereits selektierten Slice vorhergeht.

Beispiel

```
$slice = OOArticleSlice::getArticleSliceById(31);
$slice2 = $slice->getPreviousSlice();
print_r($slice2);
```

Ausgabe

```
ooarticleslice Object ( [_id] => 22
  [_re_article_slice_id] => 0
  [_value] => Array ( [1] => HOME
    [2] =>
    [3] =>
    ...
    [10] => )
  [_file] => Array ( [1] =>
    [2] =>
    [3] =>
    ...
    [10] => )
  [_link] => Array ( [1] => 0
    [2] => 0
    [3] => 0
    ...
    [10] => 0
  )
  [_php] =>
  [_html] =>
  [_article_id] => 1
  [_modultyp_id] => 6
)
```

getArticle()

Verfügbarkeit

ab 2.7

Verwendung

```
$article = OOArticleSlice::getArticle();
```

Parameter

keine

Rückgaben

OOArticle-Objekt (id, name, beschreibung, attribute, file, category_id, type_id, startpage, prior, path, status, online_von, online_bis, erstelldatum, suchbegriffe, template_id, checkbox 0 – 4)

Beschreibung

Methode; gibt Informationen aus über den Artikel, zu dem ein selektierter Slice gehört.

Beispiel

```
<?
$slice = OOArticleSlice::getArticleSliceById(31);
$article = $slice->getArticle();
print_r($article);
?>
```

Ausgabe

```
oarticle Object ( [_id] => 1
[_name] => Home
[_beschreibung] =>
[_attribute] =>
[_file] =>
[_category_id] => 1
[_type_id] => 1
[_startpage] => 1
[_prior] => 2
[_path] => -1
[_status] => 1
[_online_von] => 20040713
[_online_bis] => 20100101
[_erstelldatum] => 20040713
[_suchbegriffe] =>
[_template_id] => 1
[_checkbox01] => 0
[_checkbox02] => 0
[_checkbox03] => 0
[_checkbox04] => 0
)
```

getId()

Verfügbarkeit

ab 2.7

Verwendung

```
$id = OOArticleSlice::getId();
```

Parameter

keine

Rückgaben

Slice-ID

Beschreibung

Methode; gibt die ID des selektierten Sclices aus.

Beispiel

```
<?
$slice = OOArticleSlice::getFirstSliceForArticle(1);
$id = $slice->getId();
print_r($id);
?>
```

Ausgabe

22

getValue()

Verfügbarkeit

ab 2.7

Verwendung

```
$value = OOArticleSlice::getValue(index);
```

Parameter

Index-Wert (1 – 10)

Rückgaben

String

Beschreibung

Methode; gibt einen bestimmten Value eines selektierten Sclices aus.

Beispiel

```
<?
$slice = OOArticleSlice::getFirstSliceForArticle(8);
$value = $slice->getValue(1);
print_r($value);
?>
```

Ausgabe

OPEN BOARD

getLink()

Verfügbarkeit

ab 2.7

Verwendung

```
$link = OOArticleSlice::getLink(index);
```

Parameter

Index-Wert (1 – 10)

Rückgaben

Article_ID eines internen Links

Beschreibung

Methode; gibt die Artikel-ID eines indizierten internen Links des selektierten Sclices aus.

Beispiel

```
<?
$slice = OOArticleSlice::getFirstSliceForArticle(8);
$link = $slice->getLink(1);
print_r($link);
?>
```

Ausgabe

12

getLinkUrl()

Verfügbarkeit

ab 2.7

Verwendung

```
$url = OOArticleSlice::getLinkUrl(index);
```

Parameter

Index-Wert (1 – 10)

Rückgaben

URL eines interner Links

Beschreibung

Methode; gibt die URL eines indizierten internen Links des selektierten Sclices aus.

Beispiel

```
<?
$slice = OOArticleSlice::getArticleSliceById(52);
$url = $slice->getLinkUrl(1);
print_r($url);
?>
```

Ausgabe

index.php?article_id=2

getFile()

Verfügbarkeit

ab 2.7

Verwendung

```
$file = OOArticleSlice::getFile(index);
```

Parameter

Index-Wert (1 – 10)

Rückgaben

Dateiname

Beschreibung

Methode; gibt den Dateinamen eines indizierten Files des selektierten Sclices aus.

Beispiel

```
<?
$slice = OOArticleSlice::getArticleSliceById(52);
$file = $slice->getFile(1);
print_r($file);
?>
```

Ausgabe
banner.jpg

getFileUrl()

Verfügbarkeit

ab 2.7

Verwendung

```
$file = OOArticleSlice::getFileUrl(index);
```

Parameter

Index-Wert (1 – 10)

Rückgaben

URL

Beschreibung

Methode; gibt die URL eines indizierten Files des selektierten Sclices aus.

Beispiel

```
<?
$slice = OOArticleSlice::getArticleSliceById(52);
$file = $slice->getFileUrl(1);
print_r($file);
?>
```

Ausgabe
/banner.jpg

getHtml()

Verfügbarkeit

ab 2.7

Verwendung

```
$string = OOArticleSlice::getHtml();
```

Parameter

keine

Rückgaben

Html-Code

Beschreibung

Methode; gibt den Html-Code des selektierten Slices aus.

Beispiel

```
<?
$slice = OOArticleSlice::getArticleSliceById(52);
$html = $slice->getHtml();
print_r(htmlspecialchars($html));
?>
```

Ausgabe

```
<h2>Erstellung und Pflege</h2>
```

getPhp()

Verfügbarkeit

ab 2.7

Verwendung

```
$string = OOArticleSlice::getPhp();
```

Parameter

keine

Rückgaben

PHP-Code

Beschreibung

Methode; gibt den PHP-Code des selektierten Slices aus.

Beispiel

```
<?
$slice = OOArticleSlice::getArticleSliceById(65);
$phpcode = $slice->getPhp();
print_r (htmlspecialchars($phpcode));
?>
```

Ausgabe

```
<? echo "<h2>getArticleSliceById(</h2>"; $slice = OOArticleSlice::getArticleSliceById(31); print_r
($slice); // Ausgabe eines Einzelwertes print <<<EOT <br><h2> Einzelwert: Value (1) </h2> EOT; echo
$slice -> getValue(1); ?>
```

C7. Import & Export

Ein Import besteht in der Regel aus dem Import der Datenbank (Tabellen und Inhalte) und den Dateien, die beim Export erstellt wurden. Dabei werden in der Regel die Dateien aus den Ordnern files, css, js und pics gesichert. (Mehr unter Export)

Import

Auf der linken Seite können Sie unter **Datenbankimport** die Tabellen und deren Inhalte importieren. Dazu können Sie entweder die Redaxo-Import Dateien auswählen und importieren, oder die darunter angegebenen Importe importieren. Dabei werden die bisherigen Inhalte der Datenbank gelöscht und überschrieben!

Danach können Sie unter **Dateienimport** die benötigten Dateien importieren. Nach erfolgreichem Import werden alle Artikel sowie der Cache neu generiert. Wechseln Sie danach in die Strukturverwaltung oder auf die Webseite und sehen das Ergebnis.

Export

Auf der rechten Seite finden Sie Einstellungen und Optionen für den Export der Tabellen und Dateien ihrer Webseite.

Für Sicherung der Datenbank, wählen Sie erste Option und entscheiden unten, ob Sie die Export-Datei auf dem Server speichern oder auf Ihrer Festplatte sichern wollen.

Wählen Sie die Option **Auf dem Server speichern** wird die Datei auf dem Server gespeichert und auf der linken Seite, unter **Import** angezeigt. Sie finden die auf dem Server gespeicherte Datei im Ordner "redaxoincludeaddonsimport_exportfiles".