

sddCMS

Entwickler Schulung

Autor: Michael Sebel
Datum: 10.04.2010
Version: 2.0

Inhalte

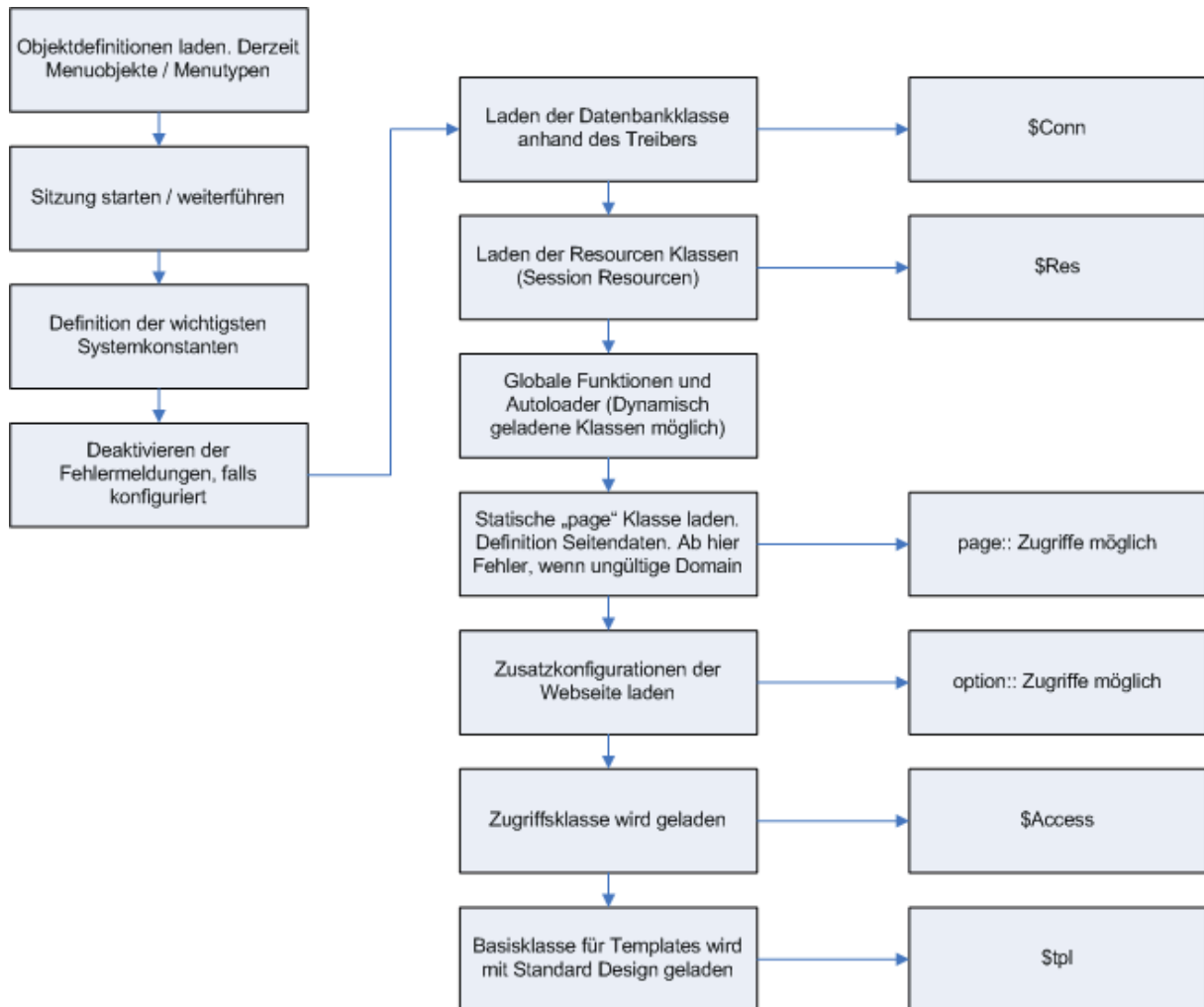
1	Einführung	3
1.1	Ladeablauf	3
1.2	Controller	3
2	Modulaufbau	5
2.1	Datenbank	5
2.1.1	Zusatzmodule / Mehrsprachigkeit.....	5
2.1.2	Kundendatenbanken	6
2.2	Dateien	6
2.3	Templates	7
2.3.1	View	7
2.3.2	Admin.....	7
2.3.3	Funktionalität	7
3	Ressourcen	8
3.1	Wichtige Objekte.....	8
3.2	Wichtige Klassen	9
3.2.1	Statische Klassen	9
3.2.2	Hilfsklassen / Objekte	10
3.2.3	Abstrakte Klassen / Interfaces	11
3.3	Dokumentation	11
3.3.1	Datenbankskripte / Referenz	11
3.3.2	API Referenz.....	11
3.3.3	Schulungsdokumente.....	11
4	Installation.....	12

1 Einführung

In diesem Kapitel erhalten Sie eine kurze Einführung in das Prinzip von sddCMS. Grundsätzlich sind alle Module danach aufgebaut. Sie werden in Kürze erfahren, wie ein normaler Ladeablauf einer Seite aussieht und wie die gesamte Navigation gesteuert ist.

1.1 Ladeablauf

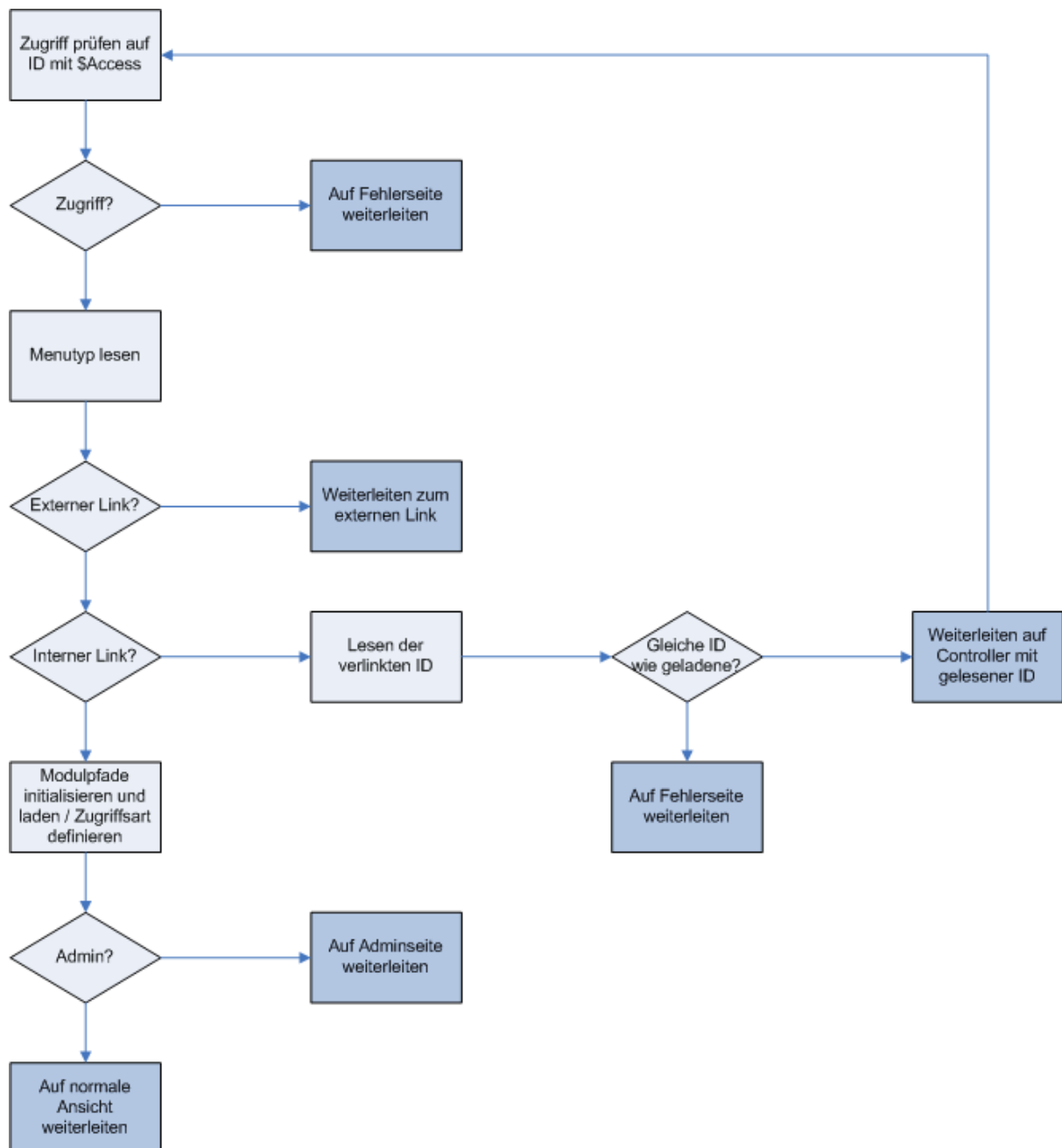
Mit dem Includieren der Datei „/system.php“ wird das gesamte System hinter sddCMS geladen. Im Kapitel 2.3 finden Sie Templates in welchen das Laden dieser Datei zusammen mit allen weiteren nötigen Dateien erledigt wird.



Die erwähnten Objekte und Klassen werden im dritten Kapitel erläutert.

1.2 Controller

Die Navigation der Webseite erfolgt über das Menu. Mit jedem Aufruf im Menu wird der „Controller“ (/controller.php) aufgerufen, welcher die Zugriffe regelt und die effektiven Module des Menus lädt.



Wie die Module aufgebaut sind und wie Sie selber auf einfache Weise neue Module entwickeln können lesen Sie im nächsten Kapitel.

2 Modulaufbau

Die folgenden Kapitel zeigen die grobe Dateistruktur von sddCMS, welche für individuelle Module genutzt werden kann.

2.1 Datenbank

In sddCMS stehen grundsätzlich eine globale Datenbank sowie eine Instanzdatenbank zur Verfügung. Sofern Sie auf einem Server mehrere Installationen von sddCMS betreiben, brauchen Sie die globalen Daten nur einmal.

2.1.1 Zusatzmodule / Mehrsprachigkeit

Wenn Sie nicht am Kern von sddCMS schrauben wollen, werden Sie sich primär nur für zwei Tabellen in der Instanzdatenbank interessieren.

Tabelle tbmenutyp

In diese Tabelle können Sie die Pfade (Admin/View) für eigens entwickelte Zusatzmodule einfüllen. Hier ein kleines Beispiel aus dem Verwaltungstool phpMyAdmin:













	typ_ID Identifiziert ein Menu	typ_Name Name des Menutyps	typ_Adminpath Pfad zur Startseite des Admin des Moduls	typ_Viewpath Pfad zur Startseite für die View des Moduls	typ_Type 0 = Normal / 1 = Nur Admin / 2 = Nur View	page_ID Da individueller Typ, nur für bestimmten Kunden gültig (Bestimmte Page, nicht Mandant)
<input type="checkbox"/>  	10001	Pageadmin	/page/1/addpage /index.php		1	1
<input type="checkbox"/>  	10002	Kontenverwaltung	/page/1 /book/index.php		1	1
<input type="checkbox"/>  	10003	Statistiken	/page/1 /book/stats.php		1	1
<input type="checkbox"/>  	10004	Kundenansicht		/page/1 /book/customer.php	1	1
<input type="checkbox"/>  	10005	Hostingverwaltung	/page/1/hosting /index.php		1	1
<input type="checkbox"/>  	10006	Umfrage SVA	/page/8/umfrage /admin.php	/page/8/umfrage /index.php	0	8

Tabelle tbresource

Dies ist die gleiche Tabelle wie sie auch in der globalen Datenbank zu finden ist. Sie wird verwendet um mit dem \$Res Objekt ein Modul mehrsprachig zu machen. Derzeit werden die Sprachen deutsch (0) und englisch (1) verwendet. Die ID (res_ID) ist selber zu vergeben und muss grösser als 100'000 sein.

	res_ID Identifiziert den Sprachinhalt mit res_Language	res_Language Identifiziert den Sprachinhalt mit res_ID	res_Text Inhalt der Resource selbst	res_Modul Name des Moduls zur wiedererkennung
<input type="checkbox"/>  	1	0	Willkommen	Standard
<input type="checkbox"/>  	1	1	welcome	Standard
<input type="checkbox"/>  	2	0	Benutzername	Standard
<input type="checkbox"/>  	2	1	username	Standard
<input type="checkbox"/>  	3	0	Passwort	Standard
<input type="checkbox"/>  	3	1	password	Standard
<input type="checkbox"/>  	4	0	Anmelden	Standard
<input type="checkbox"/>  	4	1	login	Standard

2.1.2 Kundendatenbanken

Die meisten individuellen Module werden auf irgend eine Weise einen zusätzlichen Datenzugriff benötigen. Mit der „setCustomerDB()“ Methode auf den \$Conn Objekt können Sie in die „Kundendatenbank“ wechseln. Diese müssen Sie vorher aber erstellen, wobei der Datenbankbenutzer von sddCMS Berechtigung darauf haben muss.

Die Datenbank muss im Format page[ID] sein, für die Website mit der page_ID „1“ müsste die Datenbank also page1 heissen. Falls das aufgrund der Hostingsoftware (z.B. Bei Confixx) nicht möglich ist, können Sie alternativ „setDB(name)“ verwenden um auf eine andere Datenbank zuzugreifen.

2.2 Dateien

Nun stellt sich noch die Frage, wo denn der Code für individuelle Module abgelegt werden kann und wo sich die Designs befinden. Für jeden Mandant und jede Page stehen folgende Ordner zur Verfügung.

Ordner	Beschreibung
/design/x/	Das Design können Sie in einen nummerierten Ordner einfügen. Für jede Page (tbpage) gibt es eine design_ID, von welcher die Dateien für das Design geladen werden. Nehmen Sie z.B. design_ID = 99 erstellen Sie in /design/ den Ordner "99" mit den nötigen Files. In der Grundinstallation ist bereits ein Beispieldesign vorhanden.
/page/x/	Dieser Ordner gehört der gesamten Webseite (entspricht der page_ID aus tbpage). Dort sind einige Javascript Files die automatisch eingebunden werden sowie die Datenordner welche vom Medienmanager / Datei-Bibliothek verwendet werden. Nebst diesen vorgegebenen Ordnern können Sie hier eigene anlegen und nutzen. Im Beispiel der Menutypen (Kapitel 2.2.1) sehen Sie, dass so eben auch Code für Zusatzmodule genutzt werden kann.
/mandant/x/	Im Ordner des einzelnen Mandanten (entspricht man_ID aus tbmandant, wobei der Mandant immer einer Page zugeordnet ist). Befinden sich zusätzliche Design Files (Footer, Header) sowie ein Favicon. Alles weitere kann genau gleich wie bei der Page für Module und Dateien genutzt werden.

2.3 Templates

Im folgenden zwei einfache Code Beispiele mit denen Sie ein eigenes Modul erstellen können. Hierbei gibt es jeweils eine „View“, die Benutzeransicht und einen „Admin“, die Oberfläche zur Verwaltung von Daten. Je nach Einstellung in der tbmenutyp ist aber nicht beides möglich.

2.3.1 View

```
<?php
define(BP,realpath($_SERVER['DOCUMENT_ROOT']));
require_once(BP.'/system.php');

// System abschliessen
$tpl->aC($out);
require_once(BP.'/cleaner.php');
?>
```

2.3.2 Admin

```
<?php
define(BP,realpath($_SERVER['DOCUMENT_ROOT']));
require_once(BP.'/system.php');
// Zugriff testen und Fehler melden
$Access->control();

// System abschliessen
$tpl->aC($out);
require_once(BP.'/cleaner.php');
?>
```

2.3.3 Funktionalität

Der Unterschied zum Admin ist, dass dort ein Zugriffsscheck durchgeführt wird. Wurde die Seite nicht durch den Controller geladen, erfolgt eine Fehlermeldung.

Die Konstante BP muss definiert werden, sie wird innerhalb von sddCMS mehrfach verwendet. Die Definition ist immer gleich. Mit dem inkludieren von „/system.php“ sind alle wichtigen Objekte (Kapitel 3.1) und Klassen (Kapitel 3.2) vorhanden.

Mit „\$tpl->aC(inhalt)“ füllen Sie HTML Code für die Ausgabe in das Templatesystem. Am Ende wird dies unter anderem in „/cleaner.php“ ausgegeben.

3 Ressourcen

Im letzten Kapitel werden die wichtigsten Objekte und Klassen angesprochen. Alles weitere zur Verwendung und Weiterentwicklung von sddCMS finden Sie in den Dokumentationen. Im Kapitel 3.3 wird kurz erklärt was Sie wo finden können.

3.1 Wichtige Objekte

Mit dem inkludieren der „/system.php“ stehen Ihnen automatisch einige Objekte zur Verfügung. Diese seine hier kurz erklärt. Die Komplette Funktionalität können sie der API Referenz (Kapitel 3.3.2) entnehmen.

Ordner	Beschreibung
\$Conn	Dies ist das Objekt um auf Datenbanken zuzugreifen. Es unterstützt fast alle Vorgänge, die man auch mit nativen mysql Funktionen erreichen könnte. Es gibt derzeit Treiber für mysql (4.0, 5.0) und mysqli (5.0). Weitere sind derzeit nicht in Planung.
\$Res	Mit dem \$Res Objekt lassen sich Texte aus den Ressourcen in verschiedenen Sprachen auslesen. Dazu wird die Session (Für bereits geladene Texte) und die Datenbank (tbresource) verwendet.
\$Access	Mit diesem Objekt können sie Zugriffsabfragen machen. In den meisten durchschnittlichen Modulen wird das nicht benötigt. Im Admin wird darauf meist nur die "control" Methode aufgerufen.
\$Menu	Steht nach dem Include der /system.php Datei zur Verfügung und kann für Datenoperationen mit den Menüpunkten verwendet werden. Man kann auch das selektierte Menuobjekt lesen.
\$tpl	An dieses Objekt kann HTML Code für die Ausgabe bereitgestellt werden. Weiterhin können Daten in die Menustruktur und in den Kopfbereich (CSS, JS Daten) eingebunden werden. Das ist jederzeit möglich, da die effektive Ausgabe erst am Ende des Skripts in "/cleaner.php" zusammengebaut und ausgegeben wird.
\$Meta	Wird direkt nach \$tpl zur Verfügung gestellt und erleichtert das hinzufügen von CSS und JS Dateien und weiteren Kopfdaten für die Webseite.

3.2 Wichtige Klassen

Die folgenden Kapitel sollen eine kurze Einführung in die allerwichtigsten Klassen geben. Natürlich gibt es viele weitere Funktionen, welche Sie aus der API Referenz entnehmen.

3.2.1 Statische Klassen

Um Funktionen anhand Ihrer Aufgaben zu gliedern wurden einige statische Klassen implementiert, welche allgemeine und oft wiederkehrende Aufgaben erleichtern sollen. Darunter sind die wichtigsten:

Klasse	Beschreibung
page::	Die Page Klasse hat verschiedene Methoden um auf die Eigenschaften der Domain, der Page und des Mandanten zuzugreifen. Alle Eigenschaften sind nur zum lesen gedacht. Sehr häufig werden Sie zum Beispiel den "page::menuID()" Aufruf nutzen.
option::	Bietet die Möglichkeit auf Mandantenkonfigurationen zuzugreifen. Diese werden mit einem String identifiziert.
dateOps::	Bietet Funktionen um mit Daten zu hantieren. Es werden derzeit europäische und SQL Formate unterstützt. Wenn Sie SQL Datumsstrings aus der Datenbank lesen können Sie diese hiermit zum Beispiel in ein lesbares Format wandeln.
numericOps::	Bietet einige numerische Funktionen die an Strings und Zahlen angewendet werden können.
ownerID::	Vergibt eindeutige "Owner Ids". Manche Tabellen von sddCMS haben einen über das gesamte System eindeutigen Schlüssel, welcher von dieser Klasse bezogen wird. Das sind: tbmenu, tbcontent, tbcontentsection und tbteaser
pageConfig::	Mit dieser Klasse lassen sich Module mit individuellen Parametern konfigurieren. Als Basis dazu wird die "tbkonfig" genutzt. In den Admin Modulen von sddCMS finden Sie einige Beispiele zur Anwendung (meist config.php) dieser Klasse.
stringOps::	Eine der wichtigsten Klassen, da Sie die Arbeit mit Strings, Konvertierungen und Kodierungen sehr vereinfacht.

3.2.2 Hilfsklassen / Objekte

Nebst statischen Klassen stehen auch diverse instanzierbare Klassen zur Verfügung. Die wichtigsten sind in der folgenden Tabelle erwähnt.

Klasse	Beschreibung
dologin	Behandelt User Logins für sddCMS und kann für spezielle Login Module als Grundlage verwendet werden.
fPdf	Die bekannte (leicht abgeänderte) FPDF Klasse um PDF Dateien direkt in den Browser Output zu generieren oder in Files zu speichern.
htmlControl	Ist an sich eine statische Fabrik welche diverse einfache HTML/Javascript Tools zur Verfügung stellt. Tagclouds, Flying Windows, Tooltips, Kalender selektoren und Ajax Select Boxen.
imageConverter	Simple Klasse um Bilder zwischen den Formaten PNG, GIF und JPG umher zu konvertieren (Vorsicht vor schlechter Qualität!).
imageManipulator	Kann PNG, GIF und JPG Bilder verkleinern. Es gibt eine Erweiterung imageManipulatorToBrowser, welche dasselbe tut, aber die verkleinerte Version nicht in ein File, sondern in den Browser ausgibt.
paging	Diese Klasse bietet die Möglichkeit ein SQL Query auf mehrere Seiten zu verteilen. Es wird automatisch eine Navigation ausgegeben. Sie brauchen sich daher um nichts zu kümmern.
rssDocument	Klasse um RSS Outputs an den Browser zu senden. Kann mit der rssItem Klasse mit Einträgen gefüttert werden.
tabRowExtender	Generiert ein Objekt, welches automatisch "<tr>" oder "<div>" Elemente mit abwechselnden Klassen ausstattet damit man ohne Logik Listen mit unterschiedlich farbigen Zeilen erstellen kann.
uploadFile	Klasse, welche den Upload von Dateien direkt in ein Element in der Datenbank und im Filesystem ablegt.
Dologin	Behandelt User Logins für sddCMS und kann für spezielle Login Module als Grundlage verwendet werden.

3.2.3 Abstrakte Klassen / Interfaces

Nachfolgend seine Abstrakte Klassen und Interfaces beschrieben.

Klasse	Beschreibung
teaser	Interface zur Definition von Teasermodulen. Individuelle Teasermodule müssen von diesem Interface abstammen.
commonModule	Abstrakte Klasse für Module. Alle sddCMS Module sind von dieser Klasse abgeleitet. Bietet diverse Hilfsmethoden.
additionalModule	Gleich wie commonModule ohne Hilfsfunktionen. Für individuelle Module kann eine der beiden Klasse bei Bedarf genutzt werden.

3.3 Dokumentation

Der Download von sddCMS enthält einen „/doc“ Ordner, der die Dokumentationen zum System enthält. Beim Upload auf Ihren Server sollten Sie diesen Ordner aber nicht auslassen, da der Installer auf diverse Skripte angewiesen ist.

3.3.1 Datenbankskripte / Referenz

/doc/sql/instancesdd.pdf -> Dokumentation Instanzdatenbank
/doc/sql/globalsdd.pdf -> Dokumentation globale Datenbank
/doc/sql/data/ -> Datenbankskripte für Datenbankerstellung
/doc/sql/update/ -> Updateskripte nummeriert nach Version

3.3.2 API Referenz

/doc/api/ -> Klassen- und Funktionsdokumentation

3.3.3 Schulungsdokumente

/doc/manual/ -> Schulungsdokumente (Entwickler, Admin, User)
/doc/manual/pdf -> PDF Versionen der Schulungsdokumente

4 Installation

Mit den folgenden Schritten kann sddCMS von Hand installiert werden. Danach können mit dem Installer Webseiten hinzugefügt werden.

- Hochladen aller Dateien
- Falls nicht vorhanden im Root ein File „installed“ erstellen
- Erstellen von zwei Datenbanken (instance, global)
- SQL Verbindung konfigurieren in /config.php
- SQL Skript „instancesdd.sql“ in der Instanzdatenbank ausführen
- SQL Skript „globalsdd.sql“ in der globalen Datenbank ausführen
- Aufrufen von <http://www.deinedomain.ch/install.php>
- Neue Webseite erstellen und Design angeben. Es gibt bereits ein Vorlagedesign mit der Nummer 2, welches Sie anpassen können.
- Aufrufen der neuen Webseite <http://www.deinedomain.ch>
- In der Instanzdatenbank (in tbdomain) allenfalls weitere Domains hinzufügen