



Kajona V3

Developers Handbook

Stand: 27.06.2007

3

Inhaltsverzeichnis

Inhaltsverzeichnis.....	2
Einleitung.....	4
Konzepte.....	5
Voraussetzungen.....	5
Framework.....	5
Module.....	7
Datenbank.....	7
Rechtesystem.....	8
Elemente.....	9
Direkte Elemente.....	9
Vermittelnde Elemente.....	10
Templates.....	10
Platzhalter.....	11
Seitentemplates.....	12
Elementtemplates.....	12
Master-Seite & Mastertemplate.....	14
global_includes.....	15
Administration, Portalanfrage, Adminanfrage.....	16
Portalanfrage.....	16
Adminanfrage.....	17
Installation.....	18
Download.....	18
DB- Konfiguration.....	19
Wizard-Basierte Anpassung.....	19
Old-Style Anpassung ohne Wizard.....	20
Schreibrechte.....	20
Updates.....	21
Module.....	22
Downloads.....	22
Settings.....	22
Template-API.....	22
Filemanager.....	23
Settings.....	23
Bildergalerien.....	23
Settings.....	24
Template-API.....	24
Gästebuchverwaltung.....	25
Settings.....	25
Template-API.....	25
Sprachverwaltung.....	26
Settings.....	26
Navigationsverwaltung.....	26
Settings.....	26
Template-API.....	26
Newsverwaltung.....	27
News-Feeds (RSS).....	27
Settings.....	27
Template-API.....	27

Seitenverwaltung.....	28
Settings.....	29
Template-API.....	29
Seitencache.....	29
Suche.....	30
Settings.....	30
Template-API.....	30
Statistiken.....	31
Settings.....	31
Systemverwaltung.....	31
Sperrern von Datensätzen.....	32
Settings.....	32
ModRewrite.....	32
Debug-Optionen.....	32
Benutzerverwaltung.....	33
Settings.....	33
FAQs.....	33
Settings.....	33
Template-API.....	33
Abbildungsverzeichnis.....	35

Einleitung

Was soll Kajona? Warum Kajona?

Gute Frage, nächste Frage.

So ganz einfach lässt sich diese Frage nicht beantworten. Zu Kajona führten mehrere Gegebenheiten.

Ursprünglich entstand Kajona aus der Not heraus: Bei vielen Projekten kommt man an Punkte, bei denen Probleme gelöst werden müssen, die so ähnlich bereits bei früheren Problemen gelöst wurden. Da aber oft projektspezifische Erweiterungen eingebaut werden, lässt sich der Code nicht einfach übernehmen. Das bedeutet, man muss die Lösung des Problems noch einmal finden. Was ja an sich unnötig ist.

Deshalb wurde Kajona ins Leben gerufen. Kajona sollte ursprünglich eine lose Codesammlung für ähnliche Problemstellungen werden, um damit schnell und effizient Probleme zu lösen. Schnell wurde aber aus der Codesammlung eine komplexe Bibliothek mit eigenen Schnittstellen und eigenen Strukturen. Dieser „Haufen“ wurde dann um grundlegende Funktionen erweitert, die als Vermittler zwischen den Codeblöcken fungierten – das erste kleine Framework war geschaffen.

Das dürfte ungefähr im Jahre 2004 gewesen sein, dem Geburtsjahr der Version 1.

Schnell stellte sich heraus, dass es das noch nicht gewesen sein konnte. So begann das kontinuierliche Hinarbeiten auf Version 2.0 (Codename Kaktena), welches schließlich im Jahre 2005 fertig wurde.

Während dieser Entwicklung kristallisierten sich auch die wahren Ziele des Systems heraus. Es sollte ein reines Web Content Management System (WCMS) werden, welches sich primär auf Geschwindigkeit und Usability konzentrieren sollte.

Die Version 2.1 stellte zwar nur einen kleinen Sprung der Versionsnummer dar, brachte jedoch tief greifende Änderungen auf der Ebene der Datenbank mit sich, war also mehr als lediglich ein Update. Von daher zog sich die Freigabe der Version 2.1 in den Beginn des Jahres 2006.

Von nun an begann die Arbeit an Version 3.0, gekoppelt mit dem internen Release der Version 2.2 im September 2006. Die Versionsnummer 3.0 wurde auf Grund der vielen Änderungen nötig – nicht nur das Datenbankschema wurde angepasst, auch die Klassenstruktur, die Klassenhierarchien oder auch die Klassenverbindungen wurden von Grund auf überarbeitet.

Das Resultat hiervon ist Kajona V3. Das beste Kajona aller Zeiten ;).

Die folgende Dokumentation ist an Entwickler gerichtet, die mit Kajona eine Internetseite erstellen und verwalten möchten. Hierbei wird nicht primär die Bedienung des Systems behandelt, sondern es sollen die Konzepte des Systems erläutert werden. Da diese Doku natürlich nur einen groben Einblick geben kann, dürfen offene Fragen gerne an unser Forum gerichtet werden (www.kajona.de). Ebenso natürlich gefundene Fehler oder aufgekommene Ideen und Anmerkungen.

In diesem Sinne – have fun & enjoy!

Stefan Idler Jakob Schröter

Konzepte

Voraussetzungen

Kajona ist noch jung und verwendet deshalb die aktuellen State-of-the-art Technologien PHP5 und MySQL ab 4.1. Benötigt wird:

- Webserver (Apache, IIS, ...)
- PHP 5 (ab 5.0)
- MySQL (ab 4.1)

Aber im Detail:

Um Kajona einzusetzen ist zu aller erst einmal ein Webserver nötig (wer hätte das gedacht?). Dies kann zum Beispiel ein Apache HTTPD-Server oder auch ein IIS-Server sein, kann aber auch jeder andere Server sein. Einzige Voraussetzung an den Webserver: Er sollte die Möglichkeit bieten, PHP-Skripte auszuführen.

Da Kajona in der „neuen“ Objektorientierung von PHP geschrieben wurde, muss auf dem Webserver mindestens PHP 5 zur Verfügung stehen. Mit PHP 4 lässt sich Kajona leider nicht einsetzen.

Auf der Datenbankseite unterstützt Kajona bisher MySQL, diese dann ab Version 4.1. Dabei ist es egal, ob PHP mysqli oder mysql verwendet, um auf die Datenbank zuzugreifen, beide Schnittstellen sind in Kajona als Treiber realisiert.

Weitere Datenbanken wie PostgreSQL, Oracle oder DB2 sollen folgen.

Scheinen die obigen Anforderungen erfüllt, so sollte einem erfolgreichen Einsatz von Kajona nichts mehr im Wege stehen:



Abbildung 1: Architektur (Server, PHP, DB)

Die Architektur von Kajona soll an verschiedenen Stellen erklärt werden. Hierbei soll ein kurzer Blick auf die allgemeine Struktur geworfen werden, aber auch auf andere Ebenen wie beispielsweise die Datenbank.

Framework

Klar ist, dass Kajona eine strenge Trennung der Logik von der Präsentation verfolgt. Ebenso ist die Datenhaltung / die Persistenz an sich von den beiden erwähnten Schichten zu trennen. Daraus ergibt sich ein dreistufiger Aufbau des Systems:

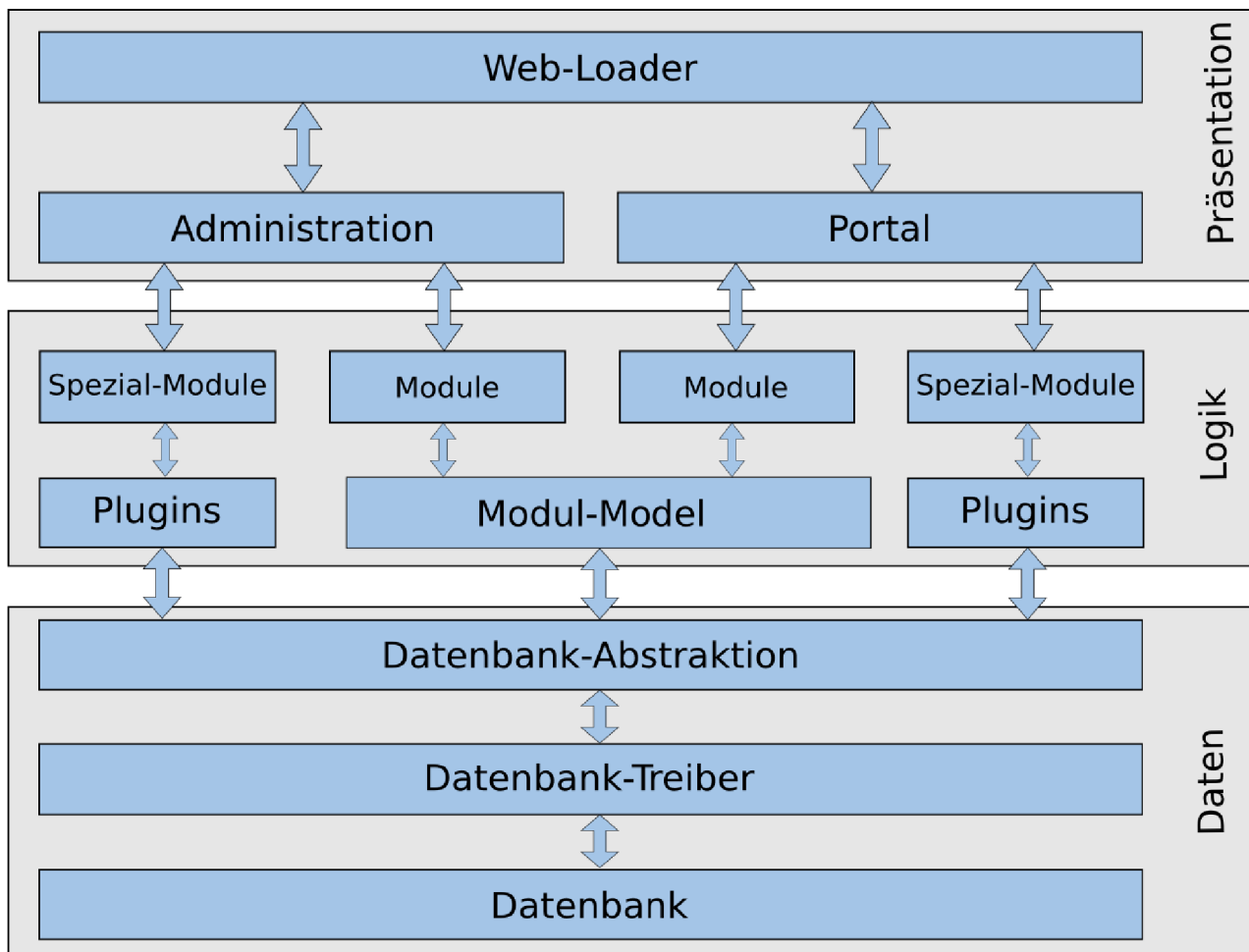


Abbildung 2: Systemarchitektur

Deutlich werden soll hier, dass es keine direkte Verbindung der Präsentationsschicht und der Datenbank gibt. Nur so können Sicherheitsmechanismen effektiv implementiert werden. Ein weiterer Vorteil ist, dass ein Entwickler der Präsentationsschicht nicht den Aufbau der Datenbank kennen muss. Er fordert die benötigten Daten einfach beim Model an, und freut sich dann des Lebens. Wie die Daten vom Model beschaffen werden, ist der Präsentation prinzipiell egal.

Analog verhält es sich mit der Relation des Models zur Datenbank. Dem Model ist es egal, ob die Datenbankschicht die Werte in eine Datenbank, oder in eine XML-Datei schreibt. So lange das Model die Daten erhält, ist alles wunderbar.

Diese Gegebenheiten prägen des Framework-Charakter des Systems. Alle Schnittstellen sind sauber definiert, so dass ein Erweitern des Systems einfach und schnell vorgenommen werden kann.

Des Weiteren wird in Kajona strikt zwischen Portal und Administration unterschieden. Das Portal wird zur Darstellung der Inhalte verwendet, die Administration zur Erfassung und Bearbeitung der Inhalte sowie zur Verwaltung des Systems.

Bei einer Anfrage an das System wird also, je nach Art der Anfrage, in das Portal oder die Administration verzweigt.

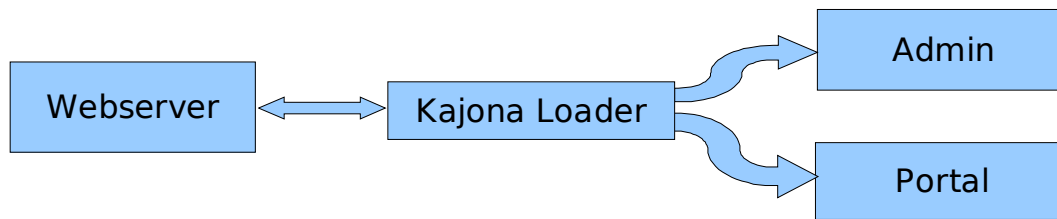


Abbildung 3: Aufteilung Portal / Administration

Module

Kajona ist modular aufgebaut. Das bedeutet, dass zusammengehörige Funktionen in Modulen gebündelt werden, und somit nur bei Bedarf in das System installiert werden. Das Erweitern eines bestehenden Systems kann auf Grund dieser Modularität jederzeit vorgenommen werden. Besteht Bedarf für eine Downloadverwaltung, so kann das Modul „Downloads“ nachträglich in das System installiert werden, um dieses um die nun benötigten Funktionen zu erweitern.

Der Aufbau eines Moduls ist hierbei klar definiert und besteht aus festen Blöcken:

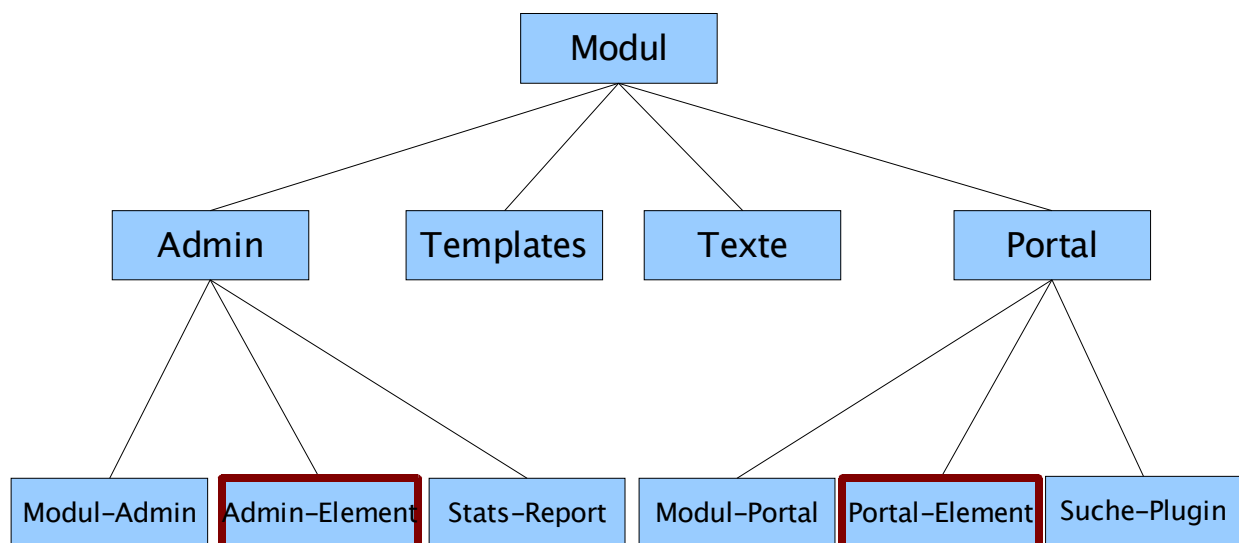


Abbildung 4: Aufbau eines Moduls

Datenbank

Kajona benötigt eine Datenbank, um die Inhalte, die über die Administration erfasst werden, speichern zu können. Aktuell existieren Treiber für MySQL, denkbar ist aber jede andere SQL-fähige Datenbank.

Das Datenbankschema von Kajona ist dabei von den nachfolgenden Eigenschaften gekennzeichnet.

Global Unique Ids (GUIDs).

Guids sind über die Systemgrenze hinweg eindeutig. Dies hat dann Vorteile, wenn man Datensätze des aktuellen Systems in ein Fremdsystem importieren möchte. Da die Erzeugung der

IDs nicht die Datenbank, sondern Kajona übernimmt, sind die IDs bereits vor dem Einfügen in die Datenbank bekannt. Ein Rückfragen der verwendeten ID nach einem Insert entfällt also. Ebenso werden keine zusätzlichen Referenzschlüssel mehr benötigt, da alle Primärschlüssel auf Grund der GUIDs ja bereits eindeutig sind und somit direkt referenziert werden können.

Systemtabelle.

Die Systemtabelle dient als zentrale Verwaltungsinanz und bildet die Strukturen des Systems ab. Verwendet wird hierfür eine Baumstruktur.

Benötigt man mehr Details zu einem Datensatz, als in der Systemtabelle vorhanden, so können diese über JOINS aus weiteren Tabellen hinzu verknüpft werden. Dazu gehören Tabellen wie die Rechte-Tabelle, Datum-Tabelle, Element-Tabelle oder die Modul-Tabelle.

Benötigt man einen Ausschnitt aus der Baumstruktur, so wird einfach die Systemtabelle mit sich selbst verknüpft.

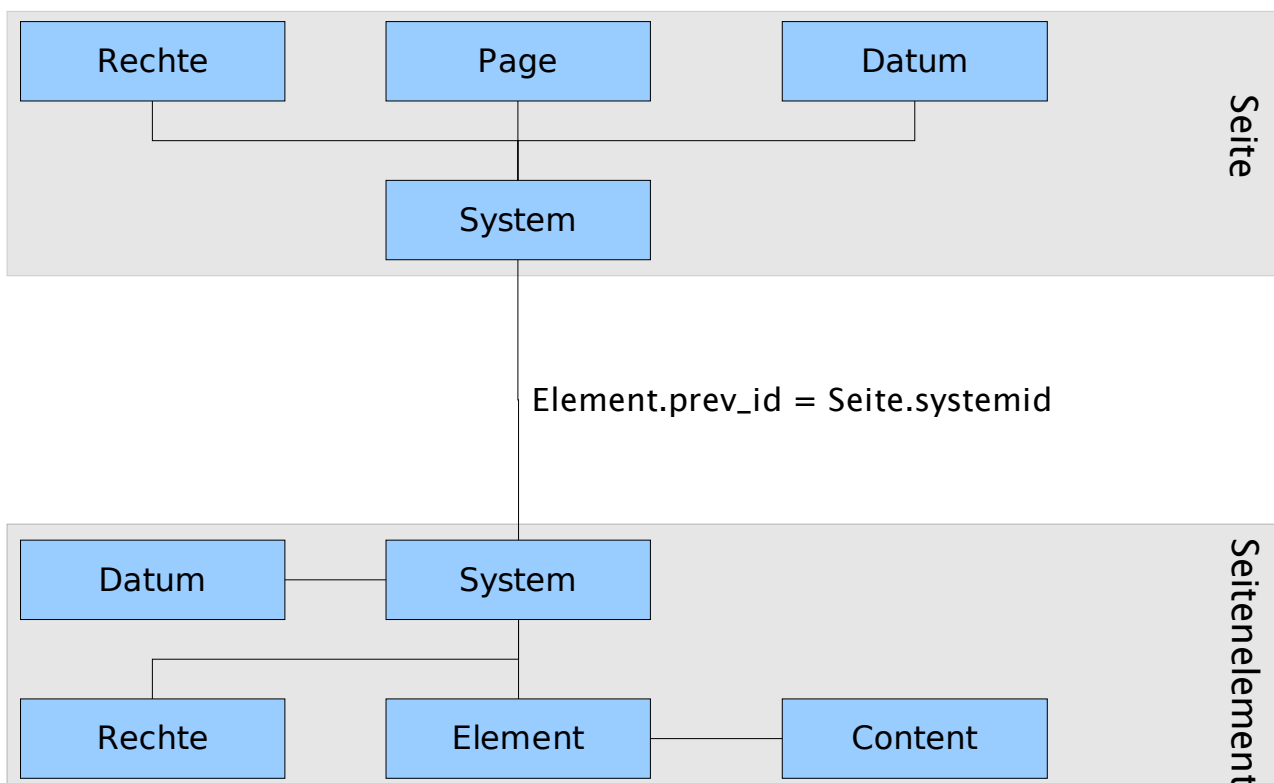


Abbildung 5: Verknüpfung eines Seitenelementes mit der übergeordneten Seite

Rechtesystem

Das Rechtesystem in Kajona orientiert sich an der bereits beschriebenen Baumstruktur der Systemtabelle. Auf Grund dieses Aufbaus ist eine rekursive Rechtevergabe innerhalb des Systems möglich, Datensätze können also von übergeordneten Datensätzen ihre Rechte erben.

Diese Kette der Vererbung kann an jeder Stelle der Hierarchie unterbrochen werden.

Angenommen, es gibt ein paar Seiten, die nur registrierten Benutzern zur Verfügung stehen sollen, so benötigen diese spezielle Rechte. Der eine Weg wäre nun, die Rechte jeder Seite anzupassen.

Der schönere Weg aber ist, alle Seiten in einem Ordner abzulegen. Diesem Ordner gibt man dann die angepassten Rechte – die darin enthaltenen Seiten und Unterordner erben dann automatisch die angepassten Rechte. Diesen Sachverhalt soll auch die nächste Grafik erläutern.

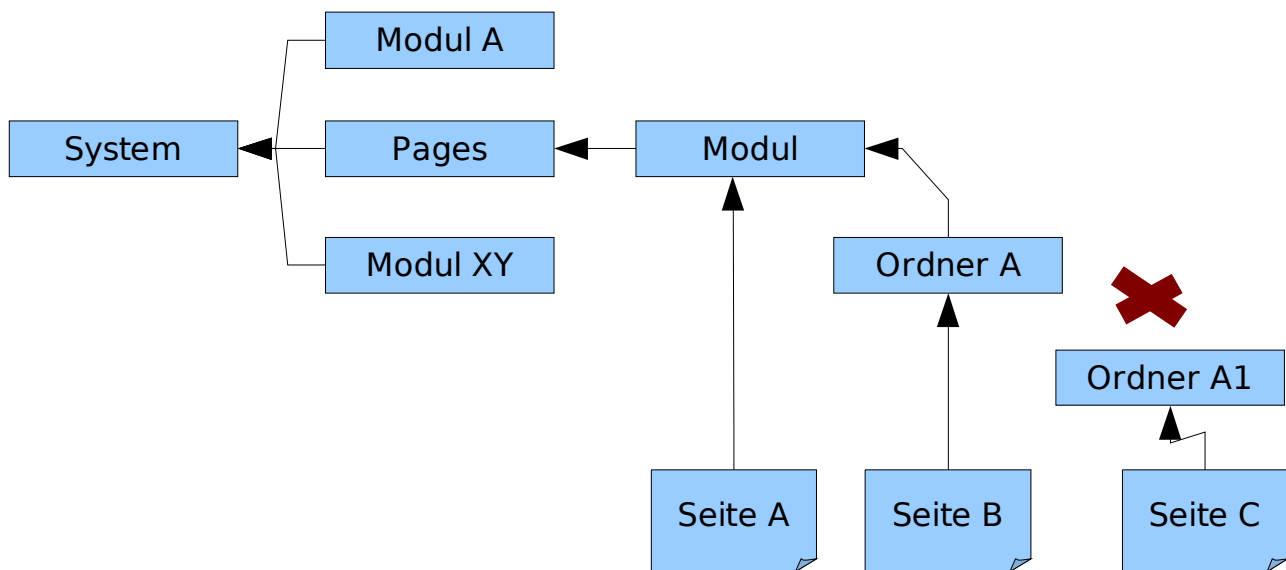


Abbildung 6: Rekursive Rechtevergabe mit Unterbrechung

Vom Konzept der Rechtevergabe her ist das Rechtesystem an dem von UNIX angelehnt. Die Vergabe von Rechten erfolgt also nicht per User, sondern per Gruppe. Hierbei gilt, das beliebig viele Benutzer in beliebig vielen Gruppen Mitglied sein dürfen.

Da die Rechte aber Gruppen abhängig vergeben werden gilt ebenso: Ein User ohne Gruppe ist ein User ohne Rechte. Dieser hat dann gegebenenfalls sogar weniger Rechte als ein Gast!

Es gibt in Kajona zwei vordefinierte Gruppen: Gäste und Administratoren. Gäste sind all die Benutzer, die nicht angemeldet sind. Mitglieder der Gruppe Administratoren hingegen haben superglobale Admin-Rechte, dürfen also alles.

Elemente

Bevor man mit dem System arbeitet, sollte man den Gedanken hinter den Elementen verstanden haben, und die zwei grundlegenden Typen von Elementen kennen.

Elemente werden immer dann verwendet, wenn über den Admin-Bereich erfasste Inhalte im Portal präsentiert werden sollen. Das kann zum Beispiel ein Text-Absatz sein, kann aber auch eine kleine Bildergalerie sein.

Hierfür unterscheidet Kajona zwischen zwei Arten von Elementen:

- direkte Elemente und
- vermittelnde Elemente.

Diese beiden Typen sollen nun genauer beleuchtet werden.

Direkte Elemente

Direkte Elemente werden immer dann eingesetzt, wenn die Inhalte direkt beim Element hinterlegt werden können. Dies ist beispielsweise bei Überschriften oder Absätzen der Fall. Als kleine Grafik könnte man das so darstellen:

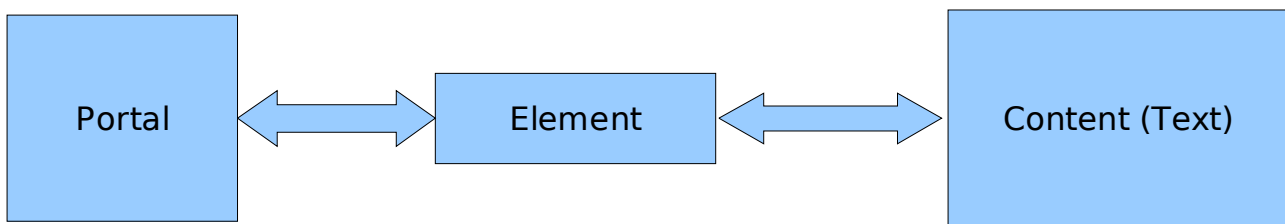


Abbildung 7: Direktes Element

Man erkennt: Wird das Portal aufgerufen, so lädt dieses das Element. Das Element kann dann sofort die zuvor hinterlegten Informationen an das Portal zurückgeben – es liefert den Content direkt.

Vermittelnde Elemente

Im Gegensatz zu direkten Elementen stellen vermittelnde Elemente lediglich eine Art Schnittstelle zu anderen Modulen dar. Das bedeutet: Der Inhalt wird in den jeweiligen Modulen hinterlegt und gepflegt. Das Element stellt dann lediglich ein Bindeglied zwischen Portal und Modul dar:

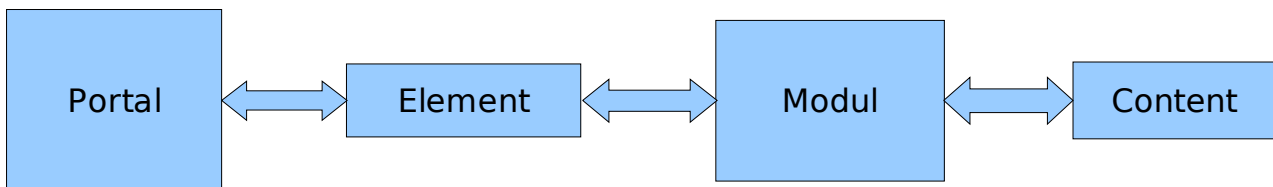


Abbildung 8: Vermittelndes Element

Es wird also deutlich: Das Element fungiert lediglich in vermittelnder Rolle.

Auch hier wird bei einem Aufruf des Portals das Element gebeten, den Inhalt zu liefern. Da dieses aber „nur“ auf ein Modul verweist, leitet das Element die Anfrage an das entsprechende Modul weiter. Das Modul kann nun eigenständig den zuvor hinterlegten Content laden und an das Element zurückgeben. Dieses leitet den Content dann an das Portal weiter, welches dann den Content in die Antwort auf die Anfrage einbettet.

Zusammenfassend:

Ein Element wird immer dann benötigt, wenn Inhalt in die Seiten des Portal geschrieben werden soll. Soll hierbei einfacher Content wie Text dargestellt werden, so kommt ein direktes Element zum Einsatz, soll aber komplexerer Content wie eine Bildergalerie oder Zeit gesteuerte News dargestellt werden, so übergibt das Element die Anfrage an das entsprechende Modul und dient lediglich als vermittelndes Element.

Templates

Bisher wurde von Elementen und Einbetten von Content gesprochen – aber wo soll das denn passieren? Genau da kommen die Templates ins Spiel!

Was also ist ein Template? Templates werden in vielen System an vielen Stellen in vielen Formaten verwendet, deshalb sollen hier nur Templates in Kajona erläutert werden.

Unter einem Template versteht man eine Art „Dokumentvorlage“. Diese liegt allen Seiten zu

Grunde und bestimmt so deren Layout. So ist es möglich, das Layout in einer Datei zu definieren, und diese Datei allen Seiten zuzuweisen. Möchte man nun das Layout verändern, so muss lediglich eine einzige Datei verändert werden.

Templates haben den Vorteil, dass durch die Verwendung von Templates die Inhalte strikt vom Layout getrennt werden. In den Templates stehen lediglich Platzhalter für Inhalte, die Inhalte befinden sich beispielsweise in der Datenbank. Bei einer Anfrage geht das System nun her und führt diese zu einer Seite zusammen:

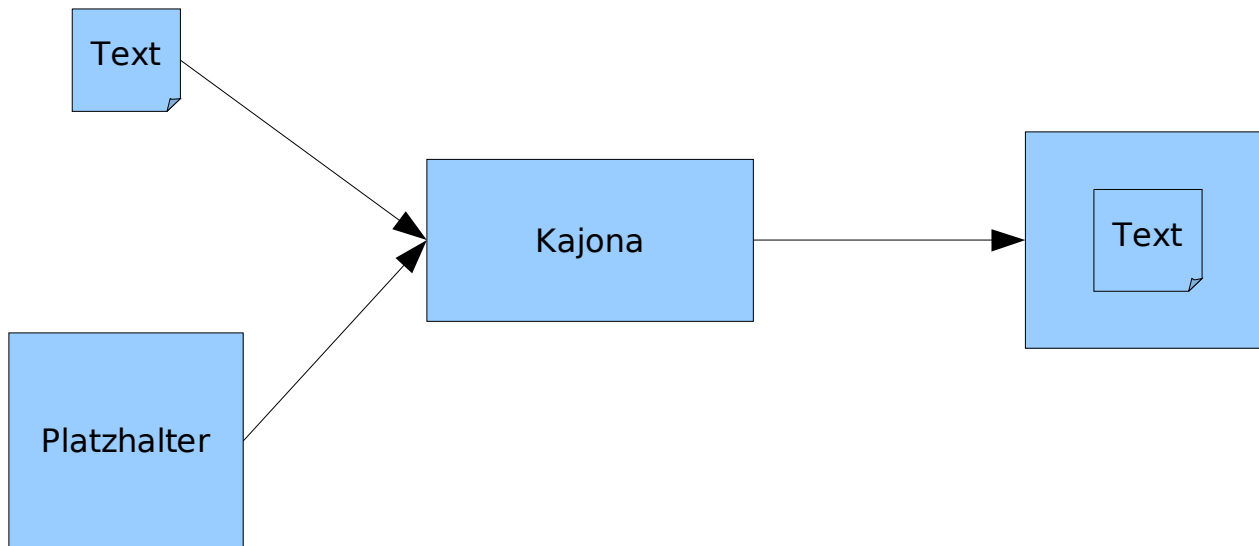


Abbildung 9: Zusammenführen von Content und Template

Platzhalter

Platzhalter werden immer dann benötigt, wenn das System aktiv werden soll.

Das bedeutet: Wenn in einem Template an einer bestimmten Stelle ein Text angezeigt werden soll, dann wird an dieser Stelle ein Platzhalter eingefügt. Dieser wird von Kajona ausgewertet und dann je nach Element mit hinterlegtem Content befüllt.

Diese Platzhalter haben in Kajona eine fest definierte Syntax. Diese soll an einem Beispiel erklärt werden:

```
%%einleitung_absatz%%
```

Als allererstes wird ersichtlich, dass Platzhalter von zwei Prozentzeichen umrahmt werden: %% leitet den Platzhalter ein und beendet diesen auch wieder. So werden Nebeneffekte mit normalem Text vermieden.

Als zweites fällt ein Art „Zweiteilung“ des Platzhalters auf: Ein Unterstrich trennt den Platzhalter in zwei Bereiche.

Vor dem Unterstrich befindet sich der Titel des Platzhalters. Dieser dient nur dem Benutzer, um die Stelle im Template sinnvoll zu benennen. Im Admin-Bereich wird dieser Titel als Platzhalter mit ausgegeben, so weiß man beim Bearbeiten einer Seite schnell, um welche Stelle es denn überhaupt geht.

Zum anderen folgt nach dem Unterstrich das Element an sich.

In Kajona können beliebig viele Seitenelemente hinterlegt werden, die alle einen Titel bekommen. In diesem Beispiel würde es sich also um einen Absatz (ein Absatzelement) handeln.

So weiß Kajona sofort, welches Element geladen werden muss, wenn die Seite in der Administration oder im Portal aufgerufen wird.

Sollen an einem Platzhalter verschiedenen Seitenelemente möglich sein, so lassen sich diese durch ein Pipe-Symbol voneinander trennen:

```
%%einleitung_absatz|bild%%
```

So können an diesem Platzhalter Bilder und Absätze angelegt werden, welche dann zueinander sortiert werden können.

Werden Platzhalter zu einem späteren Zeitpunkt erweitert, so muss das System darüber benachrichtigt werden. Dies erfolgt in der Seitenverwaltung mit Hilfe der Aktion „Platzhalter anpassen“.

Anmerkung: Wird im Platzhalter der Name eines Elements verwendet welches nicht installiert ist, so wird der Platzhalter im Admin-Bereich ausgeblendet!

Seitentemplates

Seitentemplates sind die zentralen Templates, um die es in Kajona geht. Diese bestimmen das „grobe“ Layout des Portals. An sich sind Seitentemplates zu Beginn nichts anderes als normale HTML-Seiten, jedoch anstatt als *.html Dateien gespeichert, bekommen Templates in Kajona die Endung *.tpl.

Ein weiterer Unterschied zu normalen HTML-Seiten zeigt sich im Inhalt. Bei herkömmlichen Seiten würde man seinen Content einfach an die gewünschte Stelle im HTML-Baum schreiben. Bei Templates hingegen wird lediglich ein Platzhalter hinterlegt.

Merke ;) : Anstatt des Contents wird einfach ein Platzhalter an die Stelle im HTML-Layout geschrieben. Der Content selber wird später über die Administration eingepflegt und verwaltet.

Ein Seitentemplate könnte so aussehen:

```
<html>
  <head>[...]</head>
  <body>
    [...]
    <div>
      %%headline_zeile%%
      <hr />
      %%content_absatz%%
      %%intro_bild%%
    </div>
    [...]
  </body>
</html>
```

Elementtemplates

Elementtemplates stellen eine Eigenheit von Kajona dar, die das System enorm flexibel macht. Hat man nun im Seitentemplate einen Platzhalter hinterlegt und im Admin-Bereich bereits Content für den entsprechenden Platzhalter hinterlegt, so wird dieser ab sofort bei einem Aufruf des Portals dargestellt. Um auch diesen Content layouttechnisch anzupassen, gibt es Elementtemplates, die zwischen Seitentemplates und Content geschaltet werden:

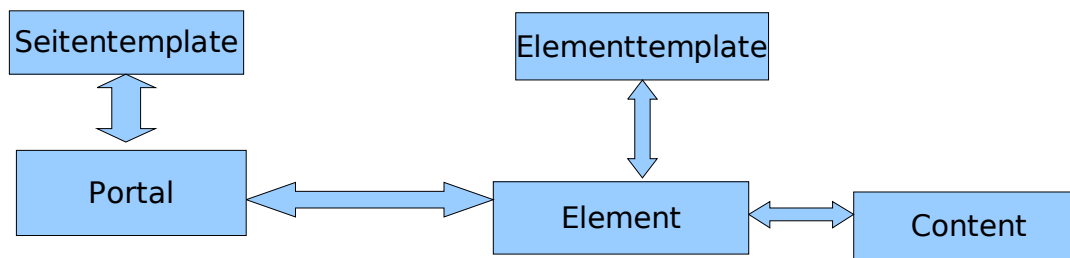


Abbildung 10: Zusammenführen der Templates zu einer Seite

In Kajona 2.0 wurden Elementtemplates auch Modultemplates genannt, da dies aber schnell zu Missverständnissen führen kann, soll diese Bezeichnung nun aus Kajona verschwinden.

Elementtemplates haben außerdem die Eigenheit, kein reguläres HTML-Dokument darzustellen, sondern lediglich einen Ausschnitt aus einem HTML-Dokument (Code-Snippet). Diese Abschnitte werden mit zuvor definierten Abschnitt-Tags eingeleitet. Ein häufiges Tag für listen ist das `<list></list>` Tag. Der Inhalt dieses Tags steuert dann, wie die Liste später aussehen soll.

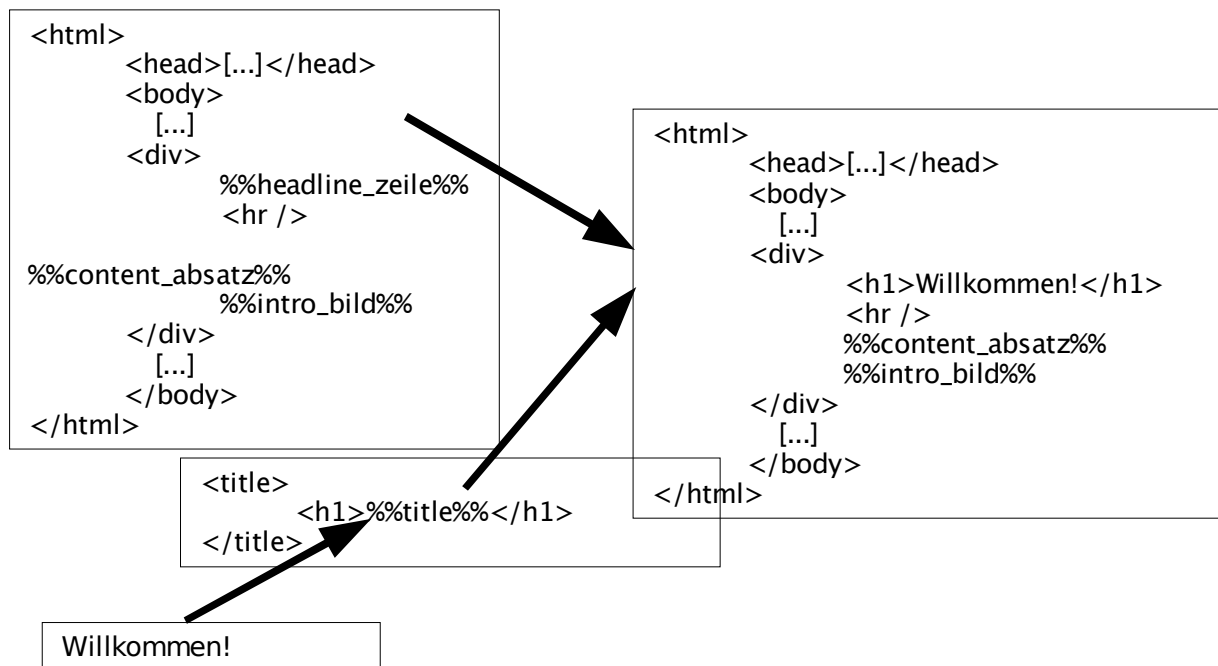
Ein Elementtemplate könnte so aussehen:

```

<list>
    Downloads: <br />
    %%rows%%
</list>

<list_row>
    &middot; %%filename%% &nbsp; &nbsp; &nbsp; %%dllink%%
</list_row>
  
```

Wenn das System nun die Seite erzeugt, dann werden die Inhalte in die Templates geschrieben, und diese nacheinander zusammengebaut. Ein Ablauf könnte folgendem Schema entsprechen:



Master-Seite & Mastertemplate

Die Masterseite ist eine Spezialseite im System. Diese trägt per Definition den Titel „master“ und kann auch nicht anders benannt werden.

Was aber ist der Sinn der Masterseite? Betrachten wir noch einmal kurz die Element-Situation: Zuvor wurde von vermittelnden Elementen gesprochen, also Elementen, die zwischen einer Seite und einem Modul vermitteln. Nun kann es durchaus Module geben, die auf jeder Seite angezeigt werden sollen. Ein Kandidat hierfür wäre das Modul Navigationen. Macht Sinn, den man möchte dem Benutzer schließlich auf jeder Seite die Navigation anbieten. Eine mögliche (und bisher die einzige Art) dies zu erreichen wäre, auf jeder Seite ein vermittelndes Element anzulegen, um die Navigation einzubinden. Diese Variante ist natürlich unschön, da wir sind ja schließlich faul ;).

Deshalb bietet Kajona für solche Fälle ein Special-Feature: Die Masterseite.

Ein paar kurze Sätze zum Prinzip der Masterseite: Auf der Masterseite werden, wie auf allen anderen Seiten auch, Elemente angelegt. Diese Elemente der Masterseite können nun aber auf allen anderen Seiten mit angezeigt werden, ohne diese erneut anlegen zu müssen. Im Beispiel der Navigation bedeutet dies, dass man die Navigation nur auf der Masterseite anlegt, auf allen anderen Seiten wird die Navigation automatisch eingebunden.

Wie man das erreicht soll der nächste Abschnitt zeigen.

Ganz klar – eine Masterseite benötigt ein spezielles Template. Unterschied zu Seitentemplates: es besteht nicht aus einem HTML-Gerüst, sondern ist an sich erst mal komplett leer. In dieses leere Template werden dann alle Platzhalter geschrieben, für die globale Elemente angelegt werden. Da es sich um Master-Elemente handelt, haben diese eine spezielle Syntax, um sie von regulären Platzhaltern zu unterscheiden.

Kurze Wiederholung eines normalen Platzhalters: %%portalnavigation_navigation%%. So würde der Platzhalter auf einer normalen Seite aussehen, um die Navigation einzubinden.

Auf der Master-Seite jedoch haben Platzhalter per Konvention das Schema %%mastertitel_element%%. Es wird also die Zeichenkette „master“ vorne angestellt. Im Beispiel der Navigation würde der Platzhalter also lauten: %%masterportalnavigation_navigation%%.

Diese Platzhalter können dann über den Admin-Bereich, wie alle andere Platzhalter auch, mit

Content befüllt werden.

Nun ist noch die Frage offen, wie denn nun die Elemente der Master-Seite auf die normale Seite kommen. Denn diese Verbindung besteht ja nicht automatisch.

Dies lässt sich aber sehr sehr einfach erreichen: An die Stelle innerhalb des normalen Seitentemplates, an die ein Element der Masterseite eingefügt werden soll, wird ein weiterer Spezialplatzhalter eingefügt, analog nach dem Schema `%%mastertitel_element%%`. (Der aufmerksame Beobachter erkennt: Genau der gleiche Platzhalter wie im Mastertemplate :)).

In unserem Beispiel würde man also an die Stelle im Seitentemplate, an der die Navigation erscheinen soll, einen Platzhalter mit dem Titel `%%masterportalnavigation_navigation%%` einfügen. Und zack, bei einem Aufruf der Seite wird der Platzhalter automatisch mit dem identisch benannten Element der Masterseite befüllt.

global_includes

Mit der Masterseite haben wir nun eine Möglichkeit kennen gelernt, Elemente nur einmal anzulegen aber überall anzuzeigen. Doch damit nicht genug, Kajona bietet eine weitere Möglichkeit, schnell und einfach kleine Elemente überall anzuzeigen.

Diese Möglichkeit sind globale Elemente. Warum aber eine weitere Möglichkeit, und wann benutzt man welche Methode, um Elemente überall anzuzeigen?

Die Elemente der Masterseite haben einen Vor- und Nachteil: Es sind reguläre Seitenelemente. Das bedeutet, bei der Erzeugung durch das Portal werden verschiedene Dinge im Hintergrund erledigt, dazu gehört die Überprüfung, ob das Element aktiv oder inaktiv ist, oder natürlich auch, ob die Rechte ausreichen, um das Element anzuzeigen. Dies kostet natürlich Laufzeit, denn das alles passiert ja nicht von alleine.

Nun gibt es aber durchaus Elemente, bei denen man darauf verzichten kann, dass die oben genannten Punkte immer abgearbeitet werden. Ein Beispiel hierfür wäre das Anzeigen einer Copyright-Zeile: „(c) 2005-2006 by mir“. Diese Zeile kann man nun selbstverständlich einfach in die Templates schreiben, also in das HTML-Layout einbauen. Was passiert aber 2007? Genau, es müssen alle Templates bearbeitet werden, um die Zeile anzupassen. Das ist natürlich Arbeit, die uns Kajona abnehmen könnte. Und genau hierfür gibt es globale Elemente. Diese werden in der Datei „global_includes.php“ im Verzeichnis /portal hinterlegt und dann im Template eingebunden. In unserem Beispiel würden wir die Datei also einfach um Folgendes erweitern:

```
$arrGlobal["copyright"] = "(c) 2005-2007 by mir";
```

In den Templates kann nun diese Zeile mit Hilfe des Platzhalters `%%copyright%%` eingefügt werden. Ist nun Silvester, und man möchte das Copyright anpassen, so muss lediglich diese eine Zeile in der Datei `global_includes.php` abgeändert werden, und schon erscheint auf allen Seiten das neue Copyright.

```
$arrGlobal["copyright"] = "&copy; 2000 – 2007 kajona.de";
```



```
<html>
  [...]
  <div class="copyright">%%copyright%%</div>
</body>
</html>
```

Administration, Portalanfrage, Adminanfrage

Bisher wurde immer von der Administration oder dem Admin-Bereich geredet. Nun soll kurz darüber geredet werden, was sich dahinter überhaupt verbirgt.

Kajona ist strikt in zwei Bereiche unterteilt: Das Portal und die Administration. Bei der Erstellung und Planung von Kajona wurde großen Wert darauf gelegt, diese Bereiche nicht zu vermischen.

Das Portal tritt immer dann in Aktion, wenn ein Internetuser die Seite aufruft. Das Portal kümmert sich also darum, die Seiten zu generieren und an den Browser auszuliefern. Und das natürlich möglichst schnell...

Die Administration hingegen funktioniert zwar auch über den Browser, dient aber nicht für den normalen Gast auf der Website, sondern den Autoren und Admins des Webauftritts. Mit der Administration oder dem Admin-Bereich können alle redaktionellen und administrativen Tätigkeiten ausgeführt werden, die im Leben einer Webseite auftreten. Dazu gehört das Anlegen, Bearbeiten oder Löschen von Seiten, das Hochladen von Dateien, Verwalten von Rechten und so weiter.

Um die Administration eines Kajona-Systems zu erreichen, genügt es im Browser die Adresse <http://www.meine-domain.ab/admin/>

einzugeben. Dann weiß Kajona sofort, dass nun die Administration und nicht das Portal zu laden ist.

Die Zugangsdaten nach der Installation des Systems befinden sich in der unten stehenden Doku des User-/System-Moduls.

Portalanfrage

Der Vollständigkeit halber sei eine Anfrage an das Portal schematisch dargestellt:

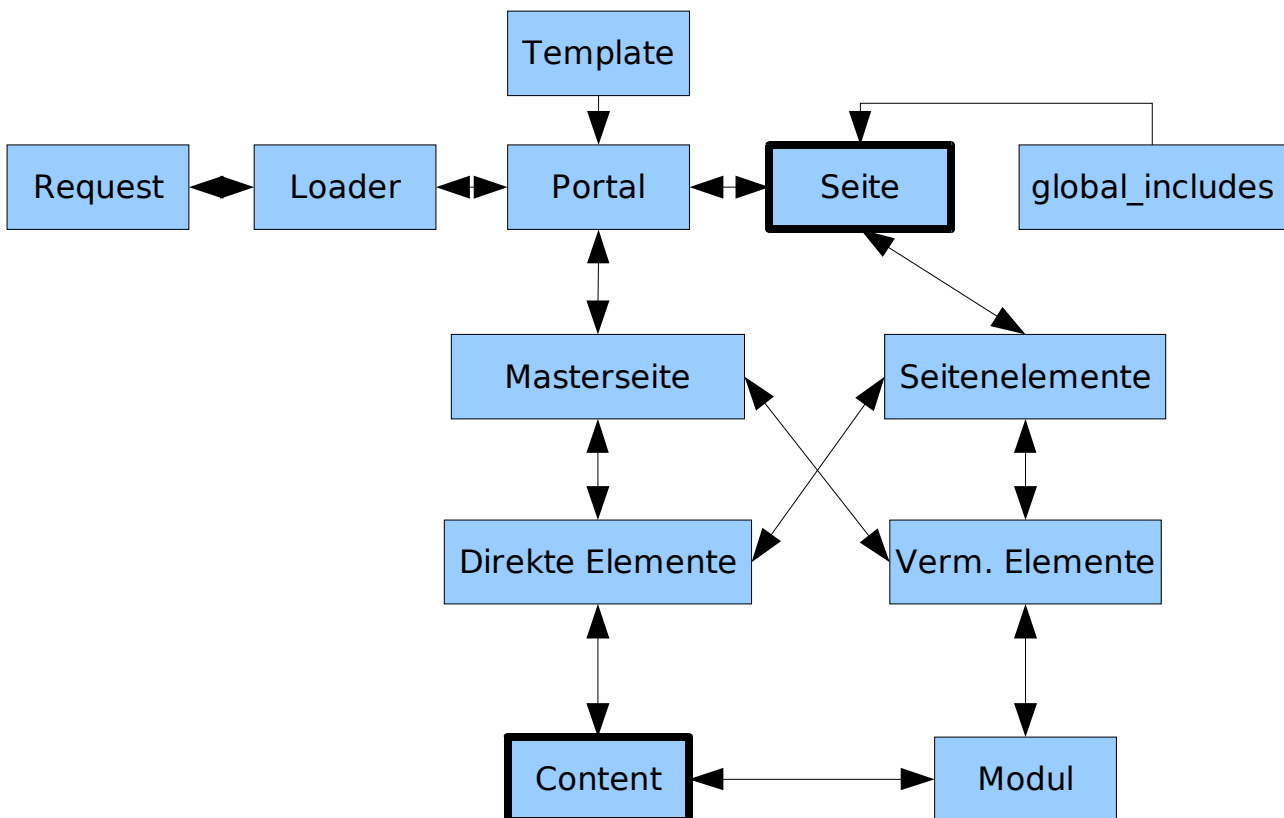


Abbildung 11: Eine Portalanfrage

Adminanfrage

Der Vollständigkeit halber sei eine Anfrage an die Administration schematisch dargestellt:

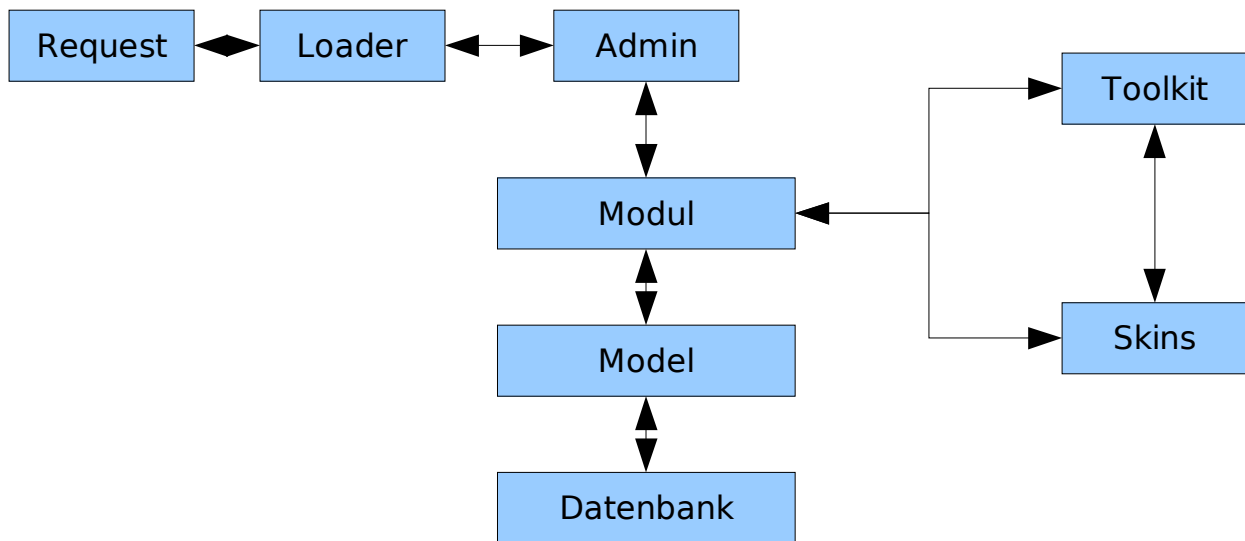


Abbildung 12: Eine Anfrage an die Administration

Installation

Nach diesen allgemeinen Worten zu Kajona geht es nun an das Interessante: Das Installieren eines Kajona Systems.

Das funktioniert recht einfach: Herunter laden, hin-und-her-kopieren, installieren, fertig. ;)

Doch Eins nach dem Anderen...

Download

Sollte dieser Schritt noch nicht passiert sein, so ist nun der richtige Zeitpunkt dafür, das System erst einmal herunterzuladen.

Die aktuellste Version des Systems befindet sich immer auf <http://www.kajona.de>.

Dort gibt es einen Download-Bereich, in dem alle offiziellen Module zum Download angeboten werden.

Da Kajona ja ein schlankes CMS sein soll, gibt es (vorerst) auch kein „Alles is drinne“-Paket, sondern alle Module können/müssen einzeln, je nach Bedarf, heruntergeladen und dann später installiert werden.

Keine Angst: Bemerkt man später, dass man die Bildergalerie ja doch gebraucht hätte, so lässt sich diese auch später noch problemlos in das System installieren.

Einzig die Versionen könnten (!) hier zum Problem werden – am besten kurz im Forum nachfragen, ob ein Update des Systems notwendig wird (und selbst das ist in den meisten Fällen kein Problem), oder nicht.

Sind also alle Module heruntergeladen und entpackt, müssen diese in einen gemeinsamen Ordner kopiert werden. Dabei kann der Dateimanager des Systems unter Umständen nachfragen, ob vorhandenen Dateien ersetzt werden sollen. Diese Frage einfach bejahen – es werden nur Ordner, aber keine Dateien ersetzt. Hat man alle Module in einen Ordner kopiert, könnte die Dateistruktur so aussehen:

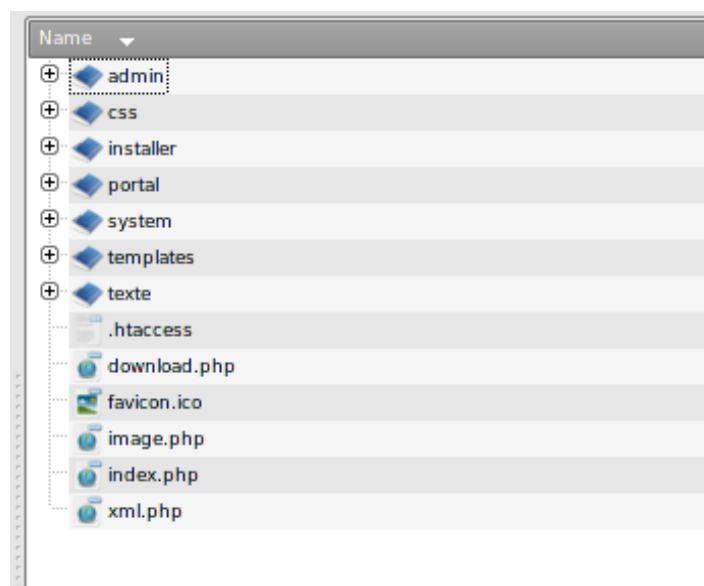


Abbildung 13: Dateistruktur nach dem Download und Kopieren

Nun geht es an das eigentliche Installieren des Systems.

Sollte man das System auf einem entfernten Server installieren wollen, dann können die Dateien nun per FTP, SSH oder was auch immer übertragen werden.

Danach kann der Installer des Systems über die URL

`http://www.meine-domain.ab/installer/installer.php`

aufgerufen werden.

Der Installer leitet schrittweise durch die Installation, so dass man an sich fast nichts falsch machen kann.

Im ersten Schritt werden die Schreibrechte und PHP-Module überprüft, die Kajona benötigt. Sollten hier einzelne Punkte rot markiert werden, sollte hier gegebenenfalls Abhilfe geschaffen werden. Sind alle Punkte grün, dann kann es mit dem nächsten Schritt weiter gehen – dem Erfassen der Datenbankeinstellungen. Hierfür gibt es nun zwei Wege zur Konfiguration.

DB- Konfiguration

Damit das System funktionsfähig wird, muss es die Zugangsdaten zur Datenbank kennen. Diese sollte übrigens im Vorfeld bereits angelegt worden sein. Bei Webhosting-Paketen lässt sich dies oft über eine Admin-Oberfläche des Providers erledigen.

Um die Config-Datei zu bearbeiten, gibt es zwei Wege. Zum einen kann man die Datei von Hand editieren, womit man mehr Flexibilität und genauere Einstellungen erhält.

Seit Kajona 2.1 gibt es aber auch einen Wizard, der die benötigten Werte in die Config-Datei schreibt.

Wizard-Basierte Anpassung

Führt man die Installation regulär mit Hilfe des Installers durch, so erscheint nun ein Formular zum Erfassen der Datenbankeinstellungen.

Wie der Installer schon besagt, muss der Webserver auf die config.php-Datei Schreibrechte besitzen, um die eingegebenen Werte in die Datei schreiben zu können. Diese Datei befindet sich im Ordner /system. Nach der erfolgreichen Installation sollten die Rechte wieder zurückgesetzt werden.

Die Rechte einer Datei können mit jedem gängigen FTP-Programm gesetzt werden (Hinweis: `chmod 0777`).

Stimmen die Rechte, können die Werte gesetzt werden. Die meisten sind selbsterklärend, ein Hinweis zum Präfix: Dieses wird jeder Tabelle vorne angestellt, so ist es möglich in einer Datenbank ohne Konflikte verschiedenen System zu installieren.

Vielleicht noch eine kleine Bemerkung zum DB-Driver, also dem Datenbank-Treiber. Momentan kann man hier zwischen `mysql` und `mysqli` auswählen. Der Unterschied zwischen beiden ist nicht all zu leicht erklären. Historisch gesehen ist `mysql` der ältere, und `mysqli` der neuere Treiber, um das System mit einer MySQL-Datenbank zu verwenden. Die Kajona-Treiber verwenden hierfür die Schnittstellen, die der Webserver und somit PHP zur Verfügung stellen, in PHP werden diese Module ebenfalls `mysql` und `mysqli` genannt. Welcher Treiber zu verwenden ist, hängt also von der Konfiguration des Webserver ab. Ob der Webserver `mysql` oder `mysqli` unterstützt, erfährt man entweder beim Betreiber des Webserver, oder über die `phpinfo()`-Funktion.

Wurden die Werte erfolgreich in die Datei geschrieben, zeigt der Installer automatisch die einzelnen Module an, die Installiert werden können.

Hinweis: Sollte es notwendig sein, leere Werte zu vergeben, beispielsweise für das

Datenbankpasswort, so muss dies von Hand in die Konfigurationsdatei eingetragen werden. Von leeren Werten wird aber aus Sicherheitsgründen unbedingt abgeraten!

Old-Style Anpassung ohne Wizard

Für alle Freunde der Config-Dateien, können die Datenbankeinstellungen für Kajona selbstverständlich auch ohne Wizard angepasst werden.

Hierfür muss die Datei config.php im Ordner /system geöffnet und bearbeitet werden.

In der Config-Datei können in verschiedenen Abschnitten beliebige viele Server hinterlegt werden. Dies hat den Vorteil, das man das System lokal und auf dem Webserver installieren kann, ohne bei einem Upload oder Download des Systems die Einstellungen erneut anpassen zu müssen.

Wichtig ist lediglich, dass der spätere Live-Server den default-Abschnitt bekommt, um unerwünschte Effekte zu vermeiden. Außerdem müssen die vorhandenen Platzhalter im Defaultabschnitt gegebenenfalls durch Leer-Strings ersetzt werden, anderenfalls beschwert sich der Installer (und zwar zurecht :)) über fehlende Werte in der Konfiguration.

Am Ende der Datei können zusätzlich weitere Debug-Optionen ein oder ausgeschaltet werden, die im Falle von Fehlern bei der Suche nach der Ursache helfen können.

Sind die Werte für die Datenbank eingetragen, so kann auch hier mit der eigentlichen Installation des Systems fortgefahren werden. Hierfür muss im Webbrowser erneut der Installer aufgerufen werden.

Im nächsten Schritt wird der Zugang zur Administration eingerichtet. Hierfür kann man einen Benutzernamen und ein Passwort wählen. Diese Daten werden dann im Verlauf der Installation während der Konfiguration des Systems verwendet.

Wurden die Login-Daten erfasst, erfolgt die Installation der einzelnen Module. Dieser Schritt sollte selbsterklärend sein, ebenso wie der darauf folgende Schritt zur Installation von Seitenelementen.

Wurde das Modul „Samplecontent“ heruntergeladen, so steht es im nächsten Schritt zur Installation bereit – anderenfalls folgt eine kleine Seite zum Abschluss des Installers.

Das System ist nun erfolgreich installiert – viel Spaß!

Schreibrechte

Auch in Kajona benötigen ein paar Dateien / Ordner Schreibrechte. Diese werden jedoch auch vom Installer überprüft.

Diese können bei entfernten Servern mit jedem gängigen FTP-Programm gesetzt werden.

Folgende Verzeichnisse benötigen Schreibrechte des Webserverns:

- Der Bilder-Cache Pfad: /portal/pics/cache
 - Hinweis: Dieser Pfad kann über die Einstellungen verändert werden.
- Der Ordner /system/debug: In diesem wird das System-Logfile und gegebenenfalls das Logfile der Datenbankabfrage abgelegt.
- Für die Sicherungen der Datenbank und den Statistiken aus der Administration heraus benötigt der Ordner /system/dbdumps Schreibrechte.
- Alle Pfade, die mit dem Filemanager (für Uploads) erreicht werden sollen.

Updates

Da Kajona ja ständig weiterentwickelt wird, kommt irgendwann der Moment, an dem man das System updaten möchte / sollte. Generell ist es zu empfehlen, das eigene System auf eine neue Version zu aktualisieren, wenn diese offiziell freigegeben wurde.

Mit einem Update werden nicht nur neue Funktionen eingespielt, sondern in der Regel auch bekannte Fehler behoben.

Ein Update des Systems erfolgt in zwei Schritten: Zu erst werden alle Dateien auf den aktuellen Stand gebracht, danach wird das System über die Datenbank aktualisiert.

Zu Beginn sollte man jedoch, wie bei jedem Software-Update, eine Sicherung des aktuellen Systems vornehmen. Dazu gehören nicht nur die Dateien, sondern auch die Datenbank. So kann man im Fall der Fälle alles wieder rückgängig machen.

Das Erneuern der Dateien erfolgt denkbar einfach: Man lädt die aktuellste Version von Kajona herunter (also alle Module, die im System installiert sind), und kopiert diese in das „alte“ System.

Wichtig ist hierbei, **NICHT** den Ordner „Templates“ mit zu kopieren! Sonst werden die in mühevoller Handarbeit erstellten Templates bei gleichem Dateinamen von den default-Templates ersetzt, und das möchte natürlich niemand. Das Gleiche gilt für die Datei „global_includes.php“ sowie für die „config.php“-Datei.

Sollten im Verlauf der Erstellung der Homepage weitere Dateien oder Klassen angepasst worden sein, so sollte man diese nur mit Vorsicht ersetzen.

Sind alle Dateien auf dem Server, so kann nun die Datenbank angepasst werden, um die für das neue System notwendigen Strukturen anzupassen oder zu installieren. Hierfür ruft man einfach in bekannter Art und Weise den Installer auf. Dieser bietet nun die Möglichkeit, die Module upzudaten. Ist dies erfolgt, so erscheinen auch im Installer und in der Systemverwaltung die Module unter der neuen Modul-Version.

Nach erfolgreichem Update des Systems sollte man wieder, wie bereits nach der Installation, den /install-Ordner löschen.

Module

Dieses Kapitel soll nun die einzelnen Module ein wenig beleuchten und erklären.

Die Doku der einzelnen Module ist im Moment leider noch ein wenig knapp gehalten, dass soll sich aber mit der Zeit verbessern.

Liefert ein Modul Settings, also Einstellungen mit, so werden diese Parameter ebenfalls erläutert. Diese können im Admin-Bereich im Modul System unter dem Punkt „Systemeinstellungen“ bearbeitet werden.

Downloads

Dient, wie der Name schon sagt, zum Verwalten von Downloads. Hierzu wird ein Archiv angelegt, das auf einen beliebigen Ordner im System zeigen kann.

Ist das Archiv in seinen Grunddaten angelegt, so kann es über die Aktion „Synchronisieren“ mit dem Dateisystem per Knopfdruck abgeglichen werden. Hierbei werden alle Dateien und Unterordner ab dem angegebenen Pfad untersucht und gegebenenfalls in die Datenbank mit aufgenommen. Es wird also das Dateisystem in der Datenbank abgebildet.

Wenn eine Datei gelöscht wurde, so wird diese nun ebenfalls aus der Datenbank gelöscht.

Die gesammelten Dateien können dann per Aktion „Archiv öffnen“ betrachtet werden. Hier lassen sich auch zusätzliche Informationen wie eine Beschreibung zu den Dateien hinterlegen.

Zu jeder Datei kann ein Geschwindigkeitslimit angegeben werden. Wenn dies ungleich 0 ist, kann die Datei nur mit der angegebenen KB-Rate heruntergeladen werden, bei 0 mit unbegrenzter Geschwindigkeit.

Zusätzlich gibt es ein Logbuch, welches die Downloads protokolliert.

Das Layout im Portal wird, wie üblich, über ein im Seitenelement angegebenes Template gesteuert.

Settings

Trefferseite: Diese Einstellung definiert, auf welche Seite die Suche verweisen soll, wenn Downloads zum angegebenen Suchwort gefunden werden. Dies ist nicht automatisch möglich, da es verschiedene Seiten geben kann, die die Downloads einbinden und somit keine eindeutiges Ziel bestimmt werden könnte.

Template-API

Abschnitt „file“:

- file_name
- file_description
- file_hits
- file_size
- file_md5sum
- file_preview (bei *.jpg, *.png, *.gif)
- file_link
- file_href

Abschnitt „folder“:

- folder_name
- folder_description
- folder_link
- folder_href

Abschnitt „list“:

- folderlist
- filelist
- pathnavigation

Abschnitt „pathnavi_entry“:

- path_level

Filemanager

Der Filemanager dient zur Dateiverwaltung im System, bspw. für den Upload von Bildern in die Bildergalerie.

Hierfür werden verschiedenen Repositories angelegt und mit verschiedenen Parametern wie einem Upload- oder einem Ansichtsfiler angelegt. Sollen z.B. im Repository „Pics“ nur jpg- und gif-Dateien als Upload erlaubt sein, so wäre eine Upload-Filter „.gif,.jpg“ zu setzen.

Das Modul Filemanager hat logischerweise keine Portalausgabe.

Installiert wird der Filemanager über das Modul System, es ist als bei einer Standardinstallation automatisch mit dabei.

Settings

Mit der Einstellung „Größe anzeigen“ wird festgelegt, ob die Größe von Ordnern rekursiv berechnet werden soll, oder nicht. Bei tiefen Verzeichnisstrukturen kann dies zu Performanceproblemen führen und kann deshalb hier deaktiviert werden.

Bildergalerien

Funktioniert analog wie das Modul Downloads. Einer der Unterschiede liegt aber im Seitenelement, dieses lässt sich mit mehr Optionen konfigurieren:

Der Parameter „Anzahl in Zeile“ gibt an, wie viele Vorschaubilder pro Zeile angezeigt werden sollen. Das Template muss hierfür angepasst werden!!!!

Außerdem lassen sich für die Vorschaubilder und die Detailansichten maximale Abmessungen angeben. Werden die Abmessungen von einem Bild überschritten, werden diese vom System automatisch klein gerechnet und gecached. Steht der Wert auf 0, so werden die Bilder nie skaliert.

Der Parameter „Inskrift“ legt in das Bild der Detailansicht einen Text an der Position X/Y, relativ zur linken, oberen Ecke.

ACHTUNG: Für die Skalierung und das Einlegen von Texten benötigt Kajona die PHP-Erweiterung GD-Lib.

Settings

Die Einstellung „Bildtypen“ beschränkt die Formate, die die Bildergalerie verarbeiten soll.

Der Wert der Trefferseite beschreibt (siehe Downloads), auf welcher Seite die Detailansicht der von der Suche gefundenen Bilder erfolgen soll.

Template-API

Abschnitt „piclist“:

- pic_(nr)
- name_(nr)
- subtitle_(nr)

Abschnitt „folderlist“:

- folder_name
- folder_description
- folder_link
- folder_href
- folder_subtitle

Abschnitt „list“:

- pathnavigation
- piclist
- folderlist

Abschnitt „picdetail“:

- pic_url
- backlink
- backlink_image_(1..3)
- backlink_image_filename_(1..3)
- backlink_image_systemid_(1..3)
- forwardlink
- forwardlink_image_(1..3)
- forwardlink_image_filename_(1..3)
- forwardlink_image_systemid_(1..3)
- overview
- pic_small
- pathnavigation
- systemid
- pic_name
- pic_description
- pic_filename
- pic_size
- pic_hits

- pic_subtitle

Abschnitt „pathnavigation_level“:

- pathnavigation_point

Gästebuchverwaltung

Das Modul Guestbook dient zum Verwalten der im System hinterlegten Gästebücher.

Dabei kann definiert werden, wie viele Einträge pro Seite im Portal erscheinen sollen, und ob neue Einträge erst freigeschaltet werden müssen, oder sofort sichtbar sind (moderiertes Gästebuch). Das Layout wird über das im Seitenelement gewählte Template festgelegt.

Zur Sicherheit gegen automatisierte Einträge von Bots lässt sich ein Captcha-Bild einbinden.

Hinweis: Damit sich Portalbesucher eintragen können, muss die Gruppe Gäste das Recht „Eintragen“ besitzen!

Settings

Auch beim Gästebuch kann eine Trefferseite definiert werden. Findet die Suche einen Suchbegriff in einem Gästebuch-Post, so leitet die Suche auf die hier angegebene Seite zur Darstellung des Posts weiter.

Template-API

Abschnitt „post“:

- post_name
- post_page
- post_text
- post_date

Abschnitt „list“:

- liste_posts
- link_newentry
- link_forward
- link_pages
- link_back

Abschnitt „entry_form“:

- action
- Erwartete Formular-Elemente:
 - gb_post_name
 - gb_post_email
 - gb_post_page
 - gb_post_text
 - gb_post_captcha

Sprachverwaltung

Seit Version 3.0 beherrscht Kajona das Verwalten verschiedener Sprachen im Portal. Hierfür wird eine Seite in der Seitenverwaltung, wie sonst auch, angelegt. Der Unterschied ist aber nun, dass sowohl die Seiteneigenschaften, also auch die Seiteninhalte sprachabhängig gespeichert werden. Die Seite „Partybilder“ kann in der Sprache Deutsch also das Template „bilder_de.tpl“, im Englischen aber „bilder_en.tpl“ als Template verwenden.

Die Sprache, die als default gewählt wurde, wird als Start-Sprache gewählt, bis der Benutzer eine andere Sprache auswählt (Anmerkung: Im Portal werden die Sprachen ausgewertet, die der Browser als default-Sprachen sendet).

Eine Spezialaktion ist „Content zuweisen“. Diese kommt nur dann in Aktion, wenn das Sprachen-Modul später in das System installiert wurde. In diesem Fall sollen ja die bisherigen Inhalte nach wie vor verwendet werden können, obwohl diese ja bisher keiner Sprache zugeordnet wurden. Mit dieser Aktion werden alle noch nicht zugeordneten Inhalte der Default-Sprache zugeordnet.

Settings

Keine vorhanden

Navigationsverwaltung

Im Modul Navigation werden die im System zur Verfügung stehenden Navigationen angelegt. Jede Navigation kann beliebig viele Unterpunkte beinhalten und auf externe oder interne Seiten verweisen.

Ist ein Navigationspunkt inaktiv, so erscheint er auch nicht im Portal, entsprechendes bei fehlenden Rechten.

Das Layout der Navigation wird über das Navigations-Element festgelegt.

Hierzu wird der Navigation ein Template zugewiesen. Zusätzlich kann eine CSS-Klasse zugewiesen werden. Diese wird später um ein paar Parameter ergänzt. An den angegebenen Namen wird das Suffix „-ebeneX“, wobei Ebene durch die aktuelle Ebene ersetzt wird, ergänzt. Ist der Navigationspunkt aktiv, so erhält er ein weiteres Suffix „-active“. So könne alle Ebenen in allen Zuständen genau definiert werden.

Seit Kajona 2.1 unterstützt Kajona die optionalen Abschnitte „first“ sowie „last“, um das Layout der Navigation noch flexibler zu halten.

Settings

Keine vorhanden

Template-API

Abschnitt „level_(x)_(active|inactive)[_first|_last]:“

- css
- text
- link
- image
- page_intern

- page_extern
- level(x+1)
- href
- target

Newsverwaltung

Das News Modul dient zum Anlegen, Steuern und Verwalten von News.

Beim Anlegen von News wird ein Titel für die News vergeben, sowie ein Start-, Ende- und Archivdatum. Außerdem wird festgelegt, in welchen Kategorien die News erscheinen soll.

Über das Seitenelement News werden die angelegten News im Portal veröffentlicht. Hierbei wird die anzuzeigende Kategorie, ein Template für die Anzeige und ein Modus definiert. Modus unterscheidet hier zwischen der Listenansicht, sowie der Detailansicht der News.

Zusätzlich gibt es den Archivmodus einer Newskategorie. In diesem landen automatisch alle News, die das Archivdatum überschritten haben, das Enddatum aber noch nicht erreicht haben. So kann man nicht mehr aktuelle News von der Top-News Liste verbannen, aber nach wie vor auf der Webseite im Archiv anbieten.

News-Feeds (RSS)

Mittlerweile beherrscht Kajona auch News-Feeds im RSS-Format. Diese werden dazu verwendet, um Inhalte der Seite abonnieren zu können, um bei neuen News diese sofort lesen zu können.

Hierfür müssen Feeds angelegt werden, die dann News aller oder einer bestimmten Kategorie anzeigen können. Dazu muss jedoch das Seitentemplate angepasst werden. Es sollte ein Abschnitt wie der folgende eingefügt werden:

```
<link rel="alternate" type="application/rss+xml" title="Kajona News" href="Adresse/zum/feed" />
```

Die Adresse des Feeds kann man sich aus der Liste der Feeds im Admin-Bereich kopieren. Das ist der Text zwischen dem Feedtitel und den Aktionen. Sollte das System mit eingeschaltetem Mod-Rewrite laufen, dann ist auch ein Aufruf des Feeds über das Schema /feedName.rss möglich. Das System erstellt daraus dann automatisch die URL zum gewünschten Feed. Um diesen Titel und den Feed-Titel an sich unterschiedlich vergeben zu können, können zwei unterschiedliche Titel vergeben werden – einer für den Aufruf, und einer als interner Feedtitel.

Das link-Tag samt Attribut wird innerhalb des Head-Bereichs des Seitentemplates platziert, also NICHT im Body-Bereich.

Settings

Wie auch bei der Bildergalerie und den Downloads wird bei den News über den Wert „Treffer-Seite“ gesteuert, auf welcher Seite die Detailansicht der von der Suche gefundenen News erfolgen soll.

Template-API

Abschnitt „news_list“:

- news_more_link
- news_start_date
- news_id
- news_title

- news_intro
- news_text
- news_image

Abschnitt „news_detail“:

- news_back_link
- news_start_date
- news_id
- news_title
- news_intro
- news_text
- news_image

Seitenverwaltung

Das Modul Pages ist das eigentliche „Herz“ des Systems, es steuert alle Portalausgaben und verwaltet die Seiten samt ihren Inhalten. Mit der Seitenverwaltung können Seiten editiert, bearbeitet, gelöscht oder mit Hilfe von Ordnern kategorisiert werden.

Diese Ordner werden lediglich virtuell angelegt, sie dienen nur für eine bessere Übersicht im Admin-Bereich und zum Vereinfachen der Rechtevergabe auf Seiten.

Über die Aktion „Neue Seite“ kann eine neue Seite im System hinterlegt werden. Dabei werden das zu verwendende Template für die Seite sowie ein paar Zusatzinformationen festgelegt. Jeder Seitenname muss im System eindeutig sein!

Nach dem Speichern erscheint die Seite in der Seitenliste.

Dort können die Seiteninhalte per Klick auf das Symbol „Seiteninhalte bearbeiten“ verwaltet werden.

Hierbei wird das verwendete Template auf Platzhalter analysiert, mit den im System hinterlegten Elementen abgeglichen und nach bereits angelegten Platzhaltern gesucht. Lässt es das Element zu (es ist also „Wiederholbar“), dann kann an einem Platzhalter ein Element mehrfach angelegt werden, sonst natürlich nur einmal.

Die Formulare der Elemente werden von den Elementen selber gesteuert, diese werden von der Seitenverwaltung aus gerufen.

Außerdem kann über die Seitenverwaltung der Status von Seiten und Elementen verändert werden: Eine Anzeige im Portal erfolgt nur mit dem Status aktiv.

Seitenelemente sind kleine Bausteine, die später den Inhalt auf der Seite darstellen. Seitenelemente bestehen immer aus einer Klasse, die die Eingabe im Admin-Bereich steuert, so wie einer Klasse, die die Ausgabe im Portal-Bereich steuert.

Neue Seitenelemente lassen sich mit der Aktion „Neues Element“ anlegen. Eine Liste der verfügbaren Seitenelemente erhält man über die Aktion „Elemente“. Die hier aufgelisteten Elemente können im zweiten Teil des Platzhalters in den Templates verwendet werden. Wird in einem Template ein Element angegeben, welches dem System unbekannt ist, so wird dieser Platzhalter später auch nicht ausgegeben.

In den Eigenschaften des Elements kann angegeben werden, ob es Wiederholbar ist, oder nicht. Wenn ein Element nicht wiederholbar ist, kann es pro Platzhalter nur einmal angelegt werden, sollte es wiederholbar sein, so kann es mehrfach angelegt werden.

Ebenfalls kann die Cachedauer des Elements bestimmt werden. Es gilt immer: Das Element mit der kleinsten Cachedauer bestimmt die maximale Zeit der Seite im Seitencache.

Settings

Die Startseite legt logischerweise fest, welche Seite als Startseite ausgeliefert wird, also immer dann, wenn kein expliziter Seitenname übergeben wurde.

Das „Standardtemplate“ legt fest, welches der vorhandenen Templates bei neuen Seiten als Template vorausgewählt sein soll.

Die Fehlerseite legt eine Seite fest, die immer dann geladen wird, wenn es zu einem Fehler kam. Dies kann der Fall sein, wenn eine Seite nicht gefunden wurde, zum Beispiel wenn keine Startseite definiert wurde oder wenn die Rechte für die angeforderte Seite nicht ausreichend sind.

Die Einstellung „Templatewechsel“ schließlich legt fest, ob das Ändern des Templates einer Seite möglich ist, wenn auf dieser bereits Elemente angelegt wurden. Wird dies zugelassen, so kann dies unter Umständen zu Geist-Datensätzen und unerwarteten Nebeneffekten führen (Was passiert mit einem Element, das zuvor im Template vorhanden war, nun aber nicht mehr vorhanden ist?).

Mit dem Wert der Einstellung „Neue Seiten inaktiv“ wird festgelegt, ob neue Seiten beim Anlegen aktiv oder inaktiv geschaltet werden sollen. Sind Seiten inaktiv, so ist eine Vorschau über den Link „Vorschau anzeigen“ in der Seitenverwaltung trotzdem möglich.

Mit der Version 2.1.1.0 bringt Kajona eine Erweiterung des Pages-Modules: Den Seitencache.

Der Seitencache dient zum Zwischenspeichern bereits erzeugter Seiten. So müssen diese, bei einer erneuten Anfrage, nicht komplett neu generiert werden, sondern können sofort ausgeliefert werden.

Aktiviert oder deaktiviert wird dieser über die Einstellung „Seitencache aktiv“.

Die Einstellung „Maximale Cachedauer“ gibt an, wie lange eine Seite (in Sekunden) maximal im Cache vorgehalten wird, bis diese verfällt. Dieser Wert kann von Seitenelementen herunter gesetzt werden.

Über den Wert der Einstellung „Portaleditor aktiv“ wird das Verhalten des Portaleditors eingestellt. Wenn der Wert auf true steht, und der aktuelle Benutzer die benötigten Rechte besitzt, dann können Seiten auch direkt im Portal editiert werden.

Template-API

Seitentemplate:

- description
- keywords
- title
- \$arrGlobal-Elemente
- [title]_[element]
- master[title]_[element]
- Systemkonstanten:
 - _indexpath_
 - _webpath_
 - _gentime_

Seitencache

Seit Version 2.1.1.0 kommt Kajona mit einem Seitencache. Dieser sorgt dafür, dass eine Seite beim ersten Aufruf und damit dem ersten Erzeugen in der Datenbank gespeichert wird. Somit wird

bei einem späteren Aufruf der Seite diese nicht neu erzeugt, sondern aus dem Cache geladen und von dort ausgeliefert. Damit der Seitencache vernünftig funktioniert, müssen die Elemente hierfür korrekt konfiguriert werden. Der Seitencache wird durch die Elemente gesteuert, diese entscheiden, wann eine Seite im Cache verfällt. Es gilt von daher: Das Element mit der kürzesten Cachedauer entscheidet über den Verwurf der Seite im Cache. Von daher kann jedem Element eine Anzahl an Sekunden mitgegeben werden. Während dieser Zeitspanne gilt das Element als gültig und die Seite wird aus dem Cache ausgeliefert. Ist die Zeitspanne überschritten, wird die Seite neu generiert. Nach der Installation sind alle Elemente auf eine Dauer von -1 eingestellt. Das bedeutet, dass keines der Elemente gecached wird. Diese Werte müssen also angepasst werden!

Jedoch sollten die Einstellungen des Seitencaches mit Vorsicht vorgenommen werden, um Nebeneffekte zu vermeiden. Zur Erklärung dient das Szenario einer Newsseite. Diese ist im Portal ganz normal verlinkt und zeigt die aktuellen News in der Listen- und Detailansicht an. Stellt man nun die Cachezeit des News-Elements auf eine lange Cachedauer, so treten verschiedene Folgeeffekte ein. Zum einen erscheinen neue News, die laut Newsgrunddaten schon seit 10 Uhr online sein sollten, erst dann, wenn die Cachezeit abgelaufen ist. Zum anderen werden nicht mehr alle Zugriffe auf die Detailansicht der News geloggt / gezählt. Dies wird verhindert, da die Seite aus dem Cache des Systems ausgeliefert wird, nicht jedoch neu erzeugt wird. Der Zugriff auf die News wird aber nur bei einem Generieren der Seite protokolliert, da nur in diesem Fall das Newsmodul in Aktion tritt.

Zusammenfassend sei gesagt, dass nicht alle Elemente pauschal eine lange Cachedauer bekommen sollten, sondern individuell für jedes Element abgeschätzt werden sollte, welche Dauer vertretbar ist.

Suche

Das Modul Suche bietet eine Suche für das Portal.

In Version 2.2 wurde die Architektur der Suche grundlegend überarbeitet. Die Suche ist nun nach einem „plug-n-play“ System aufgebaut. Jedes Modul bringt ein eigenes Such-Plugin mit, welches unter `/portal/searchplugins/` abgelegt wird.

Sofern man keine eigenen Module in das System einbindet, ist aber nur das Plugin der Pages interessant, genauer gesagt die Datei `„class_modul_pages_search.php“`. Hat man nun ein eigenes Seitenelement entwickelt und möchte, dass dieses ebenfalls durchsucht werden soll, so muss man lediglich ein paar Zeilen in der Datei einfügen. Im ersten Drittel der Datei, im Konstruktor der Klasse, werden im Array `$arrSearch` alle Tabellen samt Spalten definiert, die das Plugin durchsuchen soll.

Eine eigene Tabelle müsste als nach dem Schema `$arrSearch["pages_elements"]["table_to_search"][] = "row_to_search";` hinzugefügt werden.

Das war schon alles!

Hinweis: Verwendet ein weiteres Seitenelement eine Tabelle, die schon in der Datei definiert wurde, so muss diese NICHT erneut hinzugefügt werden.

Settings

Keine Vorhanden

Template-API

Abschnitt „search_form“:

- action
- Formular-Elemente:

- searchterm, value search_term

Abschnitt „search_hitlist_hit“:

- page_link
- page_description

Abschnitt „search_hitlist“:

- search_term
- search_nrresults
- link_forward
- link_back
- link_overview

Statistiken

Das Modul protokolliert alle Portalzugriffe auf das System. Im Admin-Bereich können diese dann ausgewertet werden und in verschiedenen Reports betrachtet werden.

In der Version 2.1.1.0 von Kajona wurden die Stats intern umgeschrieben. So ist es nun möglich, eigene Reports zu schreiben, die von Kajona automatisch eingelesen und ausgewertet werden, dies erfolgt durch ein Plugin-Konzept.

Zusätzlich wurden die Stats um so genannte „Worker“-Tasks erweitert. Diese kommen immer dann zum Einsatz, wenn die gesammelten Daten der Besucher auf irgendeine Art und Weise bearbeitet werden sollen. Eine Anwendung ist beispielsweise, die IP-Adressen der Besucher in aussagekräftigere Hostnames zu wandeln. Da dieser Vorgang aber längere Zeit in Anspruch nimmt, wurde hierfür ein extra Worker-Task erstellt. Auch der Import oder Export von Statistik-Daten im CSV-Format erfolgt über die Workertasks.

Settings

Der Wert „Anzahl Einträge“ legt fest, wie viele Zeilen in den Reports angezeigt werden sollen.

Mit dem Parameter „Anzahl Sekunden“ wird festgelegt, in welchem Zeitraum ein User als online gilt. Liegt sein letzter Zugriff auf das Portal schon länger als diese Zeitspanne zurück, dann gilt er als offline.

Über die Liste, die in „Auszuschließende Domains“ hinterlegt wird, können bestimmte Domains aus den Reports ausgeschlossen werden. Dies macht unter Anderem bei den Referrern Sinn.

Systemverwaltung

Das System-Modul bietet die Möglichkeit grundlegende Einstellungen vorzunehmen.

Es bietet zum Einen eine Liste der aktuell verfügbaren Module, zum Anderen ein paar Informationen zum Webserver.

In der Systemverwaltung lassen sich die Root-Rechte definieren. Das ist der oberste Datensatz der Hierarchie, von diesem erben die Module sowie die untergeordneten Datensätze ihre Rechte.

Unter dem Punkt „Datenbankmanagement“ können Wartungstasks der Datenbank gestartet werden. Dies kann eine Sicherung der Datenbank oder auch ein Import einer früheren Sicherung sein.

Die Option „System-Log“ gewährt einen Einblick in das Systemlogbuch. Die Granularität des

Loggens kann hierfür angepasst werden (siehe unten).

Ob das System noch aktuell ist, erfährt man unter dem Punkt „Updates“. Hier werden die lokal installierten Modulversionen mit den aktuell verfügbaren Versionen verglichen. Ein automatisches Update stellt diese Funktion bewusst nicht zur Verfügung.

Sperren von Datensätzen

Bislang nur vom Modul Pages verwendet, unterstützt Kajona das Sperren von Datensätzen, um ein zeitgleiches Bearbeiten eines Datensatzes von zwei unterschiedlichen Usern zu verhindern. Ein gesperrter Datensatz wird in der Regel vom System selber entsperrt. Sollte ein Datensatz nicht entsperrt werden, so kann ein Mitglied der Gruppe „Administratoren“ diese Sperre explizit wieder aufheben. Dies erfolgt direkt in der jeweiligen Listenansicht.

Settings

Mit der Einstellung „Portal deaktiviert“ kann das gesamte Portal ausgeschaltet werden. Ist diese Einstellung aktiviert, dann wird stattdessen die unter „Zwischenseite“ angegebene Seite geladen.

Der Eintrag „Cachepfad“ legt fest, wo die vom System berechneten Bilder zwischengespeichert werden sollen. Dieses Zwischenspeichern ist sinnvoll, da dann bei einer späteren Anfrage auf das selbe Bild dieses nicht neu berechnet werden muss, sondern direkt aus dem Cache ausgeliefert wird.

Anzahl DB-Dumps legt fest, wie viele Sicherungen der Datenbank im System vorgehalten werden sollen. Der Wert sollte nicht all zu hoch sein, da sonst schnell viel Speicherplatz verbraucht wird.

Die im Feld „Admin E-Mail“ hinterlegte E-Mail-Adresse wird dafür verwendet, um im Fehlerfall eine E-Mail zu verschicken. Wird keine Adresse angegeben, wird auch keine E-Mail verschickt.

Der Wert, der unter „maximale Sperrdauer“ definiert wird, gibt an, nach wie vielen Sekunden gesperrte Datensätze automatisch wieder entsperrt werden.

Mit der Einstellung „GZIP-Kompression“ wird festgelegt, ob die Ausgaben des Systems per gzip komprimiert werden sollen. Dies hat den Vorteil, dass die vom Server zum Client versendeten Daten geringer ausfallen, Server und Client jedoch komprimieren, bzw. entpacken müssen.

ModRewrite

Mit der Einstellung „Mod Rewrite“ wird festgelegt, ob das System die Rewrite-Extension verwenden soll. Mod Rewrite ist ein Modul des Apache-Webserver und wird in der Regel dazu verwendet, dynamische URLs in scheinbar statische html-URLs zu verwandeln. Aus `index.php?seite=bilder&action=folderImage&systemid=23hgk57gfhdK58jhgf87` wird dann `bilder.folderImage.23hgk57gfhdK58jhgf87.html`. Das ist nicht nur für den Menschen, sondern zum Beispiel auch für Suchmaschinen besser lesbar. Zusätzlich zu dieser Einstellung im System muss die Einstellung in der `.htaccess`-Datei auf „on“ gesetzt werden. In der `.htaccess`-Datei also einfach aus „RewriteEngine off“ ein „RewriteEngine on“ machen – das wars. Ist das Modul nun auch aktiviert, so werden die Links umgeschrieben.

Debug-Optionen

Oft kann es nützlich sein, bei der Suche nach einem Fehler mehr Informationen zur Verfügung zu haben. Hierfür bietet Kajona verschiedenen Debug-Optionen an. Diese werden alle in der `config.php`-Datei im `/system/` Ordner definiert.

Die Einstellungen befinden sich im Abschnitt „System-Settings“.

Einschalten lassen sich normale Debug-Ausgaben, wie die Zeit (time), Anzahl Datenbankabfragen (dbnumber) und Anzahl verarbeiteter Templates (templatnr) die zur Generierung der Seite benötigt wurden. Zusätzlich kann über die Einstellung „memory“ der Speicherverbrauch

ausgegeben werden, welcher für die Generierung einer Seite benötigt wird.

Es lassen sich aber auch Werte für das Debuggen im Hintergrund einstellen: Ist „dblog“ auf „true“ gesetzt, wird in ein Logfile ein Protokoll aller Datenbankabfragen geschrieben. Der Wert „debuglevel“ gibt an, ob Fehler wie Datenbankfehler ausgegeben werden sollen, oder stillschweigend übergangen werden sollen. Auf Live-Systemen sollte dieser Wert immer „0“ sein!

Mit der Einstellung „debuglogging“ wird festgelegt, wie detailliert das Log-File aufgebaut werden soll.

Benutzerverwaltung

Das Modul User dient zur Verwaltung der im System hinterlegten Gruppen und Benutzer.

Es können neue Benutzer und Gruppen angelegt werden, die Mitgliedschaften verändert werden, und ein Login-Protokoll ausgegeben werden.

Hinweis: Bei der Installation wird ein Benutzer mit dem Benutzernamen und Passwort angelegt. Diese Werte werden im Laufe der Installation erfasst. Mit diesem kann man sich nach der Installation an der Administration anmelden.

Settings

Der Wert des default-Skins sollte klar sein: Was wird als Standard-Skin verwendet, bis der User einen Skin explizit auswählt.

Mit dem Wert „ID der Admin-Gruppe“ kann festgelegt werden, welche Gruppen-ID als globaler Admin behandelt wird. Analog definiert der Wert der Gäste-Gruppe, was als Gast behandelt werden soll.

Der Wert „Anzahl Zeilen“ legt fest, wie viele Zeilen im Login-Protokoll ausgegeben werden sollen.

Die Einstellung „Eigene Daten“ erlaubt Anwendern, auch dann ihr Profil zu verändern, wenn der User an sich keine Rechte am Usermanagement hat. Wird der Wert auf „Nein“ gesetzt, dann benötigt der Benutzer die entsprechenden Rechte an der Benutzerverwaltung.

FAQs

Mit Hilfe des FAQ-Moduls können häufig gestellte Fragen samt Antworten auf der Webseite verwaltet werden. Hierfür bietet das Modul im Adminbereich die Möglichkeit, die FAQs zu erfassen und mit Hilfe von zuvor oder später angelegten Kategorien zu verwalten.

Diese Kategorien können dann im Portal als Listen ausgegeben werden.

Settings

Die Suche-Seite dient, analog wie bei anderen Modulen, zur Darstellung von Suchtreffern. Wird ein Frage- / Antwortpaar von der Suche gefunden, so verlinken die Treffer auf die hier angegebene Seite. Auf dieser sollten dann die Faqs dargestellt werden.

Template-API

Abschnitt „faqs_list“:

- faq_categories

Abschnitt „faq_category“:

- faq_cat_title
- faq_faqs

Abschnitt „faq_faq“:

- faq_question
- faq_answer
- faq_systemid

Abbildungsverzeichnis

Abbildung 1: Architektur (Server, PHP, DB).....	5
Abbildung 2: Systemarchitektur.....	6
Abbildung 3: Aufteilung Portal / Administration.....	7
Abbildung 4: Aufbau eines Moduls.....	7
Abbildung 5: Verknüpfung eines Seitenelementes mit der übergeordneten Seite.....	8
Abbildung 6: Rekursive Rechtevergabe mit Unterbrechung.....	9
Abbildung 7: Direktes Element.....	10
Abbildung 8: Vermittelndes Element.....	10
Abbildung 9: Zusammenführen von Content und Template.....	11
Abbildung 10: Zusammenführen der Templates zu einer Seite.....	13
Abbildung 11: Eine Portalanfrage.....	16
Abbildung 12: Eine Anfrage an die Administration.....	17
Abbildung 13: Dateistruktur nach dem Download und Kopieren.....	18