



Community-basierte Wissensportale InDeKo.Navi Atlas Installationshandbuch

**Ausarbeitung im Rahmen der Veranstaltung IT-Studienprojekt Master
SoSe 2016 - WiSe 2016/2017**

Institut für Betriebswirtschaft und Wirtschaftsinformatik,
Arbeitsgruppe Informationssysteme und Unternehmensmodellierung

Stephan Mende, Friederike Lauterbach, Konstantin Janzen, Robin Helmedach
13. März 2017

Inhaltsverzeichnis

Abkürzungsverzeichnis	iii
1 Installation	1
1.1 Installationsschritte	1
2 Projektstruktur	2
2.1 Grundsätzliche Drupal-Struktur	2
2.2 Grundsätzliche Modulstruktur	3
3 Anmerkungen zum Produktivbetrieb	4
4 Funktionalität	5
4.1 Entwickelte Custom-Module	5
4.1.1 form_wissenskarte_module	6
4.1.2 comments_shortener	7
4.1.3 imagemap_resizer	8
4.1.4 jq_maphilight	9
4.1.5 morphsearch	10
4.1.6 morphsearch_csv_export	12
4.1.7 morphsearch_sort	13
4.1.8 node_creation_links	14
4.1.9 publication_form	15
4.1.10 slider_tooltip	16
4.1.11 user_profile_elements_overview	17
4.2 Drupal Customizing	18
4.2.1 Darstellung der Suchergebnisse	18
4.2.2 Slider mit Wissenskarten als Startseite	20
4.2.3 Apache Solr Search	21
4.2.4 Inhaltstypen	22
4.2.5 Design	22
5 Fehler und deren Lösungen	23
6 Verwendete Ressourcen und deren Lizenzen	24
7 Offene Punkte	26
Glossar	26

Abbildungsverzeichnis

Abb. 1	Grundstruktur des Drupal-Projekts	2
Abb. 2	Grundstruktur Custom-Modul	3
Abb. 3	Entwickelte Custom-Module	5
Abb. 4	Custom-Modul form_wissenskarte_module	6
Abb. 5	Custom-Modul comments_shortener	7
Abb. 6	Custom-Modul imagemap_resizer	8
Abb. 7	Custom-Modul jq_maphilight	9
Abb. 8	Custom-Modul morphsearch	10
Abb. 9	Custom-Modul morphsearch Beispiel publikationsbezogene Suche	10
Abb. 10	Custom-Modul morphsearch Konfiguration morphologische Suche	11
Abb. 11	Custom-Modul morphsearch_csv_export	12
Abb. 12	Custom-Modul morphsearch_sort Beispiel	13
Abb. 13	Custom-Modul node_creation_links	14
Abb. 14	Custom-Modul publication_form	15
Abb. 15	Custom-Modul slider_tooltip	16
Abb. 16	Custom-Modul user_profile_elements_overview Beispiel	17
Abb. 17	Darstellung der Suchergebnisse Beispiel	18
Abb. 18	Darstellung der Suchergebnisse Konfiguration	18
Abb. 19	Konfiguration des Anzeigemodus „Search result highlighting input“	19
Abb. 20	Slider mit Wissenskarten als Startseite	20
Abb. 21	Konfigurationsmenü Apache Solr Search Contrib-Modul	21
Abb. 22	Falsche Position von Drupal Fehler- und Bestätigungsmeldungen	26

Abkürzungsverzeichnis

InDeKo Integratives demografiesensibles Kompetenzmanagement.

1 Installation

Dieses Kapitel beschreibt die Komplettinstallation und Konfiguration des InDeKo.Navi Atlas-Projekts. Auf die Installation und Konfiguration einzelner entwickelter Custom-Module wird in Abschnitt 4.1 eingegangen.

1.1 Installationsschritte

1. Vorausgesetzt werden mindestens ein Webserver der PHP unterstützt, PHP 5.2.5 sowie MySQL 5.0.15. Als Komplettpaket lässt sich z. B. XAMPP installieren (v5.6.12 empfohlen. In nachfolgenden Versionen wurde MySQL durch MariaDB ersetzt. Test der Installation verlief auch mit XAMPP 5.6.30 erfolgreich, aber entwickelt und getestet wurde ausschließlich mit MySQL).
2. GitHub-Projekt herunterladen (Branch KnowledgeMap) <https://github.com/KonstantinJanzen/Atlas/tree/KnowledgeMap/>. Das Projekt enthält eine komplett konfigurierte Drupal 7.53 Installation.
3. Inhalte des Atlas-Projekts in den Document Root (bzw. ein Unterverzeichnis, je nach gewünschter Konfiguration) des Webserverns entpacken (z. B. *htdocs*, *www*)
4. Drupal 7 Datenbank-Dump *db_wissenskarte.sql* aus dem Projekt-Unterverzeichnis *sql_dumps* importieren (hier beispielhaft für Datenbank Administrator „root“ ohne Passwort):
 - a) `mysqladmin -u root CREATE indeko`
 - b) `mysql -u root indeko < sql_dumps/db_wissenskarte.sql`
5. Apache Solr Server installieren:
 - a) Apache Solr Version 5.5.2 herunterladen und entpacken <http://archive.apache.org/dist/lucene/solr/5.5.2/>.
 - b) Aus dem Projekt-Unterverzeichnis *solr_config* den Ordner *drupal* nach *solr-5.5.2/server/solr/* kopieren.
 - c) Solr Server starten (z. B. `solr-5.5.2/bin/solr start`).
 - d) Der Apache Solr Server ist lokal unter <http://localhost:8983/solr> zu erreichen.
6. Apache Solr Suchindex in Drupal aktualisieren (*admin/config/search/apachesolr*):
 - a) „Delete the Search & Solr Index“ → „Delete index“
 - b) „Index all queued content“ → „Index all remaining“
7. Das Portal kann nun genutzt werden.
 - Administrator-Account „admin“ mit Passwort „pw“
 - Benutzer-Accounts „ruser“, „ruser2“, „ruser3“, „ruser4“, „ruser5“ mit Passwort „pw“

2 Projektstruktur

Dieses Kapitel beschreibt die grundsätzliche Struktur des Atlas-Projekts. Alle Änderungen im Vergleich zum InDeKo.Navi Ausgangsprojekt sind hervorgehoben und beschrieben. Abschnitt 2.1 beschreibt die Struktur des Drupal-Projekts und Abschnitt 2.2 geht auf die Struktur der entwickelten Custom-Module ein.

2.1 Grundsätzliche Drupal-Struktur

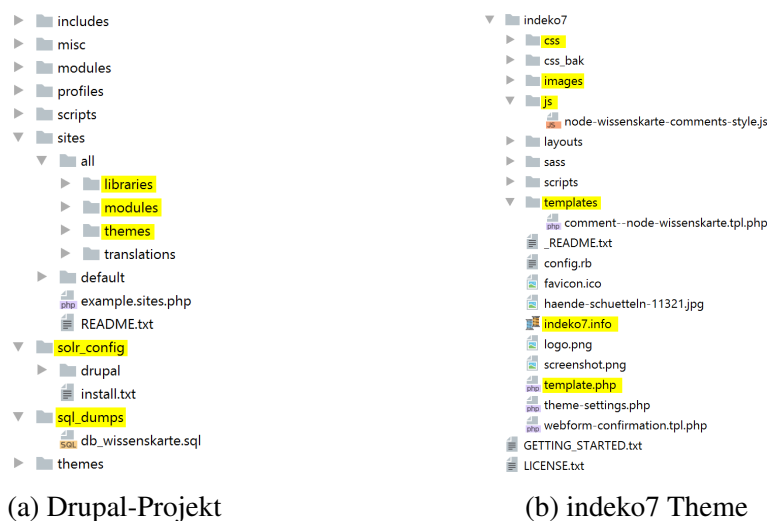


Abbildung 1: Grundstruktur des Drupal-Projekts

Abb. 1a stellt die grundsätzliche Struktur des Drupal-Projekts dar:

- *libraries*: Enthält zusätzlich installierte externe Javascript Bibliotheken (z. B. Chosen, qTip2).
- *modules*: Der Unterordner *custom* enthält alle entwickelten Module.
- *themes*: Enthält alle installierten Drupal-Themes. Für das Projekt relevant ist das adaptivetheme indeko7 (siehe Abb. 1b).
- *solr_config*: Enthält Konfigurationsdateien für den Apache Solr Server. *install.txt* beschreibt die Installation im Detail.
- *sql_dumps*: Enthält den Datenbank-Dump des InDeKo.Navi Atlas-Projekts (*db_wissenskarte.sql*).

Abb. 1b stellt die Hauptstruktur des indeko7 Themes dar */sites/all/themes/adaptivetheme/indeko7*:

- *css*: Enthält .css Dateien, die das grundlegende Design betreffen. Alle das Portal betreffenden Änderungen sind in der Datei *global.atlas.css* zusammengefasst.
- *images*: Der Unterordner *atlas* enthält Bilddateien, die speziell für das Atlas-Projekt entworfen wurden und in den .css Dateien verwendet werden (z. B. spezifische Icons für Inhaltstypen oder Aktionen).

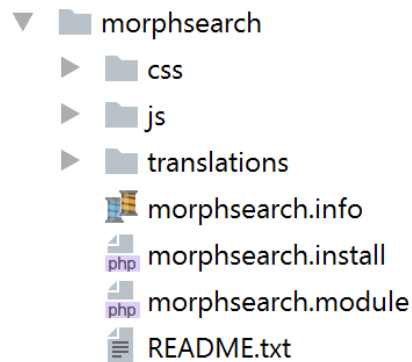


Abbildung 2: Grundstruktur Custom-Modul

- *js*: Enthält .js Dateien, die ausschließlich das Design betreffen.
- *templates*: Enthält Template-Dateien (.tpl.php), die gezielt die Darstellung der Inhalte des Portals steuern.
- *indeko7.info*: Enthält einen zusätzlichen Eintrag, um *global.atlas.css* einzubinden
(`stylesheets[screen][] = css/global.atlas.css`).
- *template.php*: Enthält Logik, die die von Drupal bereitgestellten Informationen zur Darstellung der Inhalte anpasst.

2.2 Grundsätzliche Modulstruktur

Dieser Abschnitt beschreibt die grundsätzliche Struktur der entwickelten Custom-Module (*/sites/all/modules/custom*):

- *css*: Enthält CSS Dateien, die speziell das Modul betreffen.
- *js*: Enthält JavaScript Dateien, die speziell das Modul betreffen.
- *translations*: Enthält eine Übersetzungs-Vorlage (.pot), die alle übersetzbaren Texte des Moduls enthält und eine deutsche Übersetzungsdatei (.de.po). Die Übersetzungen werden während der Installation des Moduls importiert oder können jederzeit manuell in Drupal importiert werden (*admin/config/regional/translate/import*). Templates erstellt mit dem Translation template extractor Modul (<https://www.drupal.org/project/potx>)
- *.info*: Enthält Informationen über das Modul, die von Drupal benötigt werden (Abhängigkeiten, Menüs, ...).
- *.install*: Enthält Installations- und Deinstallationsroutinen (Variablen, Datenbank-Tabellen und Einträge).
- *.module*: Enthält den PHP Quellcode des Moduls.

- *README.txt*: Enthält die Beschreibung des Moduls, die Installationsanleitung inklusive Pflicht- und optionaler Module sowie eine Beschreibung der Konfiguration (in Anlehnung an das Drupal README Template, siehe <https://www.drupal.org/node/2181737>).

3 Anmerkungen zum Produktivbetrieb

Dieser Abschnitt enthält Punkte, die bei der Überführung des InDeKo.Navi-Portals aus der Test-Umgebung in eine Live-Umgebung beachtet werden sollten.

- In der Entwicklung genutzte Benutzeraccounts deaktivieren, da die Passwörter unsicher sind (*admin/people*): „admin“, „ruser“, „ruser2“, „ruser3“, „ruser4“, „ruser5“.
- Beispiel-Wissenskarten löschen, da nicht geprüft wurde, ob die verwendeten Bilder lizenzfrei sind (*admin/content*, siehe Abschnitt 6 Punkt 3).
- Für den CSV-Export der Suchergebnisse, die aus dem Internet erreichbare URL des Apache Solr Servers eintragen (*admin/config/search/morphsearch_csv_export*).
- Anzuzeigende Fehlermeldungen deaktivieren (*admin/config/development/logging*).
- Überreste von Dummy-/Test-Daten und Benutzerprofilen löschen (*admin/content*).
- SMTP / E-Mail-Versand für Registrierung konfigurieren.

4 Funktionalität

Dieses Kapitel beschreibt die im Laufe des Projekts entwickelten Funktionalitäten. Abschnitt 4.1 beschäftigt sich mit den entstandenen Custom-Modulen. Abschnitt 4.2 beschreibt Funktionalitäten, die ausschließlich durch Drupal Contrib-Module oder Core-Module sowie Customizing in Drupal realisiert wurden.

4.1 Entwickelte Custom-Module

Wissenskarte					Suche				
Aktiviert	Name	Version	Beschreibung	Operationen	Aktiviert	Name	Version	Beschreibung	Operationen
<input checked="" type="checkbox"/>	form_wissenskarte.module		Allows the creation of knowledge maps. Beitrag: Morphosch (aktuell), Libraries (aktuell), System (aktuell), jQuery Update (aktuell), Apache Solr Framework (aktuell), Chosen (aktuell), Entity Reference (aktuell), Entity API (aktuell), Chosen tools (aktuell), Fomantic (aktuell), Yui UI (aktuell)	Hilfe	<input checked="" type="checkbox"/>	Atlas Apache Solr Search Block		Adds a block to sort Apache Solr search results and sets Atlas specific custom search parameters (ID, TTL, Boosting, Apache Solr Framework (aktuell))	Hilfe
<input checked="" type="checkbox"/>	Image Map Resizer		Makes HTML image maps responsive, so that they automatically stay scaled to the size of the image they are attached to. It detects the window being resized and updates the co-ordinates of the image map accordingly. Beitrag: Libraries (aktuell), System (aktuell)	Hilfe Konfigurieren	<input checked="" type="checkbox"/>	Atlas search result CSV export		Creates a link to export Apache Solr search results as CSV and adds an export link to the search results page. Beitrag: Apache Solr Framework (aktuell), Morphosch (aktuell), Libraries (aktuell), System (aktuell), jQuery Update (aktuell), Chosen (aktuell)	Hilfe Konfigurieren
<input checked="" type="checkbox"/>	jQuery Highlight (ATLAS VERSION)		jQuery Map Highlight adds javascript mouseover highlighting of hotspots to HTML image maps. EDIT: necessary libraries and settings only added to nodes of specified content type. Beitrag: Libraries (aktuell), System (aktuell)	Konfigurieren	<input checked="" type="checkbox"/>	Morphosch		Dynamically creates a search block in the right sidebar (fulltext search, dynamical content type search, morphological search by taxonomy terms, publication search). Beitrag: Libraries (aktuell), System (aktuell), jQuery Update (aktuell), Apache Solr Framework (aktuell), Chosen (aktuell) Beitrag von: form_wissenskarte.module (aktuell), Atlas search result CSV export (aktuell)	Hilfe Konfigurieren

(a) Module Wissenskarte

(b) Module Suche

Atlas				
Aktiviert	Name	Version	Beschreibung	Operationen
<input checked="" type="checkbox"/>	Comments Shortener		Allows trimmed comments to be displayed. New Comments will be displayed on top.	Hilfe
<input checked="" type="checkbox"/>	Dynamic node creation links		Dynamically creates node creation links (if the user has sufficient permissions) and adds them to the user menu block.	Hilfe Konfigurieren
<input checked="" type="checkbox"/>	Publication Form		Allows to make adjustments in the publication form (e.g. customized validation, mandatory fields). Beitrag: Atlas (aktuell), Taxonomy (aktuell), System (aktuell), Field (aktuell), Field UI (aktuell), Image (aktuell)	Hilfe
<input checked="" type="checkbox"/>	Slider Tooltip		Adds an image title to the tooltip and displays it on the slider.	Hilfe
<input checked="" type="checkbox"/>	User Profile Elements Overview		Allows created elements of the user to be displayed in the user profile.	Hilfe

(c) Module Atlas

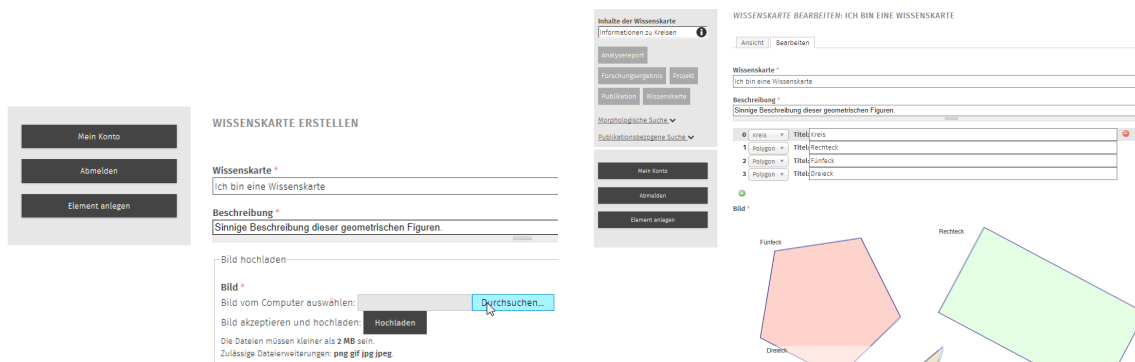
Abbildung 3: Entwickelte Custom-Module

Dieser Abschnitt beschreibt die im Laufe des Projekts entwickelten Drupal Custom-Module. Diese Module wurden in die Bereiche Wissenskarte (siehe Abb. 3a), Suche (siehe Abb. 3b) sowie Atlas für allgemeine Module (siehe Abb. 3c) unterteilt. Für jedes Modul wird der Funktionsumfang kurz beschrieben sowie auf die Punkte *Vorausgesetzte Module*, *Empfohlene Module*, *Installation* und *Konfiguration* eingegangen. *Vorausgesetzte Module* listet alle Module, die zusätzlich zum entwickelten Modul installiert sein müssen, auf. Diese Informationen sind auch in den README.txt und den [Modul].info Dateien hinterlegt (somit wird man bei der Installation auch von Drupal auf fehlende vorausgesetzte Module hingewiesen). Der Punkt *Empfohlene Module* listet Module auf, die nicht zwingend benötigt werden, aber zum Nutzen des entwickelten Moduls beitragen. Unter *Installation* werden alle nötigen Installationsschritte aufgeführt und *Konfiguration* erläutert mögliche Einstellungen am Modul bzw. das Konfigurationsmenü. Alle Texte der Module sind englisch. Jedem Modul liegen im Unterordner *translations* jeweils ein Übersetzungstemplate (mit allen übersetzbaren Textstellen) sowie eine deutsche Übersetzung bei. Diese ermöglichen bei aktiviertem Drupal Core-Modul *locale* die Konfiguration der Sprache (*admin/config/regional/translate*, importieren oder Textstellen manuell übersetzen).

Auf Implementierungsdetails wird bei der Beschreibung der Custom-Module nicht eingegangen. Diese sind ausführlich in den Quellcodedateien kommentiert.

4.1.1 form_wissenskarte_module

Dieses Modul ermöglicht es Benutzern Verweissensitive Grafiken (engl. image maps) aus Bilddateien (.jpg, .png) zu erstellen. Hierzu wird bei der Installation ein neuer Inhaltstyp „Wissenskarte“ erstellt. Dies ermöglicht es Bilder auf das Drupal-Portal zu übertragen und auf diesen geometrische Bereiche einzuzeichnen. Diese Bereiche können mit Informationen aus dem Drupal-Portal angereichert und anderen Benutzern zugänglich gemacht werden.



(a) Beispiel Wissenskarte hochladen

(b) Beispiel Wissenskarte bearbeiten

Abbildung 4: Custom-Modul form_wissenskarte_module

- Vorausgesetzte Module:

- morphsearch (siehe Abschnitt 4.1.5): Wird genutzt, um die Inhalte der gezeichneten Bereiche auszuwählen.

- Empfohlene Module:

- Entityreference (<https://www.drupal.org/project/entityreference>): Ist dieses Modul installiert, werden alle Informationen, die den einzelnen Bereichen der Wissenskarte zugeordnet sind, auch mit der gesamten Wissenskarte verknüpft. Ansonsten lassen sich Wissenskarten nur über Titel und Beschreibung per Suche finden.
- Fivestar (<https://www.drupal.org/project/fivestar>): Ermöglicht es Benutzern Wissenskarten zu bewerten.

- Installation: Dieses Modul lässt sich wie jedes andere Drupal Modul installieren.

1. Modul in das Custom-Module-Verzeichnis der Drupal-Installation kopieren (z. B. *sites/all/modules/custom*, vgl. Abb. 1a).
2. Modul auf der Seite *admin/modules* aktivieren.

- Konfiguration:

Dieses Modul verfügt über kein Konfigurationsmenü oder veränderbare Einstellungen (deutsche Übersetzungen sowie das Übersetzungstemplate liegen bei).

Alle nötigen Einstellungen werden automatisch bei der Installation durchgeführt (.install Datei).

Dieses Modul identifiziert Wissenskarten anhand, der Bilddatei zugeordneter, Bildstile (z. B. CSS-Klasse „image-style-wissenskarte“). Das verwendete Theme muss diese CSS-Klassen im HTML zur Verfügung stellen (beim Drupal 7 Standard Theme ist dies nicht der Fall).

4.1.2 comments_shortener

Dieses Modul passt das Standard-Verhalten der Drupal Kommentarsektion an:

1. Kommentare werden auf maximal 50 Zeichen gekürzt. Gekürzte Kommentare werden durch Auslassungspunkte am Ende kenntlich gemacht (siehe Abb. 5a). Ein Klick auf die Auslassungspunkte zeigt den vollständigen Kommentar an (siehe Abb. 5b).
2. Neueste Kommentare werden zuerst angezeigt (dies ließ sich nicht über die Drupal Core-Einstellungen konfigurieren).
3. Nachdem ein Kommentar abgegeben wurde, wird der Benutzer auf die erste Kommentarseite zurückgeleitet (dies ließ sich nicht über Drupal Core-Einstellungen konfigurieren).



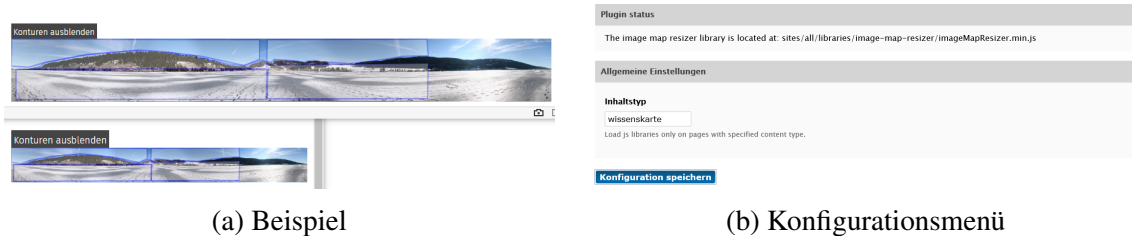
Abbildung 5: Custom-Modul comments_shortener

- Vorausgesetzte Module:
 - keine
- Empfohlene Module:
 - form_wissenskarte_module (siehe Abschnitt 4.1.1): Das comments_shortener Modul ist auf die Anforderungen an Wissenskarten zugeschnitten, funktioniert aber auch ohne dieses Modul für jede Art von Kommentar.
- Installation: Dieses Modul lässt sich wie jedes andere Drupal Modul installieren.
 1. Modul in das Custom-Module-Verzeichnis der Drupal-Installation kopieren (z. B. *sites/all/modules/custom*, vgl. Abb. 1a).
 2. Modul auf der Seite *admin/modules* aktivieren.
- Konfiguration:

Dieses Modul verfügt über kein Konfigurationsmenü oder veränderbare Einstellungen (deutsche Übersetzungen sowie das Übersetzungstemplate liegen bei).

4.1.3 imagemap_resizer

Dieses Modul sorgt dafür, dass *image maps responsive* werden. Die gezeichneten Konturen werden automatisch an die aktuelle Darstellungsgröße der *image map* angepasst (siehe Abb. 6a).



(a) Beispiel

(b) Konfigurationsmenü

Abbildung 6: Custom-Modul imagemap_resizer

- Vorausgesetzte Module:
 - Libraries (<https://www.drupal.org/project/libraries>)
- Empfohlene Module:
 - form_wissenskarte_module (siehe Abschnitt 4.1.1): Legt den Inhaltstyp *wissenskarte* an und ermöglicht es *image maps* selbst zu erzeugen.
- Installation:
 1. Modul in das Custom-Module-Verzeichnis der Drupal-Installation kopieren (z. B. *sites/all/modules/custom*, vgl. Abb. 1a)
 2. Image Map Resize plugin herunterladen <https://github.com/davidjbradshaw/image-map-resizer>
 3. *imageMapResizer.min.js* Datei in den Ordner *sites/all/libraries/image-map-resizer* entpacken.
 4. Modul auf der Seite *admin/modules* aktivieren.
- Konfiguration:

Abb. 6b stellt das Konfigurationsmenü dar. Im ersten Abschnitt lässt sich der Status der externen JavaScript Bibliothek ablesen. In dem darauffolgenden Textfeld wird der Maschinennamen des Inhaltstyps angegeben, der image maps enthält. Die für die Skalierung von image maps benötigten JavaScript Dateien werden nur auf Seiten des angegebenen Inhaltstyps geladen. Als Standardwert ist der von dem Custom-Modul form_wissenskarte_module (Abschnitt 4.1.1) erstellte Inhaltstyp *wissenskarte* definiert.

4.1.4 jq_maphilight

Dieses Modul ermöglicht es Bereiche von image maps hervorzuheben (siehe Abb. 7c).

(a) Allgemeine Einstellungen

(b) Styling Einstellungen



(c) Beispiel

Abbildung 7: Custom-Modul jq_maphilight

- Vorausgesetzte Module:
 - Libraries (<https://www.drupal.org/project/libraries>)
- Empfohlene Module:
 - form_wissenskarte_module (siehe Abschnitt 4.1.1): Legt den Inhaltstyp *wissenskarte* an und ermöglicht es *image maps* selbst zu erzeugen.
- Installation:
 1. Modul in das Custom-Module-Verzeichnis der Drupal-Installation kopieren (z. B. *sites/all/modules/custom*, vgl. Abb. 1a).
 2. jQuery Map Hilight Plugin herunterladen <https://github.com/kemayo/maphilight>
 3. *jquery.maphilight.min.js* Datei in den Ordner *sites/all/libraries/jquery.maphilight* entpacken.
 4. Modul auf der Seite *admin/modules* aktivieren.

- Konfiguration:

Abb. 7a stellt das Konfigurationsmenü dar. Im ersten Abschnitt lässt sich der Status der externen JavaScript Bibliothek ablesen. In dem darauffolgenden Textfeld wird der Maschinennamen des Inhaltstyps angegeben, der *image maps* enthält. Die für das Hervorheben von Bereichen auf *image maps* benötigten JavaScript Dateien werden nur auf Seiten des angegebenen Inhaltstyps geladen. Als Standardwert ist der von dem Custom-Modul form_wissenskarte_module (Abschnitt 4.1.1) erstellte Inhaltstyp *wissenskarte* definiert. Der letzte Abschnitt enthält allgemeine Einstellungen zu Rahmen und Füllung von Bereichen.

Abb. 7b stellt die Designereinstellungen des Konfigurationsmenüs dar. Dies umfasst mehrere Einstellungen zu Rahmen und Füllung für die Darstellung beim Endanwender (siehe Abb. 7c).

4.1.5 morphsearch

Dieses Modul stellt einen Block zur Apache Solr basierten Suche mit folgender Funktionalität zu Verfügung:

1. Volltextsuche mit individueller Syntax (siehe Abb. 8b)
2. Suche nach einzelnen Inhaltstypen (siehe Abb. 8b)
3. Morphologische Suche nach zuvor definierten Taxonomie-Begriffen (siehe Abb. 10c)
4. Publikationsbezogene Suche zur Filterung von Publikationen (siehe Abb. 9)
5. Speichern sowie Löschen von Suchanfragen (siehe Abb. 8c)



Abbildung 8: Custom-Modul morphsearch

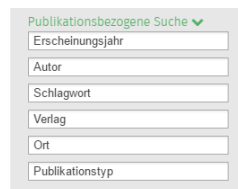


Abbildung 9: Custom-Modul morphsearch Beispiel publikationsbezogene Suche

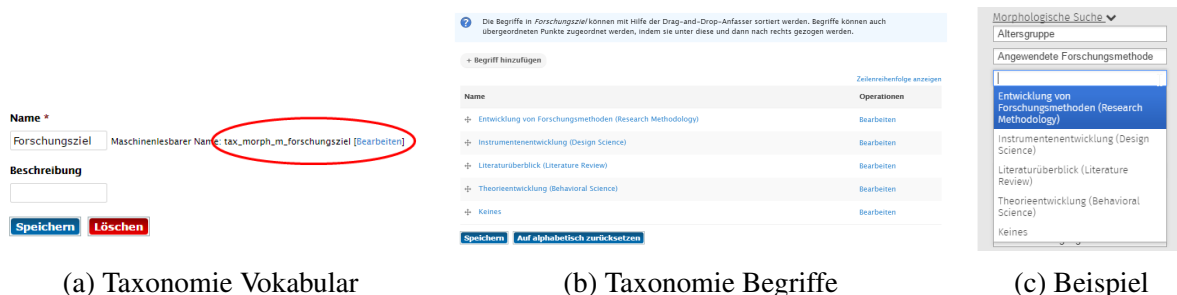
- Vorausgesetzte Module:
 - Libraries (<https://www.drupal.org/project/libraries>)
 - jQuery Update (https://www.drupal.org/project/jquery_update)
 - Apache Solr Search (<https://www.drupal.org/project/apachesolr>)
 - Chosen (<https://www.drupal.org/project/chosen>): Funktioniert theoretisch auch ohne, aber das (Interaktions-)Design ist auf Chosen Auswahllisten abgestimmt.
 - Font Awesome Icons (<https://www.drupal.org/project/fontawesome>): Nur für die Icons zum Ausklappen der *morphologischen / publikationsbezogenen Suche*.

- Empfohlene Module:

- Bibliography Module (<https://www.drupal.org/project/biblio>): Wenn *biblio* installiert ist, wird der Suchblock um die *publikationsbezogene Suche* erweitert.
- Taxonomy (Drupal Core-Modul): Stellt den Kern der morphologischen Suche dar.

- Installation:

1. Modul in das Custom-Module-Verzeichnis der Drupal-Installation kopieren (z. B. *sites/all/modules/custom*, vgl. Abb. 1a).
2. qTip2 jQuery Plugin herunterladen (<http://qtip2.com/download>)
3. *jquery.qtip.min.js* und *jquery.qtip.min.css* in den Ordner *sites/all/libraries/qtip* entpacken.
4. Modul auf der Seite *admin/modules* aktivieren.
5. Modul konfigurieren.



(a) Taxonomie Vokabular

(b) Taxonomie Begriffe

(c) Beispiel

Abbildung 10: Custom-Modul morphsearch Konfiguration morphologische Suche

- Konfiguration:

Abb. 8a stellt das Konfigurationsmenü dar (*admin/config/search/morphsearch*). Dort lässt sich einstellen, welche Inhaltstypen zur Filterung der Suchergebnisse in dem Block zur Auswahl stehen sollen (siehe Abb. 8b).

Die *publikationsbezogene Suche* (siehe Abb. 9) steht nur zur Verfügung, wenn das *Bibliography Module* installiert und aktiviert ist. Eine Anpassung, der zur Verfügung gestellten Filter (Erscheinungsjahr, Autor, ...), ist nicht vorgesehen und lässt sich somit nur durch Änderungen im Quellcode erreichen.

Die *morphologisch Suche* (siehe Abb. 10c) steht nur zur Verfügung, wenn das *Taxonomy Core-Modul* aktiviert ist. Für jede Dimension, die in der *morphologischen Suche* angezeigt werden soll, muss ein Vokabular angelegt werden (*admin/structure/taxonomy*). Hierbei ist darauf zu achten, dass ein Vokabular nur in die morphologische Suche übernommen wird, wenn der Maschinenname mit

- *tax_morph_* für Einfachauswahl
- *tax_morph_m_* für Mehrfachauswahl

beginnt (siehe Abb. 10a). Nachdem ein Vokabular angelegt wurde, können Begriffe hinzugefügt werden. Diese stehen dann zur Einfach-/Mehrfachauswahl bei der *publikationsbezogenen Suche* zur Verfügung.

4.1.6 morphsearch_csv_export

Dieses Modul fügt der Apache Solr Suchergebnisseite einen Link hinzu, der es ermöglicht alle Suchergebnisse als .csv Datei herunterzuladen (siehe Abb. 11b).

Solr server URL for network access *

Specify the Solr Server URL anyone can access from an outside network including your Solr core used for csv export (include http:// and trailing /, example: http://mysite/solr/mySolrCore/). Default value for local development: http://localhost:8983/solr/drupal/. Live test server environment: http://147.172.96.42/solr/drupal/

Konfiguration speichern

(a) Konfigurationsmenü

Suchergebnisse

InDeKo.Navi Anwendungsbeispiel

ruser

Kompetenzmanagement hat die Aufgabe Mitarbeiter-

Ergebnisse exportieren

(b) Beispiel

Abbildung 11: Custom-Modul morphsearch_csv_export

- Vorausgesetzte Module:
 - Apache Solr Search (<https://www.drupal.org/project/apachesolr>)
 - Font Awesome Icons (<https://www.drupal.org/project/fontawesome>): Nur für das csv Download Icon (siehe Abb. 11b) benötigt.
- Empfohlene Module:
 - morphsearch (siehe Abschnitt 4.1.5): Ermöglicht es die zu exportierenden Suchergebnisse einzuschränken.
- Installation: Dieses Modul lässt sich wie jedes andere Drupal Modul installieren.
 1. Modul in das Custom-Module-Verzeichnis der Drupal-Installation kopieren (z. B. *sites/all/modules/custom*, vgl. Abb. 1a).
 2. Modul auf der Seite *admin/modules* aktivieren.
- Konfiguration:

Abb. 11a stellt das Konfigurationsmenü dar (*admin/config/search/morphsearch_csv_export*). Die Suchergebnisse werden direkt vom Apache Solr-Server als CSV-Datei übertragen, daher muss die Adresse des Solr-Servers eingetragen werden. Die Adresse für die lokale Entwicklung wird automatisch während der Installation gesetzt. Für den Live-Betrieb muss die aus dem Internet erreichbare Adresse angegeben werden.

4.1.7 morphsearch_sort

Dieses Modul fügt der Drupal-Installation einen Block hinzu, der es erlaubt Apache Solr Suchergebnisse zu sortieren. Es lässt sich nach Relevanz sowie Datum und Titel (jeweils auf- und absteigend) sortieren (siehe Abb. 12).

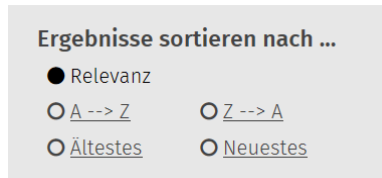


Abbildung 12: Custom-Modul morphsearch_sort Beispiel

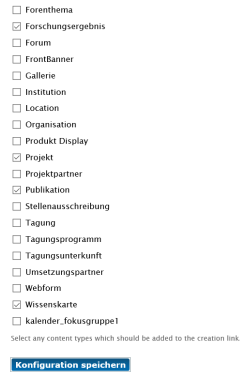
- Vorausgesetzte Module:
 - Apache Solr Search (<https://www.drupal.org/project/apachesolr>)
 - Font Awesome Icons (<https://www.drupal.org/project/fontawesome>): Nur für die Radio-Button Icons (siehe Abb. 12) benötigt.
- Empfohlene Module:
 - morphsearch (siehe Abschnitt 4.1.5): Ermöglicht es die zu sortierenden Suchergebnisse einzuschränken.
- Installation: Dieses Modul lässt sich wie jedes andere Drupal Modul installieren.
 1. Modul in das Custom-Module-Verzeichnis der Drupal-Installation kopieren (z. B. *sites/all/modules/custom*, vgl. Abb. 1a).
 2. Modul auf der Seite *admin/modules* aktivieren.
 3. Den *Apache Solr Atlas Sort Block* einer sichtbaren Region zuordnen (*admin/structure/block*).
- Konfiguration:

Dieses Modul verfügt über kein Konfigurationsmenü oder veränderbare Einstellungen (deutsche Übersetzungen sowie das Übersetzungstemplate liegen bei).

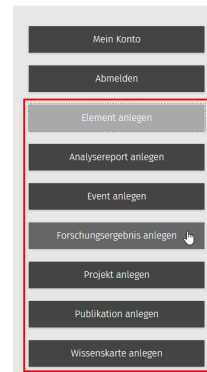
Das Aussehen des Sortierblocks wird per CSS im Theme definiert (z. B. *global.atlas.css*, siehe Abschnitt 2.1).

4.1.8 node_creation_links

Dieses Modul fügt dem *user menu* Block Links zum Anlegen neuer Inhalte hinzu. Der Eintrag *Element anlegen* ist dabei immer sichtbar und kann ausgeklappt werden (siehe Abb. 13b).



(a) Konfigurationsmenü



(b) Beispiel

Abbildung 13: Custom-Modul node_creation_links

- Vorausgesetzte Module:
 - keine
- Empfohlene Module:
 - keine
- Installation: Dieses Modul lässt sich wie jedes andere Drupal Modul installieren.
 1. Modul in das Custom-Module-Verzeichnis der Drupal-Installation kopieren (z. B. *sites/all/modules/custom*, vgl. Abb. 1a).
 2. Modul auf der Seite *admin/modules* aktivieren.
 3. Modul konfigurieren.
- Konfiguration:

Abb. 13a stellt das Konfigurationsmenü dar (*admin/config/content/node_creation_links*). Es werden alle im System definierten Inhaltstypen aufgelistet. Solange keine Inhaltstypen ausgewählt sind, ist auch der Eintrag *Element anlegen* nicht sichtbar. Ausgewählte Inhaltstypen stehen einem eingeloggten Benutzer später potentiell in dem Block unter *Element anlegen* zur Verfügung. Die in Abb. 13a dargestellten Links zum Erstellen von Inhaltstypen werden individuell pro Benutzer erzeugt (Links zu Inhaltstypen, die der aktuell eingeloggte Benutzer nicht erstellen darf, werden auch nicht erzeugt).

Das Aussehen des *Element anlegen* Blocks wird per CSS im Theme definiert (z. B. *global.atlas.css*, siehe Abschnitt 2.1).

Dieses Modul ergänzt ausschließlich den Block *user menu*. Dieser sollte daher einer sichtbaren Region zugeordnet sein (*Struktur* → *Blöcke*).

4.1.9 publication_form

Dieses Modul ergänzt fehlende Funktionalitäten des *Bibliography* Modules bei der Erstellung und Bearbeitung von Publikationen (siehe Abb. 14):

1. Validierung des Datum-Felds
2. Validierung der Autoren-Felder
3. Hinzufügen fehlender Markierungen von Pflichtfeldern

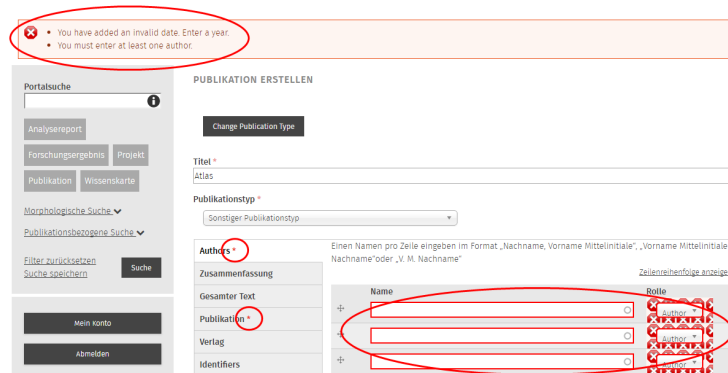


Abbildung 14: Custom-Modul publication_form

- Vorausgesetzte Module:
 - Bibliography Module (<https://www.drupal.org/project/biblio>)
- Empfohlene Module:
 - keine
- Installation: Dieses Modul lässt sich wie jedes andere Drupal Modul installieren.
 1. Modul in das Custom-Module-Verzeichnis der Drupal-Installation kopieren (z. B. *sites/all/modules/custom*, vgl. Abb. 1a).
 2. Modul auf der Seite *admin/modules* aktivieren.
- Konfiguration:

Dieses Modul verfügt über kein Konfigurationsmenü oder veränderbare Einstellungen (deutsche Übersetzungen sowie das Übersetzungstemplate liegen bei).

4.1.10 slider_tooltip

Dieses Modul fügt dem Wissenskarten Slider-Karussell (siehe Abschnitt 4.2.2) die einzelnen Bildtitel als Tooltip hinzu (siehe Abb. 15).



Abbildung 15: Custom-Modul slider_tooltip

- Vorausgesetzte Module:
 - keine
- Empfohlene Module:
 - keine
- Installation: Dieses Modul lässt sich wie jedes andere Drupal Modul installieren.
 1. Modul in das Custom-Module-Verzeichnis der Drupal-Installation kopieren (z. B. *sites/all/modules/custom*, vgl. Abb. 1a).
 2. Modul auf der Seite *admin/modules* aktivieren.
- Konfiguration:

Dieses Modul verfügt über kein Konfigurationsmenü oder veränderbare Einstellungen.

4.1.11 user_profile_elements_overview

Dieses Modul listet einem registrierten Benutzer in seinem Profil seine selbst erstellten Publikationen, Wissenskarten, Projekte, Analysereports und Forschungsergebnisse auf (siehe Abb. 16). Jedes Element verfügt über ein individuelles Icon sowie Tooltip.

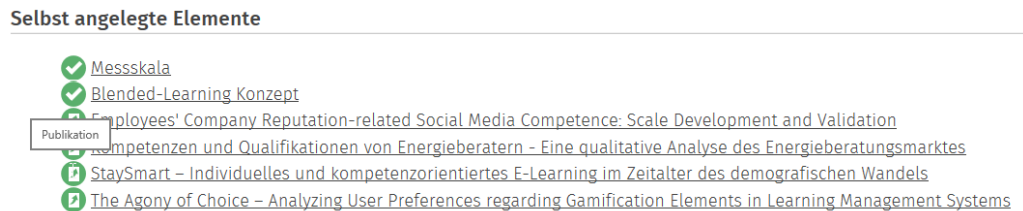


Abbildung 16: Custom-Modul user_profile_elements_overview Beispiel

- Vorausgesetzte Module:
 - keine
- Empfohlene Module:
 - Font Awesome Icons (<https://www.drupal.org/project/fontawesome>): Nur für das Standard-Icon benötigt, falls keine individuellen Icons festgelegt wurden.
- Installation: Dieses Modul lässt sich wie jedes andere Drupal Modul installieren.
 1. Modul in das Custom-Module-Verzeichnis der Drupal-Installation kopieren (z. B. *sites/all/modules/custom*, vgl. Abb. 1a).
 2. Modul auf der Seite *admin/modules* aktivieren.
- Konfiguration:

Dieses Modul verfügt über kein Konfigurationsmenü oder veränderbare Einstellungen (deutsche Übersetzungen sowie das Übersetzungstemplate liegen bei).

Sollen mehr als die fünf definierten Inhaltstypen (Publikationen, Wissenskarten, Projekte, Analysereports und Forschungsergebnisse) im Benutzerprofil angezeigt werden, muss im Quellcode Zeile 20 der Funktionsaufruf `elements_overview_html(array('biblio', 'wissenskarte', ...))` ergänzt werden.

4.2 Drupal Customizing

Dieser Abschnitt beschreibt Funktionalitäten, die ausschließlich durch Drupal Contrib-Module oder Core-Module sowie Customizing in Drupal realisiert wurden.

4.2.1 Darstellung der Suchergebnisse

Im Folgenden wird beschrieben, wie die Darstellung der Suchergebnisse umgesetzt wurde. Dies ermöglicht eine schnelle Anpassung neuer Inhaltstypen an das aktuelle Design der Suchergebnisse.

Vorausgesetzte Module:

- Display Suite (<https://www.drupal.org/project/ds>) mit aktivierten Sub-Modulen Display Suite UI, Display Suite Search und Display Suite Format

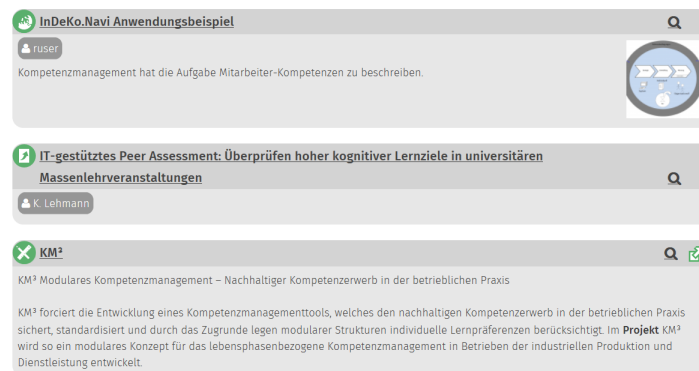


Abbildung 17: Darstellung der Suchergebnisse Beispiel

Die Darstellung der Suchergebnisse (siehe Abb. 17) erfolgt über den Anzeigemodus „Search result highlighting input“. Dieser muss in den Einstellungen für den Inhaltstyp aktiviert werden (*Struktur* → *Inhaltstypen* → *Anzeige verwalten*). Dies geschieht, wie in Abb. 18 dargestellt, unter *Benutzerdefinierte Anzeigeeinstellungen*. Nachdem der Inhaltstyp gespeichert wurde, wechselt man in den neuen Anzeigemodus „Search result highlighting input“.

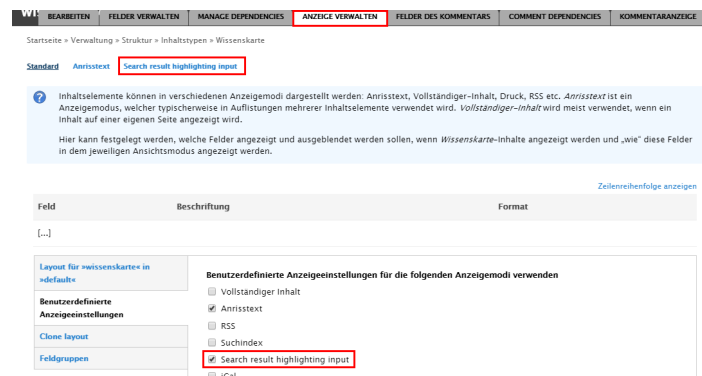


Abbildung 18: Darstellung der Suchergebnisse Konfiguration

Im Tab Layout wählt man *Fluid two column stacked* (siehe Abb. 19a). Nachdem im Tab *Custom classes* dem Layout die CSS-Klasse „search-result“ zugeordnet wurde (siehe Abb. 19b), werden die Änderungen gespeichert. Die gewünschten Felder können nun, wie in Drupal gewohnt, konfiguriert und den einzelnen Regionen des Layouts zugeordnet werden. Abb. 19a zeigt exemplarisch die Konfiguration des Inhaltstyps *Wissenskarte* (der Titel von Suchergebnissen wurde z. B. mit *h3* Wrapper und CSS-Klasse *title* konfiguriert).

The image consists of two side-by-side screenshots of the Drupal configuration interface for the 'Wissenskarte' content type.

(a) Layout: This screenshot shows the 'Layout' tab. At the top, there are tabs for 'BEARBEITEN', 'FELDER VERWALTEN', 'MANAGE DEPENDENCIES', 'ANZEIGE VERWALTEN', 'FELDER DES KOMMENTARS', 'COMMENT DEPENDENCIES', and 'KOMMENTARANZEIGE'. The 'ANZEIGE VERWALTEN' tab is active. Below the tabs, there is a breadcrumb trail: 'Startseite > Verwaltung > Struktur > Inhaltstypen > Wissenskarte > Anzeige verwalten'. The 'Standard' tab is selected. A table lists fields and their configurations:

Feld	Region	Beschriftung	Format
Kopfbereich			
Content Type Icon	Kopfbereich	- Ausgeblendet -	Standard
Titel	Kopfbereich	- Ausgeblendet -	Standard
Links			
Autor	Links	- Ausgeblendet -	Autor verlinkt zum Profil
Beschreibung	Links	- Ausgeblendet -	Getrimmt
Rechts			
Bild	Rechts	- Ausgeblendet -	Bildstil: Media thumbnail (100x100)
Fußbereich			

Below the table, there is a section for 'Layout für »wissenskarte« in »search result«'. It shows 'Current layout' as 'Fluid two column stacked'. A list of template suggestions is provided, including 'ds-2col-stacked-fluid--node.tpl.php', 'ds-2col-stacked-fluid--node-search-result.tpl.php', 'ds-2col-stacked-fluid--node-wissenskarte.tpl.php', and 'ds-2col-stacked-fluid--node-wissenskarte-search-result.tpl.php'. A dropdown menu 'Wählen Sie ein Layout' is set to 'Fluid two column stacked'.

(b) Custom classes: This screenshot shows the 'Custom classes' tab. It displays the 'Layout für »wissenskarte« in »search result«' and lists 'Custom wrappers', 'Custom classes', 'Benutzerdefinierte Felder', and 'Feldgruppen'. The 'Class for layout' is set to 'Keine search-result'. The 'Klasse für header' is set to 'Keine search-result'. The 'Klasse für left' is set to 'Keine search-result'. The 'Klasse für right' is set to 'Keine search-result'. The 'Klasse für footer' is set to 'Keine search-result'.

(a) Layout

(b) Custom classes

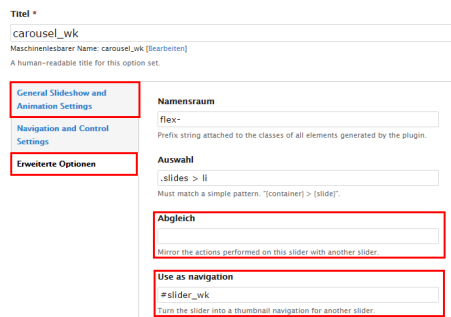
Abbildung 19: Konfiguration des Anzeigemodus „Search result highlighting input“

Display Suite ermöglicht es, zusätzlich zu den Feldern des Inhaltstyps, benutzerdefinierte Felder zu definieren (*admin/structure/ds/fields*). Für die Darstellung der Suchergebnisse wurde z. B. das benutzerdefinierte Feld *Content Type Icon* hinzugefügt. Dieses Feld fügt den Suchergebnissen ein html-Element hinzu, um in den Suchergebnissen individuelle Inhhaltstypen-Icons und Tooltips anzeigen zu können.

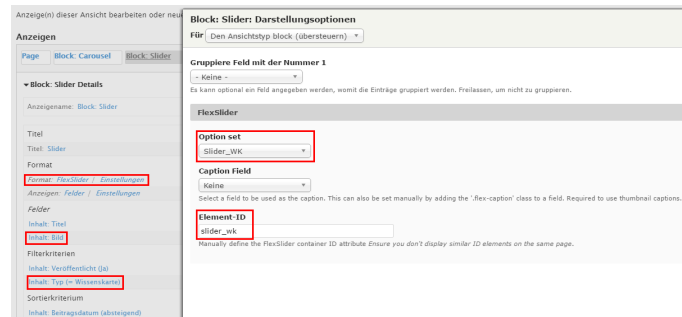
4.2.2 Slider mit Wissenskarten als Startseite

Vorausgesetzte Module:

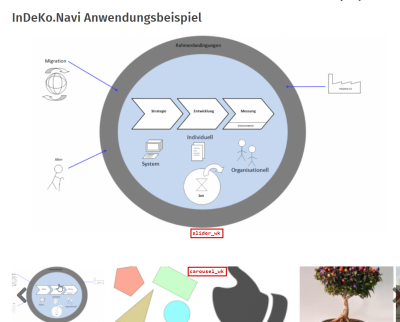
- Flex Slider (<https://www.drupal.org/project/flexslider>)
- Views (<https://www.drupal.org/project/views>)



(a) Konfiguration Flex Slider



(b) Konfiguration View



(c) Beispiel

Abbildung 20: Slider mit Wissenskarten als Startseite

Im Folgenden werden die grundlegenden Schritte beschrieben, die nötig sind um eine Startseite mit zwei zusammenhängen Bild-Slidern (Slider mit Karussell) wie in Abb. 20c anzulegen:

1. Unter den Flex Slider Einstellungen ein neues *option set* hinzufügen (*admin/config/media/flexslider/add*).
2. Als *Titel* „carousel_wk“ eintragen und bei *Animation* „Slide“ auswählen.
3. Im Tab *Erweiterte Optionen* unter der Option *Use as navigation* „#slider_wk“ eintragen (siehe Abb. 20a) und das *option set* speichern.
4. Ein zweites *option set* mit dem *Titel* „slider_wk“ erstellen und im Tab *Erweiterte Optionen* unter *Abgleich* „#carousel_wk“ eintragen.
5. Einen neuen View mit zwei Blöcken („slider“ und „carousel“) anlegen und nach Typ „Wissenskarte“ filtern (siehe Abb. 20b).
6. Für den Slider-Block *Format* auf „FlexSlider“ ändern, in den Einstellungen das zuvor erstellte „slider_wk“ *option set* auswählen und als *Element-ID* „#slider_wk“ setzen.

7. Für den Slider-Block ein Feld vom „Inhalt: Bild“ mit der Einstellung „Bilddarstellung: flexslider_full“ hinzufügen.
8. Für den Carousel-Block *Format* auf „FlexSlider“ ändern, in den Einstellungen das zuvor erstellte „carousel_wk“ *option set* auswählen und als *Element-ID* „#carousel_wk“ setzen.
9. Für den Carousel-Block ein Feld vom „Inhalt: Bild“ mit der Einstellung „Bilddarstellung: flexslider_thumbnail“ hinzufügen.
10. Beide Blöcke der Inhaltsregion zuordnen und nur auf der Seite „<front>“ anzeigen lassen.

4.2.3 Apache Solr Search

Dieser Abschnitt beschreibt wie sich zusätzliche Inhaltstypen in den Apache Solr Suchindex aufnehmen lassen. Nach den Anforderungen an das Atlas-Projekt soll nur nach den fünf Inhaltstypen Analysereport, Forschungsergebnis, Projekt, Publikation und Wissenskarte gesucht werden können.

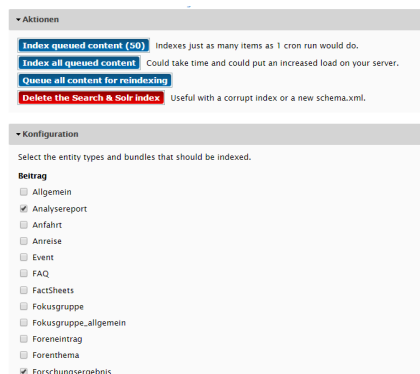


Abbildung 21: Konfigurationsmenü Apache Solr Search Contrib-Modul

Vorausgesetzte Module:

- Apache Solr Search (<https://www.drupal.org/project/apachesolr>)

Weitere Inhaltstypen lassen sich unter *admin/config/search/apachesolr* hinzufügen (siehe Abb. 21):

1. Unter „Konfiguration“ alle Inhaltstypen auswählen, die in den Suchindex aufgenommen und als Suchergebnisse angezeigt werden sollen
2. Änderungen „Speichern“
3. „Queue all content for reindexing“ → „Queue all content“
4. „Index all queued content“ → „Index all remaining“ (oder warten bis die neuen Inhalte nach und nach im Rahmen der automatischen Wartungsläufe hinzugefügt werden)

4.2.4 Inhaltstypen

Auf die im Laufe des Projekts angelegten bzw. erweiterten Inhaltstypen (Forschungsergebnis, Projekt, Umsetzungspartner, Institution, ...) wird an dieser Stelle nicht eingegangen. Die Strukturen der Inhaltstypen lassen sich direkt im System (*Administrationsmenü* → *Struktur* → *Inhaltstypen*), oder aus den entsprechenden Anforderungsdokumenten, ablesen. Zum Exportieren aller Einstellungen, um sie auf ein anderes System zu übertragen, bietet sich z. B. das Features Modul (<https://www.drupal.org/project/features>) an.

4.2.5 Design

Alle Änderungen am Design des Portals durch das Atlas-Projekt sind in der Datei *global.atlas.css* im indeko7 Theme dokumentiert (*sites/all/themes/adaptivetheme/indeko7/css*).

5 Fehler und deren Lösungen

- *ERROR 2006 (HY000) at line [...]: MySQL server has gone away*
MySQL Konfiguration (my.ini | XAMPP -> MySQL -> Config) **max_allowed_packet** erhöhen (z. B. auf 32M).
- *Fatal error: Allowed memory size of [...] bytes exhausted (tried to allocate [...] bytes) in [...]*
PHP Konfiguration (php.ini | XAMPP -> Apache -> Config) **memory_limit** hochsetzen .
- *PDOException: SQLSTATE[HY000] [2002] Es konnte keine Verbindung hergestellt werden, da der Zielcomputer die Verbindung verweigerte. in lock_may_be_available()*
MySQL Server nicht gestartet oder abgestürzt.
- *PDOException: SQLSTATE[42000]: Syntax error or access violation: 1118 The size of BLOB/TEXT data inserted in one transaction is greater than 10% of redo log size. Increase the redo log size using innodb_log_file_size.*
MySQL Konfiguration (my.ini | XAMPP -> MySQL -> Config) **innodb_log_file_size** erhöhen.
- *Warning: file_get_contents(): http:// wrapper disabled in the server configuration by allow_url_fopen = 0 in _local_parse_js_file()*
Drupal Konfiguration zur settings.php **ini_set('allow_url_fopen', 1);** hinzufügen.
- *User Warning: The following module is missing from the file system: [...]*
Fehler wurde nach Upgrade auf die neuere Drupal Version angezeigt, siehe <https://www.drupal.org/node/2487215>. Betroffene Module neu zu installieren / deinstallieren oder extra Module, die den Fehler beheben sollten halfen nicht. Nur manueller Eingriff in die Datenbank führte zum Erfolg:

```
"DELETE from system where name = 'old_module1' AND type = 'module';"
```

6 Verwendete Ressourcen und deren Lizenzen

Dieses Kapitel listet alle Ressourcen und deren Lizenzen auf, die dem InDeKo.Navi-Portal im Laufe des Projekts hinzugefügt wurden. Es dient als Übersicht was, im Vergleich zum Ursprungsprojekt, neu hinzugekommen ist und zur Versicherung, dass immer noch alles Open Source ist.

1. Drupal Contrib-Module (alle **GPLv2**)

- Apache Solr Search (<https://www.drupal.org/project/apachesolr>): Integration der Apache Solr Such-Plattform in Drupal
- Apache Solr Real-Time (https://www.drupal.org/project/apachesolr_realtime): Echtzeit Indizierung von Dokumenten
- Bibliography Module (<https://www.drupal.org/project/biblio>): Verwaltung von Publikationen
- Fivestar (<https://www.drupal.org/project/fivestar>): Bewertung von Inhalten (z. B. auf Inhaltstyp *Wissenskarte*)
- Halfstar (<https://www.drupal.org/project/halfstar>): Feingranularere Bewertung
- Voting API (<https://www.drupal.org/project/votingapi>): Voraussetzung für Bewertungen
- Display Suite (<https://www.drupal.org/project/ds>): UI zur Anpassung des Layouts von Nodes
- Flex Slider (<https://www.drupal.org/project/flexslider>): Responsive Slideshows (z. B. auf Projektstartseite)
- Chosen (<https://www.drupal.org/project/chosen>): Modernere Auswahllisten
- Conditional Fields (https://www.drupal.org/project/conditional_fields): UI zur Definition von Abhängigkeiten zwischen Feldern (z. B. auf Inhaltstyp *Forschungsergebnis*)
- Field Validation (https://www.drupal.org/project/field_validation): UI zur Definition von Validierungsregeln (z. B. auf Inhaltstyp *Projekt*, Feld *Projektzeitraum*)
- Field Group (https://www.drupal.org/project/field_group): Gruppierung von Feldern
- Entity Connect (<https://www.drupal.org/project/entityconnect>): Komfortablere Erstellung von referenzierten Inhalten (Problem mit Chosen, Links für neue Inhalte werden bei *multiselect* nicht hinter den Chosen-Elementen angeordnet. **Provisorischer Fix in *entityconnect.js* vom *Entity Connect Modul***)
- jQuery Autosize (https://www.drupal.org/project/jquery_autosize): Automatische Anpassung von Textareas an deren Inhalt (**Fix: Library pattern, Drupal behaviours angepasst**)
- Font Awesome Icons (<https://www.drupal.org/project/fontawesome>): Skalierbare sowie individuell anpassbare Vektor-Icon-Suite
- Administration menu (https://www.drupal.org/project/admin_menu): effizienteres Admin-Menü

2. JavaScript Bibliotheken

- [imgmap](https://github.com/maschek/imgmap) (<https://github.com/maschek/imgmap>) **GPLv2**
- [Maphilight](https://github.com/kemayo/maphilight) (<https://github.com/kemayo/maphilight>) **MIT**
- [Image Map Resize](https://github.com/davidjbradshaw/image-map-resizer) (<https://github.com/davidjbradshaw/image-map-resizer>) **MIT**
- [qTip²](http://qtip2.com/) (<http://qtip2.com/>) **GPLv2 & MIT**
- [Chosen](https://harvesthq.github.io/chosen/) (<https://harvesthq.github.io/chosen/>) **MIT**
- [Autosize](http://www.jacklmoore.com/autosize/) (<http://www.jacklmoore.com/autosize/>) **MIT**

3. Sonstiges (Bilder, Symbole, Anwendungen, ...)

- [Apache Solr](http://lucene.apache.org/solr/) (<http://lucene.apache.org/solr/>) **Apache License 2.0**
- [Font Awesome Fonts](http://fontawesome.io/license/) (<http://fontawesome.io/license/>) **SIL OFL 1.1**
- [Font Awesome CSS](http://fontawesome.io/license/) (<http://fontawesome.io/license/>) **MIT**
- Alle Bilder für Beispielwissenskarten **Lizenz nicht geprüft**

4. Erweiterungen indeko7-Theme (vgl. Abb. 1b)

- *global.atlas.css* enthält alle Design Anpassungen am Portal durch das Atlas-Projekt
- Unterordner *images/atlas* enthält alle für das Atlas-Projekt erstellten svg Grafiken (von Carlo Frisch erstellt, aus E-Mail vom 5.12.2016)
- *js/node-wissenskarte-comments-style.js*, *template/comment-node-wissenskarte.tpl.php* und *template.php*

5. Entwickelte Drupal-Custom-Module (siehe Abschnitt 4.1)

- *form_wissenskarte_module*, *comments_shortener*, *imagemap_resizer*, *jq_maphilight*, *morphsearch*, *morphsearch_csv_export*, *morphsearch_sort*, *node_creation_links*, *publication_form*, *slider_tooltip*, *user_profile_elements_overview*

7 Offene Punkte

- Problem mit News-Block:

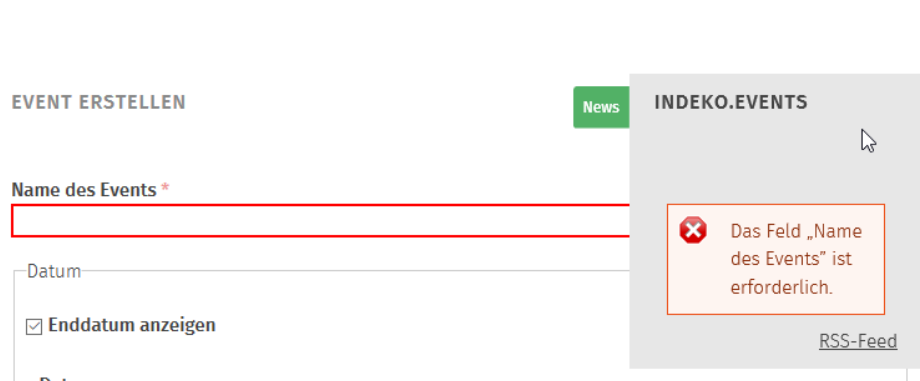


Abbildung 22: Falsche Position von Drupal Fehler- und Bestätigungsmeldungen

Wenn der News-Block leer ist (keine in der Zukunft liegenden Events mehr im System), werden alle Drupal Fehler-/Bestätigungsmeldungen in dem News-Block, statt im oberen Bereich der Seite, angezeigt (siehe Abb. 22).

Glossar

Contrib-Modul (Contributed-Modul) Modul, das von der Drupal-Community oder Drittanbietern entwickelt und zur Verfügung gestellt wird.

Core-Modul Modul, das Bestandteil der Drupal-Standardinstallation ist.

Custom-Modul Modul, das selbst entwickelt wurde, um zusätzliche Funktionalitäten zur Verfügung zu stellen oder Drupal Core-/Contrib-Funktionalitäten zu erweitern.

image map siehe Verweissensitive Grafik.

knowledge map siehe Wissenskarte.

Verweissensitive Grafik Verweis-sensitive Grafiken (englisch *image maps*) sind Grafiken, in denen der Anwender mit der Maus auf ein Detail klicken kann. Daraufhin wird ein Verweis ausgeführt. Auf diese Weise kann der Anwender in einigen Fällen wesentlich intuitiver und schneller zu Information gelangen als durch lange verbale Verweislisten.

Wissenskarte (englisch *knowledge map*) Visualisierung von Wissen.