Das Inhaltsmodell für ORM-Nutzer

Inhaltsklassen, Inhaltsobjekte, Versionen, Übersetzungen, Knoten, Datentypen, Bereiche und Objektzustände in der Sprache eines ORM erklärt - was jedes davon in der Datenbank und in PHP ist, wie sich ein typischer Blog…

Source file: `doc/ger-de/hilfe/web-framework-entwickler/das-inhaltsmodell-fuer-orm-nutzer.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/web-framework-developers/the-content-model-for-orm-users.md

Diese Seite richtet sich an Entwickler, die ein Projekt beginnen würden, indem sie Entity-Klassen und Migrationen schreiben. In Exponential sind die meisten Daten, die Redakteure pflegen, **Inhalte**: Sie definieren deren Typen in der Administrationsoberfläche (oder liefern sie als Pakete aus), und der Kernel speichert, versioniert, übersetzt, berechtigt, cacht und indiziert sie. Ihr Code liest und schreibt Inhalte über die Inhalts-API statt über eine eigene Model-Klasse.

Inhalt

1. Die Bausteine in ORM-Begriffen
2. Datentypen
3. Versionen und Übersetzungen
4. Knoten: ein Objekt, mehrere Orte
5. Bereiche, Zustände und Beziehungen
6. Ein Blog, abgebildet
7. Lesen und Schreiben aus Code

Die Bausteine in ORM-Begriffen

Exponential	Nächste ORM-Entsprechung	Klasse
Inhaltsklasse (Artikel: Titel, Einleitung, Haupttext, Bild)	eine Entity- oder Model-Definition	`eZContentClass`
Klassenattribut	eine Felddefinition mit Typ und Einstellungen	`eZContentClassAttribute`
Inhaltsobjekt	eine Zeile der Entity, mit ihrer Historie	`eZContentObject`
Version	eine Revision der Zeile: Entwurf, veröffentlicht, archiviert und weitere	`eZContentObjectVersion`
Objektattribut	der Wert eines Felds, pro Version und Sprache	`eZContentObjectAttribute`
Knoten (Platzierung)	wo die Zeile im Seitenbaum erscheint; Eltern und Kinder	`eZContentObjectTreeNode`
Remote-ID	ein stabiler externer Schlüssel, den Sie selbst wählen, anders als die numerische ID	`remote_id` von Objekten und Knoten

Die Sicht des Administrators auf dieselben Begriffe: Das Inhaltsmodell und das Bearbeiten von

Inhalten.

Datentypen

Jedes Klassenattribut hat einen **Datentyp**, der Spaltentyp, Formular-Widget, Validator und Renderer in einem ist: `ezstring` (eine Zeile), `eztext`, `ezxmltext` (Rich Text), `ezinteger`, `ezfloat`, `ezboolean`, `ezdate`, `ezdatetime`, `ezemail`, `ezurl`, `ezimage`, `ezbinaryfile`, `ezobjectrelationlist`, `ezselection`, `ezmatrix`, `ezuser` und weitere (die mitgelieferten liegen in `kernel/classes/datatypes/`). Welche für neue Attribute angeboten werden, steht in `content.ini [DataTypeSettings] AvailableDataTypes[]`. Eine neue Art von Feld ist eine Datentyp-Klasse in Ihrer Erweiterung, die `eZDataType` erweitert (Eigene Datentypen).

Versionen und Übersetzungen

- Jede Bearbeitung erzeugt eine **Version**; das Veröffentlichen macht sie aktuell und archiviert die vorherige. Entwürfe sind Versionen mit eigenem Status (`eZContentObjectVersion::STATUS_DRAFT`, `STATUS_PUBLISHED`, `STATUS_ARCHIVED` und weitere).
- Jede Version enthält die Attributwerte für jede **Sprache**, in die sie übersetzt ist. Welche Sprache ein Siteaccess anzeigt, richtet sich nach seiner `SiteLanguageList` (Inhaltssprachen).
- Historie, Vergleichen und Wiederherstellen sind inbegriffen; Sie schreiben nichts davon selbst.

Knoten: ein Objekt, mehrere Orte

Ein Objekt wird über **Knoten** im Baum platziert. Ein Objekt kann mehrere Knoten haben (derselbe Artikel unter „Nachrichten“ und unter „Sport“); einer davon ist sein Hauptknoten. Der Baum liefert Ihnen Navigation, URL-Aliase, Berechtigungen für Teilbäume und Abfragen über Teilbäume (`path_string`, `depth`). Eine eigene Tabelle für „Kategorien“ müssen Sie nicht entwerfen, es sei denn, Sie wollen eine.

Bereiche, Zustände und Beziehungen

Funktion	Zweck	Lesen
Bereich	eine Gruppe von Inhalten für Berechtigungen und Design (Standard, Benutzer, Medien ...)	Bereiche
Objektzustand	ein workflowartiger Status pro Objekt (zum Beispiel „gesperrt“, „in Prüfung“), verwendbar in Richtlinien und Request-Regeln	Objektzustände
Objektbeziehung	eine Verknüpfung von einem Objekt zu anderen, mit Rückwärtssuche gepflegt	Die Inhalts-API

Ein Blog, abgebildet

Vorgehen im Framework	Vorgehen in Exponential

Vorgehen im Framework	Vorgehen in Exponential
Tabelle `posts`, Model `Post`, Migration	Klasse `blog_post` (Titel, Einleitung, Haupttext, Bild, Tags), angelegt unter **Setup > Klassen** oder aus einem Paket installiert
Tabelle `categories`, Fremdschlüssel	Ordner im Baum, ein Beziehungsattribut oder Tags aus `eztags`
Tabelle `post_revisions`	Versionen, eingebaut
Tabelle `post_translations`	Übersetzungen, eingebaut
Spalte `slug`, eindeutiger Index	URL-Alias, eingebaut (Sprechende URLs)
CRUD-Masken für die Administration	die Administrationsoberfläche, eingebaut
Policy-Klasse pro Model	Rollen mit `content/read`, `content/edit`, eingeschränkt nach Klasse, Bereich, Teilbaum (Autorisierung)
Controller und Templates für das Frontend	Override-Templates für die Voll- und Zeilenansicht der Klasse (Template-Auflösung)

Lesen und Schreiben aus Code

PHP

```php
$node   = eZContentObjectTreeNode::fetchByURLPath( 'about_us' );
$object = eZContentObject::fetchByRemoteID( 'expdemo-settings-page' );
$title  = $object->dataMap()['title']->content();

$article = eZContentFunctions::createAndPublishObject( array(
    'creator_id'       => $userID,
    'class_identifier' => 'article',
    'parent_node_id'   => $parentNodeID,
    'remote_id'        => 'feed-item-' . $externalID,
    'attributes'       => array( 'title' => 'A new article' ) ) );
```

Regeln:

- **Schreiben Sie Inhalte nur über die Inhalts-API** (`eZContentFunctions`, Operationen), niemals per SQL in die Inhaltstabellen: Das Veröffentlichen führt Workflows aus, leert Caches und aktualisiert den Suchindex.
- **Bevorzugen Sie Remote-IDs** für Inhalte, von denen Ihr Code abhängt; numerische IDs unterscheiden sich zwischen Installationen.
- **Setzen Sie ein** `Limit` bei jedem Listen-Fetch.

Vollständige Referenz: Die Inhalts-API in PHP und Inhalte im Code anlegen und aktualisieren.

Verwandte Seiten

- ORM und Active Record: eZPersistentObject
- Ihr Inhaltsmodell entwerfen
- Lebenszyklus eines Datentyps
- Freie Dokumentation: Diese Seite darf unter der GNU FDL 1.3 frei kopiert, übersetzt und verbessert werden.

Werden Sie Exponential-Partner

Partner teilen ihre Informationen, stellen Mittel bereit, geben Beiträge zurück und entwickeln die Produkte mit uns, und so wächst Exponential. Eine Partner-Website und ein Dashboard sind für Mitte 2027 geplant: Werden Sie noch heute Partner, damit wir schneller wachsen und neue Funktionen früher liefern.

Partner: hier beginnen | Wie Sie sich bewerben

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.