

Development

Die Klassen Article und ArticleCollection

Ab Contenido Version 4.4.0 werden einige Klassen der Contenido-API mitgeliefert. Diese erleichtern das Entwickeln der Module erheblich.

Dieser Artikel befasst sich mit der Contenido-API Klassen Article und ArticleCollection, die für die Erzeugung von Artikellisten gedacht sind.

Grundkenntnisse in Objektorientierter Programmierung werden benötigt.

Die Klasse Article

Diese Klasse erleichtert den Zugriff auf Artikel und deren Eigenschaften.

Zur Erzeugung eines Objektes der Klasse Article werden 3 Parameter benötigt. Diese sind:

Code:

```
$idart // Die Artikel Id
$client // Mandanten Id
$lang // Sprach Id
```

Das Artikelobjekt wird mit dem operator new instanziiert.

Code:

```
$meinArtikel = new Article(23, 1, 1);
```

Das Objekt hat nun alle Daten des Artikels mit der Id 23, Mandant 1, Sprache 1 gespeichert.

Um die Daten zu extrahieren besitzt das Objekt 2 Methoden get und getContent.

get wird benutzt um die Artikel Eigenschaften zu extrahieren, diese sind in der Contenido Systemtabelle `con_art_lang` definiert.

Code:

```
// Artikeltitle extrahieren
$title = $meinArtikel->get("title");
// Artikelzusammenfassung extrahieren
$zusammenfassung = $meinArtikel->get("summary");
```

Development

```
// Artikel Id extrahieren  
$idart = $meinArtikel->get("idart");
```

Alle Informationen, die im Fenster Arikeleigenschaften definiert sind, können mit der Methode `get` extrahiert werden.

`getContent` wird benutzt um den Artikelinhalt zu extrahieren. Dieser setzt sich in der Regel aus verschiedenen Text und Headline Elementen zusammen..

Als ersten Parameter erwartet `getContent` die Angabe des Contenido CMS Typs als String. Gültige Typen sind in der Systemtabelle ``con_type`` definiert.

Standard sind: `head`, `text`, `html`, `htmlhead`, `img`, `imgdescr`, `link`, `linktarget`, `linkdescr`, `swf`

Die Methode verarbeitet die Kurzform `html` ebenso wie `cms_html`
Gross- und Kleinschreibung müssen nicht beachtet werden.

Code:

```
$text = $meinArtikel->getContent("html", 1);  
$headline = $meinArtikel->getContent("htmlhead", 1);
```

Die Angabe des zweiten Parameters ist optional, er definiert die Id des Content-Elementes.

Also 1 für das erste Content-Element des angeforderten Typs, 2 für das zweite, usw..
Wird er weggelassen liefert die Methode ein Array mit allen Elementen des angeforderten Typs zurück.

Code:

```
$alleHeadlines = $meinArtikel->getContent("htmlhead");  
$alleHeadlines[1] // Erste headline  
$alleHeadlines[2] // Zweite headline
```

Die Klasse *ArticleCollection*

Diese Klasse verwaltet mehrere Objekte des Typs `Article` in einer Sammlung (engl. `Collection`).

Code:

```
$artikelListe = new ArticleCollection(array("idcat"=>12));
```

Development

Als Parameter erwartet diese Klasse ein assoziatives Array mit Optionen.

Code:

```
array("idcat"=>12,  
      "client"=>1,  
      "lang"=>1,  
      "start"=>false,  
      "order"=>"title",  
      "direction"=>"asc");
```

idcat Muss angegeben werden

client Optional, standard ist der aktive Mandant

lang Optional, standard ist die aktive Sprache

start Sollen Startartikel extrahiert werden, standard ist aus

order Entspricht den Feldnamen der Tabelle `con_art_lang` nach dem sortiert werden soll

direction `asc` / `desc` für auf- oder absteigende Sortierung

Die Klasse ArticleCollection besitzt die Methoden nextArticle und startArticle. Beide Methoden erwarten keinen Parameter.

nextArticle liefert ein Objekt der Klasse Article zurück, false wenn kein Objekt mehr in der Sammlung ist.

Code:

```
while ($artikel = $artikelListe->nextArticle())  
{  
    echo $artikel->get("title");  
}
```

startArticle liefert ein Objekt der Klasse Article für den Startartikel zurück.

Code:

```
$startartikel = $artikelListe->startArticle();
```

Artikellisten-Modul mit den Klassen *Article* und *ArticleCollection*

Hier der Beispiel Code für ein Artikellisten-Modul für die Contenido Beispiel Seite:

Code:

Development

```
/**
 * Beispiel eines Artikellisten-Modules
 * mit Hilfe der Contenido-API Klassen.
 *
 * Jan Lengowski / four for business AG
 */

// Artikellisten Optionen einstellen
// Idcat ist ein MUSS field, array("idcat"=>n);
$articleListOptions = array("idcat" => 12, // Idcat (Muss angegeben werden)
    "lang" => $lang, // Sprach id (optional), standard ist die aktive Sprache
    "client"=> $client, // Mandant id (optional), standard ist der aktive Mandant
    "start" => true, // Startartikel
    "order" => "created", // Feldname nach dem sortiert wird, siehe Tabelle 'con_art_lang'
    "direction" => "asc"); // Ab- oder Aufsteigende sortierung ('asc', oder 'desc')

// Artikellisten Objekt erzeugen
$articleList = new ArticleCollection($articleListOptions);

// Artikelliste mit Head1, Head2 und HTML1 ausgeben
while ($article = $articleList->nextArticle())
{
    $head1 = $article->getContent('htmlhead', 1);
    $head2 = $article->getContent('htmlhead', 2);
    $html1 = $article->getContent('html', 1);

    echo $head1 . "
";
    echo $head2 . "
";
    echo $html1 . "

";
}

?>
```

Diese 2 Contenido-API Klassen sind ein mächtiges Werkzeug für den Contenido-Entwickler.

Sie ermöglichen einfachen Zugriff auf Artikel-Content in jeder Sprache und für jeden Mandanten.

Somit sind mehrsprachige Artikellisten, Mandantenübergreifender Content etc.. möglich.

Development

Falls sie noch mehr Informationen wünschen, entnehmen Sie diese bitte der Contenido-API Dokumentation.

Eindeutige ID: #1001

Verfasser: Timo Hummel

Letzte Änderung: 2007-06-23 18:07