

Wie benutze ich die Joomla! Datenbankklassen in meinem Script ?

Montag, 18. September 2006

Original: http://dev.joomla.org/.../how_to_use_the_database_classes_in_your_script

Joomla! bietet eine hochentwickelte Schicht zur Datenbankabstraktion an, um den Umgang für 3PDs zu vereinfachen. Diese Anleitung soll ihnen helfen diesen Database Abstraction Layer zu benutzen.

{tab=Warum}

Warum sollte ich die Joomla Datenbankklassen benutzen ?

Joomla! wurde erstellt, um verschiedene Arten von SQL-Datenbanksystemen in unterschiedlichen Umgebungen mit unterschiedlichen Tabellenpräfixen zu unterstützen. Zusätzlich zu diesen Funktionen erstellt die Klasse automatisch eine Verbindung zur Datenbank. Abgesehen davon, dass Sie das Objekt instanzieren müssen, benötigen Sie nur 2 Zeilen Code um ein Resultat der Datenbank zu erhalten - und das in einer Vielzahl von Formaten. Die Benutzung des Joomla! Datenbanklayers sichert ein Maximum an Kompatibilität und Flexibilität für Ihre Erweiterungen.

{tab=Vorbereitung}

Den Query vorbereiten

```
// Get a database object
```

```
$db = $mainframe->getDBO();
```

```
$query = "SELECT * FROM #__example_table WHERE id = 999999";
```

```
$db->setQuery($query);
```

Zuerst instanzieren wir das Datenbankobjekt, dann bereiten wir den Query vor. Sie können normale SQL Syntax benutzen Das einzige was sie ändern müssen, ist der Tabellenpräfix. Um dies so flexibel wie möglich zu machen, benutzt Joomla! einen Platzhalter für den Präfix, das #___. Im nächsten Schritt, dem setQuery(), wird diese Zeichenkette mit dem eigentlichen Präfix ersetzt.

Falls wir keine Informationen erhalten wollen, aber eine Zeile einfügen wollen, brauchen wir noch eine Funktion. Jede Zeichenkette in der SQL-Syntax muss gequotet werden. Daher müssen wir diese Werte an die Funktion \$db->Quote(\$value) übergeben.

Was immer wir auch tun wollen, wir müssen den Query mit der Funktion SetQuery() setzen. Obwohl Sie auch den Query direkt als einen Parameter der Funktion setQuery() schreiben könnten, wird dies normalerweise getan indem der Query zuerst in eine Variable gespeichert wird, normalerweise \$query, und dann wird diese Variable übergeben. Dies hilft sauberen, lesbaren Code zu schreiben.

{tab=Ausführung}

Den Query ausführen

Um den Query auszuführen, bietet Joomla! einige Funktionen, die sich in ihrem Rückgabewert unterscheiden
Einfache Queryausführung

- query

Information über die Ausführung des Querys

- getAffectedRows

- explain

- insertid

Ausführung von Insert Query

- insertObject

{tab=Resultat}Resultate des Querys

Die Datenbankklasse enthält viele Methoden um mit den Sätzen der Queryresultate zu arbeiten.
Resultat mit einfachem Wert

- loadResult

Resultat einer einzelnen Zeile

- loadObject

- loadRow

- loadResultArray

Resultat mit mehreren Zeilen

- loadObjectList

- loadRowList

- loadAssocList

Resultate diverser Methoden

- getNumRows

{tab=Tips & Tricks}Tips, Tricks & FAQ

Wir hatten kürzlich einige Leute die Subqueries wie diesen benutzen:

```
$query = "SELECT * FROM #__example WHERE id IN (SELECT * FROM #__example2);
```

Diese Sorte Queries sind nur möglich in MySQL 4.1 und darüber. Ein anderer Weg um ist den Query in zwei Teile zu teilen:

```
$query = "SELECT * FROM #__example2";
```

```
$database->setQuery($query);
```

```
$query = "SELECT * FROM #__example WHERE id IN (". implode(",", $database->loadArray()) .")";
```

tips/using_database_class.txt (1242 views) · Last modified: 2006/03/14 15:29 by Jinx {/tabs}