

Anmeldeformular mit Symbolbefehl für CMBasic

(Version 16.05.2013)

Allgemeines

Ein Symbolbefehl für ein Anmeldeformular wäre etwas praktisches.

Pflichtenheft

Mit dem Symbolbefehl wird ein Eingabeformular eingeblendet welches am Schluss der Eingabe durch eine Zusammenstellung aller angemeldeten ersetzt wird. Die Seitenüberschrift und ein eventuell nötiger Eröffnungstext sind „von Hand“ vor dem Symbolbefehl einzutragen.

Das Eingabeformular erfasst Name, Vorname, E-Mail, 3 Antwortfelder und besitzt einen Sendeknopf. Die Art der Antwortfelder lässt sich mit einem Symbolbefehlsattribut einstellen. Es sind folgende Antwortfelder mit 3 Antwortfeldern eingerichtet:

Antwortzeilen:	AW1	AW2	Mail	Attribut 1
	Name	Vorname	Mail	1
	Name Vorname	Anzahl Teilnehmer	Mail	2
	Segelnummer	Crew inkl. Skipper	Mail	3
	Segelnummer	Anzahl Personen	Mail	4

Damit mehrere Anmeldeformulare gleichzeitig verwendet werden können wird mit dem dritten Symbolbefehlsattribut jedem Anmeldeformular eine eigene Kennung abgegeben.

Die eingegebenen Daten werden nach Abschicken auf Vorhandensein überprüft, gefiltert und mit der beigefügten DB-Kennung in eine Datenbanktabelle (cmb7_data) geschrieben.

Nach erfolgreicher Speicherung auf der Datenbank wird zur Bestätigung der Eingabe automatisch der neuste Stand der Anmeldung in Tabellenform angezeigt.

Der Administrator muss die Möglichkeit besitzen Einträge zeilenweise löschen zu können.

Aufbau und Name des Symbolbefehls:

Der Aufbau soll ähnlich sein wie andere Symbolbefehle (Beispiele: {video,dok.flv [breite][höhe]} / {audio,dok.mp3} / {galerie,galerieX} etc.):

{ANMELDUNG,Typ,Kennung}

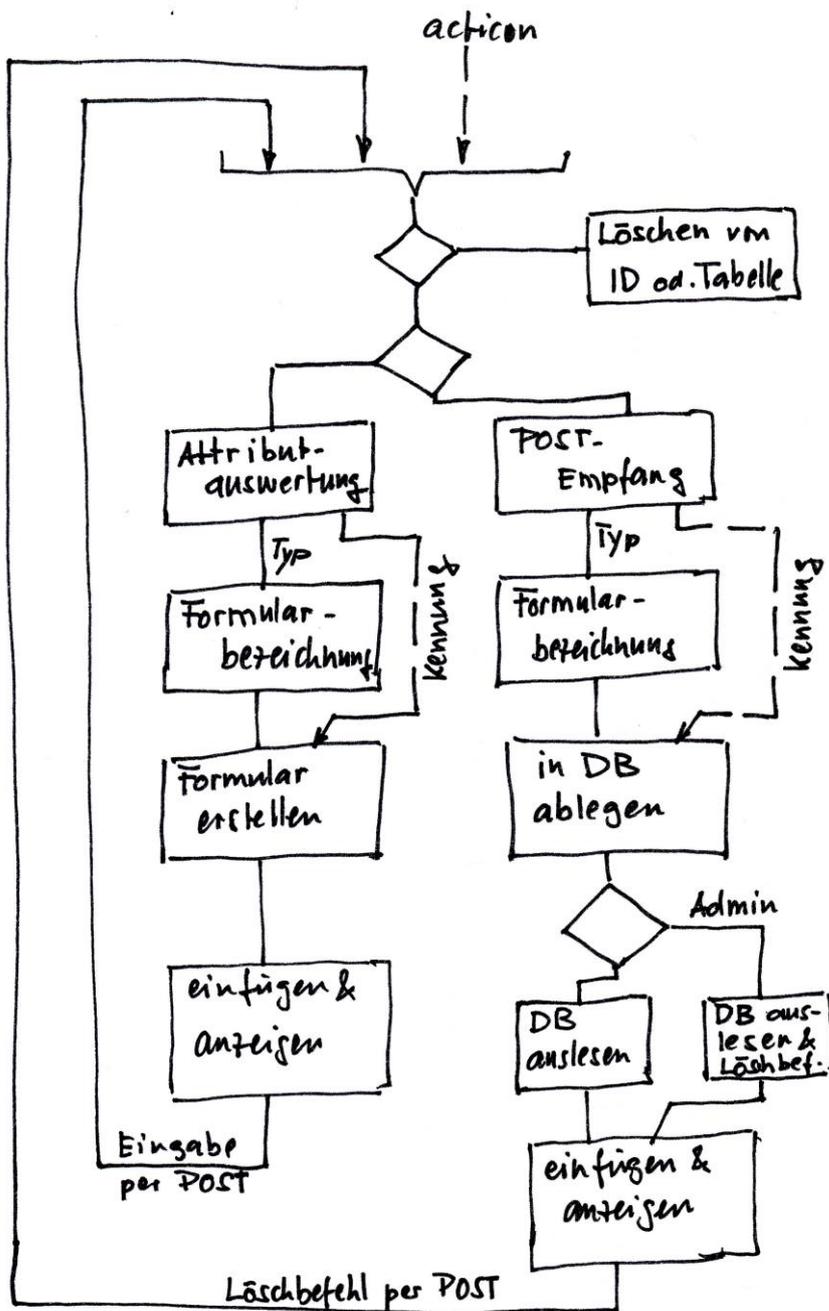
Für „Typ“ wird eine der gewünschten Antwortzeile entsprechende Zahl zwischen 1 bis 4 eingegeben. Unter „Kennung“ ist eine einmalige, einstellige Ziffer im Bereich A – Z einzugeben.

Ideen zur Verwirklichung

- Aufbau und Benennung des verwendeten Symbolbefehls „ANMELDUNG“

- Neue Zeile mit Freigabemöglichkeit des Symbolbefehls in der Datei **config.inc.php**
- Neue Zeile zur Einbindung des Symbolbefehls in der Datei **contentshow.inc.php**
- Einfügung einer neuen Datei **bot_anmeldung.inc.php** im Verzeichnis Minibot
- Anzeige des gegenwärtigen Standes durch die Datei **bot_anmeldung.inc.php** in Tabellenform unterhalb des Eingabeformulars
- Bei Adminrechten unter der Tabelle je eine Löschrzeile für eine ID (ein Eintrag) und die ganze Kennung (ganze Anmeldetabelle)

Programmablauf von bot_anmeldung.inc.php



Programmierung:

<?php

```
/* Anmelde-Einschub für CMBasic
 * zeigt Anmeldeformular am Ort des Symbolbefehls
 * speichert Eingaben auf Datenbank "data"
 * zeigt eine Zusammenstellung der Datenbankliste an.
 * @date 2013-04-07
 * @author WaloZach
 * @lizenz GNU GPL
 */

defined('CMBASIC') or die();

$dbtitel = $pfx.'data'; // Erstellen des DB-Tabellenname

if (isset($_POST['id'])) { // Löschen einzelner Anmeldungen
$id = $_POST['id'];
$sql_x = "DELETE FROM $dbtitel WHERE id = '$id'";
mysql_query($sql_x);
}
if (isset($_POST['kz'])) { // Löschen der Anmeldetabelle
$kz = $_POST['kz'];
$sql_x = "DELETE FROM $dbtitel WHERE kennung = '$kz'";
mysql_query($sql_x);
}

/* Entscheidung ob Auswertung oder Formular anzeige */
if((isset($_POST['typ'])) && (isset($_POST['kenn']))
&& (isset($_POST['aw1'])) && (isset($_POST['aw2']))
&& (isset($_POST['mail']))) {

/* Beginn Zweig für Auswertung, Speicherung und Anzeige*/
$meldung = "";
$mail_n = 'E-Mail';
$fehler = true;

/* POST-Auswertung und Umsetzung */
$typ_p = $_POST['typ']; // Versteckter Formulartyp
$muster_typ = "/^[1-4]{1,1}$/";
if (preg_match($muster_typ, $typ_p) == 0) { // Filter
$typ_1 = "";
$fehler = false;
}
else{
$typ_1 = $typ_p;
}
$kenn_p = $_POST['kenn']; // Versteckte DB-Kennung
$muster_kenn = "/^[A-Z]{1,1}$/";
if (preg_match($muster_kenn, $kenn_p) == 0) { // Filter
$kenn_1 = "";
$fehler = false;
}
else{
$kenn_1 = $kenn_p;
}
$aw1_p = $_POST['aw1']; // Feld 1 Eingang
$muster_aw1 = "/^[A-Za-zääèèöü0-9-_.*]{3,25}$/";
if (preg_match($muster_aw1, $aw1_p) == 0) { // Filter
$aw1_1 = "";
```

```

$fehler = false;
}
else{
$aw1_1 = $aw1_p;
}
$aw2_p = $_POST['aw2']; // Feld 2 Eingang
$muster_aw2 = "/^[A-Za-zàäèèöü0-9-_ &.*]{1,25}$/";
if (preg_match($muster_aw2, $aw2_p) == 0) { // Filter
$aw2_1 = "";
$fehler = false;
}
else{
$aw2_1 = $aw2_p;
}
$mail_p = $_POST['mail']; // Mail - Eingang
$muster_mail = "/^[a-zA-Z0-9-_.]+@[a-zA-Z0-9-_.]+\.[a-zA-Z]{2,4}$/";
if (preg_match($muster_mail, $mail_p) == 0) { // Filter
$mail_1 = "";
$fehler = false;
}
else{
$mail_1 = $mail_p;
}
/* Festlegen der Formularfeldtexte*/
switch($typ_1){
case(1):
$aw1_n = 'Vorname';
$aw2_n = 'Nachname';
break;case(2):
$aw1_n = 'Nachname';
$aw2_n = 'Anzahl Personen';
break;
case(3):
$aw1_n = 'Segel-Nr.';
$aw2_n = 'Crewgroesse inkl Skipper';
break;
case(4):
$aw1_n = 'Segel-Nr.';
$aw2_n = 'Anzahl Personen';
break;
}

/* schreiben der Daten in Datenbank "data" */
$sql = "INSERT INTO $dbtitel (feld1, feld2, feld3, kennung)
VALUES ('$aw1_1', '$aw2_1', '$mail_1', '$kenn_1')";
if ($fehler) { // kein Eintrag bei Fehlermeldung
mysql_query($sql);
if (mysql_affected_rows() > 0) {
$meldung .= "";
}
else {
$meldung .= "<p><h4>Eintrag ist bereits vorhanden!</h4><br />\n";
}
}
else{
$meldung .= "<p><h4>Eintrag wegen fehlender oder unerlaubter Eingabe abgebrochen!</h4><br />\n";
}
}

/* Anzeige der Daten aus der Datenbank "data" für Administratoren */
if (isset($_SESSION['rights']) && ($_SESSION['rights'] > 2)) {
$sql = "SELECT id, feld1, feld2, feld3 FROM $dbtitel ";
}

```

```

    "WHERE kennung='$kenn_1'";
$result = mysql_query($sql); // Array der Anmeldedaten
$zeilen = mysql_num_rows($result); // Anzahl Anmeldezeilen
/* Erstellen der Anzeigetabelle */
$anzeige .= "<table border='1' width='600' cellspacing='0'>\n"; // Tabellenbeginn
$anzeige .= "<tr><td width='50'>ID</td><td width='180'>{$saw1_n}</td>".
"<td width='180'>{$saw2_n}</td><td width='180'>{$mail_n}</td>". // Tabellenkopf
"</tr>\n";
/* zeilenweise auslesen und in Tabelle eintragen */
while ($row=@mysql_fetch_assoc($result)) {
    $anzeige .= "<tr>"; // Zeilenbeginn erzeugen
    foreach ($row as $key => $value) { // foreach beginn
        $anzeige .= "<td>{$value}</span></td>\n"; // Kolonnenwerte einfügen
    } // foreach ende
    $anzeige .= "</tr>"; // Zeile schliessen
} // while ende
$anzeige .= "</table><br />\n"; // Tabelle abschliessen
mysql_close(); // DB ausschalten

/* Anzeige der erstellten Anmeldetabelle */
$eingang_1 = <<<EINGANG
<p>Bis jetzt haben sich folgende {$zeilen} Mitglieder angemeldet: <br/><br />\n
{$anzeige}</p>
<p>-><a href="?home"> zurück</a></p>
<p>
<form action=" method='post'>
Löschen von Eintrag mit ID:<input type='text' name='id' size='3' />
<input type='submit' value='Ausführen' />
</form>
<form action=" method='post'>
Löschen aller Einträge dieser Anmeldeleiste <input type='hidden' name='kz' value='$kenn_1' />
<input type='submit' value='Ausführen' />
</form>
</p>
EINGANG;
$eingang = "$eingang_1"; // mit Stringzeichen versehen
$meldung = $eingang; // Meldung einsetzen
$muster2 = "|{ ANMELDUNG,$typ_1,$kenn_1}|";
$sacticon = preg_replace($muster2, $meldung, $sacticon);
}

else{ // Auslesen für Nichtadministratoren
$sql = "SELECT feld1, feld2, feld3 FROM $dbtitel ".
"WHERE kennung='$kenn_1'";
$result = mysql_query($sql); // Array der Anmeldedaten
$zeilen = mysql_num_rows($result); // Anzahl Anmeldezeilen
/* Erstellen der Anzeigetabelle */
$anzeige .= "<table border='1' width='600' cellspacing='0'>\n"; // Tabellenbeginn
$anzeige .= "<tr><td width='190'>{$saw1_n}</td><td width='190'>{$saw2_n}</td><td
width='190'>{$mail_n}</td>". // Tabellenkopf
"</tr>\n";
/* zeilenweise auslesen und in Tabelle eintragen */
while ($row=@mysql_fetch_assoc($result)) {
    $anzeige .= "<tr>"; // Zeilenbeginn erzeugen
    foreach ($row as $key => $value) { // foreach beginn
        $anzeige .= "<td>{$value}</span></td>\n"; // Kolonnenwerte einfügen
    } // foreach ende
    $anzeige .= "</tr>"; // Zeile schliessen
} // while ende
$anzeige .= "</table><br />\n"; // Tabelle abschliessen
mysql_close(); // DB ausschalten

```

```

/* Anzeige der erstellten Anmeldetabelle */
$eingang_1 = <<<EINGANG
<p>Bis jetzt haben sich folgende {$zeilen} Mitglieder angemeldet: <br/><br />\n
{$anzeige}</p>
<p>-><a href="?home"> zurück</a></p>
EINGANG;
$eingang = "$eingang_1"; // mit Stringzeichen versehen
$meldung .= $eingang; // Meldung einsetzen
$muster2 = "{ ANMELDUNG,$typ_1,$kenn_1}";
$action = preg_replace($muster2, $meldung, $action);
}
} // Ende Zweig für Auswertung

/* Beginn Zweig für Formulareinblendung */
else {
$muster_s = "{ANMELDUNG,antworttyp,db_kennung}"; // Syntax-Muster
$muster0 = "{(( ANMELDUNG),(.*?))}|si"; // Suchmuster
/* Musterdefinitionen: i = Gleichschaltung Klein- & Grossschreibung */
/* . = Jockerpunkt für Zeichen / s = Erweiterung Jockerpunkt auf alle */
/* Zeichentypen / * = alle Zeichen / ? = nur letzter Durchgang */
$preg = preg_match_all($muster0,$action,$anmfund); // action = Zeichenstring
foreach($anmfund[1] as $anmpara_komp) { // Auslesen der Parameterzeile 1
$anmpara = explode(", ", $anmpara_komp); // Parameterzeile -> Parameter aufteilen
if (!empty($anmpara[1])) { // erster Parameter (Antworttyp)
$typ_0 = $anmpara[1];
}
if (!empty($anmpara[2])) { // zweiter Parameter (DB-Kennung)
$kenn_0 = $anmpara[2];
}
}
}
switch($typ_0){ // setzen der Feldtexte
case(1):
$aw1_0 = 'Vorname';
$aw2_0 = 'Nachname';
break;
case(2):
$aw1_0 = 'Nachname';
$aw2_0 = 'Anzahl Personen';
break;
case(3):
$aw1_0 = 'Segel-Nr.';
$aw2_0 = 'Crewgroesse inkl Skipper';
break;
case(4):
$aw1_0 = 'Segel-Nr.';
$aw2_0 = 'Anzahl Personen';
break;
}
$mail_0 = 'E-Mail';
$anmcode_0 = <<<ANMELDUNG
<form action=" method='post'>
<input type='hidden' name='typ' value='{$typ_0}' maxlength='20' />
<input type='hidden' name='kenn' value='{$kenn_0}' maxlength='20' />
<table width='450px' border='0'>
<tbody>
<tr align='left' valign='middle'>
<td width='150px' align='left'>{$aw1_0}</td>
<td width='150px'></td>
<td><input type='text' name='aw1' size='20' /></td>
</tr>
<tr align='left' valign='middle'>

```

```

<td width='150px' align='left'>{$aw2_0}</td>
<td width='150px'></td>
<td><input type='text' name='aw2' size='20' /></td>
</tr>
<tr align='left' valign='middle'>
<td width='150px' align='left'>{$mail_0}</td>
<td width='150px'></td>
<td><input type='text' name='mail' size='20' /></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td width='150px'></td>
<td width='150px'></td>
<td><input type='submit' value='Absenden' /></td>
</tr>
</tbody>
</table>
</form><br />
Versenden Sie bitte die vollständig ausgefüllte Anmeldung mit dem Knopf "Absenden".
<p><br /><a href="?home"> zurück</a></p>
ANMELDUNG;
$anmcode = "$anmcode_0"; // mit Stringzeichen versehen
$muster2 = "|{ANMELDUNG,$typ_0,$kenn_0}|";
$acticon = preg_replace($muster2, $anmcode, $acticon);
}
?>

```

Anpassungen bzw. Einfügungen im bestehenden CMS-System

Datenbanktabelle 'pfx'data einfügen und auf UNIQUE (Feld1,Feld3,Kennung) einrichten

Backend / cmbasic_backup.php auf Zeile 14 mit \$pfx.'data' ergänzen

Edit/config.inc.php ergänzen

Contentshow.inc.php ergänzen (ANMELDUNG nach PWZUGANG wegen Beeinflussung!)

Bot_anmeldung.inc.php einfügen

Beschreibung fürs Handbuch

Einbinden eines Anmeldeformulars

Wenn erwünscht, lässt sich pro Seite ein einfaches Anmeldeformular mit drei Abfragefeldern einbinden deren Eingaben in einer Datenbank abgespeichert werden. Hier ein Beispiel:

NÄCHSTES KUNST-CAFÉ

Dienstag, 4. Juni 2013, 09:30 Uhr, **TERTIANUM AG**, Residenz Mitteldorfpark, Mitteldorfstrasse 16, 3072 Ostermundigen.

Anmeldung zum nächsten Kunst-Café

| | |
|---|--|
| Vorname | <input type="text" value="Didi"/> |
| Nachname | <input type="text" value="Zach"/> |
| E-Mail | <input type="text" value="didi.zach@zawadi.ch"/> |
| <input type="button" value="Absenden"/> | |

Versenden Sie bitte die vollständig ausgefüllte Anmeldung mit dem Knopf "Absenden".

Der neuste Inhalt der Datenbank wird unmittelbar nach der Eingabe angezeigt.

NÄCHSTES KUNST-CAFÉ

Dienstag, 4. Juni 2013, 09:30 Uhr, **TERTIANUM AG**, Residenz Mitteldorfpark, Mitteldorfstrasse 16, 3072 Ostermundigen.

Anmeldung zum nächsten Kunst-Café

Bis jetzt haben sich folgende 1 Mitglieder angemeldet:

| ID | Vorname | Nachname | E-Mail |
|-----|---------|----------|---------------------|
| 580 | Didi | Zach | didi.zach@zawadi.ch |

Der dazu verwendete Symbolbefehl lautet:

{ANMELDUNG,Typ,Kennung}

Das erste fixe Attribut (ANMELDUNG) zeigt dem System an, dass an Stelle des Symbolbefehls ein Anmeldeformular eingeblenet werden soll. Dieses Formular kann mit dem zweiten Attribut (Typ) wie folgt variiert werden:

| Typ | Feld 1 | Feld 2 | Feld 3 |
|-----|-------------|--------------------------|--------|
| 1 | Vorname | Nachname | E-Mail |
| 2 | Nachname | Anzahl Teilnehmer | E-Mail |
| 3 | Segelnummer | Crewgrösse inkl. Skipper | E-Mail |
| 4 | Segelnummer | Anzahl Personen | E-Mail |

Das dritte Attribut (Kennung) dient der Unterscheidung verschiedener Anmeldeformulare. Mehrere Anmeldeformulare werden hier mit einem Buchstaben zwischen A .. Z auseinandergehalten.

Nach erfolgter Eingabe werden die eingegebenen Daten kontrolliert und in einer Datenbank abgelegt. Dabei werden zur Verhinderung von Doppeleinträgen die E-Mailadressen, Vornamen bzw. Segelnummern verglichen; sind diese zwei Elemente bereits einmal verwendet worden, erfolgt kein neuer Eintrag und es wird eine rote Statusmeldung eingeblenet.

Bei fehlenden Namen wird erneut ein leeres Formular angezeigt. Bei kombinierten Vornahmen mit Leerschlägen, fehlender oder mangelhafter E-Mailadressen wird mit einer roten Statusmeldungen quittiert.

Eine erfolgreiche Anmeldung wird durch Anzeigen einer vollständigen Anmeldetabelle mit dem neusten Stand bestätigt.

Die Anmeldetabelle kann jederzeit mit dem Browser ausgedruckt werden. Der Administrator hat zudem die Möglichkeit den gesamten Inhalt oder einzelne Einträge der Tabelle zu löschen.

Bemerkung: Die benötigte Datenbank cmb7_data muss vorher eingerichtet werden. Dabei sind Feld 1, Feld 3 und Kennung auf Gleichheit („Unique“) zu prüfen! Des weitern muss garantiert werden, dass bei einem eventuellem Backup auch diese Tabellen abgesichert werden (Anpassung von cmbasic_backup.inc.php!)

Einsatz

Test -> noch machen!!

Doppeleinträge überprüfen. Verhindert mit Unique Index (Feld 1 & Feld 3 & Kennung)

Gehen alle Typen (Problem Zahlen und Buchstaben verschieden)

Was passiert bei falschen Attribut Eingaben (Typ A-Z, 5-0 / Kennung 1-0)

Was passiert bei zusammengesetzten Vornamen (Marie-Therese)

Was passiert bei zusammengesetzten durch Leerzeichen getrennte Vornamen (Walter Franz)

Sind unerlaubte Tag-Zeichen bei Eingabe verhindert?

Verhalten bei fehlenden Eingaben (Feld 1 / Feld 2 / E-Mail)

Einträge mit falschen E-Mails

Sind rote Statusmeldungen bei falschem oder fehlendem E-Mail vorhanden?

Funktionieren Löschkfunktionen für Administrator

Wieso beeinflusst bot_anmeldung.inc.php das bot_pwzugang.inc.php wenn es in contentshow.inc.php vorher eingeschlaucht wird? Die Header-Funktion von „pwzugang“ wird von „anmeldung“ ausser Betrieb gesetzt, weil ??????. Gäbe es eventuell eine bessere Funktion anstelle der Header-Funktion, die nicht so heikel ist?

Funktioniert Backup von cmb7_data ?