

Inhaltsverzeichnis

Hinzufügen von Extrafeldern (PHP)	3
\$type	3
\$elementtype	4
Linked Object	4

Verwenden von Extra Feldern

Hinzufügen von Extrafeldern (PHP)

```

int function addExtraField(
    string    $attrname           // Code of attribute
    string    $label              // label of attribute
    string    $type               // Type of attribute ('int',
    'text', 'varchar', 'date', 'datehour','link')
    int       $pos                // Position of attribute
    string    $size               // Size/length of attribute
    string    $elementtype       // Element type ('member',
    'product', 'thirdparty', 'commande_fournisseur', ...)
    int       $unique             = 0 // Is field unique or not
    int       $required           = 0 // Is field required or not
    string    $default_value     = '' // Defaulted value
    array     $param              = 0 // Params for field e.g.
    array('options'=>array('Lieferung:jheExtension/efc/lieferung.class.php'=>NULL));
    int       $alwayseditable    = 0 // Is attribute always editable
    regardless of the document status
    string    $perms              = '' // Permission to check
    int       $list               = 0 // Into list view by default (not
    used yet)
); // returns int <=0 if KO, >0 if OK

$extrafield0bj = new ExtraFields($this->db);

$extrafield0bj->addExtraField( ... );

$extrafield0bj->delete( $attrname, $elementtype );

```

\$type

Mögliche Werte:

- 'separate'
- 'boolean'
- 'price'
- 'phone'
- 'mail'
- 'select'

- 'sellist'
- 'radio'
- 'checkbox'
- 'chkbxlst'
- 'text'
- 'link'

\$elementtype

Mögliche Werte: Ergibt sich aus dem Datenbank-Tabellen-Namen. z.B. für Bestellungen von Lieferanten: 'commande_fournisseur'

Linked Object

In *Wert* ist folgendes einzutragen:

```
<ClassName>:<Pfad zu Klassen-Datei relativ zu DOL_DOCUMENT_ROOT>
```

Minimales Grundgerüst einer verwendbaren Objekt-Klasse:

```
class <ClassName> extends CommonObject {
    var $db;
    var $id; /* wird in der Datenbank abgespeichert und aufgrund $id
wird $ref ermittelt */
    var $ref; /* Wird im Eingabefeld als Wert angezeigt */

    /**
     * Constructor
     * @param DoliDB $db Database handler
     */
    function __construct($db) {
        $this->db = $db;
    }

    /**
     * Füllt Object mit Werten
     *
     * @param int $rowid
     * @param string $ref
     */
    function fetch($rowid, $ref='') {
        $this->id = ... /* muss belegt werden */
        $this->ref = ... /* muss belegt werden - Wird im Eingabefeld
```

```
als vorbelegter Wert angezeigt */
}

/**
 * Gibt anzuzeigenden String zurück
 *
 * @param int $withpicto 0=_No picto, 1=Includes the picto in the
linkn, 2=Picto only
 *
 * @return string String der angezeigt wird
 */
function getNomUrl($withpicto=0) {
    return "<string>";
}
}
```

Wird das Extrafield bearbeitet wird die Klassen-Information gelöscht. (Dolibarr Version 3.8.1) Daher muss das Feld immer erst gelöscht und neu angelegt werden.

\$id, d.h. der Wert der in der Datenbank abgelegt wird muss vom Typ **int** sein.

Beim Speichern von anderen Extra-Feldern werden alle Extra-Felder aktualisiert und ein schon gespeicherter Wert wird der fetch-Methode als Parameter \$ref übergeben.

(Dolibarr Version 3.8.1)

[web, dolibarr](#)