

Tab

Mit der Klasse `addon_tab` kann auf Add-On-spezifische Tabellen zugegriffen werden.

Methoden

`del_row_by_cond`

Syntax Bis V4.6

```
boolean del_row_by_cond(string $cond)
```

Ab V4.7

```
boolean del_row_by_cond(string $cond, bool $only_deletable = false)
```

Beschreibung `$cond` ist eine WHERE-Klausel im SQL99-Standard. Ist `$only_deletable` `true`, muss der Nutzer das Löschrecht besitzen. Zurückgegeben wird `true`, wenn das Löschen erfolgreich war.

`del_row_by_id`

Syntax

```
boolean del_row_by_id(int $id, boolean $only_deletable = false)
```

Beschreibung Ab V4.7. Löscht den Datensatz mit ID `$id`. Ist `$only_deletable` `true`, muss der Nutzer das Löschrecht besitzen. Zurückgegeben wird `true`, wenn das Löschen erfolgreich war.

`fetch_array`

Syntax

```
array fetch_array(result_set $result)
```

Beschreibung Liefert zu dem aktuellen Eintrag im Result-Set `$result` ein assoziatives Array. Wenn kein Eintrag mehr vorhanden ist, wird `false` zurückgegeben.

Beispiel

```
$table = new addon_tab('ressources_lendings');  
$result = $table->get_row_by_cond('id>0');  
while ($row = $table->fetch_array($result)) {
```

```
    crmapi::log_error($row);  
}
```

get_count_by_cond

Syntax Bis V4.6

```
int get_count_by_cond(string $cond)
```

Ab V4.7

```
int get_count_by_cond(string $cond, boolean $only_readable = false)
```

Beschreibung \$cond ist eine WHERE-Klausel im SQL99-Standard.

Ist \$only_readable true, werden nur Datensätze mit Leserecht berücksichtigt. Zurückgegeben wird die Anzahl der Treffer.

Beispiel

```
$table = new addon_tab('ressources_lendings');  
$count = $table->get_count_by_cond('id>0');  
crmapi::log_error($count);
```

get_delete_permission

Syntax

```
boolean get_delete_permission(int $oid)
```

Beschreibung Ab V4.7. Liefert das Löschrecht des aktuellen Benutzers für den Datensatz mit ID \$oid.

Beispiel

```
$table = new addon_tab('ressources_lendings');  
$perm = $table->get_delete_permission(1);  
crmapi::log_error($perm);
```

get_multi_permissions

Syntax

```
boolean get_multi_permissions(array $oids)
```

Beschreibung Ab V4.7. Liefert die Rechte des aktuellen Benutzers für mehrere Datensätze.

Beispiel

```
$table = new addon_tab('ressources_lendings');  
$perm = $table->get_multi_permissions(array(1, 2, 3));  
crmapi::log_error($perm);
```

get_oids_with_deletepermission

Syntax

```
array get_oids_with_deletepermission($returnkeys = false)
```

Beschreibung Ab V4.7. Liefert die IDs aller Datensätze, die der Nutzer löschen kann.

Ist \$returnkeys true, wird ein Array zurückgegeben, in dem die Werte die IDs sind. Sonst werden die IDs als Keys verwendet.

Beispiel

```
$table = new addon_tab('ressources_lendings');  
$perm = $table->get_oids_with_deletepermission(false);  
crmapi::log_error($perm);
```

get_oids_with_fullpermission

Syntax

```
array get_oids_with_fullpermission($returnkeys = false)
```

Beschreibung Ab V4.7. Liefert die IDs aller Datensätze, auf die der Nutzer Vollzugriff.

Ist \$returnkeys true, wird ein Array zurückgegeben, in dem die Werte die IDs sind. Sonst werden die IDs als Keys verwendet.

Beispiel

```
$table = new addon_tab('ressources_lendings');  
$perm = $table->get_oids_with_fullpermission(false);  
crmapi::log_error($perm);
```

get_oids_with_readpermission

Syntax

```
array get_oids_with_readpermission($returnkeys = false)
```

Beschreibung Ab V4.7. Liefert die IDs aller Datensätze, die der Nutzer lesen kann.

Ist \$returnkeys true, wird ein Array zurückgegeben, in dem die Werte die IDs sind. Sonst werden die IDs als Keys verwendet.

Beispiel

```
$table = new addon_tab('ressources_lendings');  
$perm = $table->get_oids_with_readpermission(false);  
crmapi::log_error($perm);
```

get_oids_with_writepermission

Syntax

```
array get_oids_with_writepermission($returnkeys = false)
```

Beschreibung Ab V4.7. Liefert die IDs aller Datensätze, die der Nutzer ändern kann.

Ist \$returnkeys true, wird ein Array zurückgegeben, in dem die Werte die IDs sind. Sonst werden die IDs als Keys verwendet.

Beispiel

```
$table = new addon_tab('ressources_lendings');  
$perm = $table->get_oids_with_writepermission(false);  
crmapi::log_error($perm);
```

get_read_permission

Syntax

```
array get_read_permission(int $oid)
```

Beschreibung Ab V4.7. Gibt an, ob der Datensatz mit ID \$oid gelesen werden kann.

Beispiel

```
$table = new addon_tab('ressources_lendings');  
$perm = $table->get_read_permission(2);  
crmapi::log_error($perm);
```

get_rights

Syntax

```
array get_rights(int $oid)
```

Beschreibung Ab V4.7. Liefert alle Berechtigungen für einen Datensatz.

Beispiel

```
$table = new addon_tab('ressources_lendings');  
$perm = $table->get_read_permission(2);  
crmapi::log_error($perm);
```

get_row_by_cond

Syntax Bis V4.6

```
result_set get_row_by_cond(string $cond)
```

Ab V4.7

```
result_set get_row_by_cond(string $cond, array $placeholder = array(),  
boolean $only_readable = false)
```

Beschreibung Bis V4.6: \$cond ist eine WHERE-Klausel im SQL99-Standard.

Ab V4.7: \$cond ist eine WHERE-Klausel im SQL-Format. Bitte beachten Sie dazu die Hinweise.

Ab V4.7: \$placeholder ist ein Array mit Platzhaltern. Bitte beachten Sie dazu die Hinweise.

Ab V4.7: Ist \$only_readable true, werden nur Datensätze mit Leserecht berücksichtigt.

Zurückgegeben wird Datenbank-Result-Set.

Beispiel

```
$table = new addon_tab('ressources_lendings');
$tab_result = $table->get_row_by_cond('lending>0');

$line = $table->fetch_array($tab_result);
while ($line) {
    //do something
    $line = $table->fetch_array($tab_result);
}
```

get_val_by_cond

Syntax Bis V4.6:

```
result_set get_val_by_cond(string $cols, string $cond)
```

Ab V4.7:

```
result_set get_val_by_cond(string $cols, string $cond, array $placeholder =
array(), boolean $only_readable = false)
```

Beschreibung \$cols ist eine komma-separierte Liste der Spalten.

Bis V4.6: \$cond ist eine WHERE-Klausel im SQL99-Standard.

Ab V4.7: \$cond ist eine WHERE-Klausel im SQL-Format. Bitte beachten Sie dazu die Hinweise.

Ab V4.7: \$placeholder ist ein Array mit Platzhaltern. Bitte beachten Sie dazu die Hinweise.

Ab V4.7: Ist \$only_readable true, werden nur Datensätze mit Leserecht berücksichtigt.

get_write_permission

Syntax

```
array get_read_permission(int $oid)
```

Beschreibung Ab V4.7. Gibt an, ob der Datensatz mit ID \$oid geändert werden kann.

Beispiel

```
$table = new addon_tab('ressources_lendings');  
$perm = $table->get_write_permission(2);  
crmapi::log_error($perm);
```

insert_row

Syntax

```
int insert_row(array $values)
```

Beschreibung Fügt einen Datensatz in die Tabelle ein. Die Daten werden als assoziatives Array übergeben, in dem die Keys den Spaltennamen entsprechen. Zurückgegeben wird die ID des Datensatzes bei Erfolg, sonst false.

Beispiel

```
$table = new addon_tab('ressources_lendings');  
$data = array();  
$data['ressource_id'] = '4';  
$data['lender']       = 'thomas';  
$data['lend_start']   = time();  
$table->insert_row($data);
```

insert_update_row

Syntax

```
bool insert_update_row(array $values)
```

Beschreibung (Ab v4.10) Fügt einen Datensatz in die Tabelle ein oder aktualisiert diesen, wenn sonst ein Duplikat entstehen würde. Die Daten werden als assoziatives Array übergeben, in dem die Keys den Spaltennamen entsprechen. Bei Erfolg wird true zurückgegeben.

Hinweis Damit diese Funktion Duplikate erkennt, muss wenigstens ein [Unique-Index](#) vorhanden sein. Ist kein Unique-Index vorhanden wird immer ein neuer Datensatz eingefügt!

Beispiel

```
$table = new addon_tab('ressources_lendings');  
$data = array();  
$data['ressource_id'] = '4';  
$data['lender']       = 'thomas';  
$data['lend_start']   = time();
```

```
$table->insert_update_row($data);
```

save_group_rights

Syntax

```
array save_group_rights(int $oid, int $gid, int $r, int $w, int $x)
```

Beschreibung Ab V4.7. Speichert zu einem Datensatz die Gruppen-Zugriffsrechte.

Beispiel

```
$table = new addon_tab('ressources_lendings');  
$perm = $table->save_group_rights(2, 1, 1, 1, 1);  
crmapi::log_error($perm);
```

save_user_rights

Syntax

```
array save_user_rights(int $oid, int $uid, int $r, int $w, int $x)
```

Beschreibung Ab V4.7. Speichert zu einem Datensatz die Benutzer-Zugriffsrechte.

Beispiel

```
$table = new addon_tab('ressources_lendings');  
$perm = $table->save_user_rights(2, 1, 1, 1, 1);  
crmapi::log_error($perm);
```

set_acl

Syntax

```
array set_acl(int $oid, int $acl_id)
```

Beschreibung Ab V4.7. Setzt zu einem Datensatz eine Zugriffsliste.

Beispiel


```
$table = new addon_tab('ressources_lendings');  
$perm = $table->set_acl(2, 1);  
crmapi::log_error($perm);
```

update_row_by_cond

Syntax Bis V4.6:

```
boolean update_row_by_cond(string $cond, array $values)
```

Ab V4.7:

```
boolean update_row_by_cond(string $cond, array $values, array $placeholder =  
array, boolean $only_writeable = false)
```

Beschreibung Aktualisiert Datensätze mit den Werten aus \$values, die durch \$cond ausgewählt werden. Die Daten werden als assoziatives Array übergeben, in dem die Keys den Spaltennamen entsprechen. Zurückgegeben wird bei Erfolg true, sonst false.

Ab V4.7: \$placeholder kann eine Liste von Werten enthalten, die in die Bedingung und auch in die Werte eingesetzt werden.

Ab V4.7: Ist \$only_writeable true, werden nur Datensätze mit Schreibrecht durch den Nutzer berücksichtigt.

Beispiel

```
$table = new addon_tab('ressources_lendings');  
$data = array();  
$data['ressource_id'] = '4';  
$data['lender'] = 'thomas';  
$data['lend_start'] = time();  
$test = $table->update_row_by_cond('id>0', $data);
```

update_row_by_id

Syntax Bis V4.6:

```
boolean update_row_by_id(int $id, array $values)
```

Ab V4.7:

```
boolean update_row_by_id(int $id, array $values, boolean $only_writeable =  
false)
```

Beschreibung Aktualisiert den Datensatz mit ID \$id mit den Werten aus \$values, die durch \$cond ausgewählt werden. Die Daten werden als assoziatives Array übergeben, in dem die Keys den Spaltennamen entsprechen. Zurückgegeben wird bei Erfolg true, sonst false.

Ab V4.7: Ist \$only_writeable true, werden nur Datensätze mit Schreibrecht durch den Nutzer berücksichtigt.

Beispiel

```
$table = new addon_tab('ressources_lendings');
$data = array();
$data['ressource_id'] = '4';
$data['lender']      = 'thomas';
$data['lend_start']  = time();
$test = $table->update_row_by_id(5, $data);
```

Hinweise zu SQL

Im Allgemeinen wird der SQL99-Standard unterstützt. Es gibt folgende Einschränkungen:

- Keine ` um Tabellennamen
- Negationen müssen der Bedingung vorangestellt werden (z.B. "NOT name LIKE '%" statt "name NOT LIKE '%")

Platzhalter

In den Abfragen können Platzhalter verwendet werden. Das ist besonders empfohlen, wenn die Abfrage mehrmals mit unterschiedlichen Werten ausgeführt werden soll.

Beispiel

```
$query = "realnumber=#realnumber OR company=#company OR id=#cid";
$placeholder = array('cid' => 1, 'realnumber' => "abc", 'company' => '0');
$table = new addon_tab('ressources_lendings');
$test = $table->get_row_by_cond($query, $placeholder);

// Ergibt diese Condition: realnumber='abc' OR company='0' OR id=1
```

From:
<https://wiki.tecart.de/> - **TecArt Developer Wiki**

Permanent link:
<https://wiki.tecart.de/addon:referenz:tab>



Last update: **2019/06/19 17:57**