

# Uitypes personalizzati

Si possono aggiungere dei nuovi tipi a quelli già esistenti e gestirli completamente senza modificare altro codice. Per crearne la procedura è:

1. Creare un nuovo campo personalizzato indicando il nuovo uitype con un valore non utilizzato (nnn).
2. Creare i file:
  - a. nnn.php in modules/SDK/examples
  - b. nnn.js in modules/SDK/examples
  - c. nnn.tpl in Smarty/templates/modules/SDK/examplesQuesti file gestiscono il comportamento del nuovo campo a seconda del contesto (list, detail...)
3. Registrare il nuovo tipo con una chiamata a SDK::setUitype.

Nel file modules/SDK/examples/VTE-SDK-2.php ci sono vari esempi di creazione campi e registrazione uitype.

In modules/SDK/doc/VTE-SDK-2.pdf alla voce **Uitypes List** è presente la lista dei principali uitype standard.

**SDK::setUitype(\$uitype, \$src\_php, \$src\_tpl, \$src\_js, \$type="", \$params=");**

*\$uitype : il numero del nuovo tipo; deve essere nnn (nome dei files)*

*\$src\_php: percorso di nnn.php*

*\$src\_tpl: percorso di nnn.tpl (senza Smarty/templates/ all'inizio)*

*\$src\_js : percorso di nnn.js*

*\$type : tipo in formato webservice ('text', 'boolean', ...)*

*\$params : non usato*

Per de-registrare un uitype è disponibile il metodo:

**SDK::unsetUitype(\$uitype);**

*\$uitype : è il numero del uitype da rimuovere (non verranno eliminati i files a esso associati)*

Si consiglia di usare uitype con valore maggiore di 2000, per evitare conflitti con futuri rilasci del CRM.

All'ingresso degli script php sono disponibili varie variabili, la prima è:

*\$sdk\_mode : le viste e funzioni che posso personalizzare per il nuovo uitype ("insert", "detail", "edit", "relatedlist", "list", "pdfmaker", "report", ecc.)*

In base al tipo di \$sdk\_mode ho a disposizione diverse variabili che posso leggere e modificare.

## **detail**

per gestire la visualizzazione del campo in DetailView

### **INPUT**

*\$module : modulo corrente*

*\$fieldlabel: etichetta del campo*

*\$fieldname : nome del campo*

*\$col\_fields: (array) valori dei campi*

### **OUTPUT**

*\$label\_fld[] : etichetta tradotta*

*\$label\_fld[] : valore da visualizzare*

## **edit**

per gestire la visualizzazione del campo in EditView

### **INPUT**

*\$module\_name : modulo corrente*

*\$fieldlabel : etichetta del campo*

*\$value : valore del campo*

### **OUTPUT**

*\$editview\_label[] : etichetta tradotta*

*\$fieldvalue[] : valore da visualizzare*

## **relatedlist, list, pdfmaker**

per gestire la visualizzazione del campo in ListView, RelatedList e PDFMaker

### **INPUT**

*\$sdk\_value : valore del campo*

### **OUTPUT**

*\$value : valore da visualizzare*

## **report**

per gestire la visualizzazione del campo nei Report

### **INPUT**

*\$sdk\_value : valore del campo*

### **OUTPUT**

*\$fieldvalue : valore da visualizzare*

Se il valore salvato nel database è diverso da quello mostrato da interfaccia (es. numero 1.000,25 che deve essere salvato come 1000.25) allora vanno gestite anche le seguenti modalità per salvare il valore nel formato corretto e cercarlo.

## **insert**

per convertire il valore nel formato da salvare nel database

### **INPUT**

*\$this->column\_fields : (array) valori dei campi*

*\$fieldname : nome del campo*

### **OUTPUT**

*\$fldvalue : valore da salvare nel database*

### **formatvalue**

per convertire il valore che arriva dalla \$\_REQUEST nel formato salvato nel database (usato nella nuova gestione dei Campi Condizionali)

*INPUT*

*\$value : valore del campo*

*OUTPUT*

*\$value : valore convertito nel formato database*

### **querygeneratorsearch**

per convertire il valore cercato dall'utente in lista e filtri

*INPUT E OUTPUT*

*\$fieldname : nome del campo*

*\$operator : operatore di confronto*

*\$value : valore cercato*

### **customviewsearch**

per gestire la conversione del valore nei filtri dei popup per i campi reference

*INPUT E OUTPUT*

*\$tablename : tabella del campo*

*\$fieldname : nome del campo*

*\$comparator : operatore di confronto*

*\$value : valore cercato*

### **popupbasicsearch**

per gestire la conversione del valore nella ricerca dei popup per i campi reference

*INPUT*

*\$table\_name : tabella del campo*

*\$column\_name : colonna del campo*

*\$search\_string : valore cercato*

*OUTPUT*

*\$where : condizione della query*

es. *\$where = "\$table\_name.\$column\_name = '".convertToDBFunction(\$search\_string)."'";*

### **popupadvancedsearch**

per gestire la conversione del valore nella ricerca avanzata dei popup per i campi reference

*INPUT E OUTPUT*

*\$tab\_col : tabella e colonna del campo*

*\$srch\_cond : operatore di confronto*

*\$srch\_val : valore cercato*

### **reportsearch**

per convertire il valore cercato dall'utente nei report

*INPUT E OUTPUT*

*\$table : tabella del campo*

*\$column : colonna del campo*

*\$fieldname : nome del campo*

*\$comparator : operatore di confronto*

*\$value : valore cercato*

### **Hooks**

data/CRMEntity.php  
include/ListView/ListViewController.php  
include/utils/crmv\_utils.php  
include/utils/EditViewUtils.php  
include/utils/DetailViewUtils.php  
include/utils/ListViewUtils.php  
include/utils/SearchUtils.php  
include/QueryGenerator/QueryGenerator.php  
modules/PDFMaker/InventoryPDF.php  
modules/Reports/ReportRun.php  
modules/Users/Users.php  
modules/CustomView/Save.php  
modules/CustomView/CustomView.php  
Smarty/templates/DisplayFieldsReadonly.tpl  
Smarty/templates/DisplayFieldsHidden.tpl  
Smarty/templates/DetailViewFields.tpl  
Smarty/templates/EditViewUI.tpl  
Smarty/templates/DetailViewUI.tpl

---

Revision #9

Created 19 February 2020 17:50:41 by ddalmaso

Updated 21 June 2024 07:37:02 by m.maporti