

# Gestione dei popup

Per la gestione delle finestre popup sono disponibili due azioni. Si può inserire uno script php prima che venga fatta la query per caricare i dati, in modo da poterne selezionare diversi da quelli standard. Inoltre è possibile inserire un altro script php prima che i dati vengano mostrati, in modo da poter modificare tali dati o il risultato alla sua chiusura.

Nel primo caso sono disponibili i 2 metodi:

**SDK::setPopupQuery(\$type, \$module, \$param, \$src, \$hidden\_rel\_fields = "");**

*\$type* : "field" o "related" per indicare un campo standard o il popup aperto da una related list

*\$module*: il modulo in cui si apre il popup

*\$param* : il nome del campo che apre il popup (deve essere uitype 10) nel caso type = "field", altrimenti il nome del modulo collegato.

*\$src* : il percorso del file php

*\$hidden\_rel\_fields* : array del tipo array(\$urlvalue => \$jscode)

*\$urlvalue* : parametro da aggiungere all'url quando si apre un popup

*\$jscode* : codice javascript eseguito quando si apre il popup, il cui valore viene assegnato a \$urlvalue

E per de-registrarlo:

**SDK::unsetPopupQuery(\$type, \$module, \$param, \$src);**

*Stessi parametri di prima*

All'interno dello script php sono disponibili le seguenti variabili:

*\$query* : la query che prende i valori da mostrare

*\$sdk\_show\_all\_button* : se true mostra il pulsante per annullare le restrizioni SDK e mostrare tutti i record

Nel secondo caso invece ci sono i seguenti metodi:

**SDK::setPopupReturnFunction(\$module, \$fieldname, \$src);**

*\$module* : il modulo che contiene il campo che apre il popup

*\$fieldname* : il nome del campo che apre il popup (solo uitype 10)

*\$src* : il file php

Per de-registrare la funzione di popup usare il metodo:

**SDK::unsetPopupReturnFunction(\$module, \$fieldname = NULL, \$src = NULL);**

Per ora gli unici campi supportati sono quelli con uitype 10 (a parte per le related list)

## Hooks

Popup.php

```
include/Utils/ListViewUtils.php
include/ListView/SimpleListView.php
```

## Esempio

```
<?php
SDK::setPopupQuery('field', 'Contacts', 'account_name', 'modules/SDK/examples/PopupQuery1.php');
SDK::setPopupQuery('related', 'Contacts', 'Products',
'modules/SDK/examples/contacts/PopupRelQuery.php');

SDK::setPopupReturnFunction('Contacts', 'vendor_id', 'modules/SDK/examples/ReturnVendorToContact
.php');

// you can set a PopupQuery and a PopupReturnFunction to a field in a table field (VTENEXT
24.08)
SDK::setPopupQuery('field', 'Accounts', 'ml1_f4',
'modules/SDK/examples/Contacts/AccountQuery.php');
// here we have a table field (vcf_3) with an account field (vcf_4) and a contact field
(vcf_5), I can view only accounts of rating Market Failed, Project Cancelled and Shutdown and
only contacts of these accounts.
$rel_fields =
array('processmaker'=>'jQuery("#processmaker").val()', 'running_process'=>'jQuery("#running_pro
cess").val()');
SDK::setPopupQuery('field', 'Processes', 'vcf_3_vcf_4',
'modules/SDK/examples/Contacts/AccountQuery.php', $rel_fields);
$rel_fields['accountid'] =
'jQuery("#vcf_3_vcf_4_"+VTE.EditView.getTableFieldCurrentRow(this)).val()';
SDK::setPopupQuery('field', 'Processes', 'vcf_3_vcf_5',
'modules/SDK/examples/Contacts/ContactsQuery.php', $rel_fields);

// you can set a PopupReturnFunction to the product or other custom fields in the inventory
product block (VTENEXT 26.01)
// view examples in modules/SDK/examples/VTE-SDK-2.php
?>
```

## AccountQuery.php

```
<?php
global $table_prefix;
```

```

// check the current position in process
if ($_REQUEST['srcmodule'] == 'Processes') {
    $processmaker = 4;
    $elementid = 'Task_0jlyxi6'; // task condition after the dynaform

    $curr_processmaker = RequestHandler::paramGetInt('processmaker');
    $curr_running_process = RequestHandler::paramGetInt('running_process');

    if ($curr_processmaker != $processmaker) return;

    require_once('modules/Settings/ProcessMaker/ProcessMakerUtils.php');
    $PMU = ProcessMakerUtils::getInstance();
    $curr_elementid = $PMU->getCurrentElementId($curr_running_process, $curr_processmaker);
    if ($curr_elementid != $elementid) return;
}

$sdk_show_all_button = true;
$query .= " and {$table_prefix}_account.rating not in ('Market Failed','Project
Cancelled','Shutdown')";

```

## ContactsQuery.php

```

<?php
global $table_prefix;
$accountid = RequestHandler::paramGetInt('accountid');

// check the current position in process
if ($_REQUEST['srcmodule'] == 'Processes') {
    $processmaker = 4;
    $elementid = 'Task_0jlyxi6'; // task condition after the dynaform

    $curr_processmaker = RequestHandler::paramGetInt('processmaker');
    $curr_running_process = RequestHandler::paramGetInt('running_process');

    if ($curr_processmaker != $processmaker) return;

    require_once('modules/Settings/ProcessMaker/ProcessMakerUtils.php');
    $PMU = ProcessMakerUtils::getInstance();
    $curr_elementid = $PMU->getCurrentElementId($curr_running_process, $curr_processmaker);
    if ($curr_elementid != $elementid) return;
}

```

```
$sdk_show_all_button = false;
```

```
$query .= " and {$table_prefix}_contactdetails.accountid = $accountid";
```

---

Revision #10

Created 2020-02-19 18:10:12 UTC by ddalmaso

Updated 2026-04-20 14:48:00 UTC by manuel.tagliapietra