10 SDK di processo

All’interno dei processi è possibile richiamare delle funzioni PHP al fine di eseguire delle operazioni od utilizzare dei dati non disponibili nello standard del motore BPMN.
In particolare, esistono 3 differenti tipologie di funzioni registrabili e richiamabili in punti differenti della configurazione di un processo: Fied Action, Task Action e Task Condition.

10.1 SDK Field Action

Questa tipologia di funzione viene normalmente utilizzata per valorizzare i campi di un record, form dinamica, ecc. con dei valori calcolati utilizzando il seguente metodo:

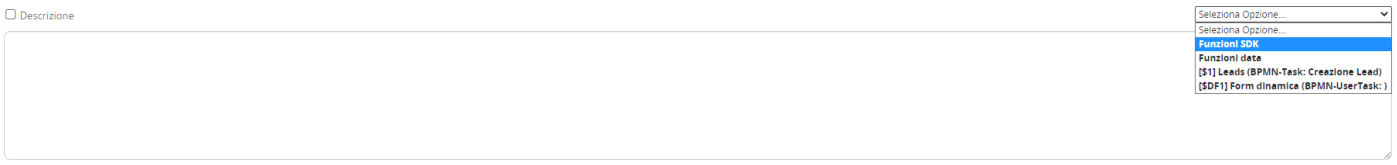
SDK::setProcessMakerFieldAction($func, $src, $label, $parameters='');
$func : nome della funzione
$src : percorso file contenente la funzione
$label : etichetta della funzione
$parameters (optional) : parametri statici (es. riferimento campo fisso)

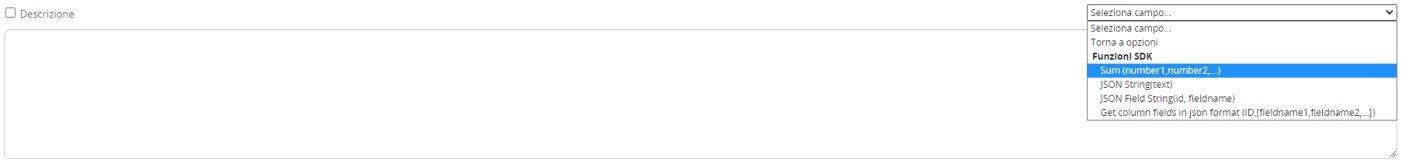
Per de-registrare il file usare:

SDK::unsetProcessMakerFieldAction($func);
$func: nome della funzione

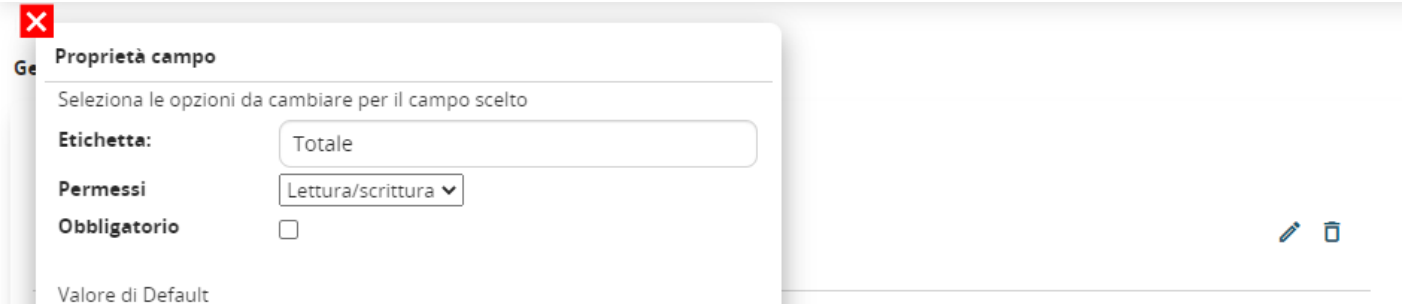
Queste funzioni possono essere richiamate dalle sezioni apposite “Funzioni SDK” disponibili nel menù a tendina “Selezione opzione” situato in alto a destra di ogni singolo campo.

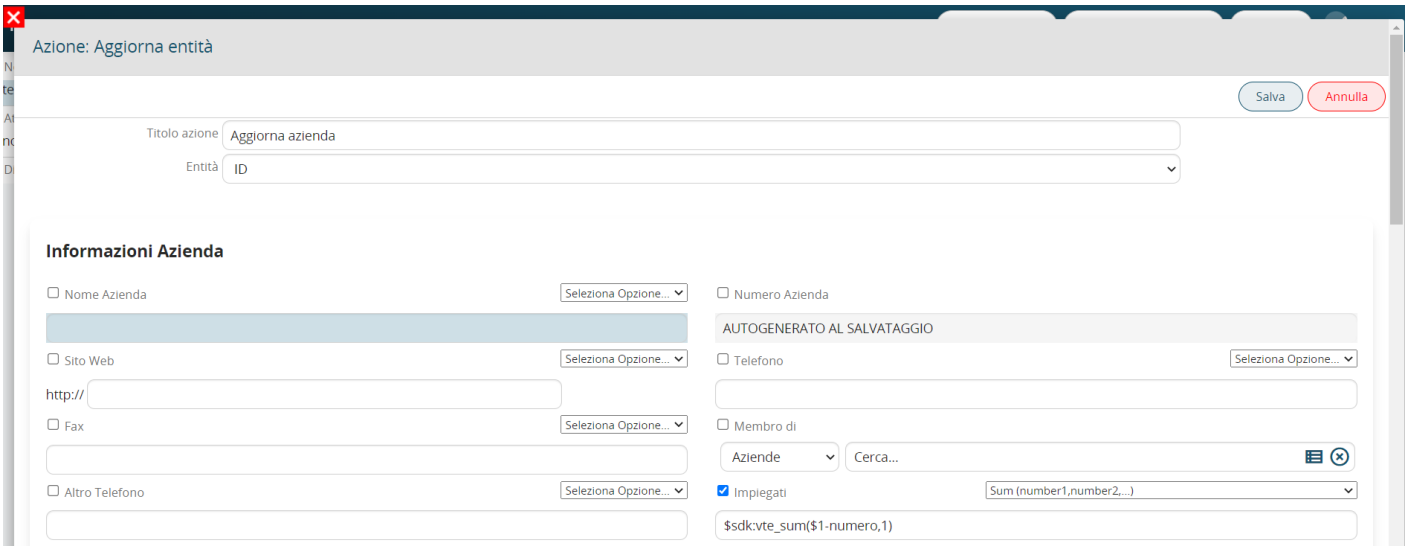
N.B: Tutte le funzioni custom registrate verranno sempre inserite nella sezione “Funzioni SDK”, la sezione “Funzioni data” viene utilizzata esclusivamente per contenere le funzione php disponibili a standard che riguardano I campi data.





Un esempio concreto di funzione SDK di questa tipologia è la funzione “sum()” che restituisce la somma dei parametri (statici o dinamici) passati.
E’ possibile memorizzare il risultato in un campo di una form dinamica di un Process helper inserendo la funzione come valore di default, oppure inserirlo direttamente in campo di un modulo sfruttando le azioni standard di crea o aggiorna entità.





Esempio

Registro una funzione per il calcolo della percentuale che avrà bisogno dei parametri in ingresso percentuale e totale.

```
SDK::setProcessMakerFieldAction('vte_calculate_percentage','modules/SDK/src/ProcessMaker/Utils.php','Calculate percentage (percentage,total)');
```

Nella funzione implementata indicherò quindi i 2 parametri e ritornerò il risultato dell'operazione.

```
function vte_calculate_percentage($percentage, $total) {
    global $engine, $current_process_actionid;

    $value = ($percentage / 100) * $total;
    return $value;
}
```

N.B. All'interno delle funzioni registrate sono disponibili le variabili globali $engine e $current_process_actionid.
Queste contengono rispettivamente l'oggetto contenente tutte le informazioni relative al processo che si sta eseguendo e l'id dell'azione corrente (es. Crea entità, Aggiorna entità, ...)

10.2 SDK Action

E’ una tipologia di funzione che viene utilizzata per creare delle azioni BPMN custom.

È possibile registrare nuove azioni con il seguente metodo:

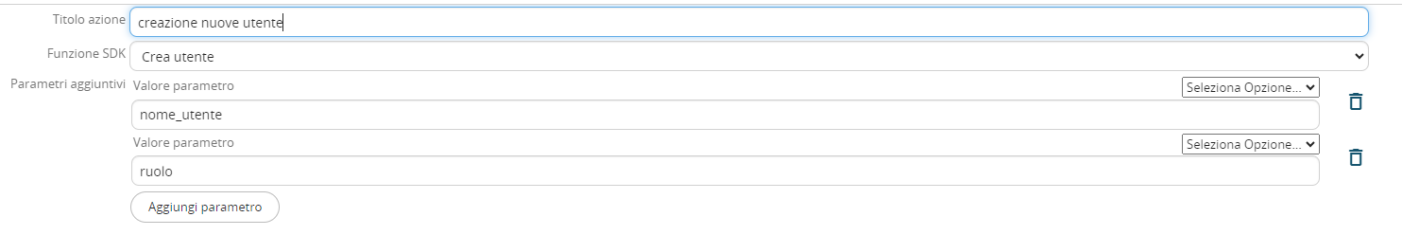
SDK::setProcessMakerAction($func, $src, $label);
$func : nome della funzione
$src : percorso file contenente la funzione
$label : etichetta della funzione

Per de-registrare il file usare:

SDK::unsetProcessMakerAction($func);
$func: nome della funzione

Una volta registrata, si rende disponibile all’utilizzo nella sezione “Azione” nella quale sono presenti tutte le altre azioni BPMN.

Inoltre una volta scelta la funzione da richiamare, se nella funzione gestiamo dei parametri di input abbiamo la possibilità di passarli direttamente da interfaccia:



Esempio

Nel seguente esempio registro una funzione per aggiungere un invitato ad un evento.

```
SDK::setProcessMakerAction('vte_add_event_invitee', 'modules/SDK/src/ProcessMaker/Utils.php', 'Add invitee to event (event, user/contact)');
```

Questa funzione necessita di due parametri: l'id dell'evento e l'id dell'invitato che può essere un utente o un contatto. Per chiarezza indico nel comando di registrazione anche i parametri nella label della funzione.

Nella funzione che andrò a sviluppare vanno aggiuti i parametri $event e $invitee che saranno popolati con i valori passati dall'interfaccia di configurazione dell'azione.
I primi due parametri invece sono fissi e sono:
$engine l'oggetto contenente tutte le informazioni relative al processo che si sta eseguendo
$actionid l'id dell'azione corrente

```
function vte_add_event_invitee($engine, $actionid, $event, $invitee) {
    $engine->log('vte_add_event_invitee', "event:$event invitee:$invitee");

    if (strpos($event,'x') !== false) list(,$event) = explode('x',$event);
    if (strpos($invitee,'x') !== false) list(,$invitee) = explode('x',$invitee);

    $parent_module = getSalesEntityType($invitee);
    if ($parent_module == 'Contacts') {
        $_REQUEST['inviteesid_con'] = $invitee;
    } else {
        $_REQUEST['inviteesid'] = $invitee;
    }
    $focus = CRMEntity::getInstance('Events');
    $focus->retrieve_entity_info_no_html($event,'Events');
    $focus->mode = 'edit';
    $focus->save('Events');
}
```

10.3 SDK Task Condition

E’ una tipologia di funzione che viene utilizzata per eseguire dei controlli custom nelle task di controllo dei processi (conditional task).

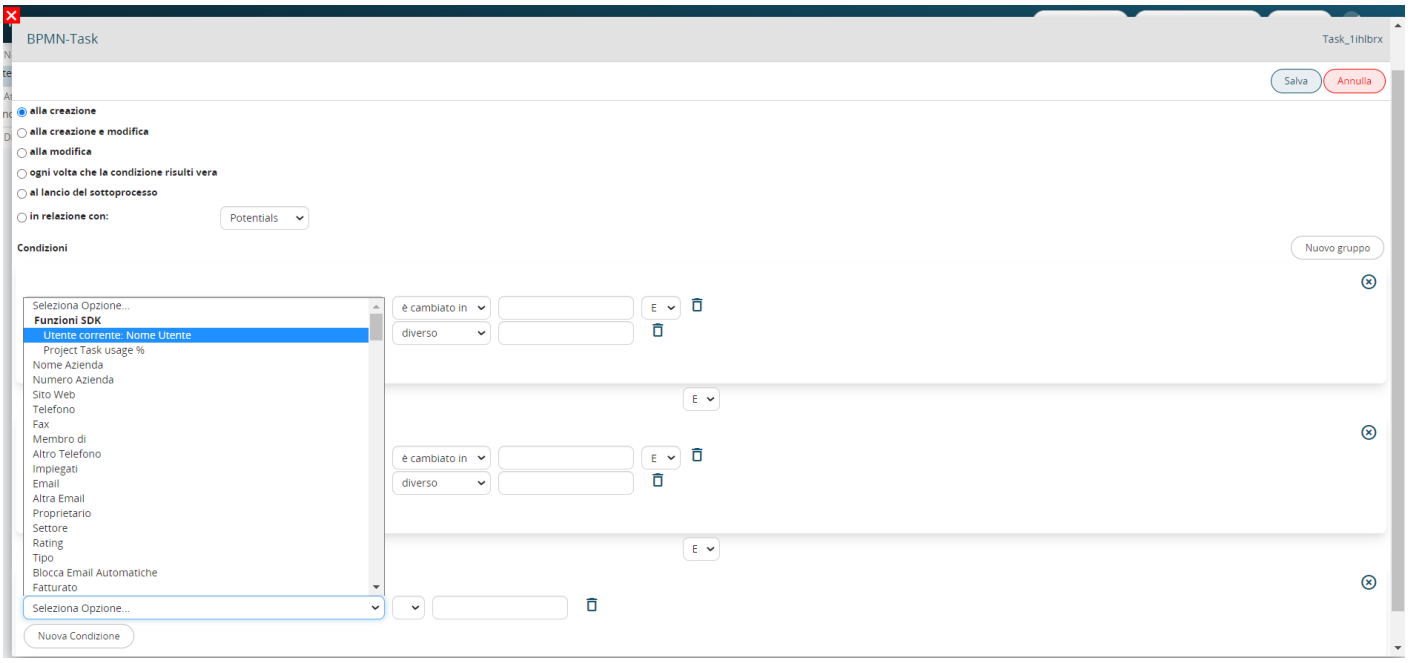
Per registrarle va utilizzato il metodo:

SDK::setProcessMakerTaskCondition($func, $src, $label)
$func : nome della funzione
$src : percorso file contenente la funzione
$label : etichetta della funzione

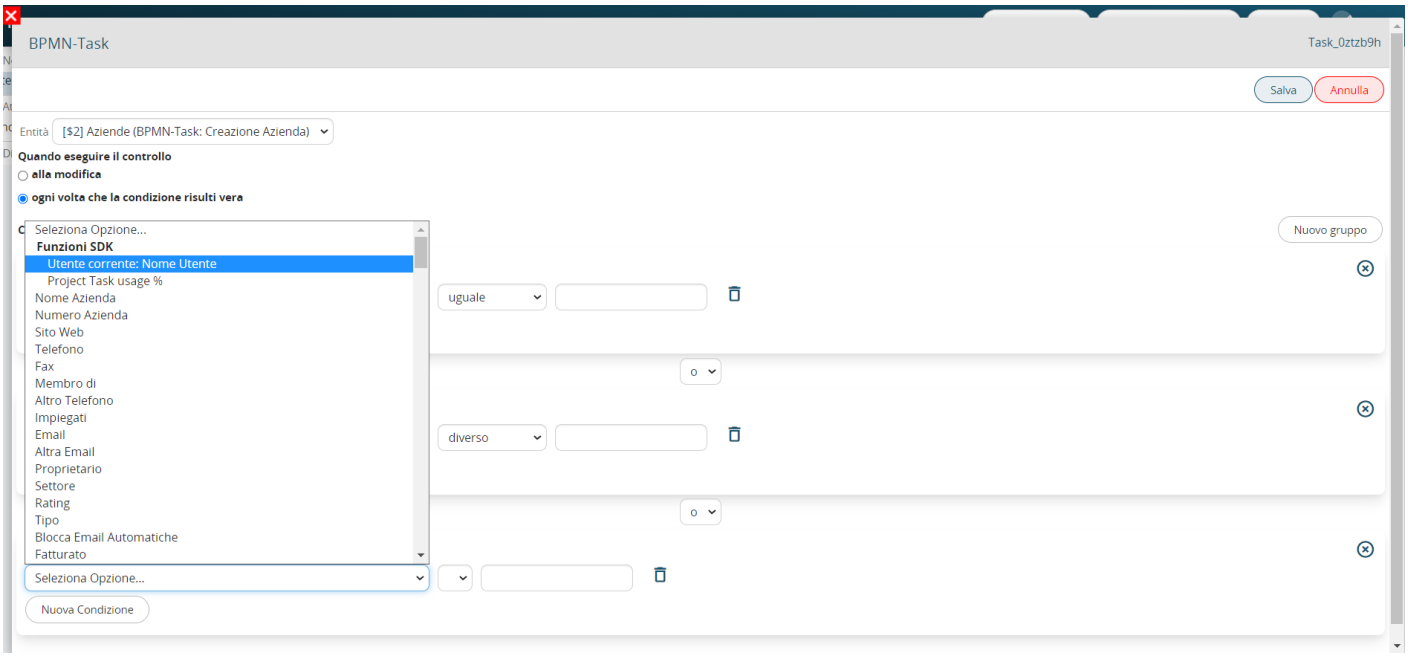
Per de-registrare il file usare:

SDK::unsetProcessMakerTaskCondition($func);
$func : nome della funzione

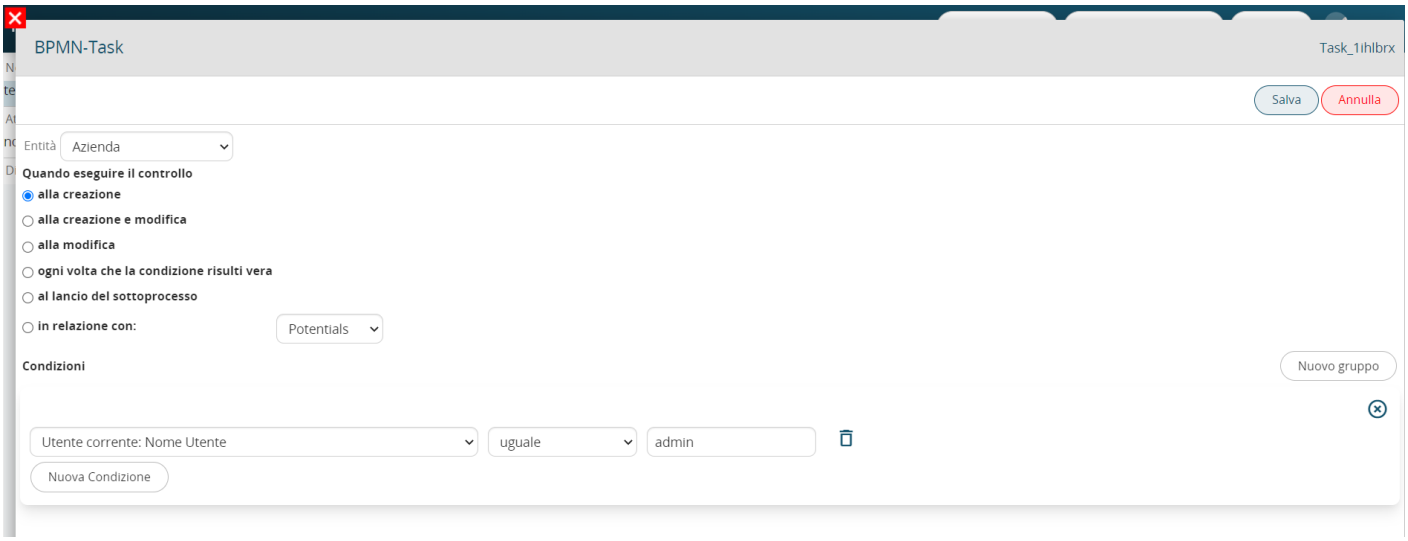
Se richiamate sulla task di “condizione iniziale” (task nella quale vengono definite le regole di partenza del processo), è possibile rendere specifica la partenza del processo in base al risultato restituito.



Se richiamate invece sulle delle semplici task di controllo permettono di far prendere al processo strade differenti in base al risultato restituito.



Le funzioni di questa tipologia saranno sempre disponibili sotto forma di variabili utilizzabili per eseguire controlli.
Un esempio concreto di funzione SDK di questa tipologia è la funzione “get_running_process_current_user” che permette di verificare se il nome dell’utente corrente è uguale o diverso dall’utente definito nella condizione.
Se utilizzata nella condizione iniziale permette di far partire il processo solamente se l’utente corrente è uguale ad un determinato utente.



Esempio

Registro una funzione per verificare se l'indirizzo di fatturazione dell'azienda è uguale quello di spedizione.

```
SDK::setProcessMakerTaskCondition('vte_compare_account_bill_ship_street',
'modules/SDK/src/ProcessMaker/Utils.php', 'Indirizzi di spedizione e fatturazione uguali [e/n]');
```

Nella funzione implementata sono presenti i seguenti parametri.
$module: modulo dell'entità su cui viene eseguita la condizione
$id: identificativo dell'entità in formato ws (es. 3x55126)
$data: array con i valori dei campi del record

La funzione deve restituire una stringa da usare per il confronto lato interfaccia.

```
function vte_compare_account_bill_ship_street($module, $id, $data) {
list($wsModId,$id) = explode('x',$id);// do it every time

if ($data['bill_street'] == $data['ship_street']) {
return 'e';
} else {
return 'n';
}
}
```

A questo punto posso configurare un processo che parte alla creazione di un'azienda quando l'indirizzo di di fatturazione è diverso da quello di spedizione.

Entità Azienda

Quando eseguire il controllo

- (●) **alla creazione**
- ( ) **alla creazione e modifica**
- ( ) **alla modifica**
- ( ) **ogni volta che la condizione risulti vera**
- ( ) **solo che la prima volta che si scatena il processo**
- ( ) **al lancio del sottoprocesso**
- ( ) **in relazione con:** Products

Condizioni

Indirizzi di spedizione e fatturazione uguali [e/n] | uguale | n

Nuova Condizione

10.4 Procedura di registrazione SDK di processo

La procedura descritta qui di seguito è la stessa per le 3 tipologie di funzioni sdk di processo.

Supponiamo di dover creare e registrare una funzione SDK di tipo “FIELD ACTION” che permetta di eseguire la divisione tra più valori.
Una volta scritta, dovrà essere inserita nel file Utils.php disponibile nel percorso modules/SDK/src/ProcessMaker.
Successivamente dovrà essere registrata attraverso il seguente comando da inserire nel file script.php disponibile nel percorso plugins/script :

SDK::setProcessMakerFieldAction('nome funzione','percorso','label funzione');

'nome funzione' → nome della funzione
‘percorso’ → percorso in cui si trova la funzione da registrare
‘label funzione’ → nome con il quale la funzione verrà visualizzata dall’utente nella sezione processi.

Inseriamo quindi il comando nel file script.php e lo compiliamo con I dati dell’sdk:

SDK::setProcessMakerFieldAction('division','modules/SDK/src/ProcessMaker/Utils.php','division(value1,value2,...)');

N.B: Ogni tipologia ha il proprio comando dedicato, quindi nel caso dovessimo caricate un sdk di tipo “ACTION” dovremmo utilizzare il seguente comando:
SDK::setProcessMakerAction('nome funzione','percorso','label funzione');

Nel caso invece dovessimo caricate un sdk di tipo “TASK CONDITION” dovremmo utilizzare il seguente comando:
SDK::setProcessMakerTaskCondition('nome funzione','percorso','label funzione');

Una volta completati questi passaggi, basterà lanciare il file script.php e la funzione verrà correttamente registrata.