

# 11.2 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.

Crea una nuova azione di tipo

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

Titolo azione	<input type="text" value="creazione nuove utente"/>
Funzione SDK	<input type="text" value="Crea utente"/>
Parametri aggiuntivi	Valore parametro <input type="text" value="nome_utente"/> <input type="button" value="Seleziona Opzione..."/>
	Valore parametro <input type="text" value="ruolo"/> <input type="button" value="Seleziona Opzione..."/>
	<input type="button" value="Aggiungi parametro"/>

## 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 aggiunti 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');
}
```

---

Revision #6

Created 3 June 2022 09:56:19

Updated 25 September 2024 09:39:16 by Riccardo