

12 SDK di Processo Standard: Descrizione e Utilizzo

- [12.1 SDK fieldAction: Somma](#)
- [12.2 SDK fieldAction: Differenza](#)
- [12.3 SDK fieldAction: Moltiplicazione](#)
- [12.4 SDK fieldAction: Divisione](#)
- [12.5 SDK fieldAction: vte_json_column_fields](#)
- [12.6 SDK fieldAction: vte_json_string](#)
- [12.7 SDK fieldAction: vte_json_field_string](#)
- [12.8 SDK fieldAction: vte_json_record](#)
- [12.9 SDK fieldAction: Format Date](#)
- [12.10 SDK fieldAction: Now Date](#)
- [12.11 SDK fieldAction: Diff Date](#)

12.1 SDK fieldAction:

Somma

Questa funzione SDK permette di eseguire la somma tra 2 o più valori che vanno passati come parametri.

Essendo una funzione di tipo fieldAction, può essere richiamata all'interno dei singoli campi dei moduli o form dinamiche.

ESEMPIO DI UTILIZZO

Per comprenderne al meglio il funzionamento, qui di seguito viene riportato un esempio di utilizzo della funzione `vte_sum()` per eseguire la somma tra i valori di due campi denominati "Stima Ore Sviluppo" e "Stima Ore Formazione" e salvarne il risultato nel campo "Totale ore stimate" (Figura 1)

The screenshot shows a form with three input fields. The first field is labeled 'Stima ore Sviluppo' and contains the number '4'. The second field is labeled 'Stima ore Formazione' and also contains the number '4'. The third field is labeled 'Totale ore stimate' and contains the number '0'.

Figura 1

All'interno di una Action Task procediamo con la configurazione un'azione di Aggiorna entità coinvolgendo il campo "Totale ore stimate".

Nello specifico, andiamo a richiamare la funzione SDK interessata attraverso la picklist "Selezione Opzione" ed accedendo alla sezione "Funzioni SDK" (Figura 2)

The screenshot shows a configuration interface. On the left, there is a section titled 'Altro' with a checkbox labeled 'Stima ore Sviluppo' and a checkbox labeled 'Totale ore stimate' which is checked. Below the 'Totale ore stimate' checkbox is a text input field containing the code '\$sdk:vte_sum()'. On the right, a dropdown menu is open, showing a list of functions under the heading 'Funzioni SDK'. The function 'Sum (number1,number2,...)' is highlighted in blue. Other functions visible include 'Get column fields in json format (ID,[fieldname1,fieldname2,...])', 'JSON String(text)', 'JSON Field String(id, fieldname)', 'JSON Record(id)', 'Difference (number1,number2)', 'Multiply (number1,number2,...)', and 'Divide (number1,number2)'.

Figura 2

Infine passiamo come parametri (separati dalla virgola) il contenuto dei due campi da sommare (Figura 3)

☒ Totale ore stimate

Seleziona Opzione... ▼

`$sdk:vte_sum($44-cf_e43_1816,$44-cf_e43_1817)`

Figura 3

12.2 SDK fieldAction:

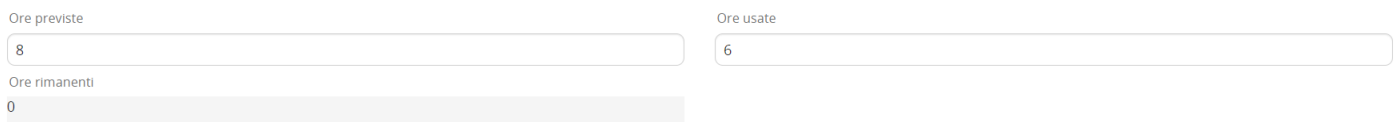
Differenza

Questa funzione SDK permette di eseguire la differenza tra 2 o più valori che vanno passati come parametri.

Essendo una funzione di tipo fieldAction, può essere richiamata all'interno dei singoli campi dei moduli o form dinamiche.

ESEMPIO DI UTILIZZO

Per comprenderne al meglio il funzionamento, qui di seguito viene riportato un esempio di utilizzo della funzione `vte_diff()` per eseguire la differenza tra i valori di due campi denominati "Ore previste" e "Ore usate" e salvarne il risultato nel campo "Ore rimanenti" (Figura 1)

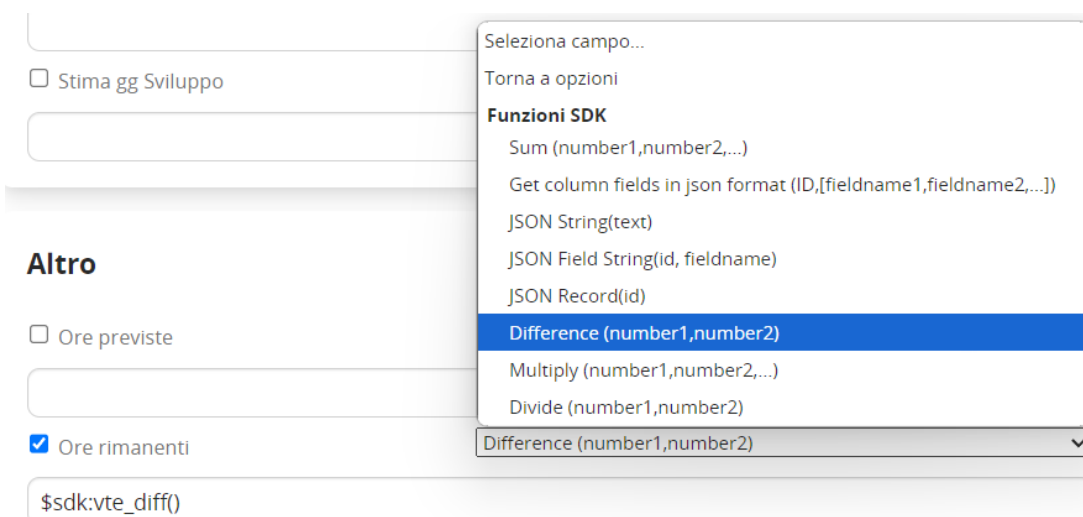


The screenshot shows a form with three input fields. The first field, labeled "Ore previste", contains the value "8". The second field, labeled "Ore usate", contains the value "6". The third field, labeled "Ore rimanenti", contains the value "0".

Figura 1

All'interno di una Action Task procediamo con la configurazione un'azione di Aggiorna entità coinvolgendo il campo "Ore rimanenti".

Nello specifico, andiamo a richiamare la funzione SDK interessata attraverso la picklist "Selezione Opzione" ed accedendo alla sezione "Funzioni SDK" (Figura 2)



The screenshot shows a form with a section titled "Altro" containing two checkboxes: "Stima gg Sviluppo" (unchecked) and "Ore previste" (unchecked). Below these is a text input field. Another checkbox, "Ore rimanenti", is checked. Below it is another text input field containing the formula `$sdk:vte_diff()`. A dropdown menu is open, showing a list of SDK functions. The function "Difference (number1,number2)" is highlighted in blue. The dropdown menu also shows a search bar "Seleziona campo...", a "Torna a opzioni" button, and a list of other functions: "Sum (number1,number2,...)", "Get column fields in json format (ID,[fieldname1,fieldname2,...])", "JSON String(text)", "JSON Field String(id, fieldname)", "JSON Record(id)", "Multiply (number1,number2,...)", and "Divide (number1,number2)".

Figura 2

Infine passiamo come parametri (separati dalla virgola) il contenuto dei due campi da sottrarre (Figura 3)

☒ Ore rimanenti

Ore usate

`$sdk:vte_diff($44-cf_e43_1824,$44-cf_e43_1822)`

Figura 3

12.3 SDK fieldAction:

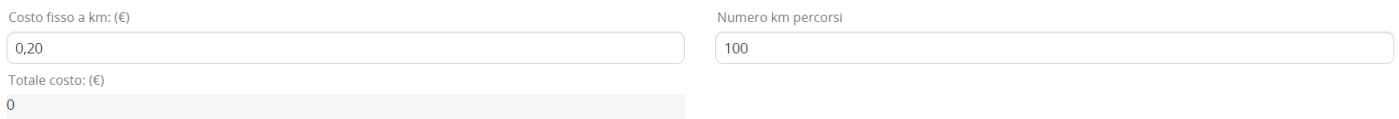
Moltiplicazione

Questa funzione SDK permette di eseguire la moltiplicazione tra 2 o più valori che vanno passati come parametri.

Essendo una funzione di tipo fieldAction, può essere richiamata all'interno dei singoli campi dei moduli o form dinamiche.

ESEMPIO DI UTILIZZO

Per comprenderne al meglio il funzionamento, qui di seguito viene riportato un esempio di utilizzo della funzione `vte_mul()` per eseguire la moltiplicazione tra il contenuto dei due campi denominati "Costo fisso a km: (€)" e "Numero km percorsi" e salvarne il risultato nel campo "Totale costo: (€)" (Figura 1)

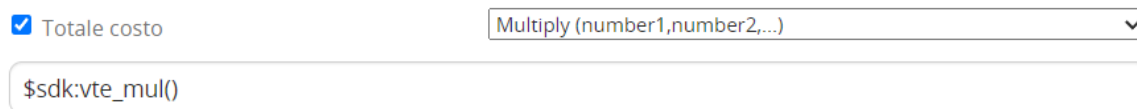


The screenshot shows a form with three input fields. The first field is labeled "Costo fisso a km: (€)" and contains the value "0,20". The second field is labeled "Numero km percorsi" and contains the value "100". The third field is labeled "Totale costo: (€)" and contains the value "0".

Figura 1

All'interno di una Action Task procediamo con la configurazione un'azione di Aggiorna entità coinvolgendo il campo "Totale costo: (€)".

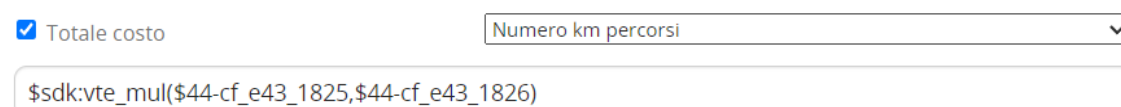
Nello specifico, andiamo a richiamare la funzione SDK interessata attraverso la picklist "Selezione Opzione" ed accedendo alla sezione "Funzioni SDK" (Figura 2)



The screenshot shows a configuration interface for a field action. It features a checkbox labeled "Totale costo" which is checked. To its right is a dropdown menu with the text "Multiply (number1,number2,...)" and a downward arrow. Below these is a text input field containing the code "\$sdk:vte_mul()".

Figura 2

Infine passiamo come parametri (separati dalla virgola) i due valori da moltiplicare (Figura 3)



The screenshot shows the same configuration interface as Figure 2, but with different values. The checkbox "Totale costo" is still checked. The dropdown menu now displays "Numero km percorsi" with a downward arrow. The text input field contains the code "\$sdk:vte_mul(\$44-cf_e43_1825,\$44-cf_e43_1826)".

Figura 3

12.4 SDK fieldAction:

Divisione

Questa funzione SDK permette di eseguire la divisione tra 2 o più valori che vanno passati come parametri.

Essendo una funzione di tipo fieldAction, può essere richiamata all'interno dei singoli campi dei moduli o form dinamiche.

ESEMPIO DI UTILIZZO

Per comprenderne al meglio il funzionamento, qui di seguito viene riportato un esempio di utilizzo della funzione `vte_div()` per eseguire la divisione tra il campo "Stima Ore Sviluppo" e un valore fisso 8 (che rappresentano le ore lavorative in un giorno) e salvarne il risultato nel campo "Stima gg Sviluppo" (Figura 1)

Stima ore Sviluppo

32

Stima gg Sviluppo

0

Figura 1

All'interno di una Action Task procediamo con la configurazione un'azione di Aggiorna entità coinvolgendo il campo "Stima gg Sviluppo".

Nello specifico, andiamo a richiamare la funzione SDK interessata attraverso la picklist "Selezione Opzione" ed accedendo alla sezione "Funzioni SDK" (Figura 2)

☐ Data Mail

(dd-mm-yyyy HH:mm:ss)

Altro

☐ Stima ore Sviluppo

☒ Stima gg Formazione

Seleziona campo...

Torna a opzioni

Funzioni SDK

- Sum (number1,number2,...)
- Get column fields in json format (ID,[fieldname1,fieldname2,...])
- JSON String(text)
- JSON Field String(id, fieldname)
- JSON Record(id)
- Difference (number1,number2)
- Multiply (number1,number2,...)
- Divide (number1,number2)**

Divide (number1,number2) ▼

Figura 2

Infine passiamo come parametri (separati dalla virgola) i due valori da dividere (Figura 3)

☒ Stima gg Formazione

Figura 3

12.5 SDK fieldAction: vte_json_column_fields

Questa funziona SDK permette di generare un codice JSON contenente le label e i valori dei 2 o più campi passati come parametri.

Come primo parametro va passato il crmid del record dal quale si vogliono estrarre le informazioni, invece come parametri successivi andranno passati i "fieldname" dei campi da includere, ovvero i nomi in cui quei campi sono registrati a Data Base.

NOTA BENE: anche se nella descrizione della funzione viene indicato di passare i parametri successivi all'id inserendo delle parentesi quadre, quest'ultime non devono essere inserite.

Viene utilizzata principalmente per formattare un subset di dati al fine di eseguire delle chiamate Web Service di tipo REST (configurabili sempre da processo tramite l'azione standard dedicata "Chiama Web Service Esterno").

Per approfondimenti sulla configurazione di chiamate API da processo si prega di consultare il capitolo 3.15 del manuale dei processi.

ESEMPIO DI UTILIZZO

Per comprenderne al meglio il funzionamento, qui di seguito viene riportato un esempio di utilizzo della funzione SDK `vte_json_column_fields()` per la generazione di un codice JSON contenente i seguenti campi e valori di un'istanza del modulo Assistenza Clienti coinvolta nel processo:

Titolo
Stato

All'interno della form dinamica di un process helper procediamo con la creazione di un campo area di testo denominato "Body JSON Format" nel quale verrà salvato il risultato della funzione. Successivamente andiamo a richiamare la funzione SDK interessata attraverso la picklist "Selezione Opzione" e ad accedere alla sezione "Funzioni SDK" (Figura 1)

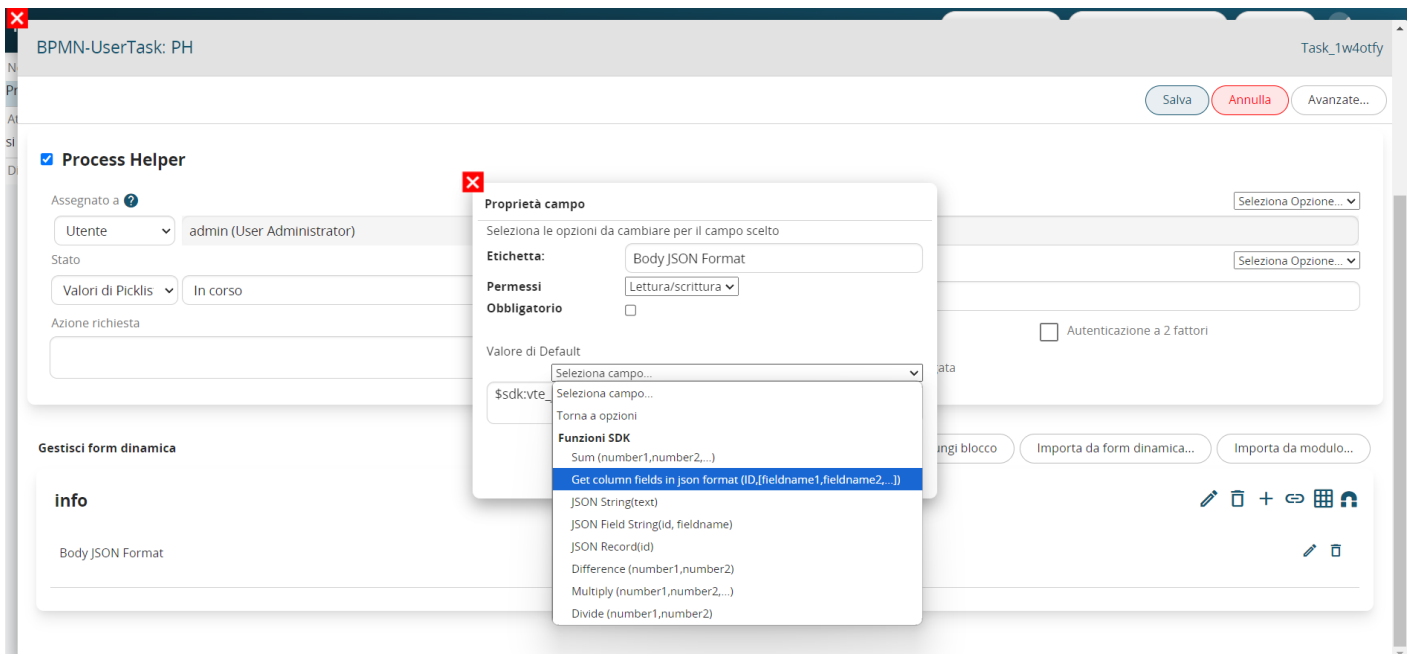


Figura 1

Infine passiamo i parametri richiesti dalla funzione, tutti separati dalla virgola. Nello specifico, come primo parametro inseriamo il crmid del record dal quale si vogliono estrarre le informazioni, quindi nel nostro caso l'id del ticket. (Figura 2)

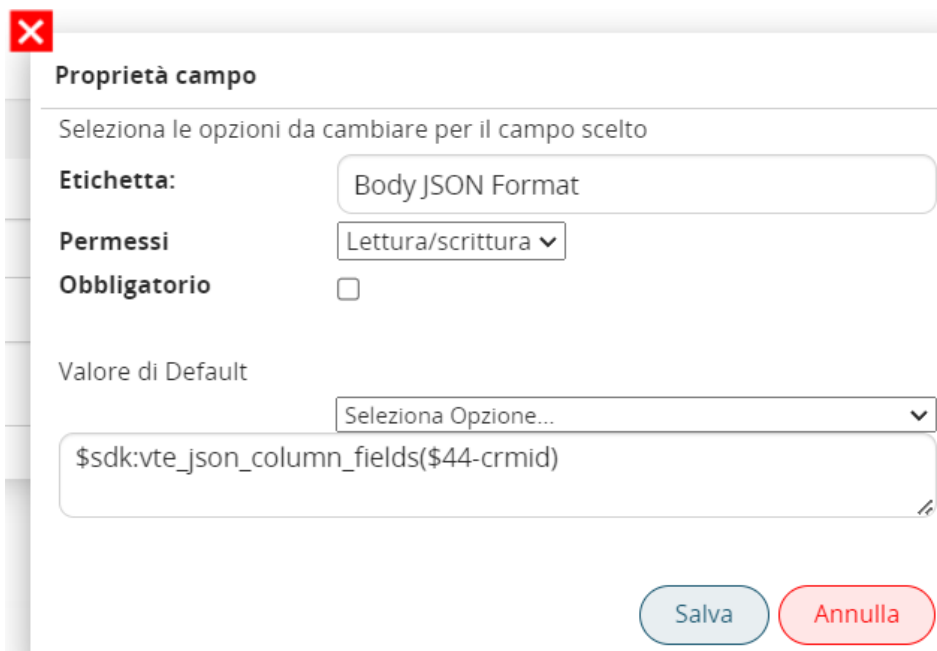
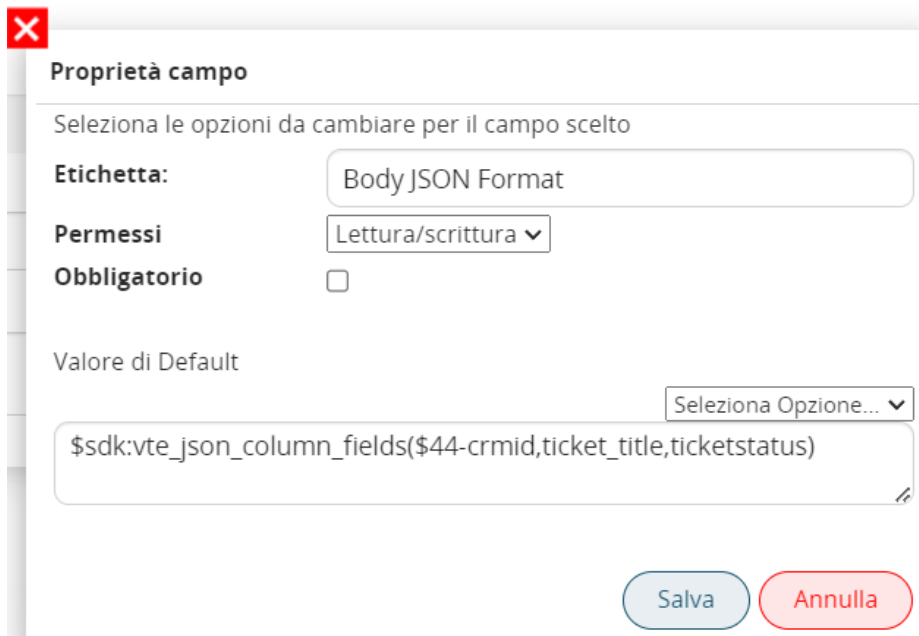


Figura 2

Invece, come parametri successivi, inseriamo i "fieldname" dei campi da coinvolgere, ovvero i nomi con cui quei campi sono registrati a Data Base. Nel nostro caso specifico saranno "ticket_title", "ticketstatus" e "ticketpriorities". (Figura 3)



Proprietà campo

Seleziona le opzioni da cambiare per il campo scelto

Etichetta: Body JSON Format

Permessi Lettura/scrittura ▼

Obbligatorio ☐

Valore di Default

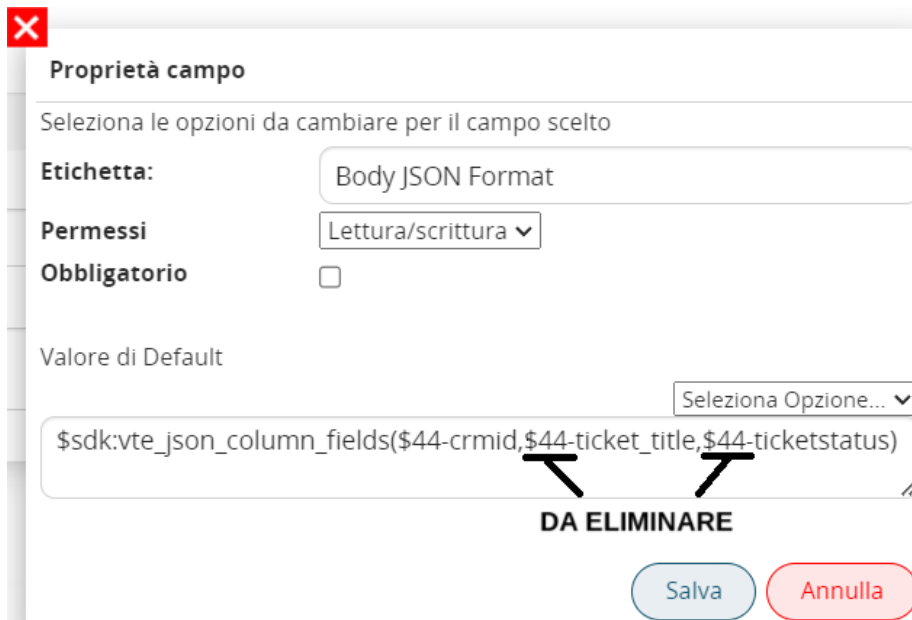
Seleziona Opzione... ▼

`$sdk:vte_json_column_fields($44-crmid,ticket_title,ticketstatus)`

Salva Annulla

Figura 3

Per ricavarli facilmente, sarà sufficiente selezionare il campo relativo dalla picklist "Selezione Opzione" ed eliminare il riferimento all'istanza coinvolta nel processo, ovvero il metaid (Figura 4).



Proprietà campo

Seleziona le opzioni da cambiare per il campo scelto

Etichetta: Body JSON Format

Permessi Lettura/scrittura ▼

Obbligatorio ☐

Valore di Default

Seleziona Opzione... ▼

`$sdk:vte_json_column_fields($44-crmid,$44-ticket_title,$44-ticketstatus)`

DA ELIMINARE

Salva Annulla

Figura 4

Il codice JSON generato sarà come quello mostrato in Figura 5

Body JSON Format

```
{ "ticket_title": "demo_vtenext", "ticketstatus": "Open", "ticketpriorities": "Bassa" }
```

Figura 5

12.6 SDK fieldAction: vte_json_string

Questa funziona SDK permette di convertire una stringa in formato JSON gestendo la compatibilità con i caratteri speciali, in modo da formattarli correttamente ed evitare che entrino in conflitto con esso.

Viene utilizzata principalmente per formattare delle stringhe al fine di eseguire delle chiamate Web Service di tipo REST (configurabili sempre da processo tramite l'azione standard dedicata "Chiama Web Service Esterno") senza riscontrare errori di sintassi sul JSON.

Nello specifico, nel caso in cui nella stringa fossero presenti dei caratteri speciali utilizzati nella sintassi del JSON come le doppie virgolette ("), lo slash (/) e il backslash (\), la funzione andrà ad inserire in automatico ulteriori backslash per evitare che il JSON generi un errore.

Come unico parametro, quindi, passiamo il contenuto di un campo qualsiasi (purché sia una stringa) oppure un semplice valore statico.

ESEMPIO DI UTILIZZO

Per comprenderne al meglio il funzionamento, qui di seguito viene riportato un esempio di utilizzo della funzione SDK `vte_json_string()` per la formattazione del campo "Descrizione" contenente la seguente stringa:

E \ a quel punto / gli dissi: "Addio!" (Figura 1)

Descrizione

E \ a quel punto / gli dissi: "Addio!"

Figura 1

All'interno di una Action Task procediamo con la configurazione un'azione di Aggiorna entità coinvolgendo il campo "Descrizione".

Nello specifico, andiamo a richiamare la funzione SDK interessata attraverso la picklist "Selezione Opzione" ed accedendo alla sezione "Funzioni SDK" (Figura 2)

The image shows a web form with two columns of input fields. Each column has a header row with a checkbox and a label, followed by a text input field and a dropdown menu labeled 'Seleziona Opzione...'. The left column is for 'Fatturazione' (Billing) and the right column is for 'Spedizione' (Shipping). Below the input fields is a section titled 'Informazioni Descrizione' (Description Information) with a checkbox labeled 'Descrizione' (Description) and a text input field containing the code '\$sdk:vte_json_string()'. On the right side, a dropdown menu is open, showing a list of functions under the heading 'Funzioni SDK' (SDK Functions). The functions listed are: 'Sum (number1,number2,...)', 'Get column fields in json format (ID,(fieldname1,fieldname2,...))', 'JSON String(text)', 'JSON Field String(id, fieldname)', 'JSON Record(id)', 'Difference (number1,number2)', 'Multiply (number1,number2,...)', and 'Divide (number1,number2)'. The 'JSON String(text)' option is currently selected.

Figura 2 (cliccare sull'immagine per una risoluzione grafica maggiore)

Infine passiamo come unico parametro il contenuto del campo da formattare (Figura 3)

The image shows a form with a checkbox labeled 'Descrizione' (Description) which is checked. Below the checkbox is a text input field containing the code '\$sdk:vte_json_string(\$41-description)'. The input field has a light blue border and a small shadow.

Figura 3

Il risultato generato dalla funzione sarà come quello mostrato in Figura 4

The image shows a form with a checkbox labeled 'Descrizione' (Description) which is checked. Below the checkbox is a text input field containing the code '\$sdk:vte_json_string(\$41-description)'. The input field has a light blue border and a small shadow.

Figura 4

12.7 SDK fieldAction: vte_json_field_string

E' una funzione analoga a `vte_json_string()`, infatti permette allo stesso modo di convertire una stringa in formato JSON gestendo la compatibilità con in caratteri speciali, in modo da formattarli correttamente ed evitare che entrino in conflitto con esso.

Anch'essa viene utilizzata principalmente per formattare delle stringhe al fine di eseguire delle chiamate Web Service di tipo REST (configurabili sempre da processo tramite l'azione standard dedicata "Chiama Web Service Esterno") senza riscontrare errori di sintassi sul JSON.

Nello specifico, nel caso in cui nella stringa fossero presenti dei caratteri speciali utilizzati nella sintassi del JSON come le doppie virgolette ("), lo slash (/) e il backslash (\), la funzione andrà ad inserire in automatico ulteriori backslash per evitare che il JSON generi un errore.

Per approfondimenti sulla configurazione di chiamate API da processo si prega di consultare il capitolo 3.15 del manuale dei processi.

La vera differenza consiste nei parametri inseribili in input.

Come primo parametro va passato il `crmid` del record dal quale si vogliono estrarre le informazioni, invece come secondo parametro va passato il "fieldname" del campo da includere, ovvero il nome in cui quel campo è registrato a Data Base.

ESEMPIO DI UTILIZZO

Per comprenderne al meglio il funzionamento, qui di seguito viene riportato un esempio di utilizzo della funzione SDK `vte_json_field_string()` per la formattazione in codice JSON del campo "Città (Fatturazione)" di un'istanza del modulo Azienda coinvolta nel processo contenente la seguente stringa:

"Verona" (Figura 1)

Città (Fatturazione)

"Verona"

Figura 1

Il risultato poi verrà salvato all'interno del campo "Descrizione".

All'interno di una Action Task procediamo con la configurazione un'azione di Aggiorna entità coinvolgendo il campo "Descrizione".

Nello specifico, andiamo a richiamare la funzione SDK interessata attraverso la picklist "Selezione Opzione" ed accedendo alla sezione "Funzioni SDK" (Figura 2)

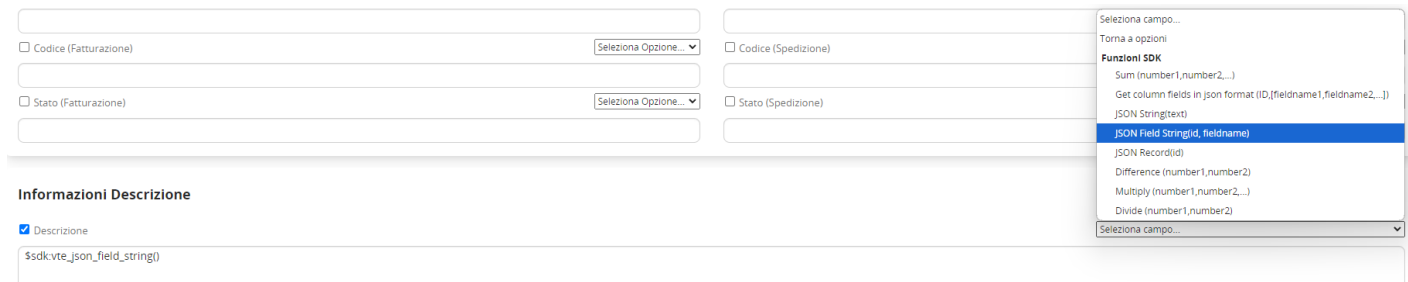


Figura 2 (cliccare sull'immagine per una risoluzione grafica maggiore)

Infine passiamo i parametri richiesti dalla funzione, tutti separati dalla virgola.

Nello specifico, come primo parametro inseriamo il crmid del record dal quale si vogliono estrarre le informazioni, quindi nel nostro caso l'id dell'azienda. (Figura 3)

☒ Descrizione

`$sdk:vte_json_field_string($41-crmid)`

Figura 3

Invece, come secondo parametro, inseriamo il "fieldname" del campo da coinvolgere, ovvero il nome con cui quel campo è registrato a Data Base.

Nel nostro caso specifico sarà "bill_city". (Figura 4)

☒ Descrizione

`$sdk:vte_json_field_string($41-crmid,bill_city)`

Figura 4

Per ricavarlo facilmente, sarà sufficiente selezionare il campo relativo dalla picklist "Selezione Opzione" ed eliminare il riferimento all'istanza coinvolta nel processo, ovvero il metaid (Figura 5).

☒ Descrizione

`$sdk:vte_json_field_string($41-crmid,$41-bill_city)`

T
DA ELIMINARE

Figura 5

Il risultato generato dalla funzione sarà come quello mostrato in Figura 6

Descrizione

\\"Verona\\"

Figura 6

12.8 SDK fieldAction: vte_json_record

Questa funziona SDK permette di generare un codice JSON contenente le label e i valori di tutti i campi di un'istanza di un modulo (record) passata in input.

Come unico parametro va passato il crmid del record dal quale si vogliono estrarre le informazioni da inserire all'interno del codice JSON.

Viene utilizzata principalmente per generare facilmente un codice JSON al fine di eseguire delle chiamate Web Service di tipo REST (configurabili sempre da processo tramite l'azione standard dedicata "Chiama Web Service Esterno").

Per approfondimenti sulla configurazione di chiamate API da processo si prega di consultare il capitolo 3.15 del manuale dei processi.

ESEMPIO DI UTILIZZO

Per comprenderne al meglio il funzionamento, qui di seguito viene riportato un esempio di utilizzo della funzione SDK vte_json_record() per la generazione di un codice JSON contenente tutti i campi e valori di un'istanza del modulo Aziende coinvolta nel processo.

All'interno di una Action Task procediamo con la configurazione un'azione di Aggiorna entità coinvolgendo il campo "Descrizione".

Nello specifico, andiamo a richiamare la funzione SDK interessata attraverso la picklist "Selezione Opzione" ed accedendo alla sezione "Funzioni SDK" (Figura 1)

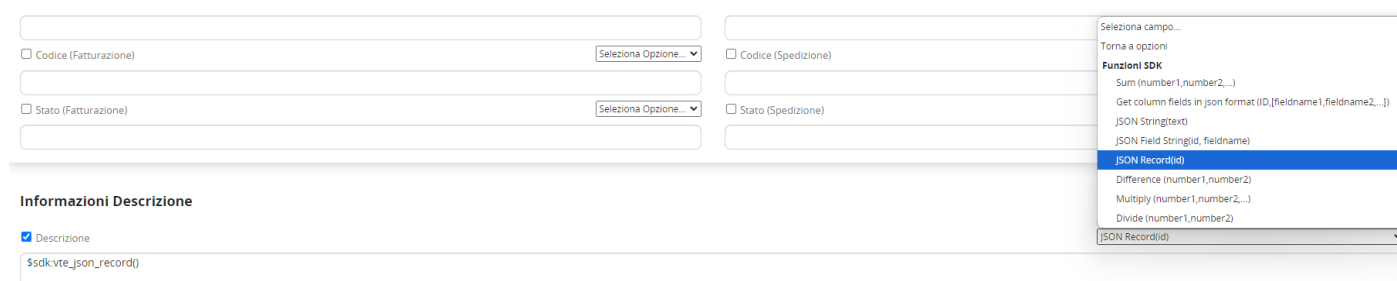


Figura 1 (cliccare sull'immagine per una risoluzione grafica maggiore)

Infine passiamo come unico parametro il crmid del record dal quale si vogliono estrarre le informazioni, in questo caso dell'Azienda (Figura 2)

☒ Descrizione

`$sdk:vte_json_record($41-crmid)`

Figura 2

Il risultato generato dalla funzione sar  come quello mostrato in Figura 3

Descrizione

```
{
  "accountname": "demo_vtenext",
  "account_no": "ACC1374",
  "phone": "",
  "website": "",
  "account_id": "0",
  "email1": "",
  "employees": "0",
  "email2": "",
  "ownership": "",
  "rating": "None",
  "industry": "None",
  "accounttype": "None",
  "annual_revenue": "0",
  "crmv_bankdetails": "",
  "crmv_vat_registration_number": "",
  "crmv_social_security_number": "",
  "external_code": "",
  "emailoptout": "0",
  "newsletter_unsubscrpt": "1",
  "assigned_user_id": "1",
  "createdtime": "2024-10-07 16:17:55",
  "modifiedtime": "2024-10-07 16:17:55",
  "creator": "1",
  "bill_street": "",
  "ship_street": "",
  "bill_city": "",
  "ship_city": "",
  "bill_state": "",
  "ship_state": "",
  "bill_code": "",
  "ship_code": "",
  "bill_country": "",
  "ship_country": "",
  "description": "",
  "campaignrelstatus": "",
  "daily_cost": "0.00",
  "cf_e43_1299": "0.00",
  "contacted": "",
  "mail_view": "",
  "accesscount": "",
  "formato": "0",
  "cf_e43_1354": "Azienda",
  "bu_mcd": "",
  "classe_anagrafica": "Standard",
  "ml2": "0",
  "cf_e43_1682": "0",
  "cf_e43_1697": "Cliente",
  "cf_e43_1815": "0",
  "record_id": "130029",
  "record_module": "Accounts"
}
```

Figura 3 (cliccare sull'immagine per una risoluzione grafica maggiore)

12.9 SDK fieldAction: Format Date

Questa funzione SDK permette di convertire una qualsiasi data passata in input in un formato specifico.

Come primo parametro va passato il valore di un campo di un modulo o form dinamica contenente la data che si vuole formattare, invece come secondo parametro va passato il formato secondo le seguenti sintassi:

giorno / mese / anno(completo) -> d-m-Y

Esempio: 25-10-2024

mese / giorno / anno(completo) -> m-d-Y

Esempio: 10-25-2024

anno(completo) / mese / giorno -> Y-m-d

Esempio: 2024-10-25

anno(completo) / giorno / mese -> Y-d-m

Esempio: 2024-25-10

giorno / mese / anno(parziale) -> d-m-y

Esempio: 25-10-24

mese / giorno / anno(parziale) -> m-d-y

Esempio: 10-25-24

anno(parziale) / mese / giorno -> y-m-d

Esempio: 24-10-25

anno(parziale) / giorno / mese -> y-d-m

Esempio: 24-25-10

giorno / mese / anno(completo) ore / minuti / secondi -> d-m-Y H:i:s

Esempio: 25-10-2024 16:32:20

mese / giorno / anno(completo) ore / minuti / secondi -> m-d-Y H:i:s

Esempio: 10-25-2024 16:32:20

anno(completo) / mese / giorno ore / minuti / secondi -> Y-m-d H:i:s

Esempio: 2024-10-25 16:32:20

anno(completo) / giorno / mese ore / minuti / secondi -> Y-d-m H:i:s

Esempio: 2024-25-10 16:32:20

ore / minuti / secondi -> H:i:s

Esempio: 16:32:20

solo giorno -> d

Esempio: 25

solo mese -> m

Esempio: 10

solo anno(completo) -> Y

Esempio: 2024

solo anno(parziale) -> y

Esempio: 24

solo ore -> H

Esempio: 16

solo minuti -> i

Esempio: 32

solo secondi -> s

Esempio: 20

N.B: un risultato parziale non sarà gestibile all'interno di campi data o datetime, questo perché accettano solamente valori aventi almeno un giorno, mese e anno.

ESEMPIO DI UTILIZZO

Per comprenderne al meglio il funzionamento, qui di seguito viene riportato un esempio di utilizzo della funzione SDK `formatDate()` per la formattazione del valore del campo "Periodo Creazione" di un'istanza del modulo Aziende e il suo salvataggio all'interno di un campo di un process helper.

All'interno della form dinamica procediamo con la creazione di un campo data denominato "Data creazione" nel quale verrà salvato il risultato della funzione.

Successivamente andiamo a richiamare la funzione SDK interessata attraverso la picklist "Selezione Opzione" e ad accedere alla sezione "Funzioni data" (Figura 1)

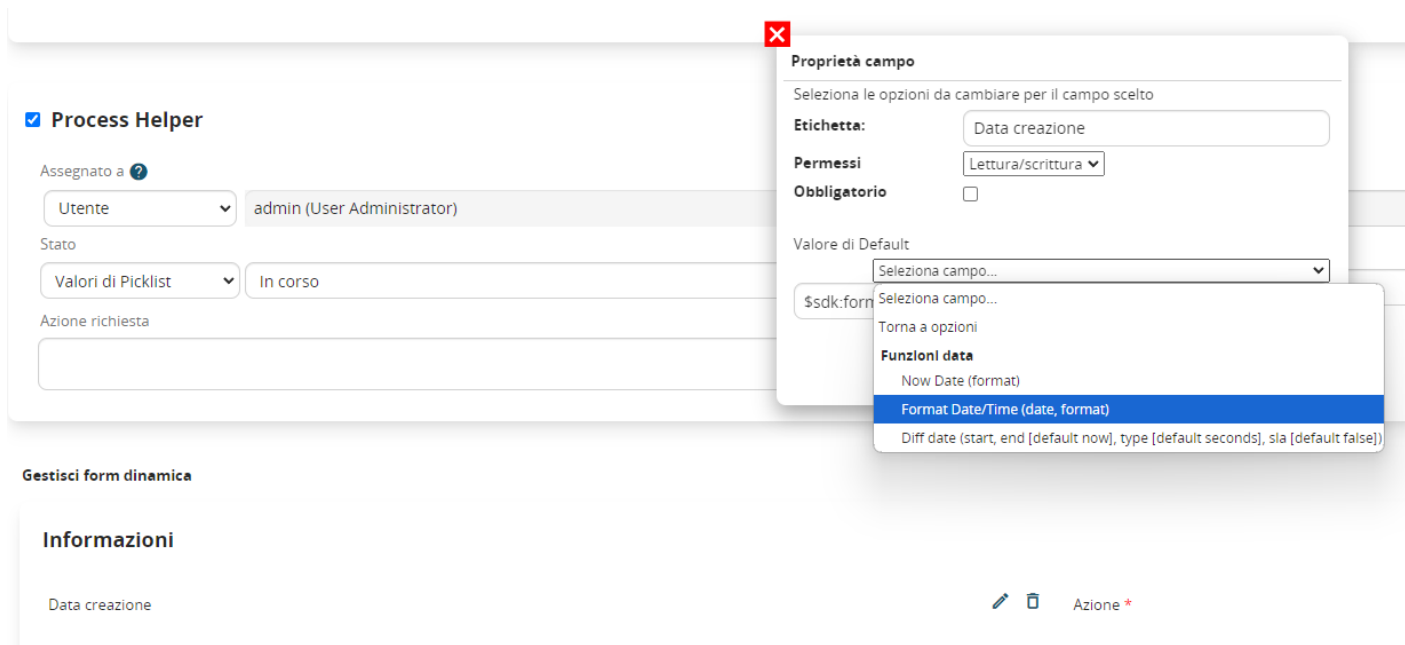


Figura 1

Infine passiamo i parametri richiesti dalla funzione, tutti separati dalla virgola. Nello specifico, come primo parametro inseriamo valore del campo "Periodo Creazione" (Figura 2)

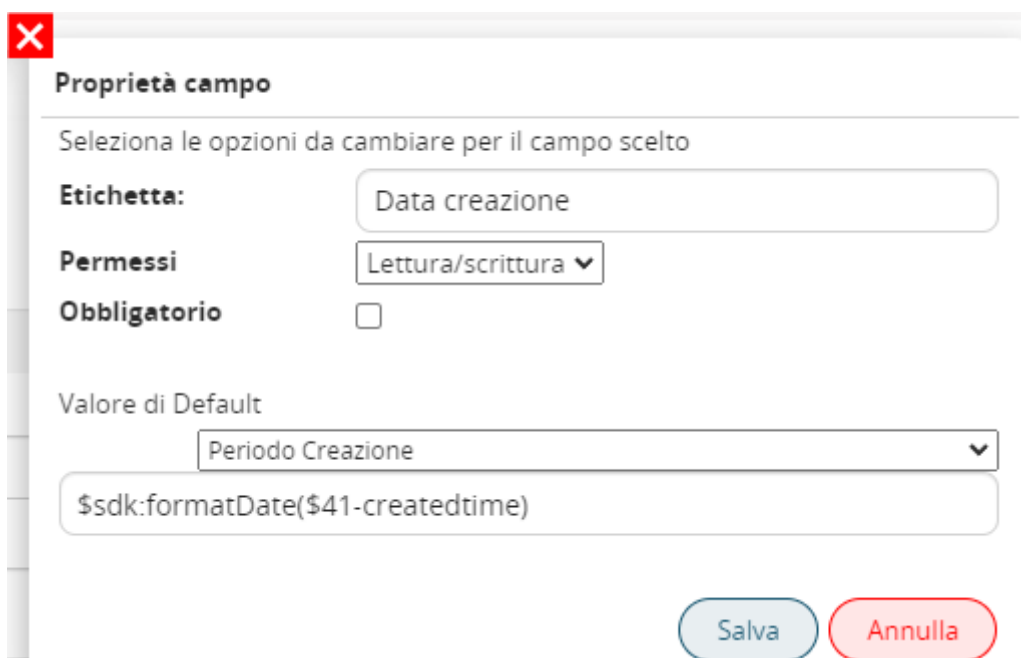



Figura 2

Invece, come secondo parametro, passiamo il formato con il quale la funzione deve restituire la data, quindi nel nostro caso d-m-Y (Figura 3)



Proprietà campo

Seleziona le opzioni da cambiare per il campo scelto

Etichetta:

Permessi


Obbligatorio ☐

Valore di Default

Figura 3

Il risultato sarà come quello mostrato in figura 4

Data creazione



(dd-mm-yyyy)

Figura 4

12.10 SDK fieldAction: Now Date

Questa funzione SDK permette di ottenere e salvare la data odierna all'interno di un qualsiasi campo di tipo testo, area di testo, data e datetime.

Di default (quindi senza passare alcun parametro), il risultato verrà sempre restituito nel formato "d-m-Y", per ottenere invece la data in un formato differente andrà passato il formato come unico parametro della funzione, per visionare nel dettaglio la sintassi di ogni casistica disponibile consultare il capitolo ... del manuale dei processi.

ESEMPIO DI UTILIZZO

Per comprenderne al meglio il funzionamento, qui di seguito viene riportato un esempio di utilizzo della funzione SDK `date_now()` per il salvataggio della data odierna all'interno di un campo di un process helper.

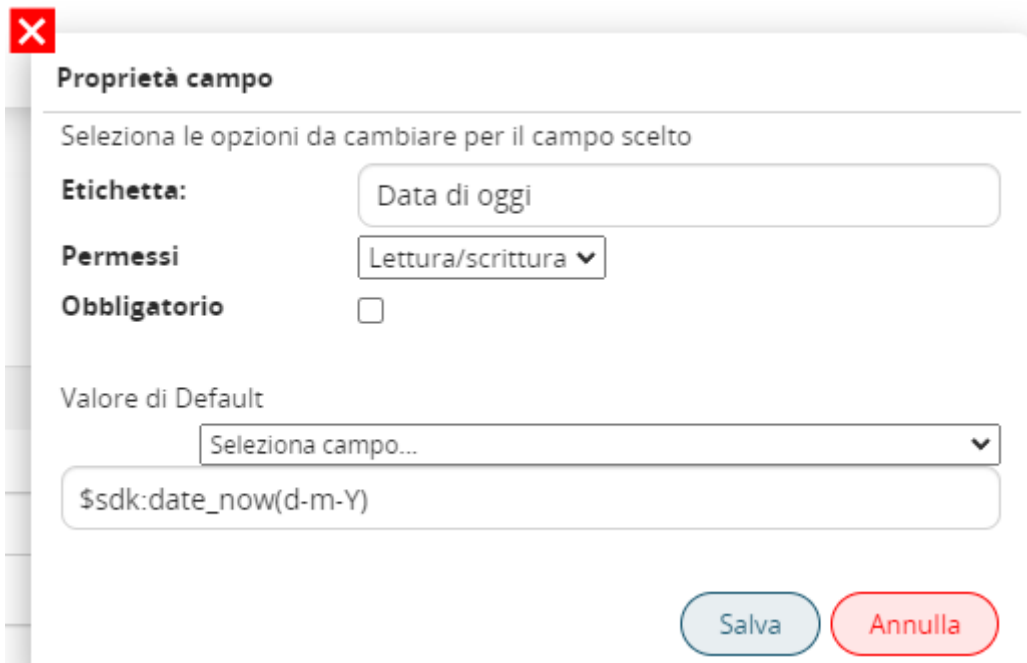
All'interno della form dinamica procediamo con la creazione di un campo data denominato "Data di oggi" nel quale verrà salvato il risultato della funzione.

Successivamente andiamo a richiamare la funzione SDK interessata attraverso la picklist "Selezione Opzione" e ad accedere alla sezione "Funzioni data" (Figura 1)

The screenshot displays the 'Process Helper' configuration interface. On the left, the 'Process Helper' section is active, showing fields for 'Assegnato a' (set to 'admin (User Administrator)'), 'Stato' (set to 'In corso'), and 'Valori di Picklist'. Below these is an 'Azione richiesta' field. On the right, a 'Proprietà campo' dialog box is open for the field 'Data di oggi'. The dialog includes fields for 'Etichetta' (set to 'Data di oggi'), 'Permessi' (set to 'Lettura/scrittura'), and 'Obbligatorio' (unchecked). Below these are 'Valore di Default' and '\$sdk:date' dropdowns, both set to 'Seleziona campo...'. A 'Torna a opzioni' button is also present. At the bottom of the dialog, the 'Funzioni data' section is expanded, showing options: 'Now Date (format)' (highlighted), 'Format Date/Time (date, format)', and 'Diff date (start, end [default now], type [default seconds], sla [default false])'. At the bottom of the main interface, the 'Gestisci form dinamica' section is visible, with an 'Informazioni' tab showing the field 'Data di oggi' and an 'Azione' button.

Figura 1

Infine, se necessario, passiamo il formato con il quale deve essere restituita la data (Figura 2)



Proprietà campo

Seleziona le opzioni da cambiare per il campo scelto

Etichetta: Data di oggi

Permessi Lettura/scrittura ▼

Obbligatorio ☐

Valore di Default

Seleziona campo... ▼

\$sdk:date_now(d-m-Y)

Salva Annulla

Figura 2

Il risultato sarà come quello mostrato in figura 3

Data di oggi

07-10-2024 📅

(dd-mm-yyyy)

Figura 3

12.11 SDK fieldAction: Diff Date

Questa funziona SDK permette di eseguire la differenza tra due date passate come parametri in input.

Come primi due parametri, separati dalla virgola, andranno passati i valori dei due campi di tipo data da coinvolgere nella differenza, tenendo però in considerazione che la funzione esegue il seguente calcolo:

$\text{parametro2} - \text{parametro1}$

Il secondo parametro data non è obbligatorio, quindi nel caso in cui si passasse solamente un valore data (parametro1), il calcolo verrà eseguito prendendo in automatico come secondo parametro (parametro2) la data-ora odierna.

Di default (quindi senza passare un terzo parametro), la funzione restituirà sempre il risultato del calcolo in secondi.

Invece, per ottenere il risultato in giorni, si dovrà passare la stringa fissa "days".

ATTENZIONE! di default la funzione restituirà sempre e solo risultati positivi, per abilitare la restituzione di risultati negativi di dovrà applicare una modifica al codice php.

ESEMPIO

Per comprenderne al meglio il funzionamento, qui di seguito viene riportato un esempio di utilizzo della funzione SDK diffDate() per eseguire il calcolo della differenza tra i valori di due campi denominati "Data Inizio" e "Data Fine" e salvarne il risultato (in giorni) nel campo "Intervallo Temporale" (Figura 1)

Data Inizio

01-10-2024



(dd-mm-yyyy)

Data Fine

31-10-2024



(dd-mm-yyyy)

Figura 1

All'interno di una Action Task procediamo con la configurazione un'azione di Aggiorna entità coinvolgendo il campo "Intervallo Temporale".

Nello specifico, andiamo a richiamare la funzione SDK interessata attraverso la picklist "Selezione Opzione" ed accedendo alla sezione "Funzioni data" (Figura 2)

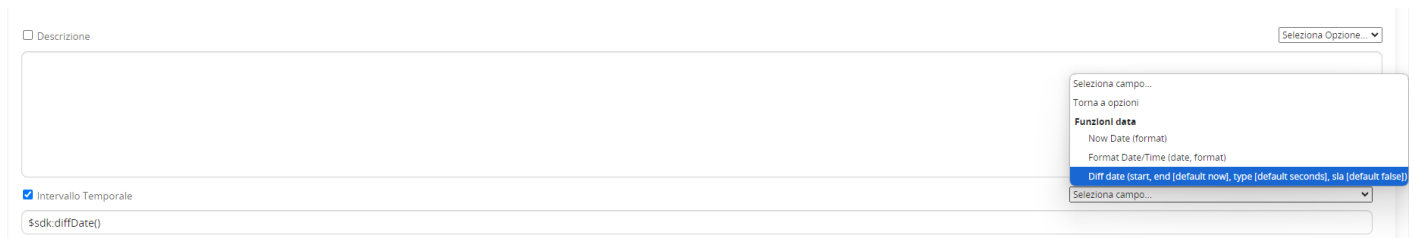


Figura 2 (cliccare sull'immagine per una risoluzione grafica maggiore)

N.B: nella label della funzione selezionabile nella picklist "Seleziona opzione.." viene indicata la possibilità di passare un ulteriore parametro denominato "sla" che di default viene impostato a false, di fatto si tratta solamente di un refuso, pertanto risulta essere un parametro non gestibile.

Come primo e secondo parametro (separati dalla virgola) passiamo i valori dei campi "Data Inizio" e "Data Fine" nell'ordine appena citato (Figura 3)

☒ Intervallo Temporale

`$sdk:diffDate($41-cf_e43_1816,$41-cf_e43_1817)`

Figura 3

Infine come terzo parametro passiamo la stringa fissa "days" (Figura 4)

☒ Intervallo Temporale

`$sdk:diffDate($41-cf_e43_1816,$41-cf_e43_1817,days)`

Figura 4

Il risultato sarà come quello mostrato in figura 5

Intervallo Temporale

30

Figura 5

