راهنمای استفاده از Occurrences System .

شرح راهکار :

اگر نیاز دارید که ارتباط سیستم های دیگر با CRM شما بصورت کاملا پویا باشد و منطق آن سمت CRM کنترل شود این راهکار برای سازمان شما سودمند خواهد بود. متدهای نوع ....Post,Put,Delete همگی می توانند سمت CRM توسط Post,Put,Delete, توسعه یا تغییر در API وجود کود که نیاز به توسعه یا تغییر در API وجود نیاز به توسعه یا تغییر در API وجود نداشته باشد.

مثال :

۱-نیاز است یک متد سمت CRM فراخوانی گردد که شناسه ی مشتری را جستجو نموده (فیلد یکتا) سپس اطلاعات خاصی را آیدیت نماید

۲-نیاز است یک Contact سمت CRM با دیتای مشخصی ساخته شود.

۳-نیاز است یک Account سمت CRM ساخته شود یا تغییر کند.

تمام این موارد نیاز دارد که برای هرکدام یک متد جداگانه توسعه داده شود و یک توسعه دهنده با صرف زمان مشخصی این فعالیت را انجام دهد. حال فرض کنید فقط <mark>یک</mark> متد پویا توسعه داده شود اما تمام این بیزینس ها سمت CRM بصورت Action توسعه داده شود و توسعه دهنده ی CRM کل منطق را در این اکشن ها کنترل نماید.

Occurrences System دقیقا همین کار را برای شما انجام می دهد. یک نوع تعریف کنید و از آن رکورد بسازید این رکورد ها عملیاتی هستند که انجام می شوند و لاگ شما را نگهداری می کنند.

# توسعه ی فنی :

این راهکار از دو موجودیت و یک پلاگین تشکیل شده است که کنترل اصلی سیستم برعهده ی این پلاگین می باشد.

:OccurrenceType

هر رکورد از این موجودیت نوع بیزینس و منطق را تعریف می کند مثلا ساخت مشتری ، تغییر در مشخصات یک Order. در واقع اکشن مورد نظر در این رکورد تعریف می شود و متد پویا رکورد دیگری از این نوع می سازد.

| ew Occurrence           |     |                              | Owner & Mohammad Hasan<br>Arvinfar |
|-------------------------|-----|------------------------------|------------------------------------|
| General                 |     |                              |                                    |
| General                 |     |                              |                                    |
| Name *                  | 0   | Type Id *                    |                                    |
| Trigger Action *        |     | Completion<br>Action(Global) |                                    |
| Enable on deactive mode | No  |                              |                                    |
| Description             |     |                              |                                    |
| Extended data           |     |                              |                                    |
| Enable Target           | Yes |                              |                                    |
| Target Entity *         |     |                              |                                    |
| Target Field *          |     |                              |                                    |
|                         |     |                              |                                    |

| Name                    | نام منطق یا بیزینسی که تعریف می کنید                        |
|-------------------------|---|
| Typeld                  | یک کد منحصر بفرد که باید توسط متد پویا فراخوانی گردد        |
| Trigger Action          | اکشنی که بیزینس و منطق را کنترل می کند                      |
| Enable on deactive mode | درصورت غيرفعال بودن Type ،قابل استفاده باشد                 |
| Description             | توضيحات   |
| Completion              | اکشن Global که پس از اجرای موفق TriggerAction اجرا خواهد شد |
| Enable Target           | اگر Yes باشد یعنی TriggerAction دارای یک Target می باشد     |

| Target Entity | موجودیتی که اکشن برروی آن اجرا می شود                 |
|---------------|---|
| Target Field  | فیلد مربوط به موجودیت که اکشن باید برروی آن اجرا گردد |

### موجودیت Occurrence:

این موجودیت فرامین را براساس Type مشخص شده اجرا می کند و پس از انجام موفق لاگ را ثبت می نماید. در واقع متد پویا باید یک رکورد از نوع این موجودیت و یک فیلد خاص با دیتای مورد نظر ایجاد نماید. ساخت موفقیت آمیز این رکورد و کد وضعیت Done به معنای اجرا موفق دستورات می باشد.

در زیر نمونه هایی ذکر شده است که مسیر اجرای این فرایند را بطور کامل و شفاف توضیح می دهد :

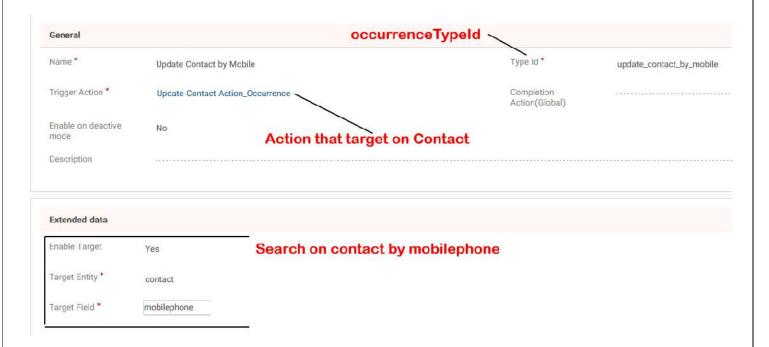
#### مثال اول :

فرض کنید نیاز به آپدیت اطلاعات یک Contact می باشد بطوریکه یک مشتری حقیقی با شماره موبایل 09112320258 را جستجو کند و اطلاعات نام، نام خانوادگی، نام پدر، جنسیت و شرکت را تغییر دهد. در نتیجه نیاز به تعریف یک Json می باشد که ورودی ها را مشخص نماییم.

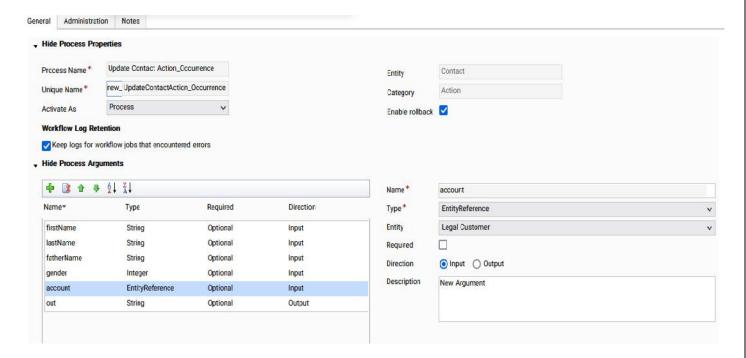
```
{
    "occurrenceTypeId":"update_contact_by_mobile",
    "targetValue":"09112320258",
    "firstName":"Alice",
    "lastName":"Portman",
    "gender":0,
    "account":"6A534348-6FBE-E811-80DB-005056B6C839"
}
```

در Json بالا Typeld مربوط به OccurrenceType را مشخص نموده و یک شناسه ی یکتا بعنوان Target تعریف می کنیم یا تغییرات برروی آن رکورد انجام شود. مابقی اطلاعات نیز بر اساس نیاز در Json اضافه می شود.

## : OccurrenceType



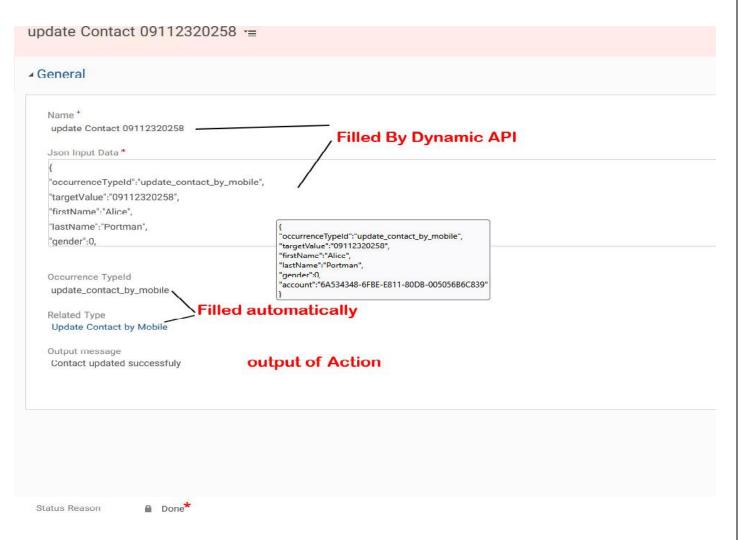
### اکشن در OccurrenceType به صورت زیر می باشد :



پارامترهای اکشن باید دقیقا هم نام با ورودی های Json باشد و بهتر است پارامترهای اکشن از نوع Optional باشد. برای اکشن می توان خروجی نیز تعریف کرد تا پس از اجرای موفق مقدار خروجی نیز لاگ شود. پس از تعریف OccurrenceType باید یک رکورد Occurrence تعریف کنیم و Json تعریف شده را در فیلد new\_jsoninputdata قرار دهیم. (این بخش باید توسط متد پویا ایجاد گردد) به محض ایجاد این رکورد Plugin کنترل کننده اجرا می شود و OccurrenceTypeld را از Json استخراج نموده و Action مورد نظر را اجرا می کند و پس از انجام موفق پروسه خروجی مورد نظر را در فیلد new\_outputmessage قرار داده و وضعیت رکورد را در حالت Done قرار می دهد. چنانچه در OccurrenceTypel فیلد CompletionAction تعریف شده باشد پس از اجرای موفق پروسه آن اکشن نیز اجرا شده و فرایند دلخواه را انجام

می دهد.

نکته : چنانچه در ورودی های اکشن پارامتری با نام self تعریف شود Guid مربوط رکورد Occurrence تولید شده در این پارامتر قرار می گیرد تا در صورت نیاز بتوان برای ارتباط بین رکوردها از آن استفاده شود.



```
مثال دوم :
```

نیاز به ساخت یک Contact با مقادیر دلخواه می باشد.

ابتدا Json مورد نظر را پیاده سازی می کنیم. مشخص است که در این منطق نیازی به فیلد Target نداریم و اکشن مورد نظر نیز باید Global باشد.

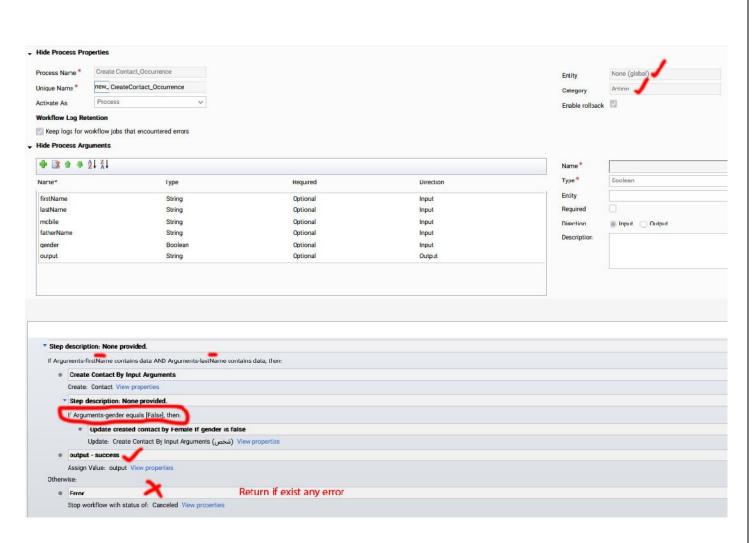
```
{
    "occurrenceTypeId":"create_contact",
    "firstName":"David",
    "lastName":"Simon",
    "fatherName":"Jack",
    "gender":false,
    "mobile":"09112320258"
```

| General                 |                           |                              |                |
|-------------------------|---------------------------|------------------------------|----------------|
| Name *                  | Create Contact            | Type Id *                    | create_contact |
| Trigger Action *        | Create Contact_Occurrence | Completion<br>Action(Global) |                |
| Enable on deactive mode | No                        |                              |                |
| Description             |                           |                              |                |

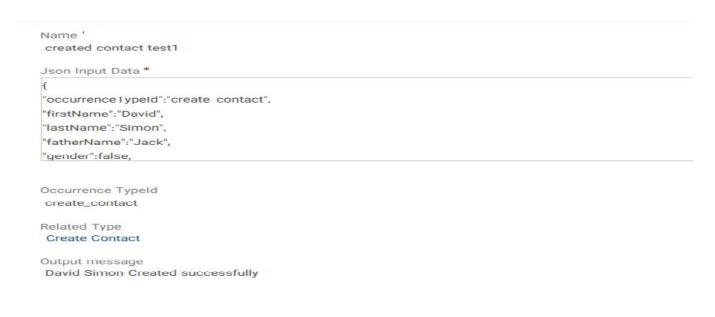
| Extended data | Action is global and does not require to Target |
|---------------|---|
| Enable Target | No  |

ساخت Occurrence Type با یک ID ساخت

تعریف اکشن مورد نظر



اکنون باید Json ساخته شده را به پارامتر متد پویا منتقل کنیم تا رکورد Occurrence را ایجاد کند و پس از ساخت رکورد Occurrence مشتری نیز ساخته می شود.



این سیستم هر Data Type استانداردی که در CRM وجود داشته باشد را پشتیبانی نموده و حتی در مواردی Cast مورد نیاز را با در نظر گرفتن DataType مقصد(اکشن) انجام می دهد. مثلا در Json زیر چنانچه نوع داده ی پارامترهای اکشن از نوع حاده ی پارامترهای اکشن از نوع داده ی پارامترهای اکشن از نوع داده بطور خودکار انجام می نوع داده بطور خودکار انجام می شود و درصد خطا کاهش می یابد.

```
"occurrenceTypeId":"xxxxxx",
"firstName":"Tim",
"lastName":"Marson",
"birthDate":"1988-02-16T12:51:07.397Z",
"deposit":"50000000",
"gender":7
}
```

{

تبدیل String birthdate به String deposit تبدیل Money به String deposit تبدیل Boolean True به int gender

در صورت نیاز به توضیحات بیشتر و یا پیشنهاد جهت بهبود این سیستم از طریق زیر در دسترس هستم

Mobile: +989112320258

Email: aria.rvn@gmail.com

LinkedIn: https://linkedin.com/in/ariarvn

GitHub: https://github.com/marvinfar