

راهنمای استفاده از Occurrences System :

شرح راهکار :

اگر نیاز دارید که ارتباط سیستم های دیگر با CRM شما بصورت کاملاً 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 CRM Developer توسعه داده شود و تغییر ورودی و خروجی سمت داخل رخ دهد و 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 آپدیت 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 یک 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 دقیقاً 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.

موجودیت 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.

OCURRENCE TYPE : INFORMATION

New Occurrence Type

Owner\*  
Mohammad Hasan  
Arvinfar

General

General

Name\*

Type Id\*

Trigger Action\*

Completion  
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	در صورت غیرفعال بودن Type ، قابل استفاده باشد
Description	توضیحات
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 بر روی آن اجرا می شود
Target Field	فیلد مربوط به موجودیت که اکشن باید بر روی 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 بر روی آن رکورد انجام شود. مابقی اطلاعات نیز بر اساس نیاز در Json اضافه می شود.

تعریف OccurrenceType :

**General**

**occurrenceTypeId**

Name \*

Update Contact by Mobile

Type Id \*

update\_contact\_by\_mobile

Trigger Action \*

Update Contact Action\_Occurrence

Completion Action(Global)

Enable on deactive mode

No

Description

**Action that target on Contact**

**Extended data**

Enable Target

Yes

Target Entity \*

contact

Target Field \*

mobilephone

**Search on contact by mobilephone**

اکشن در OccurrenceType به صورت زیر می باشد :

General

Administration

Notes

**Hide Process Properties**

Process Name \*

Update Contact: Action\_Occurrence

Entity

Contact

Unique Name \*

new\_UpdateContactAction\_Occurrence

Category

Action

Activate As

Process

Enable rollback

☒

**Workflow Log Retention**

☒ Keep logs for workflow jobs that encountered errors

**Hide Process Arguments**

+

✖

↑

↓

A↕Z

Z↕A

Name	Type	Required	Direction
firstName	String	Optional	Input
lastName	String	Optional	Input
fatherName	String	Optional	Input
gender	Integer	Optional	Input
account	EntityReference	Optional	Input
out	String	Optional	Output

Name \*

account

Type \*

EntityReference

Entity

Legal Customer

Required

☐

Direction

☒ Input ☐ Output

Description

New Argument

پارامترهای اکشن باید دقیقاً 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 occurrenceTypeId 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 OccurrenceType 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.

update Contact 09112320258

General

Name +  
update Contact 09112320258

Json Input Data \*

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

Occurrence TypeId  
update\_contact\_by\_mobile

Related Type  
Update Contact by Mobile

Output message  
Contact updated successfully

Status Reason Done

Filled By Dynamic API

Filled automatically

output of Action

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 Action(Global)	.....
Enable on deactive mode	No		
Description	.....		

Extended data	
Action is global and does not require to Target	
Enable Target	<input type="text" value="No"/>
<small>Enable Target-If this event should happen on a particular record.</small>	

ساخت Occurrence Type با یک ID یکتا

تعریف اکشن مورد نظر

**Hide Process Properties**

Process Name \*

Unique Name \*

Activate As

**Workflow Log Retention**

☒ Keep logs for workflow jobs that encountered errors

**Hide Process Arguments**

Name*	Type	Required	Direction
firstName	String	Optional	Input
lastName	String	Optional	Input
mobile	String	Optional	Input
fatherName	String	Optional	Input
gender	Boolean	Optional	Input
output	String	Optional	Output

Name \*

Type \*

Entity

Category

Enable rollback ☒

Entity

Type

Entity

Required ☐

Direction ☒ Input ☐ Output

Description

**Step description: None provided.**

If Arguments-firstName contains data AND Arguments-lastName contains data, then:

- Create Contact By Input Arguments**  
Create: Contact [View properties](#)
- Step description: None provided.**  
If Arguments-gender equals [False], then:
  - Update created contact by female if gender is false**  
Update: Create Contact By Input Arguments (شخص) [View properties](#)
- output - success** ✓  
Assign Value: output [View properties](#)

Otherwise:

- Error** ✗ Return if exist any error  
Stop workflow with status of: Canceled [View properties](#)

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.

Name \*

created contact test1

Json Input Data \*

```
{
  "occurrenceTypeId": "create contact",
  "firstName": "David",
  "lastName": "Simon",
  "fatherName": "Jack",
  "gender": false,
}
```

Occurrence TypeId

create\_contact

Related Type

Create Contact

Output message

David Simon Created successfully

این سیستم هر Data Type استاندارد 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 DateTime,Money,Boolean باشد و مقادیر ورودی Json به شکل زیر باشد تبدیل نوع داده بطور خودکار انجام می شود و درصد خطا کاهش می یابد.

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

تبدیل String birthdate به DateTime

تبدیل String deposit به Money

تبدیل int gender به Boolean True



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](mailto:aria.rvn@gmail.com)

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

GitHub: <https://github.com/marvinfar>