

How to expose RegisterUser events as a specific implementation of System Event Notifications

Created by Ilka Guigova, last modified on Jul 21, 2015

Status	DECIDED
Stakeholders	@Matt McLean @Jordan Glassco @Esteban @Darren Tong
Outcome	Use a db based solution to define and query rules
Due date	
Owner	@Ilka Guigova

Background

The following plan is a little rough in the edges but I am proposing it for a solution to the system notifications (email) for registered user events.

Consider the following tables

- RULES_FORMULATION - defines the means / formula to extract the necessary data to determine whether the rule condition is met
- RULES_PARAMETERS - defines the parameters that will be passed to the rule formula
- RULES_DEFINITION - defines the formula to be used and the template for the email notification (ASSUMPTION: the action for the rule is a simple email notification)
- RULES - combines a rule definition with values for the rule parameters (this is a one to many relationship between the rule definition and rule parameters tables)

Here is an example of how this will come together. The design supports more than just registered user events but does make certain assumptions:

1. ASSUMPTION: the action for the rule is a simple email notification
2. ASSUMPTION: the rule formulation ALWAYS includes information about the email_address of the recipient and the pen_guid of the pen the user belongs to
3. ASSUMPTION: The email is sent to the email_address field to be included with each record in the rule formulation dataset
4. ASSUMPTION: In the code we iterate through the rule formulation dataset and call the template for each record

```
CREATE TABLE RULES_FORMULATION
(
    ID INT IDENTITY,
    [DESC] NVARCHAR(500),
    FORMULA NVARCHAR(MAX)
)
-- DROP TABLE RULES_FORMULATION
INSERT INTO RULES_FORMULATION ([DESC], FORMULA) VALUES ('List of users that have registered x number of days ago', 'S
INSERT INTO RULES_FORMULATION ([DESC], FORMULA) VALUES ('List of metrics for a given user', '... spGetUserMetrics @pa
INSERT INTO RULES_FORMULATION ([DESC], FORMULA) VALUES ('List of users that have been invited but have not yet regist
SELECT * FROM RULES_FORMULATION
CREATE TABLE RULES_PARAMETERS
(
    ID INT IDENTITY,
    NAME NVARCHAR(50)
)
-- DROP TABLE RULES_PARAMETERS
INSERT INTO RULES_PARAMETERS(NAME) VALUES ('param1')
INSERT INTO RULES_PARAMETERS(NAME) VALUES ('param2')
CREATE TABLE RULES_DEFINITION
(
    ID INT IDENTITY,
    [DESC] NVARCHAR(500),
    FORMULA_ID INT, -- FK RULES_FORMULATION
    [NOTIFICATION] NVARCHAR(MAX) -- ASSUMPTION: In the code we iterate through the rule formulation dataset and call
    -- ASSUMPTION: The email is sent to the email_address field to be included with each record in the rule formulati
```

```


)
-- DROP TABLE RULES_DEFINITION
INSERT INTO RULES_DEFINITION ([DESC], FORMULA_ID, [NOTIFICATION]) VALUES ('List of users that have registered 5 days
SELECT * FROM RULES_DEFINITION
CREATE TABLE RULES
(
    ID INT IDENTITY,
    DEFINITION_ID INT, -- FK RULES_DEFINITION
    PARAM_ID INT, -- FK RULES_PARAMETERS
    VALUE NVARCHAR(MAX)
)
-- DROP TABLE RULES
INSERT INTO RULES (DEFINITION_ID, PARAM_ID, VALUE) VALUES (1, 1, 5)

```

So this proposition requires almost no code changes - just proper configuration of the rules in the database and the addition of a Collection API call to go through the RULES and send the notifications.

It is not the final version... what are you thoughts?

Action items

 [Like](#) Be the first to like this

decisions

14 Comments



Esteban

I haven't read the proposal yet!!!! So, this is not a comment on the ideal itself! 😊

But I would like to make a side comment about table name convention. Please use camelized notation for table and column names. I missed this when the Settings* tables were created. The reason for this is that EF will generate odd looking classes with this convention and ideally we want that code to be consistent. Thanks!



Jordan Glassco

In principle this is awesome [@Ilka Guigova](#). Maybe security consideration with executing SQL stored in a record as [@Matt McLean](#) suggested... but im not sure its a showstopper.

Say this is a go.. here are some pieces we should discuss:

- SWS trigger job - requirements?

-CollectionApi endpoint to be invoked by SWS

-CollectionApi Rule evaluation engine. This is the code that executes the data models and whose results get funneled into email notifications sender

-Async email queue ? (i'm assuming this is not a requirement for june, because this approach is CollectionApi)



Ilka Guigova

Make sure that the subject if the email notification (at this point the only action when a rule condition is met) is configurable per rule



Ilka Guigova

Make sure that the templates we use for the email notification (at this point the only action when a rule condition is met) supports the same features/capabilities as the other templates in the system. Pay attention in particular to the "placeholders" that can be used in a template and get dynamically replaced by actual data. (e.g., service name, etc)



Ilka Guigova

Added rules_tags and rules_definition_tags to support tagging. Each rule can have more than one tag. Tags can help grouping/filtering rules.

**Ilka Guigova**

Wired the CollectionAPI call: <https://bitbucket.org/cirius/em2platform/commits/4a287c36f4e58955ee3582e9ec5e30a156f4c073>

- input includes an optional Tag parameter
- output includes the response status only at the moment - we may want to extend it with things like RulesCheckedCount, NotificationsSentCount, or similar
- it seems that my line endings differ from the rest and the diff tool shows the entire csproj files changed (after they get autoupdated in my environment)... Any hints?

Let me know what you think of the above - it is subject to change, I am hoping to have the repository function completed shortly as well.

**Ilka Guigova**

Few new properties have been added:

- active BIT DEFAULT 1 to rules_definition
- param_type NVARCHAR to rules_parameters
 - Needs to match exactly the strings in the SqlDbType enumeration (c#)

**Ilka Guigova**

Added type column to rules_formulation. Default: Text. Needs to match C# CommandType enumeration: Text, StoredProcedure, DirectTable

**Ilka Guigova**

Logging has been enabled to a file named "rules-{timestamp}"

**Ilka Guigova**

Please pay attention to the new assumption that all formulas return the service_guid as well as the email_address of ... It is required in order to send the basic email; and to CTemplate.ParseTemplate the subject and body templates.

**Ilka Guigova**

Body and Subject templates are parsed twice. The first time from the dataset returned by the formula given the set of parameters for the rule. For this to work use {n} to indicate which field of the record set to use for the value. The second time from the oService.service.encData as all the other templates (defined in the admin console for example).

**Ilka Guigova**

No longer the case, the formula response is stored as CEncoding and passed as a parameter to CTemplate.ParseTemplate now.

**Ilka Guigova**

Should we be using guids instead of ids?

**Ilka Guigova**

Added the following to the CollectionAPI response:

```
"body": {  
  "stats": [  
    {  
      "ruleDefinitionId": 2,  
      "emailsMatched": 3,  
      "emailsSent": 3  
    }  
  ]  
},
```