INDICE

EXPRESION CRON

La documentación fue tomada de https://www.baeldung.com/cron-expressions

Cron Expression

Entendamos la expresión cron.

Consta de cinco campos:

```
<minute> <hour> <day-of-month> <month> <day-of-week> <command>
```

Caracteres especiales en la expresión

- *** (all)** specifies that event should happen for every time unit. For example, "*" in the <minute>* field means "for every minute."
- **? (any)** is utilized in the <*day-of-month*> and <*day-of -week*> fields to denote the arbitrary value and thus neglect the field value. For example, if we want to fire a script at "5th of every month" irrespective of what day of the week falls on that date, we specify a "?" in the <*day-of-week*> field.
- **– (range)** determines the value range. For example, "10-11" in the field means "10th and 11th hours."
- **, (values)** specifies multiple values. For example, "MON, WED, FRI" in <*day-of-week*> field means on the days "Monday, Wednesday and Friday."
- **/ (increments)** specifies the incremental values. For example, a "5/15" in the <*minute*> field means at "5, 20, 35 and 50 minutes of an hour."
- **L (last)** has different meanings when used in various fields. For example, if it's applied in the <*day-of-month*> field, it means last day of the month, i.e. "31st of January" and so on as per the calendar month. It can be used with an offset value, like "L-3", which denotes the "third to last day of the calendar month." In <*day-of-week*>, it specifies the "last day of a week." It can also be used with another value in <*day-of-week*>, like "6L", which denotes the "last Friday."
- **W (weekday)** determines the weekday (Monday to Friday) nearest to a given day of the month. For example, if we specify "10W" in the <*day-of-month*> field, it means the "weekday near to 10th of that month." So if "10th" is a Saturday, the job will be triggered on "9th," and if "10th" is a Sunday, it will trigger on "11th." If we specify "1W" in <*day-of-month*> and if "1st" is Saturday, the job will be triggered on "3rd," which is Monday, and it will not jump back to the previous month.
- **#** specifies the "N-th" occurrence of a weekday of the month, for example, "third Friday of the month" can be indicated as "6#3".

Ejemplos de expresiones Cron

Let's see some examples of *cron* expressions by using the fields and special characters combinations:

At 12:00 p.m. (noon) every day:

```
0 12 * * ?
```

Every five minutes starting at 1 p.m. and ending at 1:55 p.m. and then starting at 6 p.m. and ending at 6:55 p.m., every day:

```
0/5 13,18 * * ?
```

Every minute starting at 1 p.m. and ending at 1:05 p.m., every day:

```
0-5 13 * * ?
```

At 1:15 p.m. and 1:45 p.m. every Tuesday in the month of June:

```
15,45 13 ? 6 Tue
```

At 9:30 a.m. every Monday, Tuesday, Wednesday, Thursday and Friday:

```
30 9 ? * MON-FRI
```

At 9:30 a.m. on the 15th day of every month:

```
30 9 15 * ?
```

At 6 p.m. on the last day of every month:

```
0 18 L * ?
```

At 6 p.m. on the third to last day of every month:

```
0 18 L-3 * ?
```

At 10:30 a.m. on the last Thursday of every month:

```
30 10 ? * 5L
```

At 10 a.m. on the third Monday of every month:

```
0 10 ? * 2#3
```

At 12 midnight on every day for five days starting on the 10th day of the month:

```
0 0 10/5 * ?
```

Cron Special Strings

In addition to the fields specified in the cron expression, there's also support for some special, predefined values that we can use instead of the fields:

- ***@reboot*** – run once at the startup
- ***@yearly*** or ***@annualy*** – run once a year
- ***@monthly*** – run once a month
- ***@weekly*** – run once a week

- ***@daily*** or ***@midnight*** – run once a day
- ***@hourly*** – run hourly

Conclusion

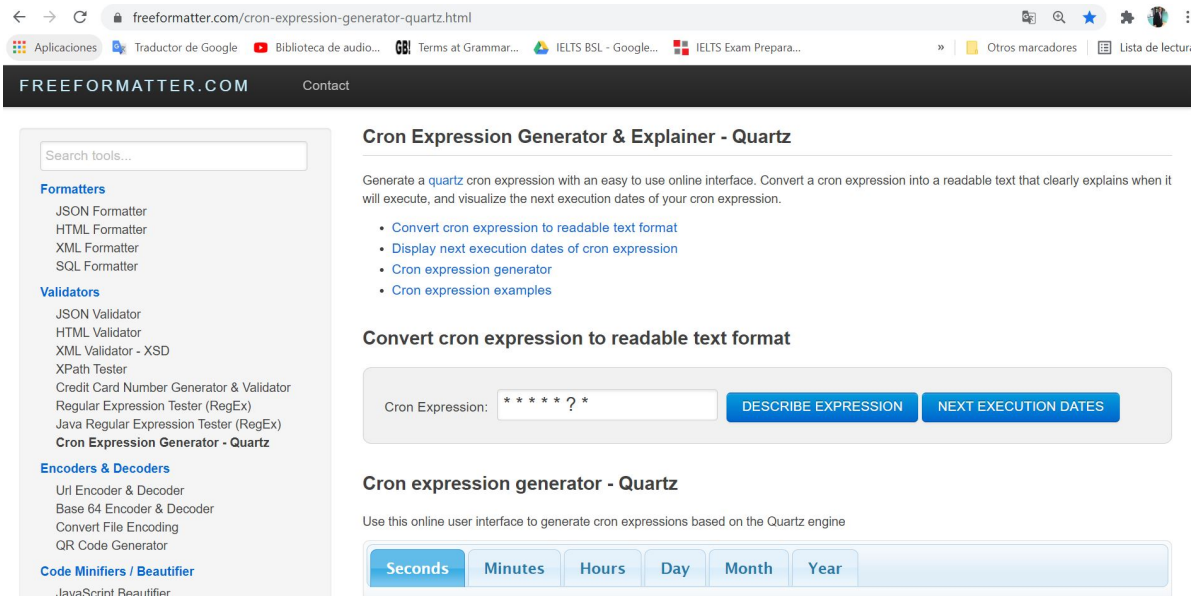
In this quick article, we've explored *cron* jobs and *crontab*.

We've also seen a number of expression examples we can use in our daily work or simply infer other expressions from.

PRUEBA ONLINE DE EXPRESION CRON

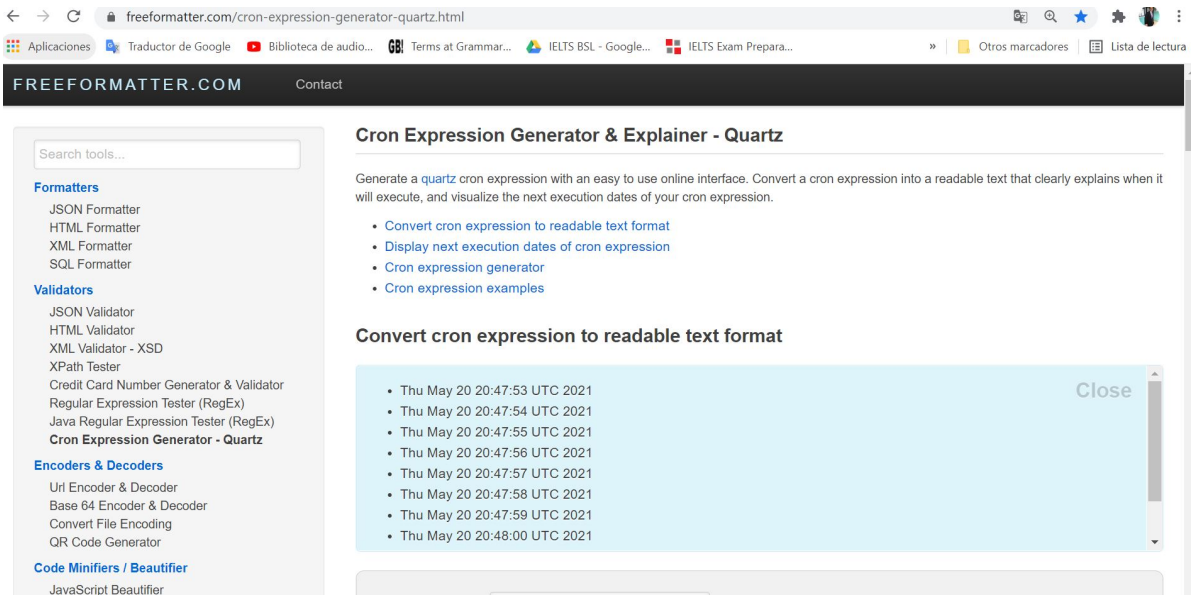
Los ejemplos anteriores pueden ser probados Online.

https://www.freeformatter.com/cron-expression-generator-quartz.html



Al hacer clic en "NEXT EXCECUTION DATES" tendremos algo como esto.

De esta forma podemos probar que nuestra expresión se ejecutará exitosamente.



EXPRESION CRON EN QUICKWIN.

Internamente **QuickWin** arma una expresión Cron de la siguiente manera tomando datos de las configuraciones. Esta configuración aplica solo si la columna **SCHEDULE_DETAIL.EXPRESSION** es NULL o Vacía. Si existe configuración, se tomará directamente esa configuración.

```
// <minute> <hour> <day-of-month> <month> <day-of-week>
//Solo si expresion es NULL
            if("".equals(expression)) {

                expression = "* "+minutoFrom+"-"+minutoUntil+" "+horaFrom+"-
"+horaUntil+" "+dayOfMonth+" "+month+" "+dayOfWeek;

            }
```

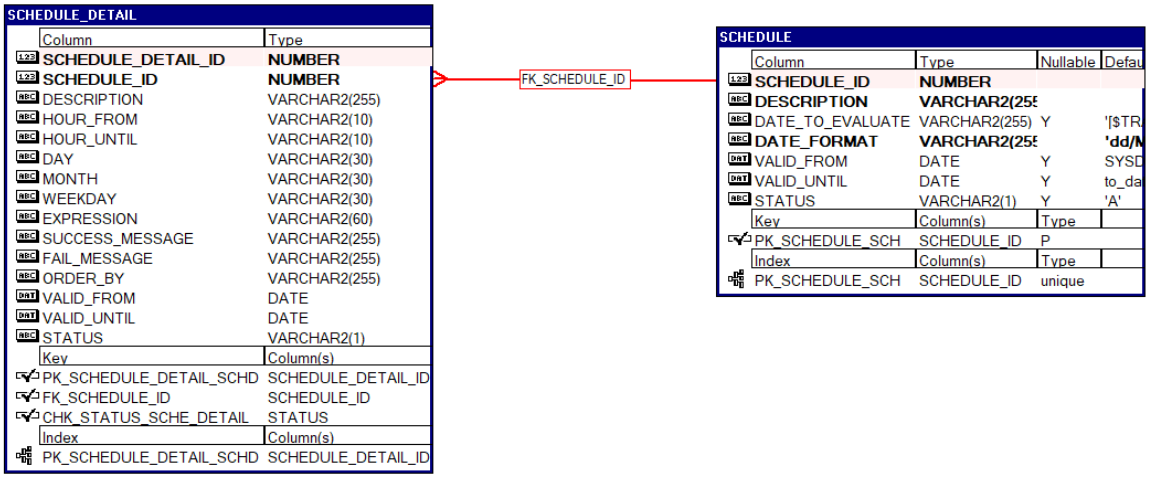
Para validar que una fecha aplica a la expresión Cron, lo que se hace es que a la fecha evaluada se le resta una mínima unidad de tiempo. Si la fecha esta dada con segundo, se le resta 1 segundo, y si está dada en minutos se le resta un minuto. Es decir, la fecha a evaluar siempre debe estar en un formato que soporte **minutos y segundos**. Una vez que la fecha se le resta una unidad de tiempo, se evalúa esa fecha con la expresion Crob para extraer la siguiente fecha. Si la siguiente fecha coincide con la original, entonces el Schedule se cumple. Debe recalcar que también se valida VALID_FROM y VALID_UNTIL del registro.

SCHEDULE

Los **Schedules** sirven para validar rangos de fechas.

Se soportan múltiples fechas u horarios a nivel de configuraciones. Cuando las configuraciones requieren algo mucho mas especifico, se provee un campo llamado EXPRESSION que es la EXPRESSION **CRONTAB soportada por SPRING BOOT** y al estar configurado es quien tiene prioridad.

MODELO ENTIDAD RELACION



QUICKWIN_MDS.SCHEDULE

NAME	TYPE	NULLABLE	DEFAULT	COMMENTS
SCHEDULE_ID	NUMBER	N		Schedule ID
DESCRIPTION	VARCHAR2(255)	N		Descripcion
DATE_TO_EVALUATE	VARCHAR2(255)	Y	'[$TRANSACTIONDATE$]'	Entre corchetes se puede poner el subscriber property
DATE_FORMAT	VARCHAR2(255)	N	'dd/MM/yyyy HH:mm:ss'	Formato de la fecha
VALID_FROM	DATE	Y	SYSDATE	Valido desde
VALID_UNTIL	DATE	Y	to_date('20/03/2987 23:59:59','DD/MM/YYYY HH24:MI:SS')	Valido hasta
STATUS	VARCHAR2(1)	Y	'A'	Estado Activo o Inactivo

SCHEDULE_ID: Id numérico.

DESCRIPTION: Descripción de lo que hace el Schedule.

DATE_TO_EVALUATE: Por defecto [$TRANSACTIONDATE$]. Toma el **Subscriber Properties** .$TRANSACTIONDATE$ y obtiene la fecha. Esta fecha corresponde al formato Java: dd/MM/yyyy HH:mm:ss

DATE_FORMAT: El formato que se espera es formato JAVA. Por ejemplo: **dd/MM/yyyy HH:mm:ss**

VALID_FROM: Desde cuando es aplica.

VALID_UNTIL: Hasta cuando es aplica.

STATUS: Estado A de Activo, I de Inactivo.

QUICKWIN_MDS.SCHEDULE

NAME	TYPE	NULLABLE	DEFAULT	COMMENTS
SCHEDULE_DETAIL_ID	NUMBER	N		
SCHEDULE_ID	NUMBER	N		
DESCRIPTION	VARCHAR2(255)	Y		
HOUR_FROM	VARCHAR2(10)	Y	'ALL'	Minuto en que inicia
HOUR_UNTIL	VARCHAR2(10)	Y	'ALL'	Hora en que inicia 0-23
DAY	VARCHAR2(30)	Y		Dia de la semana 1-31
MONTH	VARCHAR2(30)	Y	'ALL'	Month and weekday names can be given as the first three letters of the English names
WEEKDAY	VARCHAR2(30)	Y	'ALL'	Month and weekday names can be given as the first three letters of the English names
EXPRESSION	VARCHAR2(60)	Y		Expresion Crontab
SUCCESS_MESSAGE	VARCHAR2(255)	Y	'El schedule ha sido validado'	
FAIL_MESSAGE	VARCHAR2(255)	Y	'El schedule no cumple'	
ORDER_BY	VARCHAR2(255)	Y	10	
VALID_FROM	DATE	Y	SYSDATE	
VALID_UNTIL	DATE	Y	to_date('20/03/2987 23:59:59','DD/MM/YYYY HH24:MI:SS')	
STATUS	VARCHAR2(1)	Y	'A'	

SCHEDULE_DETAIL_ID: Id numérico.

SCHEDULE_ID: Id de Schedule.

DESCRIPTION: Descripcion de Schedule.

HOUR_UNTIL: Si es NULL ,Vacío u 'ALL' es: '*'. 0-23 horas. Debe estar entre minutos y segundos.

DAY: Dia del mes. Si es NULL ,Vacío es: '?'. Si es 'ALL' es '*': Todos los días del mes. Debe estar entre minutos y segundos.

MONTH: Si es NULL ,Vacío u 'ALL' es: '*': Todos los meses.

WEEKDAY: Si es NULL ,Vacío u 'ALL' es '*' . Todos los días del semana. Cuando se elije un día de la semana: MON, TUE etc. DAY debe ser vacío o ?, puesto que el día de MES se debe ignorar. El formato usado es mayúsculas los tres primero letras. Tambien puede usarse numeros en vez de nombres.

```
0 - Sun      Sunday
1 - Mon      Monday
2 - Tue      Tuesday
3 - Wed      Wednesday
4 - Thu      Thursday
5 - Fri      Friday
6 - Sat      Saturday
7 - Sun      Sunday
```

EXPRESSION: Debe ser null o vacia, para que las configuraciones de los campos anteriores funcionen, caso contrario se toma la expresion configurada y se ingnora las configuracion anterior. Por ejemplo: 0 18 L * ?. ***Nota : Para mayor información de ejemplos, revise los 'Ejemplos de Expresion Cron' en esta documentación.***

SUCCESS_MESSAGE: Mensaje de exito de la expresión cron.

ORDER_BY: Orden de ejecución

VALID_FROM: Desde cuando es aplica.

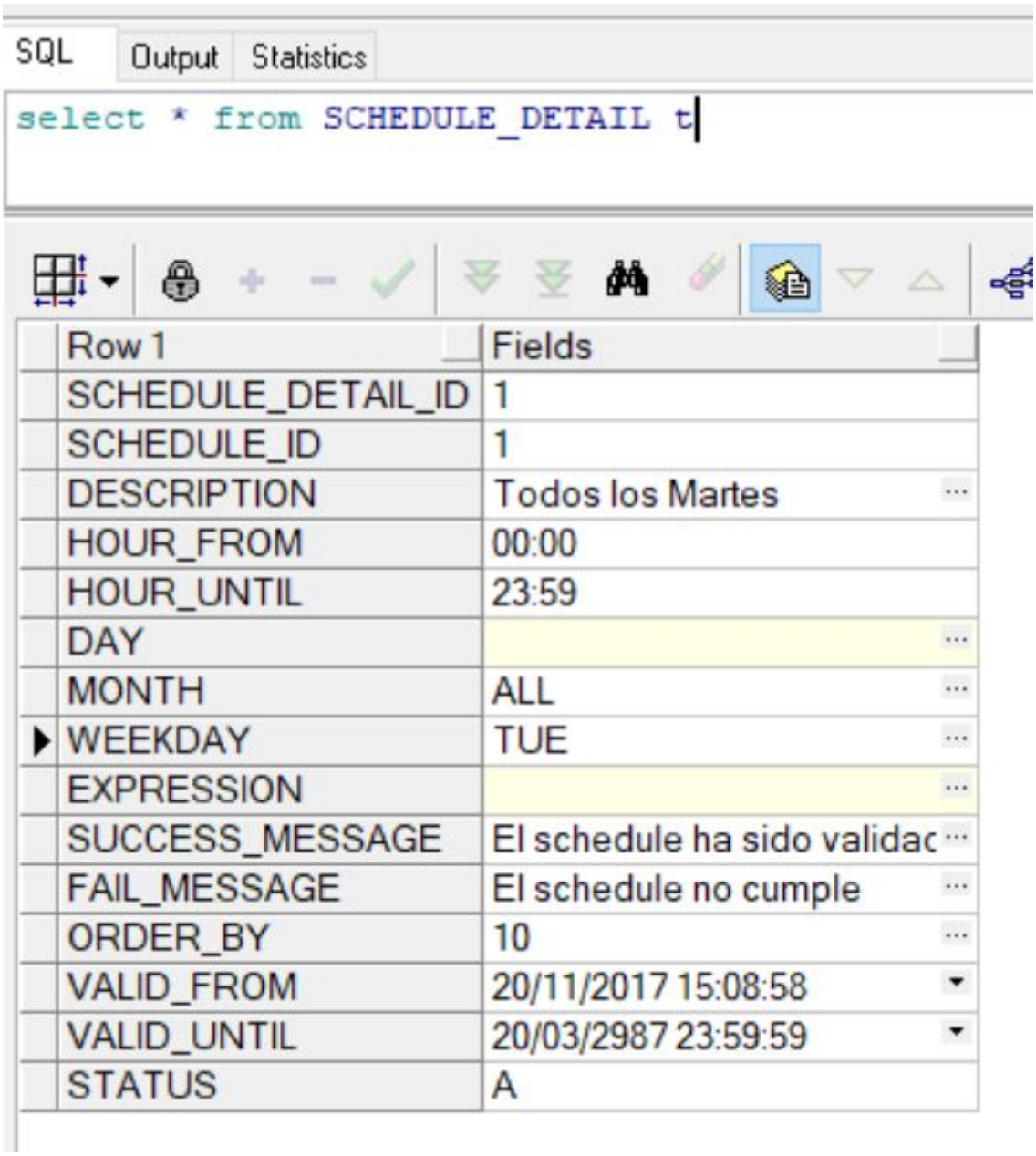
VALID_UNTIL: Hasta cuando es aplica.

STATUS: Estado A de Activo, I de Inactivo.

EJEMPLO

En el siguiente ejemplo veremos la configuración de un Schedule que aplica todos los martes de 0 horas a 23 horas con 59 minutos.

Nota: Se pueden configurar mas de una, y todas se evalúan, y si una de ellas se cumple, el Schedule es valido.



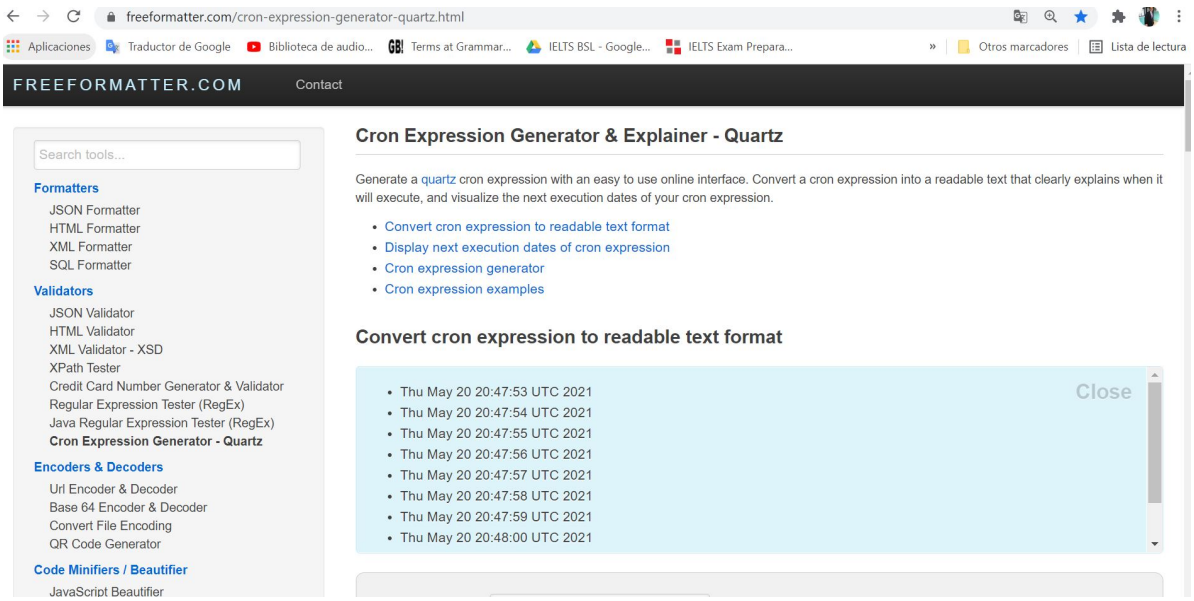
Row 1	Fields
SCHEDULE_DETAIL_ID	1
SCHEDULE_ID	1
DESCRIPTION	Todos los Martes
HOUR_FROM	00:00
HOUR_UNTIL	23:59
DAY	
MONTH	ALL
WEEKDAY	TUE
EXPRESSION	
SUCCESS_MESSAGE	El schedule ha sido validac
FAIL_MESSAGE	El schedule no cumple
ORDER_BY	10
VALID_FROM	20/11/2017 15:08:58
VALID_UNTIL	20/03/2987 23:59:59
STATUS	A

En el siguiente ejemplo la expresión configurada corresponde.

```
expression = "* "+minutoFrom+"-"+minutoUntil+" "+horaFrom+"-"+horaUntil+"
"+dayOfMonth+" "+month+" "+dayOfWeek
```

Expresión Cron resultante:

```
expression = "* 00-59 00-23 ? * TUE"
```



REQUEST

POST http://192.168.37.146:8101/quickWin/checkSchedule

```
{
  "scheduleId": "1",
  "invokerName": "Test de los dias martes",
  "cacheOptions":"0",
  "sessionData": {
    "externalSubscriberProperties": [
      {"id": "$PAYMENTDATE$","value": ["26/03/2019 00:00:00"]}
    ]
  }
}
```

scheduleId: Representa el ID del Schedule configurado.

invokerName: Representa el nombre de quien lo ejecuta.

cacheOptions: "0" representa tomar datos de la memoria caché y "1" recuperarlo directo de la base. Esta última no es una opción recomendada.

sessionData: Como ya hemos explicao en la sesion de ***Subscriber Properties***, ***sessionData*** es quien contiene la lista de ***Subscriber Properties***. Normalmente este request siempre debe tener un subscriber de fechas en el formato "dd/MM/yyyy HH:mm:ss" debería incluir minimo segundos o minutos para funcionar.

Por defecto internamente en QuickWin utiliza **$TRANSACTIONDATE$** que un Subscriber Properties con el formato **dd/MM/yyyy HH:mm:ss** incializado por la campaña que lo usa. Si se requiere usarse desde el contrato por fines de prueba se debería proveer: `{"id": "$TRANSACTIONDATE$","value": ["26/03/2019 00:00:00"]}`

RESPONSE

Response con Error.

En este caso espera que se envíe el parámetro delimitado por []; que es es $PAYMENTDATE$

```
{
   "met": false,
   "mensaje": "Error de Conversion de Formato de Fechas para ScheduleId. Revise que los subscriber esten entre [SP] 1: Unparseable date: \"[$PAYMENTDATE$]\"",
   "code": -4
}
```

Response con Éxito true

```
{
   "met": true,
   "mensaje": "ScheduleId 1. El schedule ha sido validado",
   "code": 0
}
```

Response con Éxito false

```
{
   "met": false,
   "mensaje": "ScheduleId 1. El schedule no cumple",
   "code": 0
}
```

met: true, el Schedule se cumple. False si la Schedule , no se cumple o hay algun error.

mensaje: Mensaje de éxito cuando el Schedule se cumple tambien mensaje de falla cuando el Schedule no se cumple.

code: Codigo 0 de Exito cuando la Schedule se cumple o no se cumple. Negativo si existe algun error.

CLEAR CACHE

Para refrescar datos de memoria use la URL http://192.168.37.146:8101/quickWin/clearScheduleById/1

Nota: Debe cambiar el id respectivo.

GET CACHE DATA

Para consultar datos de memoria use la URL http://192.168.37.146:8101/quickWin/findScheduleById/1

Nota: Debe cambiar el id respectivo.

PLSQL

CREAR SCHEDULE

```
declare

  ln_schedule_id  number := '';
  lv_identifier varchar(15) := '[QWV1-MGR] ';
begin

  -- CONDITION
select nvl(max(t.SCHEDULE_ID), 0) + 1 into ln_schedule_id from SCHEDULE t;



 -- Si son mas valores, agregarlos con comas, (select 'a,b,c' from dual)
insert into SCHEDULE (SCHEDULE_ID, DESCRIPTION, DATE_TO_EVALUATE, DATE_FORMAT)
values (ln_schedule_id, lv_identifier||'Evaluacion los dias martes',
'[$PAYMENTDATE$]', 'dd/MM/yyyy HH:mm:ss');

-- SCHEDULE_DETAIL
insert into SCHEDULE_DETAIL (SCHEDULE_DETAIL_ID, SCHEDULE_ID, DESCRIPTION,
HOUR_FROM, HOUR_UNTIL, DAY, MONTH, WEEKDAY, EXPRESSION, SUCCESS_MESSAGE,
FAIL_MESSAGE, ORDER_BY)
values ((SELECT NVL(MAX(BS.SCHEDULE_DETAIL_ID),0) + 1  FROM SCHEDULE_DETAIL BS),
ln_schedule_id, 'Todos los Martes', '00:00', '23:59', '', 'ALL', 'TUE', '', 'El
schedule ha sido validado', 'El schedule no cumple', (SELECT
NVL(MAX(T.Order_By),0) + 10  FROM SCHEDULE_DETAIL T WHERE T.SCHEDULE_ID =
ln_schedule_id));

dbms_output.put_line('ln_schedule_id: ' || ln_schedule_id);

dbms_output.put_line('---------------REFRESH-------------------------------------
--');
dbms_output.put_line('http://192.168.37.146:8101/quickWin/clearScheduleById/'||l
n_schedule_id);

dbms_output.put_line('---------------SELECT--------------------------------------
-');
dbms_output.put_line('SELECT *  FROM SCHEDULE_DETAIL t where t.SCHEDULE_ID  =
'''||ln_schedule_id||''';');
dbms_output.put_line('SELECT *  FROM SCHEDULE t where t.SCHEDULE_ID =
'''||ln_schedule_id||''';');
dbms_output.put_line('http://192.168.37.146:8101/quickWin/findScheduleById/'||ln
_schedule_id);

dbms_output.put_line('--------------UPDATE STATUS-------------------------------
--------');
```

```
dbms_output.put_line('UPDATE  SCHEDULE_DETAIL t set status = ''I'' where
t.SCHEDULE_ID  = '''||ln_schedule_id||''';');
dbms_output.put_line('UPDATE  SCHEDULE t set status = ''I'' where t.SCHEDULE_ID
= '''||ln_schedule_id||''';');

dbms_output.put_line('--------------DELETE--------------------------------------
-');
dbms_output.put_line('DELETE FROM SCHEDULE_DETAIL t where t.SCHEDULE_ID  =
'''||ln_schedule_id||''';');
dbms_output.put_line('DELETE FROM SCHEDULE t where t.SCHEDULE_ID =
'''||ln_schedule_id||''';');

dbms_output.put_line('---------------REQUEST-------------------------------------
----------');

dbms_output.put_line('http://192.168.37.146:8101/quickWin/checkSchedule');

dbms_output.put_line('---------------POST REST-----------------------------------
----');

dbms_output.put_line('{
   "scheduleId": "'||ln_schedule_id||'",
   "invokerName":"Test de los dias martes",
   "cacheOptions":"0",
   "sessionData": {
      "externalSubscriberProperties": [');

FOR subs IN (SELECT T.DATE_TO_EVALUATE FROM SCHEDULE T WHERE T.SCHEDULE_ID =
ln_schedule_id) LOOP
dbms_output.put_line('           {"id":
"'||substr(subs.DATE_TO_EVALUATE,2,length(subs.DATE_TO_EVALUATE)-2)||'","value":
["26/03/2019 00:00:00"]},');
END LOOP;
-- DATOS DE LA FUNCION
dbms_output.put_line('           {"id": "'||'DUMMY'||'","value": [""]}');


dbms_output.put_line('       ]');
dbms_output.put_line('   }
}');


--chequear los erroes
exception when others then
    rollback;
    dbms_output.put_line(sqlerrm);
end;
```

SALIDA

```
ln_schedule_id: 3
---------------REFRESH--------------------------------------
http://192.168.37.146:8101/quickWin/clearScheduleById/3
---------------SELECT---------------------------------------
SELECT *  FROM SCHEDULE_DETAIL t where t.SCHEDULE_ID  = '3';
SELECT *  FROM SCHEDULE t where t.SCHEDULE_ID = '3';
http://192.168.37.146:8101/quickWin/findScheduleById/3
```

```
--------------UPDATE STATUS----------------------------------------
UPDATE  SCHEDULE_DETAIL t set status = 'I' where t.SCHEDULE_ID  = '3';
UPDATE  SCHEDULE t set status = 'I' where t.SCHEDULE_ID = '3';
--------------DELETE----------------------------------------
DELETE FROM SCHEDULE_DETAIL t where t.SCHEDULE_ID  = '3';
DELETE FROM SCHEDULE t where t.SCHEDULE_ID = '3';
---------------REQUEST-------------------------------------------------
http://192.168.37.146:8101/quickWin/checkSchedule
---------------POST REST-------------------------------------
{
   "scheduleId": "3",
   "invokerName":"Test de los dias martes",
   "cacheOptions":"0",
   "sessionData": {
      "externalSubscriberProperties": [
           {"id": "$PAYMENTDATE$","value": ["26/03/2019 00:00:00"]},
           {"id": "DUMMY","value": [""]}
       ]
   }
}
```

DELETE SCHEDULE

El siguiente script elimina Schedule.

```
-- Created on 20/05/2021 by MGARCIAR
declare
lv_description varchar(255) := '[QWV1-MGR]%';
deleted_cant number := 0;
BEGIN


select count(t.schedule_id) into deleted_cant FROM QUICKWIN_MDS.Schedule_Detail
T
 WHERE T.Schedule_Detail_Id IN
       (SELECT T.Schedule_Id FROM QUICKWIN_MDS.Schedule T
         WHERE T.DESCRIPTION LIKE lv_description);

dbms_output.put_line('cantidad de Schedule_Detail a eliminar: ' || deleted_cant
);

DELETE FROM QUICKWIN_MDS.Schedule_Detail T
 WHERE T.Schedule_Detail_Id IN
       (SELECT T.Schedule_Id FROM QUICKWIN_MDS.Schedule T
         WHERE T.DESCRIPTION LIKE lv_description);

select count(t.schedule_id) into deleted_cant FROM QUICKWIN_MDS.Schedule T WHERE
T.DESCRIPTION LIKE lv_description;
dbms_output.put_line('cantidad de Schedule a eliminar: ' || deleted_cant );

DELETE FROM QUICKWIN_MDS.Schedule t WHERE T.DESCRIPTION LIKE lv_description;
--chequear los erroes
exception when others then
    rollback;
    dbms_output.put_line(sqlerrm);
end;
```

SALIDA

```
cantidad de Schedule_Detail a eliminar: 1
cantidad de Schedule a eliminar: 1
```

Support

QukickWin is a Claro's project. It can grow thanks to feeback and support of other actors inside the company

Stay in touch

- Leader Architec - TIC Manuel García
- Mail - mgarciar@claro.com.ec
- Sponsor - TIC Guillermo Proaño
- Mail - gproano@claro.com.ec
- Developers - Fernando Andrade
- Mail - fernando.andrade@gizlocorp.com