

# INDICE

#

## CONDITIONS/CONDICIONES

Los **Conditions** en **QuickWin** se las usa para validar cierta lógica y esto depende de cada necesidad en específica. Algunas otras funcionalidades de **QuickWin** las requieren. Por lo tanto, es importante primero siempre configurar las condiciones necesarias. Las **Conditions** en QuickWin pueden probarse y configurarse de forma independiente. Tienen su propio contrato de ejecución así como también endpoints para refrescar caché.

Las Condiciones dependen de los SubscriberProperties, ya que la lógica de condición recae sobre ellos. Por lo pronto, antes de empezar a configurar Condiciones, es importante que tenga noción de como funcionan los **SubscriberProperties**.

Como ya lo mencionamos previamente, las condiciones requieren a los SubscriberProperties, por lo tanto su contrato los contienen.

## REQUEST

POST <http://192.168.37.146:8101/quickWin/executeCondition> HTTP/1.1

```
{
  "conditionId": "269",
  "invokerName": "PRUEBA",
  "cacheOptions": "0",
  "sessionData": {
    "externalSubscriberProperties": [
      {
        "id": "PRUEBA_GRUPO_COND1",
        "value": [
          "SxI"
        ]
      },
      {
        "id": "TMP1",
        "value": [
          "SI"
        ]
      },
      {
        "id": "CODIGO_ID",
        "value": [
          "100"
        ]
      }
    ]
  }
}
```

Ud puede formatear json en el siguiente enlace <https://jsonformatter.curiousconcept.com/>

**conditionId:** Representa el ID de la condicion configurada.

**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 explicado en la sesión de **SubscriberProperties**, **sessionData** es quien contiene la lista de **SubscriberProperties**.

## RESPONSE

---

La condicion, contiene dos posibles escenarios, cumplida o no.

```
{
  "met": false,
  "mensaje": "El valor de PRUEBA_GRUPO_COND1=SxI es diferente a SI",
  "code": 100100
}
```

```
{
  "met": true,
  "mensaje": "La condicion PRUEBA_GRUPO_COND1=SI concide con TMP1=SI",
  "code": -100100
}
```

**met:** true, si la condicion se cumple. False si la condición, no se cumple.

**mensaje:** Mensaje de éxito cuando la condicion se cumple tambien mensaje de falla cuando la condición no se cumple.

**code:** Código de Exito cuando la condicion se cumple y tambien Código de Falla cuando la condición no se cumple.

## CLEAR CACHE

---

Para refrescar los cambios ejecute:

GET <http://192.168.37.146:8101/quickWin/clearConditionById/269>

```
{"refreshMessage":"Se ha actualizado Condition con Id: 269","updated":true}
```

## GET DATA FROM CACHE

---

Para consultar datos cargados en memoria ejecute:

GET <http://192.168.37.146:8101/quickWin/findConditionById/269>

```
{
  "refreshMessage": "Se ha actualizado Condition con Id: 269",
  "updated": true,
  "conditionProperties": [
    {
      "conditionPropertyId": 411,
      "subscriberProperties": {
        "subscriberPropertyId": "PRUEBA_GRUPO_COND1",
        "description": "Grupo de condiciones",
        "status": "A",
        "orderBy": 10,
        "domain": {
          "domainId": 377,
          "description": "Condition_Group: SI",
          "status": "A",
          "condition": {
            "conditionId": 269,
            "description": "Condicion SI/NO",
            "validFrom": "2020-01-16",
            "validUntil": "2987-03-20",
            "status": "A",
            "dataType": "VARCHAR",
            "expressionCondition": "[SP=DM]",
            "minNumDomainElements": 1,
            "maxNumDomainElements": 1,
            "validFrom": "2020-01-16",
            "validUntil": "2987-03-20",
            "failMessage": "SEP=#PRUEBA_GRUPO_COND1#El valor de PRUEBA_GRUPO_COND1=%s es diferente a SI",
            "successMessage": "SEP=#PRUEBA_GRUPO_COND1#TMP1#La condicion PRUEBA_GRUPO_COND1=%s concide con TMP1=%s",
            "successCode": "SEP=|CODIGO_ID|-100%s",
            "failCode": "SEP=|CODIGO_ID|100%s",
            "status": "A"
          }
        },
        "domainElementsRepos": [
          {
            "domainElements": [
              {
                "domainElementId": 805,
                "domain": {
                  "domainId": 377,
                  "description": "Condition_Group: SI",
                  "status": "A",
                  "elementValue": "SI",
                  "status": "A"
                }
              }
            ],
            "domainValueRangesRepos": [
              {
                "domainValueRanges": [
                  {}
                ]
              }
            ]
          }
        ]
      }
    }
  ]
}
```

## COMPORTAMIENTO DE LAS CONDICIONES

Una condición evalúa a una lista de forma individual de **SubscriberProperty** VS **Domain** a través de una expresión de evaluación de condición definida a nivel de configuración de tablas. Por ejemplo IGUALDAD, DESIGUALDAD, CONTIENE, NO CONTIENE etc. La evaluación se ejecuta **SubscriberProperties** por **SubscriberProperties** evaluando todos los valores que cada una contienen contra todos los elementos del **Dominio** configurado, en un orden específico definido. La evaluación se detiene cuando una no se cumple o todas se cumplen. En ese momento determina si se cumplieron todas las evaluaciones de todos los **SubscriberProperties** para dar por éxito a la condición por completo. Si la condición fue interrumpida ya que un **SubscriberProperty** no se cumplió contra su dominio, entonces muestra mensaje y código de falla configurados. Si es exitosa, es decir llegó al final, entonces muestra mensaje y código de éxito de la última propiedad definida. En este punto el mensaje de falla es el más importante, ya que éste nos da feedback en caso de detenerse la evaluación, mientras que el mensaje de éxito siempre puede ser el mismo para todas las propiedades evaluadas y será siempre el mensaje del último **SubscriberProperty** evaluado contra su dominio.

## ANATOMIA DE LAS CONDICIONES



## QUICKWIN\_MDS.DOMAIN\_ELEMENTS

NAME	TYPE	NULLABLE	DEFAULT
DOMAIN_ELEMENT_ID	NUMBER	N	
ELEMENT_VALUE	VARCHAR2(255)	Y	
DOMAIN_ID	NUMBER	N	
STATUS	VARCHAR2(1)	N	'A'

**DOMAIN\_ELEMENT\_ID:** Id numérico.

**DOMAIN\_ID:** Id numérico de dominio.

**ELEMENT\_VALUE:** Contiene el valor del elemento.

**STATUS:** Estado A de Activo, I de Inactivo.

## QUICKWIN\_MDS.DOMAIN\_VALUE\_RANGES

**Nota:** Para que la configuración de **DOMAIN\_VALUE\_RANGES** funcione se debe crear por lo menos un elemento en la tabla **DOMAIN\_ELEMENTS** y ese elemento nunca se encuentre como valor esperado sólo así se consideran los rangos.

Se puede definir múltiple rangos de evaluación para un mismo dominio.

NAME	TYPE	NULLABLE	DEFAULT
DOMAIN_VALUE_RANGE_ID	NUMBER	N	
DOMAIN_ID	NUMBER	Y	
FROM_VALUE	VARCHAR2(255)	Y	
TO_VALUE	VARCHAR2(255)	Y	
ORDER_BY	NUMBER	N	
STATUS	VARCHAR2(1)	N	'A'

**DOMAIN\_VALUE\_RANGE\_ID:** Id numérico.

**DOMAIN\_ID:** Id numérico de dominio.

**FROM\_VALUE:** Valor Inicial, normalmente numérico, o fechas.

**TO\_VALUE:** Valor Inicial, normalmente numérico, o fechas.

**ORDER\_BY:** Valor numérico que indica prioridad de menor a mayor. 100,200,300 etc.

**STATUS:** Estado A de Activo, I de Inactivo.

## QUICKWIN\_MDS.CONDITION\_PROPERTIES

NAME	TYPE	NULLABLE	COMMENTS
SUBSCRIBER_PROPERTY_ID	VARCHAR2(255)	N	
ORDER_BY	NUMBER	Y	10
DATA_TYPE	VARCHAR2(255)	N	
DATA_TYPE_FORMAT	VARCHAR2(255)	Y	
EXPRESSION_CONDITION	VARCHAR2(255)	Y	
FAIL_MESSAGE	VARCHAR2(255)	N	'La condicion no cumple'
DOMAIN_ID	NUMBER	N	
MIN_NUM_DOMAIN_ELEMENTS	NUMBER	Y	
MAX_NUM_DOMAIN_ELEMENTS	NUMBER	Y	
VALID_FROM	DATE	N	SYSDATE
VALID_UNTIL	DATE	N	to_date('20/03/2987 23:59:59','DD/MM/YYYY HH24:MI:SS')
STATUS	VARCHAR2(1)	N	'A'
SUCCESS_MESSAGE	VARCHAR2(255)	Y	'La condición ha sido validada'
FAIL_CODE	VARCHAR2(255)	Y	'-121'
SUCCESS_CODE	VARCHAR2(255)	Y	'0'

## COMENTARIOS

NAME	COMMENTS
CONDITION_PROPERTY_ID	
CONDITION_ID	
SUBSCRIBER_PROPERTY_ID	
ORDER_BY	
DATA_TYPE	
DATA_TYPE_FORMAT	
EXPRESSION_CONDITION	
FAIL_MESSAGE	MENSAJE CUANDO ES FALLIDA LA EVALUACION DEL CONDICION, PARA TOMAR DESDE UN SUBSCRIBER PROPERTY SE DEBE COLOCAR ENTRE LLAVES [MI_MENSAJE]
DOMAIN_ID	
MIN_NUM_DOMAIN_ELEMENTS	
MAX_NUM_DOMAIN_ELEMENTS	
VALID_FROM	
VALID_UNTIL	
STATUS	
SUCCESS_MESSAGE	MENSAJE CUANDO ES EXITOSA LA EVALUACION DEL CONDICION, PARA TOMAR DESDE UN SUBSCRIBER PROPERTY SE DEBE COLOCAR ENTRE LLAVES [MI_MENSAJE]
FAIL_CODE	CODIGO CUANDO FALLA LA EVALUACION DE LA CONDICION, PARA TOMAR DESDE UN SUBSCRIBER PROPERTY SE DEBE COLOCAR ENTRE LLAVES [MI_CODIGO]
SUCCESS_CODE	CODIGO CUANDO ES EXITOSA LA EVALUACION DE LA CONDICION, PARA TOMAR DESDE UN SUBSCRIBER PROPERTY SE DEBE COLOCAR ENTRE LLAVES [MI_CODIGO]

**CONDITION\_PROPERTY\_ID:** ID de numérico.

**CONDITION\_ID:** Id de Condición.

**SUBSCRIBER\_PROPERTY\_ID:** Nombre o ID de **SubscriberProperty**.

**ORDER\_BY:** Valor numérico que indica prioridad de menor a mayor. 100,200,300 etc.

**DATA\_TYPE:** DATA\_TYPE IN('NUMERIC','DATE','VARCHAR'). Especifica el tipo de datos que se va a evaluar.

- Si va a comparar **STRING**, use **VARCHAR**.
- Si va a compara **NÚMEROS** use **NUMERIC**.
- Si va a comparar **FECHAS**, use **DATE**.

**DATA\_TYPE\_FORMAT:** Este campo se lo usa cuando **DATA\_TYPE** está definido como **DATE**.

- Se utiliza para especificar el formato de las fechas que se usa para comparar con fechas. Se usan los formatos que el lenguaje JAVA soporta. Por ejemplo: **dd/MM/yyyy HH:mm:ss**
- Recuerde que HH, representa 24 Horas. Es recomendable visitar la web para mas informacion.

- - y = year (yy or yyyy)
  - M = month (MM)
  - d = day in month (dd)
  - h = hour (0-12) (hh)
  - H = hour (0-23) (HH)
  - m = minute in hour (mm)
  - s = seconds (ss)
  - S = milliseconds (SSS)
  - z = time zone text (e.g. Pacific Standard Time...)
  - Z = time zone, time offset (e.g. -0800)

Ejemplos de formatos de fechas:

```

yyyy-MM-dd           (2009-12-31)

dd-MM-YYYY           (31-12-2009)

yyyy-MM-dd HH:mm:ss  (2009-12-31 23:59:59)

HH:mm:ss.SSS         (23:59:59.999)

yyyy-MM-dd HH:mm:ss.SSS  (2009-12-31 23:59:59.999)

yyyy-MM-dd HH:mm:ss.SSS Z  (2009-12-31 23:59:59.999 +0100)
  
```

**EXPRESSION\_CONDITION:**

**SP** representa **SubscriberProperties**, mientras que **DM** representa **Dominio**.

- **[SP<>DM]:** El valor o valores de SP no corresponde a ningun valor del dominio
- **[SP^DM]** Representa cuantos elementos de **SP** deben estar en **DM**. En logica de conjunto SP intercepta a DM indistintamente, pero en las columnas **MIN\_NUM\_DOMAIN\_ELEMENTS** y **MAX\_NUM\_DOMAIN\_ELEMENTS** definimos cuantos minios deben interceptar y cuantos maximos. Por defecto deberia ser uno.
- **[SP=DM]** Los valores del SP deben coincidir con los valores de **DM**.
- **[SP<=DM]** Los valores de SP pueden ser iguales a todos los valores del Dominio o no todos por completo.
- **[SP>=DM]** Los valores de SP corresponden a todos los valores del dominio y no importa si traen más de lo que no están.

**MIN\_NUM\_DOMAIN\_ELEMENTS:** Mínimo número de elementos que deben coincidir con los valores de DM. Válido para expresión **[SP^DM]**

**MAX\_NUM\_DOMAIN\_ELEMENTS:** Máximo que pueden coincidir. Válido para expresión **[SP^DM]**

**DOMAIN\_ID:** Id de dominio con el cual se hace la evaluación de **EXPRESSION\_CONDITION**.



**VALID\_FROM:** Desde cuando es aplica.

**VALID\_UNTIL:** Hasta cuando es aplica.

**STATUS:** Estado A de Activo, I de Inactivo.

**SUCCESS\_MESSAGE:** ver comentario de tabla. Adicional este campo soporta **INTERPOLACION DE STRING**.

**FAIL\_MESSAGE:** ver comentario de tabla. Adicional este campo soporta **INTERPOLACION DE STRING**.

**FAIL\_CODE:** ver comentario de tabla. Adicional este campo soporta **INTERPOLACION DE STRING**.

**SUCCESS\_CODE:** ver comentario de tabla. Adicional este campo soporta **INTERPOLACION DE STRING**.

## DOMINIO CON ELEMENTOS DINAMICO

Para hacer una codición dinámica debe alterar la siguiente propiedad.

**QUICKWIN\_MDS.DOMAIN\_ELEMENTS.ELEMENT\_VALUE**

Este valor puede recibir el nombre de un **SubscriberProperty** entre corchete.

[**SUBSCRIBER\_PROPERTY**]. En este caso los valores del **SubscriberProperty** pasan a formar parte del los elementos del dominio y se uniran a los que ya exista.

## DOMINIO CON ELEMENTOS RANGO DINAMICOS

**Nota:** Para que la configuracion de **DOMAIN\_VALUE\_RANGES** funcione se debe crear por lo menos un elemento en la tabla **DOMAIN\_ELEMENTS** y ese elemento nunca se encuentre como valor esperado sólo así se consideran los rangos.

La tabla de **DOMAIN\_VALUE\_RANGES** puede contener multiples rangos para un mismo dominio.

Los rangos tambien se pueden definir con **SubscriberProperties**, delimitados por [] en las columnas **FROM\_VALUE**, y **TO\_VALUE**.

```
SELECT * FROM QUICKWIN_MDS.DOMAIN_VALUE_RANGES
```

	DOMAIN_VALUE_RANGE_ID	DOMAIN_ID	FROM_VALUE	TO_VALUE	ORDER_BY	STATUS
1	1	1	14/09/2018 00:00:00 ...	14/09/3018 00:00:00 ...	10	A
2	2	2	sysdate	14/09/3018 00:00:00 ...	20	A
3	3	18	[FECHA_INI]	[FECHA_FIN]	10	A

Nota: Es recomendable que tanto la propiedad de **FROM\_VALUE**, y **TO\_VALUE** contengan la misma cantidad de elementos.

### Ejemplo 1.

```
FECHA_INI = ["FECHA1", "FECHA3"]  
FECHA_FIN = ["FECHA2", "FECHA4"]
```

Tendriamos algo así.

## QUICKWIN\_MDS.DOMAIN\_VALUE\_RANGES

FECHA_INI	FECHA_FIN
FECHA1	FECHA2
FECHA3	FECHA4

## Ejemplo 2.

```
FECHA_INI = ["FECHA1", "FECHA3"]  
FECHA_FIN = ["FECHA2"]
```

## QUICKWIN\_MDS.DOMAIN\_VALUE\_RANGES

FECHA_INI	FECHA_FIN
FECHA1	FECHA2
FECHA3	FECHA2

## CONDITIONS GROUP

En la mayoría de las configuraciones de **Quickwin**, se utilizan condiciones, pero hay caso donde necesitamos evaluar si una u otra condicion se cumple o ambas se cumplen, ya sea con AND u OR, para continuar. A este concepto le llamamos grupos de condiciones donde unimos varias condiciones con operador **AND** u **OR**.

**Nota:** En los casos donde se requiera configurar **CONDITIONS** o **CONDITION\_GROUPS**, **CONDITION\_GROUPS** tiene preferencia, y se ignora la condicion si esta tambien se encuentra configurada.

Condition Group puede configurarse por separado y mantiene su propio contrato de servicio.

## REQUEST

POST <http://192.168.37.146:8101/quickWin/executeConditionGroup> HTTP/1.1

```
{  
  "conditionGroupId": "1",  
  "invokerName": "PRUEBA",  
  "cacheOptions": "0",  
  "sessionData": {  
    "externalSubscriberProperties": [  
      {  
        "id": "PRUEBA_GRUPO_COND2",  
        "value": [  
          "1"  
        ]  
      },  
      {  
        "id": "PRUEBA_GRUPO_COND1",  
        "value": [  
          "1"  
        ]  
      }  
    ]  
  }  
}
```

```

    "SI"
  ]
}
]
}
}

```

Ud puede formatear json en el siguiente enlace <https://jsonformatter.curiousconcept.com/>

**conditionGroupId:** Representa el ID del grupo de condición 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 explicado en la sesión de **SubscriberProperties**, **sessionData** es quien contiene la lista de **SubscriberProperties**.

## RESPONSE

```

{
  "met": true,
  "mensaje": ["La condicion PRUEBA_GRUPO_COND1=SI concide con TMP1=TMP1,
Resultado: true"],
  "code": 0
}

```

```

{
  "met": false,
  "mensaje": [
    "El valor de PRUEBA_GRUPO_COND1=NO es diferente a SI, Resultado: false",
    "El valor de PRUEBA_GRUPO_COND2 es diferente a 1, Resultado: false"
  ],
  "code": -121
}

```

**met:** true, si la evaluación del grupo de condición se cumple. False si no se cumple.

**mensaje:** Mensaje de las condiciones evaluadas.

**code:** Código de éxito o fracaso cuando las condiciones se evalúan

## CLEAR CACHE

Para refrescar los cambios ejecute:

GET <http://192.168.37.146:8101/quickWin/clearConditionGroupById/1>

```

{"refreshMessage":"Se ha limpiado ConditionGroup con Id:
1","updated":true,"conditionGroup":null,"conditionGroupElements":null}

```

## GET DATA FROM CACHE

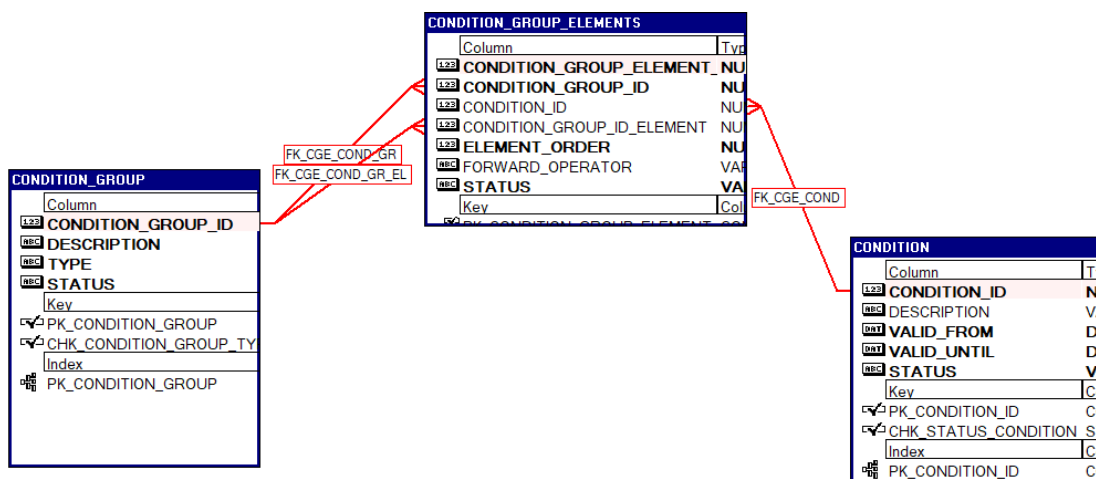
Para consultar datos cargados en memoria ejecute:

GET <http://192.168.37.146:8101/quickWin/findConditionGroupById/1>

```
{"refreshMessage":"Se ha consultado ConditionGroup con Id: 1", "updated":true, "conditionGroup":{"conditionGroupId":1, "description":"Prueba OR", "status":"A", "type":"CONDITION"}, "conditionGroupElements": [{"conditionGroupElementId":1, "elementOrder":1, "forwardOperator":"OR", "status":"A", "condition":{"conditionId":269, "description":"Condicion SI/NO", "validFrom":"2020-01-16", "validUntil":"2987-03-20", "status":"A"}, "conditionGroup":{"conditionGroupId":1, "description":"Prueba OR", "status":"A", "type":"CONDITION"}}, {"conditionGroupElementId":2, "elementOrder":2, "status":"A", "condition":{"conditionId":270, "description":"CONDICION 1/0", "validFrom":"2020-01-16", "validUntil":"2987-03-20", "status":"A"}, "conditionGroup":{"conditionGroupId":1, "description":"Prueba OR", "status":"A", "type":"CONDITION"}}]}
```

## ANATOMIA DE CONDITIOS GROUP

Es importante, primero crear las condiciones y probarlas de forma independiente. A continuación mostramos la relacion de **Grupo de Condiciones** con condiciones.



## QUICKWIN\_MDS.CONDITION\_GROUP

Tabla en la que se configuran los grupos de condiciones.

NAME	TYPE	NULLABLE	DEFAULT	COMMENTS
CONDITION_GROUP_ID	NUMBER	N		Clave Primaria
DESCRIPTION	VARCHAR2(255)	N	"	Descripcion
TYPE	VARCHAR2(15)	N		Tipo de Agrupacion: CONDITION , CONDITION_GROUP
STATUS	VARCHAR2(1)	N	'A'	Estado logico: A, I

## QUICKWIN\_MDS.CONDITION\_GROUP\_ELEMENTS

Tabla en la que se configuran todas las condiciones u otros grupos de condiciones.

NAME	TYPE	NULLABLE	DEFAULT
CONDITION_GROUP_ELEMENT_ID	NUMBER	N	
CONDITION_GROUP_ID	NUMBER	N	
CONDITION_ID	NUMBER	Y	
CONDITION_GROUP_ID_ELEMENT	NUMBER	Y	
ELEMENT_ORDER	NUMBER	N	
FORWARD_OPERATOR	VARCHAR2(3)	Y	
STATUS	VARCHAR2(1)	N	'A'

**CONDITION\_GROUP\_ELEMENT\_ID:** Id numérico.

**CONDITION\_GROUP\_ID:** Id numérico de grupo de condición.

**CONDITION\_GROUP\_ID\_ELEMENT:** Id de otro grupo de condiciones, en este caso, evaluar grupo de grupos de condiciones.

**ELEMENT\_ORDER:** Orden de ejecución de la evaluación.

**FORWARD\_OPERATOR:** AND u OR. Operador lógico.

**STATUS:** Estado A de Activo, I de Inactivo.

## CÓDIGO DE EXITO Y MENSAJE DE ERROR

### ESTATICOS

Por defecto los códigos de éxito **SUCCESS\_CODE**, código de error **FAIL\_CODE**, mensaje de éxito **SUCCESS\_MESSAGE** y mensaje de error **FAILURE\_MESSAGE** han sido estático como vemos en el siguiente ejemplo.

### Ejemplo de configuracion

COLUMNAS	VALOR
CONDITION_PROPERTY_ID	249
CONDITION_ID	175
SUBSCRIBER_PROPERTY_ID	LINEA_ESTA_EN_AXIS
ORDER_BY	10
DATA_TYPE	VARCHAR
DATA_TYPE_FORMAT	
EXPRESSION_CONDITION	[SP<>DM]
FAIL_MESSAGE	El numero si se encuentra
DOMAIN_ID	233
MIN_NUM_DOMAIN_ELEMENTS	1
MAX_NUM_DOMAIN_ELEMENTS	1
VALID_FROM	17/10/2019 11:26
VALID_UNTIL	20/03/2987 23:59
STATUS	A
SUCCESS_MESSAGE	ERROR: No se encontro el numero de telefono
FAIL_CODE	-121
SUCCESS_CODE	0

## DINÁMICOS

Existe dos formas de hacer que **FAIL\_CODE,FAILURE\_MESSAGE,SUCCESS\_CODE y SUCCESS\_MESSAGE** sean dinámicos.

- Haciendo uso de un Subscriber Properties. En vez de un valor estático podemos poner un Subscriber Properties entre corchetes: [MY\_SUBSCRIBER\_PROPERTIES]. Recuerde que solo se toma el primer valor de la lista.
  - Asumiendo que MY\_SUBSCRIBER\_PROPERTIES = ["HOLA MUNDO","HELLO WORLD"], entonces la salida seria "**HOLA MUNDO**".
- Haciendo uso de una funcion en linea denominada Interpolacion de mensaje.
  - Esta funcion necesita definir un caracter separador de elemento de una lista despues de '**SEP=**'. Si elegimos usar ';', entonces empieza de esta forma: SEP=;
  - Recibe la lista delimitada por el separador.
    - SEP=;SUBSCRIBER\_PROPERTIES1;SUBSCRIBER\_PROPERTIES2;
    - Si no encuentra algún subscriber properties definido en la lista este es un valor estático.
  - El ultimo elemento es el formato del String que se desea basado en FORMAT del String en Java usando '%'

1. SEP=;SUBSCRIBER\_PROPERTIES1;SUBSCRIBER\_PROPERTIES2;**El valor de % no es igual a %.**

1. En este caso toma el primer elemento del Subscriber y lo presenta como mensaje. Asumiendo que SUBSCRIBER\_PROPERTIES1 = ["100","500"] y SUBSCRIBER\_PROPERTIES2 = ["200","3000"], entonces seria **El valor de 100 no es igual a 200.**

2. Si el SUBSCRIBER\_PROPERTIES2 no existiese. **El valor de 100 no es igual a SUBSCRIBER\_PROPERTIES2.**

3. E incluso SEP=;[SUBSCRIBER\_PROPERTIES1];%

1. Si [SUBSCRIBER\_PROPERTIES1] no existe entonces tendríamos [SUBSCRIBER\_PROPERTIES1].

2. Pero si quisieramos lograr lo mismo con otro subscriber, pudieramos hacer lo siguiente. Asumiendo que SUBSCRIBER\_PROPERTIES2 = ["SUBSCRIBER\_PROPERTIES1"] entonces SEP=;SUBSCRIBER\_PROPERTIES2;[%] es igual a ["SUBSCRIBER\_PROPERTIES1"]

**Nota:** Si una expresión de interpolacion de mensaje da como resultado un subscriber properties entre corchete, finalmente se presentará el mensaje de ese Subscriber Properties.

## Ejemplos de configuracion.

En este caso, "|" es el separador de la lista, mientras que PRUEBA\_GROUP\_COND1 contiene el valor a mostrar.

SQLOutputStatistics

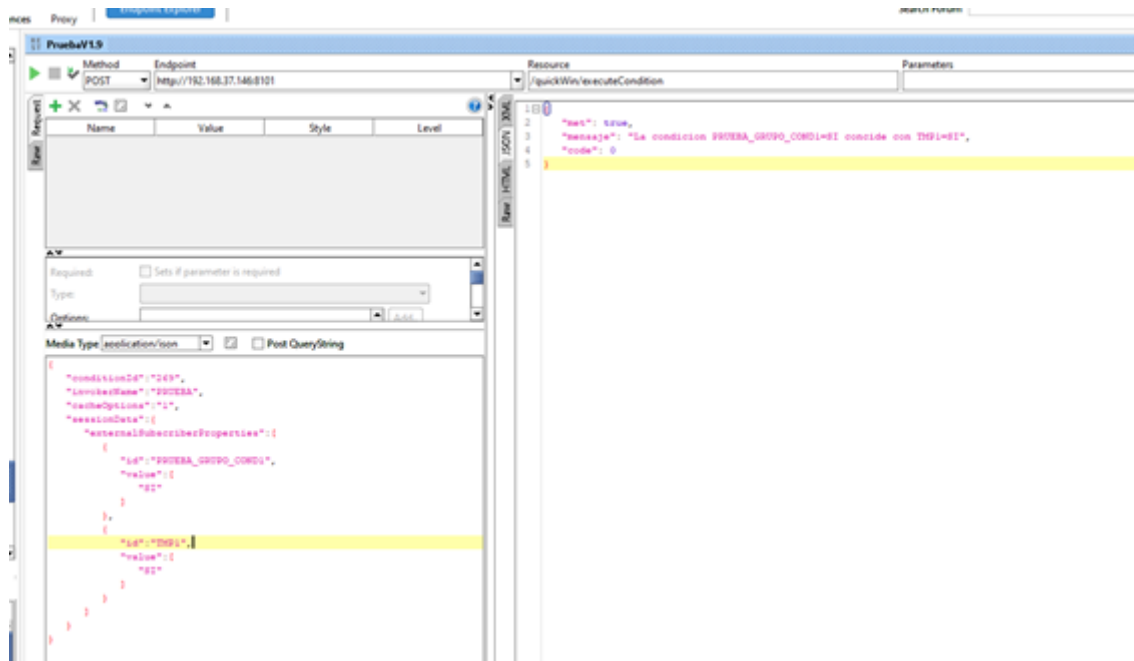
```
select T.*,ROWID from condition_properties t where t.condition_id = 269;
```

<

El mismo formato de configuracion aplica para **FAIL\_CODE,FAILURE\_MESSAGE,SUCCESS\_CODE y SUCCESS\_MESSAGE.**

STATUS	SUCCESS_MESSAGE	FAIL_CODE
A	SEP=#PRUEBA_GRUPO_COND1#TMP1#La condicion PRUEBA_GRUPO_COND1=%s concide con TMP1=%s	-121

En este ejemplo vemos el mensaje como resultado de la interpolacion.



# PLSQL SCRIPT

## CREATE CONDITIONS PLSQL

**Nota:** La primera vez que se crean las configuraciones en **QuickWin** no requieren actualizar memoria antes de consumirse por primera vez el servicio. Cambios a la configuracion posterior, requiere actualizar caché. Algunos servicios como conditions, invokes tienen su propia URL de refresh de memoria. Si la condición está atada a una campaña, se puede refrescar la campaña, sin embargo, es un proceso ineficiente, ya que el refresh de la campaña actualizará todo.

```
declare

l_subscriber_property_id varchar(32767) := '';
l_condition_id number := 0;
l_domain_id number := 0;
l_domain_value_range_id number := 0;
l_condition_property_id number := 0;
lv_identifier varchar(15) := '[QWV1-MGR] ';
begin

-- CONDITION
select nvl(max(c.condition_id), 0) + 1 into l_condition_id from CONDITION c;
insert into CONDITION (CONDITION_ID, DESCRIPTION)
values (l_condition_id, lv_identifier||'Comprobar si
CHANNEL==PROVISIONING_AMCO');

-- SUBSCRIBER_PROPERTIES
l_subscriber_property_id := 'EVENT_NAME';

--insert into SUBSCRIBER_PROPERTIES (SUBSCRIBER_PROPERTY_ID, DESCRIPTION,
STATUS) values (l_subscriber_property_id, 'Canal de Origen', 'A');

-- DOMAIN
```



```

select nvl(max(d.domain_id), 0) + 1 into l_domain_id from DOMAIN d;
insert into DOMAIN (DOMAIN_ID, DESCRIPTION, STATUS)
values (l_domain_id,lv_identifier|| 'Tipo de Evento', 'A');

-- Si son mas valores, agregarlos con comas, (select 'a,b,c' from dual)
INSERT INTO DOMAIN_ELEMENTS
SELECT rownum +
      (SELECT NVL(MAX(DE.DOMAIN_ELEMENT_ID),0) + 1 FROM DOMAIN_ELEMENTS DE)
DOMAIN_ELEMENT_ID,
      trim(regexp_substr(valor, '^[,]+', 1, level)) ELEMENT_VALUE,
      l_domain_id DOMAIN_ID, --Dominio al que pertenece
      'A' STATUS
FROM (select 'PROVISIONING_AMCO' valor from dual) t
CONNECT BY instr(valor, ',', 1, level - 1) > 0;

-- DOMAIN_VALUE_RANGES
/* select nvl(max(dvr.domain_value_range_id), 0) + 1 into
l_domain_value_range_id from DOMAIN_VALUE_RANGES dvr;
insert into DOMAIN_VALUE_RANGES (DOMAIN_VALUE_RANGE_ID, DOMAIN_ID, FROM_VALUE,
TO_VALUE, ORDER_BY, STATUS)
values (l_domain_value_range_id, l_domain_id, '1', '10', 10, 'A');*/

-- CONDITION_PROPERTIES
select nvl(max(cp.condition_property_id), 0) + 1 into l_condition_property_id
from CONDITION_PROPERTIES cp;
insert into CONDITION_PROPERTIES (CONDITION_PROPERTY_ID, CONDITION_ID,
SUBSCRIBER_PROPERTY_ID, DATA_TYPE, EXPRESSION_CONDITION, DOMAIN_ID,
MIN_NUM_DOMAIN_ELEMENTS, MAX_NUM_DOMAIN_ELEMENTS,SUCCESS_MESSAGE,FAIL_MESSAGE)
values (l_condition_property_id, l_condition_id, l_subscriber_property_id,
'VARCHAR', '[SP=DM]', l_domain_id, null, null,'EL evento Recibido es
Cancelar','El evento no es CANCELAR');

dbms_output.put_line('l_domain_id' || l_domain_id);
dbms_output.put_line('l_condition_id ' || l_condition_id);
dbms_output.put_line('SUBSCRIBER_PROPERTIES ' || l_subscriber_property_id);

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

dbms_output.put_line('http://192.168.37.146:8101/quickwin/clearConditionById/' ||
l_condition_id);
dbms_output.put_line('-----GUARDE ESTE SCRIPT POR MAYOR SEGURIDAD-----
-----');

--DELETE FROM CONDITION_PROPERTIES t where t.condition_property_id = ;

dbms_output.put_line('-----SELECT-----
-');
dbms_output.put_line('SELECT * FROM DOMAIN_VALUE_RANGES t where t.domain_id =
'''||l_domain_id||''';');

```

```

dbms_output.put_line('SELECT * FROM DOMAIN_ELEMENTS t where t.domain_id =
'''||l_domain_id||''';');
dbms_output.put_line('SELECT * FROM DOMAIN t where t.domain_id =
'''||l_domain_id||''';');
dbms_output.put_line('SELECT * FROM CONDITION_PROPERTIES T WHERE T.CONDITION_ID
= '''||l_condition_id||''';');
dbms_output.put_line('SELECT * FROM SUBSCRIBER_PROPERTIES t where
t.Condition_Id = '''||l_condition_id||''';');
dbms_output.put_line('SELECT * FROM CONDITION t where t.condition_id =
'''||l_condition_id||''';');
dbms_output.put_line('http://192.168.37.146:8101/quickwin/findConditionById/'||l_
_condition_id);

```

```

dbms_output.put_line('-----UPDATE STATUS-----
-----');
dbms_output.put_line('UPDATE DOMAIN_VALUE_RANGES t set status = 'I' where
t.domain_id = '''||l_domain_id||''';');
dbms_output.put_line('UPDATE DOMAIN_ELEMENTS t set status = 'I' where
t.domain_id = '''||l_domain_id||''';');
dbms_output.put_line('UPDATE DOMAIN t set status = 'I' where t.domain_id =
'''||l_domain_id||''';');
dbms_output.put_line('UPDATE CONDITION_PROPERTIES T set status = 'I' WHERE
T.CONDITION_ID = '''||l_condition_id||''';');
dbms_output.put_line('UPDATE SUBSCRIBER_PROPERTIES t set status = 'I' where
t.Condition_Id = '''||l_condition_id||''';');
dbms_output.put_line('UPDATE CONDITION t set status = 'I' where
t.condition_id = '''||l_condition_id||''';');
dbms_output.put_line('http://192.168.37.146:8101/quickwin/clearConditionById/'||l_
condition_id);

```

```

dbms_output.put_line('-----DELETE-----
-');
dbms_output.put_line('DELETE FROM DOMAIN_VALUE_RANGES t where t.domain_id =
'''||l_domain_id||''';');
dbms_output.put_line('DELETE FROM DOMAIN_ELEMENTS t where t.domain_id =
'''||l_domain_id||''';');
dbms_output.put_line('DELETE FROM DOMAIN t where t.domain_id =
'''||l_domain_id||''';');
dbms_output.put_line('DELETE FROM CONDITION_PROPERTIES T WHERE T.CONDITION_ID =
'''||l_condition_id||''';');
--dbms_output.put_line('DELETE FROM SUBSCRIBER_PROPERTIES t where
t.Condition_Id = '''||l_condition_id||''';');
dbms_output.put_line('DELETE FROM CONDITION t where t.condition_id =
'''||l_condition_id||''';');

```

```

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

```

```

dbms_output.put_line('http://192.168.37.146:8101/quickwin/executeCondition');

```

```

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

```

```

dbms_output.put_line('{

```

```

"conditionId": "||l_condition_id||",
"invokerName": "TestQuickwin",
"cacheOptions": "0",
"sessionData": {
    "externalSubscriberProperties": []);

FOR subs IN (SELECT T.SUBSCRIBER_PROPERTY_ID FROM CONDITION_PROPERTIES T WHERE
T.CONDITION_ID = l_condition_id) LOOP
dbms_output.put_line('          {"id":
"||subs.subscriber_property_id||", "value": [""],');
END LOOP;
-- DATOS DE LA FUNCION
dbms_output.put_line('          {"id": "||'DUMMY'||", "value": [""],');

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

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

```

Al ejecutar este Script te dará un **request**:

```

{
  "conditionId": "1147",
  "invokerName": "TestQuickwin",
  "cacheOptions": "0",
  "sessionData": {
    "externalSubscriberProperties": [
      {"id": "EVENT_NAME", "value": [""],
      {"id": "DUMMY", "value": [""]}
    ]
  }
}

```

## CREATE CONDITION GROUPS PLSQL

```

declare
-- DELETE INVOKES, DEBERIA SER POR ID DE GRUPO O NOMBRE UNICO
lv_condition_description varchar(255) := '[QWV1-MGR] ';
-- REEMPLAZAR POR LAS CONDICIONES
lv_condition_id1 number := 0;
-- REEMPLAZAR POR LAS CONDICIONES
lv_condition_id2 number := 0;
lv_condition_group_id number := 0;

```

```

BEGIN
select nvl(max(c.condition_group_id), 0) + 1 into ln_condition_group_id from
condition_group c;
insert into condition_group (CONDITION_GROUP_ID, DESCRIPTION,TYPE,STATUS)
values (ln_condition_group_id, lv_condition_description||'Comprobar si el Canal
es SGA y Oferta CANONICA PRIMARIA', 'CONDITION', 'A');
-- NOTE QUE PUEDE USAR AND U OR, PARA FORWARD_OPERATOR
insert into condition_group_elements (CONDITION_GROUP_ELEMENT_ID,
CONDITION_GROUP_ID, CONDITION_ID, ELEMENT_ORDER, FORWARD_OPERATOR, STATUS)
values ((select nvl(max(cp.condition_group_element_id), 0) + 1 from
condition_group_elements cp), ln_condition_group_id, ln_condition_id1, 10,
'AND', 'A');

insert into condition_group_elements (CONDITION_GROUP_ELEMENT_ID,
CONDITION_GROUP_ID, CONDITION_ID, ELEMENT_ORDER, FORWARD_OPERATOR, STATUS)
values ((select nvl(max(cp.condition_group_element_id), 0) + 1 from
condition_group_elements cp), ln_condition_group_id, ln_condition_id2, 20, '',
'A');

dbms_output.put_line('ln_condition_group_id number :=
'||ln_condition_group_id||');

dbms_output.put_line('-----SELECT-----
-');
dbms_output.put_line('SELECT * FROM QUICKWIN_MDS.condition_group_elements t
where t.CONDITION_GROUP_ID = ''||ln_condition_group_id||''');
dbms_output.put_line('SELECT * FROM QUICKWIN_MDS.condition_group t where
t.CONDITION_GROUP_ID = ''||ln_condition_group_id||''');
dbms_output.put_line('http://192.168.37.146:8101/quickwin/findConditionGroupById
/'||ln_condition_group_id);

dbms_output.put_line('-----UPDATE STATUS-----
-----');
dbms_output.put_line('UPDATE QUICKWIN_MDS.condition_group_elements t set status
= ''I'' where t.CONDITION_GROUP_ID = ''||ln_condition_group_id||''');
dbms_output.put_line('UPDATE QUICKWIN_MDS.condition_group t set status = ''I''
where t.CONDITION_GROUP_ID = ''||ln_condition_group_id||''');
dbms_output.put_line('http://192.168.37.146:8101/quickwin/clearConditionGroupByI
d/'||ln_condition_group_id);

dbms_output.put_line('-----DELETE-----
-');
dbms_output.put_line('DELETE FROM condition_group_elements t where
t.CONDITION_GROUP_ID = ''||ln_condition_group_id||''');
dbms_output.put_line('DELETE FROM condition_group t where t.CONDITION_GROUP_ID =
''||ln_condition_group_id||''');

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

dbms_output.put_line('http://192.168.37.146:8101/quickwin/executeDefinedFunction
s');

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

```

```

dbms_output.put_line('{
  "conditionGroupId": "'||ln_condition_group_id||'",
  "invokerName":"TestQuickwin",
  "cacheOptions":"0",
  "sessionData": {
    "externalSubscriberProperties": [']);

FOR subs IN (SELECT DISTINCT CP.SUBSCRIBER_PROPERTY_ID FROM
CONDITION_PROPERTIES CP, CONDITION_GROUP_ELEMENTS CE , CONDITION_GROUP CG WHERE
CP.CONDITION_ID = CE.CONDITION_ID AND CE.CONDITION_GROUP_ID =
CG.CONDITION_GROUP_ID AND CG.CONDITION_GROUP_ID = ln_condition_group_id ) LOOP

dbms_output.put_line('          {"id":
"'||subs.subscriber_property_id||'", "value": [""]}');
END LOOP;
-- DATOS DE LA FUNCION
dbms_output.put_line('          {"id": "'||'DUMMY'||'", "value": [""]}');

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

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

```

Al ejecutar este Script te dará el siguiente **request**:

```

{
  "conditionGroupId": "112",
  "invokerName":"TestQuickwin",
  "cacheOptions":"0",
  "sessionData": {
    "externalSubscriberProperties": [
      {"id": "DUMMY", "value": [""]}
    ]
  }
}

```

## DELETE CONDITIONS AND CONDITION GROUPS PLSQL

```

declare
-- DELETE INVOKES, DEBERIA SER POR ID DE GRUPO O NOMBRE UNICO
lv_condition_description varchar(255) := '[QWV1-MGR]';

BEGIN

```

```

-- DELETE CONDITION GROUPS ELEMENT
DELETE FROM CONDITION_GROUP_ELEMENTS T
WHERE T.CONDITION_GROUP_ID IN
    (SELECT G.CONDITION_GROUP_ID
     FROM CONDITION_GROUP G
     WHERE G.DESCRPTION LIKE lv_condition_description);

-- ELIMINAR CONDITION GROUPS ELEMENTS
DELETE FROM CONDITION_GROUP T WHERE T.DESCRPTION LIKE lv_condition_description;

-- DELETE CONDITION PROPERTIES
DELETE FROM CONDITION_PROPERTIES CP
WHERE CP.CONDITION_ID IN
    (SELECT T.CONDITION_ID
     FROM CONDITION T
     WHERE T.DESCRPTION LIKE lv_condition_description);

-- DELETE DOMAIN ELEMENTE
DELETE FROM DOMAIN_ELEMENTS DE
WHERE DE.DOMAIN_ID IN
    (SELECT D.DOMAIN_ID
     FROM DOMAIN D WHERE D.DESCRPTION LIKE lv_condition_description);

-- DELETE DOMAIN
DELETE FROM DOMAIN D
WHERE D.DOMAIN_ID IN
    (SELECT D.DOMAIN_ID
     FROM DOMAIN D WHERE D.DESCRPTION LIKE lv_condition_description);

-- ELIMINAR CONDITION
DELETE FROM CONDITION T WHERE T.DESCRPTION LIKE lv_condition_description;

-- OPCIONAL, PERO SE RECOMIENDA BORRAR ESPECIALMENTE SI SOLO LAS USAN LAS
CONDITIONS
/*DELETE FROM SUBSCRIBER_PROPERTIES T WHERE T.SUBSCRIBER_PROPERTY_ID =
'CANAL_OPERACION'; */
--chequear los errores
exception when others then
    rollback;
    dbms_output.put_line(sqlerrm);
end;

```

## Support

---

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

## Stay in touch

---

- Leader Architec - TIC Manuel García
- Mail - [mgarcia@claro.com.ec](mailto:mgarcia@claro.com.ec)
- Sponsor - TIC Guillermo Proaño
- Mail - [gproano@claro.com.ec](mailto:gproano@claro.com.ec)
- Developers - Fernando Andrade

- Mail - [fernando.andrade@gizlocorp.com](mailto:fernando.andrade@gizlocorp.com)