

INDICE

DYNAMIC FUNCTIONS o FUNCIONES DINAMICAS

Las **funciones** dinámicas o **functions** en **QuickWin** sirven para realizar lógica que **QuickWin** no puede resolver. Son dinámicas porque su parámetro "**INPUT**" es cualquier **JSON** válido. Las funciones por si sola no pueden ser usadas en **QuickWin**. Para poderlas usar deberían ser configuradas vía **INVOKE**. Sin embargo existe otro componente llamado **DEFINED FUNCTIONS**, que permite llamar a cualquiera de ellas sin necesidad de configurar un **INVOKE** de forma interna por **QuickWin**.

REQUEST

Ejemplo de una llamada a una función dinámica.

POST <http://192.168.37.148:8101/quickWin/executeDynamycFunctions> HTTP/1.1

```
{
  "idFunction": "1",
  "input": {
    "param1": 3,
    "param2": 7
  }
}
```

idFunction: Id de la función configurada.

input: recibe un objeto JSON con los parámetros. Este objeto JSON es enviado a la función configurada.

Nota: el nombre de parámetro de la función del ejemplo (**jsonParams**) puede ser cualquiera, sin embargo todas estas funciones solo reciben 1 parámetro.

```
function suma(jsonParams) {
  var params = JSON.parse(jsonParams);
  var a = params.param1;
  var b = params.param2;
  var sum = a + b;
  var text = '{ "suma" : '+sum+' }';
  return text;
}
```

RESPONSE

```
{
  "success": true,
  "message": "OK",
  "code": 0,
  "result": {"suma": 10}
}
```

success: true si fue exitosa la ejecución.

message: mensaje de éxito o error.

result: es un objeto JSON dinámico formado por el RETURN de la función. También puede ser un solo valor.

CLEAR CACHE

Cualquier cambio en la función requiere **refresh**.

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

```
{"refreshMessage":"Refresh en progreso","updated":true,"functions":null}
```

##

REQUEST DE FUNCION USADA POR FUNCION DEFINIDA.

Nota: Una función para que pueda ser llamada por un **DEFINED FUNTION** debe cumplir cierto formato, como explicamos mas adelante. Por lo pronto, aquí hay un ejemplo de un contrato de servicio.

EJEMPLO

POST <http://192.168.37.148:8101/quickWin/executeDynamycFunctions> HTTP/1.1

```
{
  "idFunction": "26",
  "input": {
    "externalSubscriberProperties": [
      {
        "id": "CLAROVIDEO_OFFERID",
        "value": [
```

```

        [{"offerCode":"3653","name":"Claro video costo
GPON","offerType":"GeneralOffer","startDate":"23/02/2021","endDate
":"","origin":"LOCAL","price":
{"taxes":"","discounts":"","unit":"DOLLARS","amountBeforeTaxes":10
.0,"total":8.8},"complexProducts":"","products":{"product":
{"productId":"PACK-
DIAS","description":"3655","shortName":"10000030","expiryTime":30,"e
xpiryTimeUnit":"dias"}},{"fundingInfo":"","alias":{"category":
{"name":"AMCO","description":"Ofertas Externas","alias":{"item":
{"name":"Costo","identifier":"10002002"}}}},{"locations":"","offe
rDetail":{"detail":
{"longDescription":"","productType":"3236","additionalComment":"Adici
onal","featuresDetails":"FIJA"}},{"mediaInfo":[]},
{"offerCode":"185","name":"Claro video costo
HFC","offerType":"GeneralOffer","startDate":"23/02/2021","endDate\
":"","origin":"LOCAL","price":
{"taxes":"","discounts":"","unit":"DOLLARS","amountBeforeTaxes":10
.0,"total":8.8},"complexProducts":"","products":{"product":
{"productId":"PACK-
DIAS","description":"3654","shortName":"10000030","expiryTime":30,"e
xpiryTimeUnit":"dias"}},{"fundingInfo":"","alias":{"category":
{"name":"AMCO","description":"Ofertas Externas","alias":{"item":
{"name":"Costo","identifier":"10002002"}}}},{"locations":"","offe
rDetail":{"detail":
{"longDescription":"","productType":"2600","additionalComment":"Adici
onal","featuresDetails":"FIJA"}},{"mediaInfo":[]},
{"offerCode":"3238","name":"Claro video
GPON","offerType":"GeneralOffer","startDate":"23/02/2021","endDate
":"","origin":"LOCAL","price":
{"taxes":"","discounts":"","unit":"DOLLARS","amountBeforeTaxes":10
.0,"total":8.8},"complexProducts":"","products":"","fundingInfo":"","
","alias":{"category":[{"name":"AMCO","description":"Ofertas
Externas","alias":{"item":
{"name":"Costo","identifier":"10002002"}}}}}},{"locations":"","offe
rDetail":{"detail":
{"longDescription":"","productType":"3232","additionalComment":"Adici
onal","featuresDetails":"FIJA"}},{"mediaInfo":[]},
{"offerCode":"400132377","name":"Suscripcion Claro video mensual 6,39
dolares","offerType":"GeneralOffer","startDate":"22/10/2020","endD
ate":"","origin":"LOCAL","price":
{"taxes":"","discounts":"","unit":"DOLLARS","amountBeforeTaxes":10
.0,"total":8.8},"complexProducts":"","products":{"product":
{"productId":"PACK-
DIAS","description":"400132408","shortName":"10000030","expiryTime":3
0,"expiryTimeUnit":"dias"}},{"fundingInfo":"","alias":{"category":
{"name":"AMCO","description":"Ofertas Externas","alias":{"item":
{"name":"Costo","identifier":"10002002"}}}}}},{"locations":"","offe
rDetail":{"detail":
{"longDescription":"","productType":"","additionalComment":"Adicional
","featuresDetails":"MOVIL"}},{"mediaInfo":
{"key":"image.jpg","value":"","/dockerappsvol/dockerappdata/component
//images//"}]}]
    },
    {
        "id": "ID_PLAN",
        "value": [
            "2600"

```

```

    ]
  },
  {
    "id": "CRM",
    "value": [
      "SGA"
    ]
  }
]
}
}
}

```

RESPONSE

```

{
  "success": true,
  "message": "OK",
  "code": 0,
  "sessionData": {"externalSubscriberProperties": [ {
    "id": "OFFER_ID",
    "value": ["185"]
  }]}
}

```

FUNCION JAVASCRIPT USADA POR FUNCION DEFINIDA.

idFunction: 26

```

function getOfferCode(jsonParams) {
  var params = JSON.parse(jsonParams);
  var res;
  var offerCodeId;
  var offerDetailObject1;
  var detailObject;
  var productType;
  var arrayProductType;
  var i;
  var p;
  //var c;
  var externalsubs = params.externalSubscriberProperties;
  var jsonOffersSP = externalsubs.filter(function(x) {
    return x.id === 'CLAROVVIDEO_OFFERID';
  });

  var id_plan = externalsubs.filter(function(x) {
    return x.id === 'ID_PLAN';
  });

  var jsonOffers = jsonOffersSP[0];
  jsonOffers.value[0] = jsonOffers.value[0].replace(/\n/g, "\\n")
    .replace(/\\/g, "\\")
    .replace(/"/g, "\\")
    .replace(/&/g, "\\&")
    .replace(/r/g, "\\r")
    .replace(/t/g, "\\t")
    .replace(/b/g, "\\b")

```

```

        .replace(/\\f/g, "\\f");

var array = jsonOffers.value[0];
var cadena = JSON.parse(array)
var detailObj;
var plan = id_plan[0].value;
for (i = 0; i < cadena.length; i++) {
    //offerCodeId = (JSON.parse(cadena)[i]).offerCode;
    offerCodeId = (cadena[i]).offerCode;
    if ((cadena[i]).hasOwnProperty("offerDetail")) {
        offerDetailObject1 = cadena[i].offerDetail;
        if ((offerDetailObject1).hasOwnProperty("detail")) {
            //detailObj = offerDetailObject1.detail;
            detailObject = offerDetailObject1.detail;
            for (p = 0; p < detailObject.length; p++) {
                if ((detailObject[p]).hasOwnProperty("productType")) {
                    productType = detailObject[p].productType;
                    //if ((productType).hasOwnProperty("-")) {
                    arrayProductType = productType.split("-");
                    var x=0;
                    var valores="";
                    for (var c = 0; c < arrayProductType.length; c++) {

                        if ((arrayProductType[c]) == plan) {
                            //res = JSON.stringify({externalSubscriberProperties: [{id:
                            [(offerCodeId)] }]});
                            //valores=valores+"-"+arrayProductType[c];
                            res = JSON.stringify({externalSubscriberProperties: [{id:
                            "OFFER_ID",value: [offerCodeId] }]});
                            return res;
                            x++;
                        }
                    }
                }
            }
        }
    }
}
res = JSON.stringify({externalSubscriberProperties: [{id: "OFFER_ID",value:
["0"] }]});
return res;

}

```

ANATOMIA DE FUNCIONES DINAMICAS

MODELO DE ENTIDADES

FUNCTION_DOMAINS		
Column	Type	
FUNCTION_DOM_ID	NUMBER	
DESCRIPTION	VARCHAR2(255)	
FUNCTION_FILE	VARCHAR2(255)	
SOURCE_CODE	CLOB	
STATUS	VARCHAR2(1)	
FUNCTION_TYPE	VARCHAR2(255)	
Key	Column(s)	
FUNCTION_DOM_ID_PK	FUNCTION_DOM	

FUNCTIONS		
Column	Type	Nullabl
FUNCTION_ID	NUMBER	
FUNCTION_NAME	VARCHAR2(255)	
FUNCTION_DOM_ID	NUMBER	
STATUS	VARCHAR2(1)	Y
RESULT_KEY	VARCHAR2(255)	Y
Key	Column(s)	Type
FUNCTION_ID_PK	FUNCTION_ID	P
FUNCTION_DOM_ID_FK	FUNCTION_DOM_ID	R
CHK_STATUS_FUNCTIONS	STATUS	C
Index	Column(s)	Type
FUNCTION_ID_PK	FUNCTION_ID	unique

QUICKWIN_MDS.FUNCTION_DOMAINS

NAME	TYPE	NULLABLE	DEFAULT
FUNCTION_DOM_ID	NUMBER	N	
DESCRIPTION	VARCHAR2(255)	Y	
FUNCTION_FILE	VARCHAR2(255)	Y	
SOURCE_CODE	CLOB	Y	
STATUS	VARCHAR2(1)	Y	'A'
FUNCTION_TYPE	VARCHAR2(255)	Y	

FUNCTION_DOM_ID: Id numérico.

DESCRIPTION: Descripción de Función.

FUNCTION_FILE: Descontinuado.

SOURCE_CODE: contiene la función a ejecutar. Pueden haber varias funciones en este misma configuración. Depende del campo **FUNCTION_TYPE**, para **JavaScript** o

STATUS: Estado A de Activo, I de Inactivo.

FUNCTION_TYPE: Representa el tipo de lenguaje de la función: **JavaScript** o **groovy**.

groovy

Ejemplo Groovy

```
import groovy.json.JsonSlurper
def factorial(params) {
    def slurper = new JsonSlurper().parseText(params.toString())
    def n = slurper.param1
    def n2 = slurper.param2
    def val = n + n2;
    return "{\"suma2\":\""+val+"\", \"valor\": [\"valor\"]}";
}
```

Ejemplo JavaScript

```
function suma(jsonParams) {  
    var params = JSON.parse(jsonParams);  
    var a = params.param1;  
    var b = params.param2;  
    var sum = a + b;  
    var text = '{ "suma" : '+sum+' }';  
    return text;  
}
```

QUICKWIN_MDS.FUNCTIONS

NAME	TYPE	NULLABLE	DEFAULT
FUNCTION_ID	NUMBER	N	
FUNCTION_NAME	VARCHAR2(255)	N	
FUNCTION_DOMAIN_ID	NUMBER	N	
STATUS	VARCHAR2(1)	Y	'A'
RESULT_KEY	VARCHAR2(255)	Y	'output'

FUNCTION_ID: Id numérico de la función.

FUNCTION_NAME: El nombre de la función a ejecutar.

FUNCTION_DOMAIN_ID: Id de dominio de función.

STATUS: Estado A de Activo, I de Inactivo.

RESULT_KEY: **sessionData** es un nombre obligatorio cuando se llama a la función a través de **DEFINED FUNCTIONS**. Por defecto es **output**.

STANDARD QUE DEBE CUMPLIR UNA FUNCION LLAMADA A TRAVES DE DEFINED FUNCTION

La función debe devolver un objeto con una lista de **SUBSCRIBER_PROPERTIES** identificado por **externalSubscriberProperties**.

Nota: Funciones definida a mayor detalle se explica mas adelante.

```
function createArrayOfOffers(jsonParams) {
  var params = JSON.parse(jsonParams);
  var res;
  var externalsubs = params.externalSubscriberProperties;

  var jsonOffersSP = externalsubs.filter(function(x) {
    return x.id === 'OFFER_ID_HW2';
  });

  res = JSON.stringify({externalSubscriberProperties: [{id: "OFFER_ID_HW",value:
[JSON.stringify(jsonOffersSP[0].value)] }]});

  return res;
}
```

Finalmente {"externalSubscriberProperties":[]} se embebe dentro del objeto **sessionData** que es el **KEY_RESULT** y pasan a ser parte del objeto **SessionData** de **QuickWin**.

```
{
  "success": true,
  "message": "OK",
  "code": 0,
  "sessionData": {"externalSubscriberProperties": [ {
    "id": "OFFER_ID",
    "value": ["185"]
  ]}]
}
```

Nota: Al ser una función llamada por **DEFINED FUNCTIONS** los parámetros (**jsonParams**) representan una lista de parámetros de **SUBSCRIBER_PROPERTIES**. Por ejemplo.

```
[
  {
    "id": "MESSAGE",
    "value": [
      "MESSAGESEQ"
    ]
  },
  {
    "id": "NAMES",
    "value": [
      "KAREN"
    ]
  }
]
```

Los parámetros pueden ser enviados a través del contrato de servicio de **DEFINED FUNCTIONS**. Internamente cuando la función definida es usada por **QuickWin**, éste pasa la lista de todos los **subscriber properties** de **QuickWin** a estas funciones. Recuerde que todas estas funciones solo se inicializan con un solo parámetro. Este parámetro debe recorrerse para extraer los **SUBSCRIBER_PROPERTIES** enviados.


```

var params = JSON.parse(jsonParams);
var res;
var externalsubs = params.externalSubscriberProperties;

var jsonOffersSP = externalsubs.filter(function(x) {
return x.id === 'OFFER_ID_HW2';
});

```

Las funciones definidas tienen sus propios contratos de ejecución. Sin embargo se puede probar a la función que llama a través del contrato de la función.

La forma de enviar los parámetros es dentro del objeto **input.externalSubscriberProperties.***

```

{
  "idFunction": "26",
  "input": {
    "externalSubscriberProperties": [
      {
        "id": "MESSAGE",
        "value": [
          "MESSAGESEQ"
        ]
      },
      {
        "id": "NAMES",
        "value": [
          "KAREN"
        ]
      }
    ]
  }
}

```

Nota: También puede probar la función usando **NODE JS**.

Por ejemplo para probarlo desde NODE, puede guardar el contenido dentro de un documento llamado **"test.js"** y ejecutarlo vía CMD.

test.js

```

function shearchOfferCode(jsonParams) {
var params = JSON.parse(jsonParams);
var res;
var externalsubs = params.externalSubscriberProperties;

var jsonOffersSP = externalsubs.filter(function(x) {
return x.id === 'OFFERID';
});
res = JSON.stringify({externalSubscriberProperties: [{id: "ID_OFFER", value:
[JSON.stringify(jsonOffersSP[0].value)] }]});
return res;
}

// PRUEBA DE PARAMETROS VIA CONSOLE

```

```
console.log(shearchOfferCode(JSON.stringify({externalSubscriberProperties: [{id:
"OFFERID",value: ["1234","5687"]} ]})))
```

Si tiene **node** instalado ubíquese en la ruta y ejecute **node test.js**

```
E:\Arquitectura\Frameworks\Quickwin\Dynamyc Functions and Defined Functions>node
test.js
{"externalSubscriberProperties":[{"id":"ID_OFFER","value":["
["1234","\\"5687\\""]"]}]}
```

PLSQL

CREAR FUNCION DINAMICAS PLSQL

```
declare
-- DELETE INVOKES, DEBERIA SER POR ID DE GRUPO O NOMBRE UNICO
lv_description varchar(255) := '[QWV1-MGR] ';
ln_function_dom_id number := 0;
ln_function_id number := 0;

BEGIN

select nvl(max(t.function_dom_id), 0) + 1 into ln_function_dom_id from
FUNCTION_DOMAINS t;

insert into QUICKWIN_MDS.FUNCTION_DOMAINS (FUNCTION_DOM_ID, DESCRIPTION,
FUNCTION_FILE, SOURCE_CODE, STATUS, FUNCTION_TYPE)
values (ln_function_dom_id, lv_description||'Test de Funcion - Formato para usa
en Defined Function', '', 'function testFunctionForDefinedFunction(jsonParams) {
  var params = JSON.parse(jsonParams);
  var res;
  var externalsubs = params.externalSubscriberProperties;

  var jsonOffersSP = externalsubs.filter(function(x) {
    return x.id === 'OFFERID';
  });
  res = JSON.stringify({externalSubscriberProperties: [{id: "ID_OFFER",value:
[JSON.stringify(jsonOffersSP[0].value)] }]});
  return res;

}', 'A', 'JavaScript');

select nvl(max(f.FUNCTION_ID), 0) + 1 into ln_function_id from
QUICKWIN_MDS.FUNCTIONS f;
-- nombre de funcion: testFunctionForDefinedFunction
insert into QUICKWIN_MDS.FUNCTIONS (FUNCTION_ID, FUNCTION_NAME, FUCTION_DOM_ID,
STATUS, RESULT_KEY)
values (ln_function_id, 'testFunctionForDefinedFunction', ln_function_dom_id,
'A', 'sessionData');

dbms_output.put_line('ln_function_dom_id number := '||ln_function_dom_id||');
dbms_output.put_line('ln_function_id number := '||ln_function_id||');
```

```

dbms_output.put_line('-----SELECT-----');
-');
dbms_output.put_line('SELECT * FROM QUICKWIN_MDS.FUNCTIONS t where
t.fuction_dom_id = '''||ln_function_dom_id||''';');
dbms_output.put_line('SELECT * FROM QUICKWIN_MDS.FUNCTION_DOMAINS t where
t.function_dom_id = '''||ln_function_dom_id||''';');
--
dbms_output.put_line('http://192.168.37.146:8101/quickwin/findDynamycFunctionByI
d/'||ln_function_id);

dbms_output.put_line('-----UPDATE STATUS-----');
-');
dbms_output.put_line('UPDATE QUICKWIN_MDS.FUNCTIONS t set status = 'I' where
t.fuction_dom_id = '''||ln_function_dom_id||''';');
dbms_output.put_line('UPDATE QUICKWIN_MDS.FUNCTION_DOMAINS t set status = 'I'
where t.function_dom_id = '''||ln_function_dom_id||''';');
dbms_output.put_line('http://192.168.37.146:8101/quickwin/clearDynamycFunctionBy
Id/'||ln_function_id);

dbms_output.put_line('-----DELETE-----');
-');
dbms_output.put_line('DELETE FROM FUNCTIONS t where t.fuction_dom_id =
'''||ln_function_dom_id||''';');
dbms_output.put_line('DELETE FROM FUNCTION_DOMAINS t where t.function_dom_id =
'''||ln_function_dom_id||''';');

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

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

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

dbms_output.put_line('{
  "idFunction": '''||ln_function_id||'',
  "input": {
    "externalSubscriberProperties": [
      {
        "id": '''||'OFFERID'||'',
        "value": ["123","456"]
      }
    ]
  }
}');
dbms_output.put_line('  }
}');

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

```

Request:

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

```
{
  "idFunction": "44",
  "input": {
    "externalSubscriberProperties": [
      {
        "id": "OFFERID",
        "value": ["123","456"]
      }
    ]
  }
}
```

Al ejecutar esa función, deberíamos tener una salida así.

```
{
  "success": true,
  "message": "OK",
  "code": 0,
  "sessionData": {"externalSubscriberProperties": [ {
    "id": "ID_OFFER",
    "value": ["\"[\"123\", \"456\"]\""]
  }]}
}
```

ELIMINAR FUNCIONES DINAMICA PLSL

Nota, en caso de **DELETE**, se recomienda hacer una prueba SQL, para asegurase de no borrar otros registros en caso que los filtros estén mal realizados.

```
declare
lv_description varchar(255) := '[QWV1-MGR]%' ;

BEGIN

DELETE FROM QUICKWIN_MDS.FUNCTIONS T
WHERE T.FUNCTION_DOM_ID IN
(SELECT T.FUNCTION_DOM_ID
FROM QUICKWIN_MDS.FUNCTION_DOMAINS T
WHERE T.DESCRPTION LIKE lv_description);

-- ELIMINAR CONDITION GROUPS ELEMENTS
DELETE FROM QUICKWIN_MDS.FUNCTION_DOMAINS T WHERE T.DESCRPTION LIKE
lv_description;
-- RECUPERAR DATOS BORRADOS
-- SELECT * FROM QUICKWIN_MDS.FUNCTIONS AS OF TIMESTAMP (SYSTIMESTAMP - INTERVAL
'3' HOUR);

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

DEFINED FUNCTIONS CON DYNAMIC FUNCTIONS

Hasta ahora, hemos hablado de **DEFINED FUNCTIONS** de forma general, pero que son?

Las funciones definidas se llaman así, porque son funciones que corren en el mismo Core de la aplicación de **Quickwin**. Estas funciones definidas algunas tienen un propósito muy finito. Por ejemplo sumar dos NUMEROS dado dos SUBSCRIBER_PROPERTIES, sumar fechas dado dos SUBSCRIBER_PROPERTIES, y por que no mencionar, EJECUTAR CUALQUIER FUNCION. Tienen un objetivo común, reciben SUBSCRIBER_PROPERTIES y devuelven nuevos **SUBSCRIBER_PROPERTIES** inicializados que se agregan a los datos de Sesión de **QuickWin**.

Esta una lista de todas las funciones definidas que existen en **QuickWin**.

EXECUTE_ANY_FUNCTION es la función definida que usaremos para llamar a las funciones.

DESCRIPCION DE FUNCIONES DEFINIDAS DE QUICKWIN

DEFINED_FUNCTION_ID	DESCRIPTION
ADD_DATE	Agrega una cantidad en una unidad de tiempo especifico a una fecha
CONCAT	Concatena 2 textos ya sean SP o valor fijo
EVALUAR_FECHA	Evalúa si una fecha esta dentro de un rango a partir de 2 fechas
EVALUAR_NUMERO_RANGO	Evalúa si un numero está dentro de un rango a partir de 2 numeros
EVALUAR_TEXTO	Evalúa si el texto cumple una expresión regular
EXECUTE_ANY_FUNCTION	Ejecuta Cualquier Función. Las funciones deben devolver un KEY_RESULT: sessionData y una lista de SUBSCRIBER PROPERTIES en el RETURN. La función recibe una solo parámetro que representa una lista de subscriber properties dentro de la propiedad de objeto de parámetros {externalSubscriberProperties: [{id:"SP",value: ["123"]}]}
FORMAT_DATE	formatea una fecha a partir de un SimpleDateFormat
LENGTH	Extrae la longitud de una cadena
LOOKUP	Busca un valor y lo homologa por otro
MATH_SUBSCRIBERS	Hace operaciones matemáticas con dos SP, suma, resta, divide
REPLACE	Reemplaza un valor por otro
STRING_INTERPOLATION	Ejecuta INTERPOLACION de STRING. Recibe el formato STRING como '%s', y le aplica una lista de SUBCRIB ER PROPERTIES delimitados por un SEPARADOR
SUBSCRIBER-PROPERTY_CONTAINS	Busca un valor dentro del subscriber properties
SUBSTRING	Extrae una parte de una cadena de texto a partir de los índices
TRIM	Elimina los espacios en blanco

Los **defined functions** tienen sus propios contratos de servicios por las cuales se puede probar que el defined function funcione correctamente.

REQUEST

Ejemplo de la ejecución de un Defined Function.

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

```
{
  "definedFunctionCallId": 9,
  "sessionData": {
    "externalSubscriberProperties": [
```

```

{
  "id": "CLAROVVIDEO_OFFERID",
  "value": [
    "[{"offerCode":"3653","name":"Claro video costo
GPON","offerType":"GeneralOffer","startDate":"23/02/2021","endDate
":"","origin":"LOCAL","price":
{"taxes":"","discounts":"","unit":"DOLLARS","amountBeforeTaxes":10
.0,"total":8.8},"complexProducts":"","products":{"product":
{"productId":"PACK-
DIAS","description":"3655","shortName":"10000030","expiryTime":30,"e
xpiryTimeUnit":"dias"}},{"fundingInfo":"","alias":{"category":
{"name":"AMCO","description":"Ofertas Externas","alias":{"item":
{"name":"Costo","identifier":"10002002"}}}}},"locations":"","offe
rDetail":{"detail":
{"longDescription":"","productType":"3236","additionalComment":"Adici
onal","featuresDetails":"FIJA"}},{"mediaInfo":[]},
{"offerCode":"185","name":"Claro video costo
HFC","offerType":"GeneralOffer","startDate":"23/02/2021","endDate\
":"","origin":"LOCAL","price":
{"taxes":"","discounts":"","unit":"DOLLARS","amountBeforeTaxes":10
.0,"total":8.8},"complexProducts":"","products":{"product":
{"productId":"PACK-
DIAS","description":"3654","shortName":"10000030","expiryTime":30,"e
xpiryTimeUnit":"dias"}},{"fundingInfo":"","alias":{"category":
{"name":"AMCO","description":"Ofertas Externas","alias":{"item":
{"name":"Costo","identifier":"10002002"}}}}},"locations":"","offe
rDetail":{"detail":
{"longDescription":"","productType":"2600","additionalComment":"Adici
onal","featuresDetails":"FIJA"}},{"mediaInfo":[]},
{"offerCode":"3238","name":"Claro video
GPON","offerType":"GeneralOffer","startDate":"23/02/2021","endDate
":"","origin":"LOCAL","price":
{"taxes":"","discounts":"","unit":"DOLLARS","amountBeforeTaxes":10
.0,"total":8.8},"complexProducts":"","products":"","fundingInfo":"","
","alias":{"category":[{"name":"AMCO","description":"Ofertas
Externas","alias":{"item":
{"name":"Costo","identifier":"10002002"}}}}}}},"locations":"","offe
rDetail":{"detail":
{"longDescription":"","productType":"3232","additionalComment":"Adici
onal","featuresDetails":"FIJA"}},{"mediaInfo":[]},
{"offerCode":"400132377","name":"Suscripcion Claro video mensual 6,39
dolares","offerType":"GeneralOffer","startDate":"22/10/2020","endb
ate":"","origin":"LOCAL","price":
{"taxes":"","discounts":"","unit":"DOLLARS","amountBeforeTaxes":10
.0,"total":8.8},"complexProducts":"","products":{"product":
{"productId":"PACK-
DIAS","description":"400132408","shortName":"10000030","expiryTime":3
0,"expiryTimeUnit":"dias"}},{"fundingInfo":"","alias":{"category":
{"name":"AMCO","description":"Ofertas Externas","alias":{"item":
{"name":"Costo","identifier":"10002002"}}}}}}},"locations":"","offe
rDetail":{"detail":
{"longDescription":"","productType":"","additionalComment":"Adicional
","featuresDetails":"MOVIL"}},{"mediaInfo":
{"key":"image.jpg","value":"","/dockerappsvol/dockerappdata/component
/\\/images/\\}}}]
  ]
},

```

```

    {
      "id": "ID_PLAN",
      "value": [
        "2600"
      ]
    },
    {
      "id": "CRM",
      "value": [
        "SGA"
      ]
    }
  ]
}

```

definedFunctionCallId: Representa el ID de la **Función Definida Configurada**. Para este caso representa una instancia de función definida: message.

sessionData: Es el objeto que contiene una lista de **Subscriber Properties**. dentro de la propiedad **externalSubscriberProperties**. Aquí se envían los parámetros que necesita la función y los que necesita en si la función definida. Internamente **QuickWin** cuando llama a la función definida le pasa todos los subscriber properties de la sesion.

RESPONSE

```

{
  "functionResult": {
    "id": "$DYNAMIC_FUNCTION_RESULT$",
    "value": [{"externalSubscriberProperties": [{"id": "OFFER_ID", "value": ["185"]}]}]
  },
  "success": true,
  "message": "Funcion ejecutada correctamente",
  "code": 0
}

```

functionResult contiene un objeto con dos propiedades. **ID** representa por defecto **\$DYNAMIC_FUNCTION_RESULT\$** la propiedad que almacena el resultado de la ejecución. (VALUE).

El valor de ID, puede ser cualquier otra propiedad, en el caso de configurarse en los parámetros por defecto del **Defined Functions** para la función definida con id **EXECUTE_ANY_FUNCTION**.

NOTA: Recuerde que si se usa una propiedad que inicialmente esta siendo usada, después de la ejecución se **sobreescribe** con el nuevo valor, cuando **QuickWin** la llama. Es recomendable usar diferentes propiedades si así la necesidad lo amerita.

success: true o false.

message: mensaje de éxito o error en caso de alguno.

code: 0 para éxito, menores que 0 representan errores de configuración.

REFRESH DATA

Para hacer **refresh**.

GET <http://192.168.37.148:8101/quickWin/clearDefinedFunctionCallById/4>

```
{"refreshMessage":"Refresh en progreso","updated":true}
```

GET DATA FROM CACHE

Para consultar datos de función definida.

GET <http://192.168.37.148:8101/quickWin/findDefinedFunctionCallById/4>

```
{"refreshMessage":"Refresh en progreso","updated":true,"definedFunctionBindings":
[{"definedFunctionBindingId":8,"definedFunctionCall":
{"definedFunctionCallId":4,"definedFunction":
{"definedFunctionId":"ADD_DATE","definedFunctionName":"ADD_DATE","status":"A"},"
description":"LLAMADA A ADD DATE","status":"A"},"definedFunctionParams":
{"definedFunctionParamId":12,"definedFunction":
{"definedFunctionId":"ADD_DATE","definedFunctionName":"ADD_DATE","status":"A"},"
name":"UNIDAD","defaultValue":"SECOND","way":"=>","dataType":"TEXT","status":"A"
},"bindingType":"FV","fixedValue":"MINUTE","status":"A"},
{"definedFunctionBindingId":9,"definedFunctionCall":
{"definedFunctionCallId":4,"definedFunction":
{"definedFunctionId":"ADD_DATE","definedFunctionName":"ADD_DATE","status":"A"},"
description":"LLAMADA A ADD DATE","status":"A"},"definedFunctionParams":
{"definedFunctionParamId":13,"definedFunction":
{"definedFunctionId":"ADD_DATE","definedFunctionName":"ADD_DATE","status":"A"},"
name":"VALOR","way":"=>","dataType":"NUMBER","status":"A"},"bindingType":"SP","s
ubscriberProperties":
{"subscriberPropertyId":"$UNIT$","description":"Unit","status":"A"},"status":"A"
}}]}
```

EJEMPLOS DE DEFINED FUNCTIONS

ADD DATE

Configuraciones

Parámetros

DEFINED_FUNCTION_PARAM_ID	DEFINED_FUNCTION_ID	NAME	DEFAULT_VALUE	WAY	DATA_TYPE	STATUS
10	ADD_DATE	FECHA_ENTRADA	\$TRANSACTIONDATE\$	=>	DATE	A
11	ADD_DATE	FORMATO_ENTRADA	dd/MM/yyyy HH:mm:ss	=>	TEXT	A
12	ADD_DATE	UNIDAD	SECOND	=>	TEXT	A
13	ADD_DATE	VALOR		=>	NUMBER	A
14	ADD_DATE	VARIABLE_SALIDA	\$ADDDATE_RESULT\$	<=	TEXT	A

```
select * from DEFINED_FUNCTION_CALL t
where defined_function_id like 'ADD_DATE';
```

DEFINED_FUNCTION_CALL_ID	DEFINED_FUNCTION_ID	DESCRIPTION	STATUS
4	ADD_DATE	LLAMADA A ADD DATE	A

```
-- resultado
select * from DEFINED_FUNCTION_BINDINGS t
where t.defined_function_call_id = 4;
```

DEFINED_FUNCTION_BINDING_ID	DEFINED_FUNCTION_CALL_ID	DEFINED_FUNCTION_PARAM_ID	BINDING_TYPE	SUBSCRIBER_PROPERTY_ID	FIXED_VALUE	STATUS
177	4	10	SP	\$TRANSACTIONDATE_WS\$		A
194	4	11	FV		yyyy-MM-dd'T'HH:mm:ssXXX	A
195	4	14	SP	RESULTDESC		A
8	4	12	FV		MINUTE	A
9	4	13	SP	\$UNIT\$		A

Request

Se suman 5 minutos

```
{
  "definedFunctionCallId": 4,
  "sessionData": {
    "externalSubscriberProperties": [
      {
        "id": "$UNIT$",
        "value": [
          "5"
        ]
      },
      {
        "id": "$TRANSACTIONDATE_WS$",
        "value": [
          "2021-05-12T18:55:29-05:00"
        ]
      },
      {
        "id": "RESULTDESC",
        "value": [
          "OLD_RESULT_ID"
        ]
      }
    ]
  }
}
```

Response

. La suma de la fecha, se la hace en formato Webservice

```
{
  "functionResult": {
    "id": "OLD_RESULT_ID",
    "value": ["2021-05-12T19:00:29-05:00"]
  },
  "success": true,
  "message": "Funcion ejecutada correctamente",
  "code": 0
}
```

STRING INTERPOLATION

Configuraciones

Parámetros

```
select * from DEFINED_FUNCTION_PARAMS t
where defined_function_id like 'STRING_INTERPOLATION';
```

DEFINED_FUNCTION_PARAM_ID	DEFINED_FUNCTION_ID	NAME	DEFAULT_VALUE	WAY	DATA_TYPE	STATUS
65	STRING_INTERPOLATION	OUTPUT_VARIABLE	\$INTERPOLATION_VALUE\$	<=	TEXT	A
52	STRING_INTERPOLATION	STRING		=>	TEXT	A
55	STRING_INTERPOLATION	VALUES		=>	TEXT	A
50	STRING_INTERPOLATION	VALUES_SEPARATOR		=>	TEXT	A

```
select * from DEFINED_FUNCTION_CALL t
where defined_function_id like 'STRING_INTERPOLATION';
```

DEFINED_FUNCTION_CALL_ID	DEFINED_FUNCTION_ID	DESCRIPTION	STATUS
73	STRING_INTERPOLATION	GENERA TOKEN CLARO VIDEO	A

```
select * from DEFINED_FUNCTION_BINDINGS t
where defined_function_call_id = 73;
```

DEFINED_FUNCTION_BINDING_ID	DEFINED_FUNCTION_CALL_ID	DEFINED_FUNCTION_PARAM_ID	BINDING_TYPE	SUBSCRIBER_PROPERTY_ID	FIXED_VALUE	STATUS
200	73	52	FV		%s;phone-context=claro.ec	A
201	73	55	FV		\$\$SUBSCRIBERID\$	A
202	73	65	FV		TOKEN	A

Request

```
{
  "definedFunctionCallId": 73,
  "sessionData": {
    "externalSubscriberProperties": [
      {
        "id": "$SUBSCRIBERID$",
        "value": [
          "123344455"
        ]
      }
    ]
  }
}
```

Response

```
{
  "functionResult": {
    "id": "TOKEN",
    "value": ["123344455;phone-context=claro.ec"]
  },
  "success": true,
  "message": "Funcion ejecutada correctamente",
  "code": 0
}
```

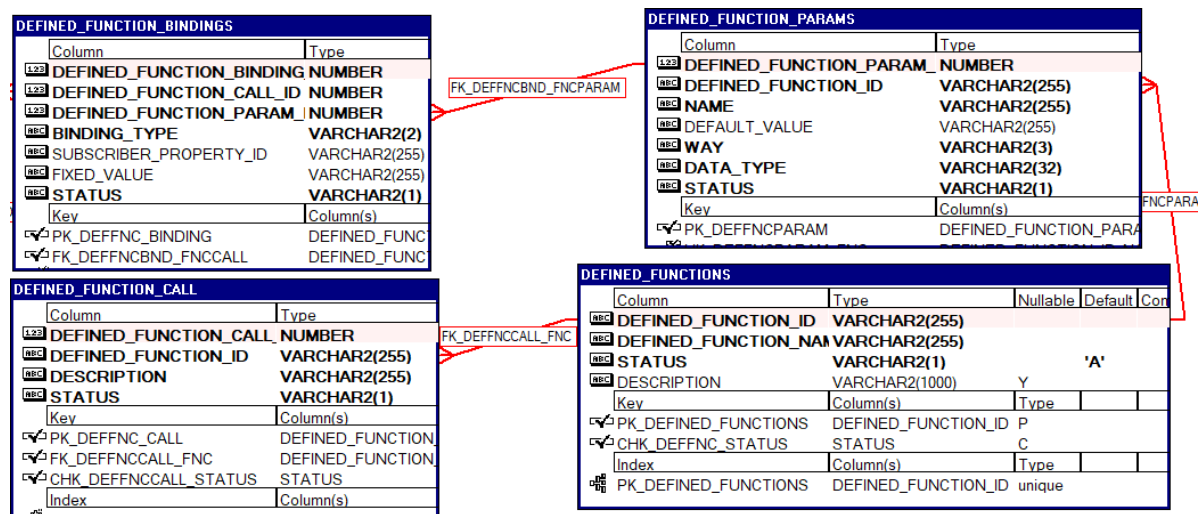
LOOKUP TABLE

Configuraciones

Parámetros

ANATOMIA DE DYNAMIC FUNCTIONS

MODELO DE ENTIDADES



QUICKWIN_MDS.DEFINED_FUNCTIONS.

Esta es una tabla de catálogos de funciones. Normalmente todas las funciones definida comparten la misma filosofía de configuración.

NAME	TYPE	NULLABLE	DEFAULT
DEFINED_FUNCTION_ID	VARCHAR2(255)	N	
DEFINED_FUNCTION_NAME	VARCHAR2(255)	N	
STATUS	VARCHAR2(1)	N	'A'
DESCRIPTION	VARCHAR2(1000)	Y	

DEFINED_FUNCTION_ID: ID de función definida.

DEFINED_FUNCTION_NAME: Nombre de función definida.

STATUS: Estado A de Activo, I de Inactivo.

DESCRIPTION: Descripción de lo que la función hace.

QUICKWIN_MDS.DEFINED_FUNCTION_PARAMS

En esta tabla se definen una lista parámetros de cada función representados por **DEFINED_FUNCTION_PARAM_ID** . Cada parámetros se puede sobrescribir con sus valores por defecto si así se desea en la tabla **DEFINED_FUNCTION_BINDINGS**. Algunos parámetros son obligatorios y no hay valores por defectos.

NOTA: Los Id de **DEFINED_FUNCTION_PARAM_ID** no siempre corresponden en producción. Los **DEFINED_FUNCTION_PARAM_ID** deberían ser consultados por **NAME** y **DEFINED_FUNCTION_ID**.

PARAMETROS DE FUNCIONES DEFINIDAS

DEFINED_FUNCTION_PARAM_ID	DEFINED_FUNCTION_ID	NAME	DEFAULT_VALUE	WAY	DATA_TYPE	STATUS
10	ADD_DATE	FECHA_ENTRADA	\$TRANSACTIONDATE\$	=>	DATE	A
11	ADD_DATE	FORMATO_ENTRADA	dd/MM/yyyy HH:mm:ss	=>	TEXT	A
12	ADD_DATE	UNIDAD	SECOND	=>	TEXT	A
13	ADD_DATE	VALOR		=>	NUMBER	A
14	ADD_DATE	VARIABLE_SALIDA	\$ADDDATE_RESULT\$	<=	TEXT	A
4	CONCAT	TEXTO_A_CONCATENAR		=>	TEXT	A
3	CONCAT	TEXTO_ENTRADA		=>	TEXT	A
5	CONCAT	VARIABLE_SALIDA	\$CONCAT_RESULT\$	<=	TEXT	A
39	EVALUAR_FECHA	FECHA_ENTRADA	\$TRANSACTIONDATE\$	=>	DATE	A
40	EVALUAR_FECHA	FORMATO_ENTRADA	dd/MM/yyyy HH:mm:ss	=>	TEXT	A
41	EVALUAR_FECHA	UNIDAD	SECOND	=>	TEXT	A
42	EVALUAR_FECHA	VALOR		=>	NUMBER	A
43	EVALUAR_FECHA	VARIABLE_SALIDA	\$EVALUAR_FECHA_RESULT\$	<=	TEXT	A
32	EVALUAR_NUMERO_RANGO	NUMERO_ENTRADA		=>	NUMBER	A
37	EVALUAR_NUMERO_RANGO	SEPARADOR_DECIMALES	.	=>	TEXT	A
35	EVALUAR_NUMERO_RANGO	VALOR_MAXIMO		=>	NUMBER	A
36	EVALUAR_NUMERO_RANGO	VALOR_MAXIMO_INCLUSIVO	SI	=>	TEXT	A
33	EVALUAR_NUMERO_RANGO	VALOR_MINIMO		=>	NUMBER	A
34	EVALUAR_NUMERO_RANGO	VALOR_MINIMO_INCLUSIVO	SI	=>	TEXT	A
38	EVALUAR_NUMERO_RANGO	VARIABLE_SALIDA	\$EVALUAR_NUMERO_RESULT\$	<=	TEXT	A
47	EVALUAR_TEXTO	EXPRESION_REGULAR		=>	TEXT	A
46	EVALUAR_TEXTO	TEXTO_ENTRADA		=>	TEXT	A
48	EVALUAR_TEXTO	VARIABLE_SALIDA	\$EVALUARTEXTO_RESULT\$	<=	TEXT	A
58	EXECUTE_ANY_FUNCTION	ADD_SESSION_DATA	YES	=>	TEXT	A
57	EXECUTE_ANY_FUNCTION	FUNCTION_ID		=>	TEXT	A
60	EXECUTE_ANY_FUNCTION	KEY_OUTPUT	sessionData	=>	TEXT	A
59	EXECUTE_ANY_FUNCTION	RESULT_VALUE	\$DYNAMYC_FUNCTION_RESULT\$	<=	TEXT	A
6	FORMAT_DATE	FECHA_ENTRADA	\$TRANSACTIONDATE\$	=>	DATE	A
7	FORMAT_DATE	FORMATO_ENTRADA	dd/MM/yyyy HH:mm:ss	=>	TEXT	A
8	FORMAT_DATE	FORMATO_SALIDA		=>	TEXT	A
9	FORMAT_DATE	VARIABLE_SALIDA	\$FORMATDATE_RESULT\$	<=	DATE	A
44	LENGTH	TEXTO_ENTRADA		=>	TEXT	A
45	LENGTH	VARIABLE_SALIDA	\$LENGTH_RESULT\$	<=	NUMBER	A
30	LOOKUP	CATALOGO		=>	TEXT	A
29	LOOKUP	SEPARADOR_CLAVE_VALOR	:	=>	TEXT	A
28	LOOKUP	SEPARADOR_ELEMENTOS	,	=>	TEXT	A
27	LOOKUP	VALOR_BUSQUEDA		=>	TEXT	A
31	LOOKUP	VARIABLE_SALIDA	\$LOOKUP_RESULT\$	<=	TEXT	A
49	MATH_SUBSCRIBERS	DATA_TYPE_CASTING	INT	=>	TEXT	A
54	MATH_SUBSCRIBERS	OPERATION	ADD	=>	TEXT	A
56	MATH_SUBSCRIBERS	RESULT_VALUE	\$SP_MATH_RESULT\$	<=	TEXT	A
51	MATH_SUBSCRIBERS	VALUE1	500	=>	TEXT	A
53	MATH_SUBSCRIBERS	VALUE2	600	=>	TEXT	A
20	REPLACE	EXPRESION_A_REEMPLAZAR		=>	TEXT	A
21	REPLACE	EXPRESION_REEMPLAZO		=>	TEXT	A
19	REPLACE	TEXTO_ENTRADA		=>	TEXT	A
22	REPLACE	VARIABLE_SALIDA	\$REPLACE_RESULT\$	<=	TEXT	A
65	STRING_INTERPOLATION	OUTPUT_VARIABLE	\$INTERPOLATION_VALUE\$	<=	TEXT	A
52	STRING_INTERPOLATION	STRING		=>	TEXT	A
55	STRING_INTERPOLATION	VALUES		=>	TEXT	A
50	STRING_INTERPOLATION	VALUES_SEPARATOR		=>	TEXT	A
16	SUBSCRIBER-PROPERTY_CONTAINS	DOMINIO		=>	TEXT	A
17	SUBSCRIBER-PROPERTY_CONTAINS	SEPARADOR	,	=>	TEXT	A
15	SUBSCRIBER-PROPERTY_CONTAINS	SUBSCRIBER_PROPERTY		=>	TEXT	A
18	SUBSCRIBER-PROPERTY_CONTAINS	VARIABLE_SALIDA	\$CONTAINS_RESULT\$	<=	TEXT	A
25	SUBSTRING	INDICE_FIN		=>	NUMBER	A
24	SUBSTRING	INDICE_INICIO		=>	NUMBER	A
23	SUBSTRING	TEXTO_ENTRADA		=>	TEXT	A
26	SUBSTRING	VARIABLE_SALIDA	\$SUBSTRING_RESULT\$	<=	TEXT	A
1	TRIM	TEXTO_ENTRADA		=>	TEXT	A

DEFINED_FUNCTION_PARAM_ID	DEFINED_FUNCTION_ID	NAME	DEFAULT_VALUE	WAY	DATA_TYPE	STATUS
2	TRIM	VARIABLE_SALIDA	\$TRIM_RESULTS\$	<=	TEXT	A

QUICKWIN_MDS.DEFINED_FUNCTION_PARAMS.

NAME	TYPE	NULLABLE	DEFAULT	COMMENTS
DEFINED_FUNCTION_PARAM_ID	NUMBER	N		Clave Primaria
DEFINED_FUNCTION_ID	VARCHAR2(255)	N		Referencia a la tabla DEFINED_FUNCTIONS
NAME	VARCHAR2(255)	N		Nombre del parametro
DEFAULT_VALUE	VARCHAR2(255)	Y		Valor por defecto en caso que no exista un binding
WAY	VARCHAR2(3)	N		Sentido de la variable, Entrada = => Salida = <=
DATA_TYPE	VARCHAR2(32)	N		Tipo de datos de la variable TEXT,NUMBER,DATE
STATUS	VARCHAR2(1)	N	'A'	Estado logico

QUICKWIN_MDS.DEFINED_FUNCTION_BINDINGS.

NAME	TYPE	NULLABLE	DEFAULT	COMMENTS
DEFINED_FUNCTION_BINDING_ID	NUMBER	N		Clave Primaria
DEFINED_FUNCTION_CALL_ID	NUMBER	N		Estado Logico
DEFINED_FUNCTION_PARAM_ID	NUMBER	N		Foranea a defined Function Param
BINDING_TYPE	VARCHAR2(2)	N		Tipo de valor que tomar el parametro en el binding, SP SUBSCRIBERPROPERTY, FV: VALOR FIJO
SUBSCRIBER_PROPERTY_ID	VARCHAR2(255)	Y		Foranea del subscriber property a usar
FIXED_VALUE	VARCHAR2(255)	Y		Valor fijo del binding
STATUS	VARCHAR2(1)	N	'A'	

SCRIPT PLSQL

CREATE DEFINED FUNCTION EXECUTE_ANY_FUNCTION PLSQL

```

declare
lv_description varchar(255) := '[QWV1-MGR] ';
lv_defined_function_id varchar(255) := 'EXECUTE_ANY_FUNCTION';
ln_defined_function_call_id number := 0;
ln_id_param number := 0;

BEGIN

```

```

select nvl(max(t.DEFINED_FUNCTION_CALL_ID), 0) + 1 into
ln_defined_function_call_id from DEFINED_FUNCTION_CALL t;

insert into DEFINED_FUNCTION_CALL (DEFINED_FUNCTION_CALL_ID,
DEFINED_FUNCTION_ID, DESCRIPTION, STATUS)
values (ln_defined_function_call_id, lv_defined_function_id, lv_description ||
'Ejemplo de Creacion de Defined Functions', 'A');

-- LOS PARAMATROS QUE SE NECESITAN..
-- PARAMETRO QUE PIDE EL ID DE LA FUNCION A EJECUTAR..
SELECT t.defined_function_param_id into ln_id_param
FROM DEFINED_FUNCTION_PARAMS t
where t.defined_function_id = lv_defined_function_id
and t.name = 'FUNCTION_ID';

-- ESTE VALOR PUEDE SER FIJO, O ENVIADO A TRAVES DE UN SUBSCRIBER_PROPERTIES.
-- BINDING_TYPE ES SP cuando el valro viene en un SUBSCRIBER PROPERTY, Y FV, para
valores FIJOS.
-- PARA ESTE CASO PROBARERMOS LA FUNCION DINAMICA #44 CREADA PREVIAMENTE
insert into DEFINED_FUNCTION_BINDINGS (DEFINED_FUNCTION_BINDING_ID,
DEFINED_FUNCTION_CALL_ID, DEFINED_FUNCTION_PARAM_ID, BINDING_TYPE,
SUBSCRIBER_PROPERTY_ID, FIXED_VALUE, STATUS)
values ((select nvl(max(T.DEFINED_FUNCTION_BINDING_ID), 0) + 1 from
DEFINED_FUNCTION_BINDINGS T), ln_defined_function_call_id, ln_id_param, 'SP',
'FUNCTION_ID', '', 'A');

SELECT t.defined_function_param_id into ln_id_param
FROM DEFINED_FUNCTION_PARAMS t
where t.defined_function_id = lv_defined_function_id
and t.name = 'ADD_SESSION_DATA';

-- este insert es opcional, a pesar que no se envia por parametro o es fijo, no
es necesario, por defecto es YES
insert into DEFINED_FUNCTION_BINDINGS (DEFINED_FUNCTION_BINDING_ID,
DEFINED_FUNCTION_CALL_ID, DEFINED_FUNCTION_PARAM_ID, BINDING_TYPE,
SUBSCRIBER_PROPERTY_ID, FIXED_VALUE, STATUS)
values ((select nvl(max(T.DEFINED_FUNCTION_BINDING_ID), 0) + 1 from
DEFINED_FUNCTION_BINDINGS T), ln_defined_function_call_id, ln_id_param, 'FV',
'', 'YES', 'A');

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

dbms_output.put_line('-----SELECT-----');
dbms_output.put_line('SELECT * FROM QUICKWIN_MDS.DEFINED_FUNCTION_BINDINGS t
where t.defined_function_call_id = ''||ln_defined_function_call_id||''');
dbms_output.put_line('SELECT * FROM QUICKWIN_MDS.DEFINED_FUNCTION_CALL t where
t.defined_function_call_id = ''||ln_defined_function_call_id||''');
dbms_output.put_line('http://192.168.37.146:8101/quickwin/findDefinedFunctionCa
lById/'||ln_defined_function_call_id);

dbms_output.put_line('-----UPDATE STATUS-----');
dbms_output.put_line('UPDATE QUICKWIN_MDS.DEFINED_FUNCTION_BINDINGS t set
status = 'I' where t.defined_function_call_id =
''||ln_defined_function_call_id||''');

```



```

dbms_output.put_line('UPDATE QUICKWIN.MDS.DEFINED_FUNCTION_CALL t set status =
''I'' where t.defined_function_call_id =
'''||ln_defined_function_call_id||''');');
dbms_output.put_line('http://192.168.37.146:8101/quickwin/clearDefinedFunctionCa
llById/'''||ln_defined_function_call_id);

dbms_output.put_line('-----DELETE-----
-');
dbms_output.put_line('DELETE FROM DEFINED_FUNCTION_BINDINGS t where
t.defined_function_call_id = '''||ln_defined_function_call_id||''');');
dbms_output.put_line('DELETE FROM DEFINED_FUNCTION_CALL t where
t.defined_function_call_id = '''||ln_defined_function_call_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('{
  "definedFunctionCallId": '''||ln_defined_function_call_id||'',
  "sessionData": {
    "externalSubscriberProperties": ['');

FOR subs IN (SELECT T.SUBSCRIBER_PROPERTY_ID FROM DEFINED_FUNCTION_BINDINGS T
WHERE T.BINDING_TYPE = 'SP' AND T.DEFINED_FUNCTION_CALL_ID =
ln_defined_function_call_id ) LOOP
dbms_output.put_line('          {"id":
'''||subs.subscriber_property_id||','"value": ["44"]},');
END LOOP;
-- DATOS DE LA FUNCION
dbms_output.put_line('          {"id": '''||'OFFERID' ||','"value":
["1111","2222"]},');

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 nos dará un **request**:

Note que usamos la función con id, 44 que creamos anteriormente. También podemos enviar cualquier función que cumpla el Standard definido y ese valor puede ser fijo en la función oculto por defecto. Adicional enviamos los **SUBSCRIBER_PROPERTIES** que necesita la función a ejecutarse. Para este ejemplo. **OFFERID**.

```
{
  "definedFunctionCallId": "78",
  "sessionData": {
    "externalSubscriberProperties": [
      {"id": "FUNCTION_ID", "value": ["44"]},
      {"id": "OFFERID", "value": ["1111", "2222"]}
    ]
  }
}
```

Con el siguiente resultado.

```
{
  "functionResult": {
    "id": "$DYNAMIC_FUNCTION_RESULT$",
    "value": [{"\"{\"externalSubscriberProperties\": [
      [{\"id\": \"ID_OFFER\", \"value\": [\"1111\", \"2222\"]}]}]}\""]
    ],
    "success": true,
    "message": "Funcion ejecutada correctamente",
    "code": 0
  }
}
```

Nota: Recuerde que internamente **QuickWin** cuando usa las funciones definidas, envía todos los **suscriber properties** con los cuales esta configurado.

ELIMINAR DEFINED FUNCTION PLSQL

```
declare
lv_description varchar(255) := '[QWV1-MGR]';
BEGIN

DELETE FROM QUICKWIN_MDS.DEFINED_FUNCTION_BINDINGS T
WHERE T.DEFINED_FUNCTION_CALL_ID IN
      (SELECT T.DEFINED_FUNCTION_CALL_ID FROM
QUICKWIN_MDS.DEFINED_FUNCTION_CALL T
      WHERE T.DESCRPTION LIKE lv_description);

DELETE FROM DEFINED_FUNCTION_CALL t WHERE T.DESCRPTION LIKE lv_description;
--chequear los errores
exception when others then
  rollback;
  dbms_output.put_line(sqlerrm);
end;
```

Support

QuckickWin 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
- Sponsor - TIC Guillermo Proaño
- Mail - gproano@claro.com.ec
- Developers - Fernando Andrade
- Mail - fernando.andrade@gizlocorp.com