

QR Code: QR Code - Create

```
# -----  
# Purpose:      QR Code - Create  
# Creator:      The Scarpetta Group - James Scarpetta  
# Date:         2024-Apr-10  
# Version:      1.0  
# Notes:  
# Parameters:  
# -----
```

Commit Records/Requests

[No dialog]

#Get parameter.

Set Variable [\$bgColor; Value:QRC::_g_bgcolor]

Set Variable [\$color; Value:QRC::_g_color]

Set Variable [\$data; Value:QRC::_g_data]

Set Variable [\$description; Value:QRC::_g_description]

Set Variable [\$format; Value:QRC::_g_format]

Set Variable [\$size; Value:QRC::_g_size]

#Set api parameters.

Set Variable [\$apiName; Value:"QR Code"]

Set Variable [\$baseUrl; Value:apiBaseUrl (\$apiName)]

#QR code settings.

Set Variable [\$qrCodeSettings; Value:qrCodeSettings (\$data ; \$size ; \$color ; \$bgColor ; \$format)]

#Setup url.

Set Variable [\$url; Value:\$baseUrl & \$qrCodeSettings]

#Create record.

New Record/Request

Set Field [QRC::data; \$data]

Set Field [QRC::description; \$description]

#Perform api call.

Insert from URL [QRC::qr code; \$url]

[Select; No dialog]

#Clear globals

Set Field [QRC::_g_bgcolor; null]

Set Field [QRC::_g_color; null]

Set Field [QRC::_g_data; null]

Set Field [QRC::_g_description; null]

Set Field [QRC::_g_format; null]

Set Field [QRC::_g_size; null]

Commit Records/Requests

[No dialog]

Exit Script []

Weather API: Weather - Process { json }

```
# -----
# Purpose:           Weather - Process { json }
# Creator:           The Scarpetta Group - James Scarpetta
# Date:              2024-Apr-08
# Version:           1.00
# Notes:
# Parameters:
# -----

#Sample JSON.
#JSONSetElement ( "" ;

    [ "location" ; value ; JSONString ] ;
    [ "date1" ; value ; JSONString ] ;
    [ "date2" ; value ; JSONString ]

)

#Get parameters.
Set Variable [ $json; Value:Get ( ScriptParameter ) ]
Set Variable [ $location; Value:JSONGetElement ( $json ; "location" ) ]
Set Variable [ $date1; Value:Let (

    d = JSONGetElement ( $json ; "date1" )

    ;

    date.format ( d ; "yyyy-mm-dd" )

)]
Set Variable [ $date2; Value:Let (

    d = JSONGetElement ( $json ; "date2" )

    ;

    date.format ( d ; "yyyy-mm-dd" )

)]

#Set api parameters.
Set Variable [ $apiName; Value:"Weather" ]
Set Variable [ $apiKey; Value:apiKey ( $apiName ) ]
Set Variable [ $baseUrl; Value:apiBaseUrl ( $apiName ) ]

#Setup base endpoint.
Insert Text [ $endpoint ]
[ Select ]
Set Variable [ $endpoint; Value:Substitute ( $endpoint ;

    [ "{location}" ; $location ] ;
    [ "{date1}" ; $date1 ] ;
    [ "{date2}" ; $date2 ] ;
    [ "{apiKey}" ; $apiKey ]

)]
```

Weather API: Weather - Process { json }

#Setup curl.

Insert Text [\$curl]
[Select]

#Setup url.

Set Variable [\$url; Value:\$baseUrl & \$endpoint]

#Perform api call.

Insert from URL [\$response; \$url; cURL options: \$curl]
[Select; No dialog]

#Display formatted.

Set Variable [\$response; Value:JSONFormatElements (\$response)]

#Create weather record.

New Record/Request

Set Field [WEA::date_start; WEA::_g_date_start]

Set Field [WEA::date_end; WEA::_g_date_end]

Set Field [WEA::location; WEA::_g_location]

Commit Records/Requests
[No dialog]

#Create API call log record.

Set Variable [\$acJSON; Value:JSONSetElement ("" ;
["id" ; WEA::_WEA ; JSONString] ;
["curl" ; \$curl ; JSONString] ;
["request_json" ; null ; JSONString] ;
["request_type" ; "Weather" ; JSONString] ;
["response" ; \$response ; JSONString] ;
["response_header" ; \$header ; JSONString] ;
["table" ; Get (LayoutTableName) ; JSONString] ;
["trace" ; \$trace ; JSONString] ;
["url" ; \$url ; JSONString]
)]

Perform Script ["API Call Log - Create Record { json }"; Parameter: \$acJSON]

#Process days.

Perform Script ["Weather - Create Days { response }"; Parameter: \$response]

#Clear globals.

Set Field [WEA::_g_date_start; null]

Set Field [WEA::_g_date_end; null]

Set Field [WEA::_g_location; null]

Exit Script [Result: \$response]

SendGrid: SendGrid - Get Templates

```
# -----  
# Purpose:           SendGrid - Get Templates  
# Creator:           The Scarpetta Group - James Scarpetta  
# Date:              2024-Apr-11  
# Version:           1.00  
# Notes:  
# Parameters:  
# -----
```

Set Error Capture [On]

#Set api parameters.

Set Variable [\$apiName; Value:"SendGrid"]

Set Variable [\$apiKey; Value:apiKey (\$apiName)]

Set Variable [\$baseUrl; Value:apiBaseUrl (\$apiName)]

#Setup endpoint.

Set Variable [\$endpoint; Value:"/v3/templates?generations=dynamic"]

#Setup curl.

Insert Text [\$curl]
[Select]

Set Variable [\$curl; Value:Substitute (\$curl ; "@\$apiKey" ; \$apiKey)]

#Setup url.

Set Variable [\$url; Value:\$baseUrl & \$endpoint]

#Perform api call.

Insert from URL [\$response; \$url; cURL options: \$curl]
[Select; No dialog]

Set Variable [\$formatted; Value:JSONFormatElements (\$response)]

#Check for keys in response.

Set Variable [\$templates; Value:JSONGetElement (\$response ; "templates")]

Set Variable [\$keys; Value:JSONListKeys (\$response ; "templates")]

Set Variable [\$keysCount; Value:ValueCount (\$keys)]

#Check if templates returned.

If [\$keysCount > 0]

#Process in card window.

New Window [Style: Card; Using layout: "Processing_SGT" (SGT); Close: No; Minimize: No; Maximize: No; Resize: No; Menu Bar: No; Dim parent window: Yes; Toolbars: No]

Pause/Resume Script [Duration (seconds): .01]

#Loop and create templates.

Set Variable [\$i; Value:1]

Loop [Flush: Always]

#Get template data.

Set Variable [\$value; Value:JSONGetElement (\$templates ; GetValue (\$keys ; \$i))]

SendGrid: SendGrid - Get Templates

```
Set Variable [ $id; Value:JSONGetElement ( $value ; "id" ) ]
Set Variable [ $name; Value:JSONGetElement ( $value ; "name" ) ]
Set Variable [ $generation; Value:JSONGetElement ( $value ; "generation" ) ]
Set Variable [ $versionId; Value:JSONGetElement ( $value ; "versions[id" ) ]

#Check for duplicate template.
Enter Find Mode [ ]
Set Field [ SGT::id; $id ]
Perform Find [ ]

#No record found.
If [ Get ( LastError ) > 0 ]
    New Record/Request
End If

#Set template fields.
Set Field [ SGT::reponse_json; $value ]
Set Field [ SGT::id; $id ]
Set Field [ SGT::name; $name ]
Set Field [ SGT::generation; $generation ]

#Get version record.
Set Variable [ $SGTid; Value:SGT::__SGT ]
Set Variable [ $versionJSON; Value:JSONSetElement ( "" ;

    [ "SGTid" ; $SGTid ; JSONString ] ;
    [ "templateId" ; $id ; JSONString ] ;
    [ "versionId" ; $versionId ; JSONString ]

    ) ]
Perform Script [ "SendGrid - Get Template Version { json }"; Parameter: $versionJSON ]

#Exit when all templates created.
Exit Loop If [ Let (

    $i = $i + 1

    ;

    $i > $keysCount

    ) ]

#Back to template table.
Go to Layout [ "Processing_SGT" (SGT) ]

#Clear variables.
Set Variable [ $value; Value:null ]
Set Variable [ $id; Value:null ]
Set Variable [ $name; Value:null ]
Set Variable [ $versionId; Value:null ]
Set Variable [ $SGTid; Value:null ]
```

SendGrid: SendGrid - Get Templates

Set Variable [\$versionJSON; Value:null]

End Loop

End If

#Cleanup.

Close Window [Current Window]

Refresh Window

#Finish Up

Show Custom Dialog [Title: "Success"; Message: "Templates have been imported."; Default Button: "OK", Commit: "Yes"]

Exit Script []

SendGrid: SendGrid - Get Template Version { json }

```
# -----  
# Purpose:           SendGrid - Get Template Version { json }  
# Creator:           The Scarpetta Group - James Scarpetta  
# Date:              2024-Apr-11  
# Version:           1.00  
# Notes:  
# Parameters:  
# -----
```

Set Error Capture [On]

#Get parameters.

```
Set Variable [ $json; Value:Get ( ScriptParameter ) ]  
Set Variable [ $SGTid; Value:JSONGetElement ( $json ; "SGTid" ) ]  
Set Variable [ $templateId; Value:JSONGetElement ( $json ; "templateId" ) ]  
Set Variable [ $versionId; Value:JSONGetElement ( $json ; "versionId" ) ]
```

#Set api parameters.

```
Set Variable [ $apiName; Value:"SendGrid" ]  
Set Variable [ $apiKey; Value:apiKey ( $apiName ) ]  
Set Variable [ $baseUrl; Value:apiBaseUrl ( $apiName ) ]
```

#Setup endpoint.

```
Insert Text [ $endpoint ]  
[ Select ]  
Set Variable [ $endpoint; Value:Substitute ( $endpoint ;  
  
    [ "{template_id}" ; $templateId ] ;  
    [ "{version_id}" ; $versionId ]  
  
    ) ]
```

#Setup curl.

```
Insert Text [ $curl ]  
[ Select ]  
Set Variable [ $curl; Value:Substitute ( $curl ; "@$apiKey" ; $apiKey ) ]
```

#Setup url.

```
Set Variable [ $url; Value:$baseUrl & $endpoint ]
```

#Perform api call.

```
Insert from URL [ $response; $url; cURL options: $curl ]  
[ Select; No dialog ]
```

```
Set Variable [ $formatted; Value:JSONFormatElements ( $response ) ]
```

#Check for duplicate version.

```
Enter Find Mode [ ]  
Go to Layout [ "Processing_STV" (STV) ]  
Set Field [ STV::id; $versionId ]  
Perform Find [ ]
```


SendGrid: SendGrid - Get Template Version { json }

#No record found.

If [Get (LastError) > 0]

New Record/Request

End If

#Set template version fields.

Set Field [STV::reponse_json; \$response]

Set Field [STV:: f_SGT; \$SGTid]

Set Field [STV:: f_SGT; \$SGTid]

Set Field [STV::active; JSONGetElement (\$response ; "active")]

Set Field [STV::editor; JSONGetElement (\$response ; "editor")]

Set Field [STV::generate_plain_content; JSONGetElement (\$response ; "generate_plain_content")]

Set Field [STV::html_content; JSONGetElement (\$response ; "html_content")]

Set Field [STV::test_data; JSONGetElement (\$response ; "test_data")]

Set Field [STV::id; JSONGetElement (\$response ; "id")]

Set Field [STV::templateName; JSONGetElement (\$response ; "name")]

Set Field [STV::plain_content; JSONGetElement (\$response ; "plain_content")]

Set Field [STV::subject; JSONGetElement (\$response ; "subject")]

Set Field [STV::template_id; JSONGetElement (\$response ; "template_id")]

Set Field [STV::updated_at; JSONGetElement (\$response ; "updated_at")]

Set Field [STV::user_id; JSONGetElement (\$response ; "user_id")]

Set Field [STV::versionName; JSONGetElement (\$response ; "name")]

Commit Records/Requests

[No dialog]

Exit Script []