



Oracle Responsys

28-Jul-2022

©2022 Oracle Corporation. All rights reserved

Event Notification REST API Guide

REST API v1.3

Update 20D

⚠ NOTICE: These interfaces are only available to Oracle Customers that have received Responsys Management approval.

Information in this document is subject to change without notice. Data used as examples in this document is fictitious. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, without prior written permission of Oracle Responsys. Address permission requests, comments, or suggestions about Oracle Responsys documentation by creating a My Oracle Support (MOS) Service Request at <https://support.oracle.com>.

Contents

What's new to the Event Notification API	6
Changes for 22B	6
Changes for 22A	6
Changes for 21D	6
Changes for 21C	7
Changes for 21B	7
Changes for 21A	7
Changes for 20D	8
Overview	9
Conventions used in this document	9
Before you begin	10
About Event Notification APIs	11
Use Case Example	11
Callbacks	14
Create Callback	14
Verify a Callback	19
Retrieve a Callback	21
Update a Callback	23
Delete a Callback	28
Subscriptions	31
Create Subscription	31
Get a Subscription	36
Get all Subscriptions	38
Get all Subscriptions for a Callback	42

Update a Subscription	46
Event List	52
Retrieve a List of Supported Events	52
Event Responses	54
Email Bounced	54
Email Clicked	54
Email Complaint	55
Email Failed	55
Email Opt In	56
Email Opt Out	56
Email Skipped	57
MMS Failed	57
MMS Skipped	58
Push Bounced	58
Push Failed	58
Push Opt In	59
Push Opt Out	59
Push Skipped	59
SMS Failed	60
SMS MO FW	61
SMS Opt In	61
SMS Opt Out	61
SMS Receipt	62
SMS Skipped	62
Web Push Bounced	63
Web Push Closed	63

Web Push Failed	63
Web Push Opt In	64
Web Push Opt Out	64
Web Push Skipped	64
Frequently Asked Questions	66
How are custom event properties sent from Responsys to my callback URL? ..	66
What event types are supported?	67
Are there any plans to support other events in the future?	67
Is Basic Authentication the only authentication method supported?	67
How can I whitelist the callback service?	67
Do callbacks expire?	67
How does the callback service retry connections?	67

What's new to the Event Notification API

This section describes changes and enhancements to the Oracle Responsys Event Notification APIs in product updates during the past year.

- [Update 22B](#)
- [Update 22A](#)
- [Update 21D](#)
- [Update 21C](#)
- [Update 21B](#)
- [Update 21A](#)

Changes for 22B

There are no significant updates to the Event Notification API.

Changes for 22A

There are no significant updates to the Event Notification API.

Changes for 21D

There are no significant updates to the Event Notification API.

Changes for 21C

There are no significant updates to the Event Notification API.

Changes for 21B

There are no significant updates to the Event Notification API.

Changes for 21A

- We've added 13 new event types to enable you to extend the capabilities of the Event Notification API. The new event types are:
 - Web Push Failed (WEBPUSH_FAILED)
 - Web Push Skipped (WEBPUSH_SKIPPED)
 - Web Push Bounced (WEBPUSH_BOUNCED)
 - Web Push Closed (WEBPUSH_CLOSED)
 - Email Opt Out (EMAIL_OPTOUT)
 - Email Opt In (EMAIL_OPTIN)
 - Email Complaint (EMAIL_COMPLAINT)
 - Push Opt in (PUSH_OPT_IN)
 - Push Opt Out (PUSH_OPT_OUT)
 - SMS Opt In (SMS_OPT_IN)

- SMS Opt Out (SMS_OPT_OUT)
- Web Push Opt In (WEBPUSH_OPTIN)
- Web Push Opt Out (WEBPUSH_OPTOUT)

You can see the list of supported events using the [Retrieve a List of Supported Events](#) endpoint.

Changes for 20D

- A new property, filters, has been added for subscriptions APIs that allows events to only be triggered for certain campaigns. See [Subscriptions](#) for more details.

Overview

This document is for customers that use the Oracle Responsys Web Services REST APIs. It is a supplement to the standard Responsys Web Services REST API Guide.

Please refer to either of the following for information about the standard API:

- [Oracle Responsys REST API Developer's Guide](#) – A PDF version of our standard Responsys Web Services REST API Guide.
- [REST API for Oracle Responsys Marketing Cloud Service](#) – An HTML version of our standard Responsys Web Services REST API Guide. This guide contains the full request and response parameter descriptions for the API endpoints.

Conventions used in this document

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates terms defined in the text or emphasis.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates URLs, code, text that appears on the screen, or text that you enter.
{endpoint variable}	Indicates a variable in the REST endpoint; for example {campaignName} should be substituted with the name

Convention	Meaning
	of the campaign as defined in Responsys.
<code variable>	Indicates a variable in the header, request, or response.

Before you begin

Ensure that your Responsys account is properly configured by the Oracle Responsys Support team to have access to the Responsys Web Services Event Notifications REST API.

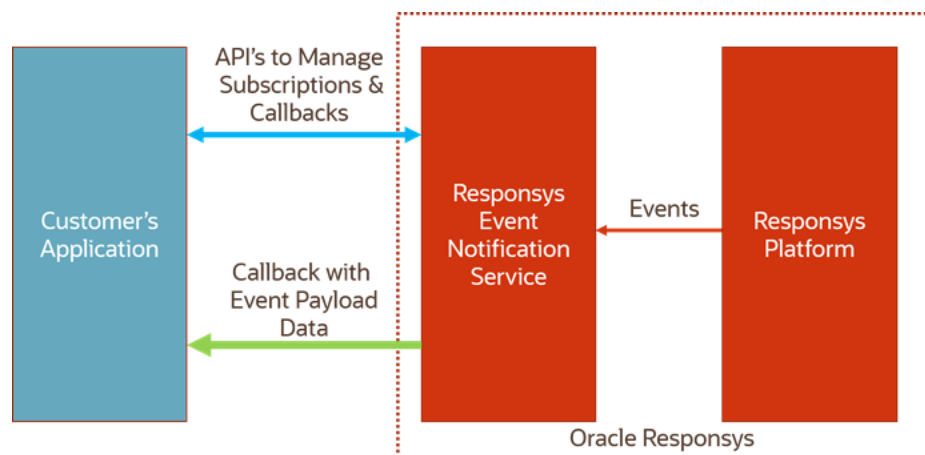
About Event Notification APIs

Real-time campaign event data is now available with the new Responsys Event Notifications API endpoints. Using this service, when events happen in Responsys, Responsys notifies you at a URL you specify.

This provides newer automation opportunities, real-time customer data synchronization, and more marketing intelligence capabilities.

The general flow to use the Event Notification API endpoints follows:

- 1. **Create** and **verify** your callback URL.
- 2. **Create** a subscription to your callback URL to events.
- 3. When events are triggered, Responsys calls your callback URL with the payloads outlined in **Event Responses**.



Use Case Example

A fictitious company named Durham Denim wants to be notified at <https://www.durhamdenim.in/callback> whenever an email message fails, bounces, or

skips. Durham Denim creates a callback.

```
POST /rest/api/v1.3/notifications/callbacks
{
  "name": "Email_Callback",
  "url": "https://www.durhamdenim.in/callback",
  "batchSize": 100
  "authentication": {
    "type": "B",
    "data": {
      "userName": "john_doe",
      "password": "Oracle1234"
    }
  }
}
```

Next, the callback is verified.

```
GET /rest/api/v1.3/notifications/callbacks/Email_Callback?action=verify
```

Durham Denim creates a subscription, subscribing to email bounces, fails, and skips.

```
POST /rest/api/v1.3/notifications/subscriptions
{
  "name": "Email_Subscription",
  "callBackName": "Email_Callback",
  "eventTypes": [
    "EMAIL_BOUNCED",
    "EMAIL_FAILED",
    "EMAIL_SKIPPED"
  ]
}
```

If a Durham Denim email campaign message bounces, fails, or skips, Responsys will send a notification to Durham Denim's callback URL. The notification for email bounces resembles:

```
{  
  "campaignId": "193897761",  
  "riid": "3738409741",  
  "failureReason": "MAILBOX_FULL",  
  "customerId": "10",  
  "recipient": "example@oracle.com",  
  "failureType": "EMAIL_BOUNCED",  
  "channelType": "Email",  
  "campaignName": "Summer Promotions"  
}
```

In this case, the email bounced because the recipient's mailbox was full. Now that you have an understanding of how these APIs work, details about each endpoint is detailed below.

Callbacks

The Callback API is used to manage callbacks. Callbacks inform Responsys of the URL you want to receive notifications for events that happen in Responsys.

Callbacks will tell Responsys that after an event happens, send an event notification to a callback URL. Here are some use cases where you might implement callbacks:

- In the event a failed, bounced, or skipped email event happens, notify the callback URL so that marketers can take action
- After successful email delivery, notify the callback URL so that marketers can implement subsequent marketing activities

Create Callback

Use this endpoint to create a callback. You must specify details about the callback, such as the callback URL and the authentication details used to connect to the URL.

After creating a callback, it must be verified before it can be used in an event subscription.

A successful response returns the details of the callback such as the callback name, callback URL, and batch size.

Service URL:

`/rest/api/v1.3/notifications/callbacks`

Request Method:

POST

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Request Body - Required Properties:

- **name** – Name of the callback to create.
- **url** – URL location for the callback. Responsys must be able to connect and authenticate to this URL.
- **batchSize** – Determines the maximum number of events sent in a callback. Must be positive integer. Maximum 200.
- **authentication** – Authentication details used to authenticate to the callback URL. Basic authentication (B) is supported. The userName and password credentials will be used to authenticate to the callback url.

Request Body - Optional Properties:

- **backupUrl** – A back up callback URL for Responsys to call, in the event the **url** is unreachable. After 3 unsuccessful attempts at the **url**, Responsys will call the back up URL. The back up URL can be on the same domain as the primary URL, but cannot be the exact location.

Sample Request Body

```
{
  "name": "<callback_name>",
  "url": "<callback_url>",
  "backupUrl ": "<backupCallbackUrl>",
  "batchSize": <batch_size>,
  "authentication": {
```

```
"type": "B",
"data": {
  "userName": "<username>",
  "password": "<password>"
}
}
```

Sample Response Body - Success

```
{
  "name": "<callback_name>",
  "url": "<callback_url>",
  "backupUrl ": "<backupCallbackUrl>",
  "batchSize": 1,
  "links": [
    {
      "rel": "self",
      "href": "<callback_url>",
      "method": "PUT"
    },
    {
      "rel": "getCallBack",
      "href": "<callback_url>",
      "method": "GET"
    },
    {
      "rel": "deleteCallBack",
      "href": "<callback_url>",
      "method": "DELETE"
    }
  ]
}
```

Sample Response Body - Failure

400 Bad Request

Callback already exists: Requests fail if a callback with the same name specified already exists. The error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "Callback with name [<callback_name>] already exists",
  "errorDetails": []
}
```

Invalid callback URL: Requests fail if the system cannot connect to the callback URL. Error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "Could not connect to callback url [<callback_url> ]",
  "errorDetails": []
}
```

URL already exists: Requests fail if a callback with the URL specified already exists. The error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "Callback url [<callback_url> ] already exists",
  "errorDetails": []
}
```

Invalid credentials: Requests fail if the credentials specified in the request cannot be used to connect to the URL specified. The error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "Could not connect to callback url [<callback_url>]",
  "errorDetails": []
}
```

Invalid authentication: Requests fail if the authentication method specified is invalid.

Only Basic Authentication (B) is supported. The error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "Only Basic Authentication is supported currently and value should be 'B' ",
  "errorDetails": []
}
```

Empty batch size: Requests fail if the batch size specified is less than 0. The error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "Batch Size cannot be less than 0",
  "errorDetails": []
}
```

Empty callback name: Requests fail if a callback name is not specified. The error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "callback name cannot be empty or null",
  "errorDetails": []
}
```

Empty URL: Requests fail if a callback URL is not specified. The error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "callback url cannot be empty or null",
  "errorDetails": []
}
```

Empty authentication: Requests fail if an authentication type is not specified. The error resembles:

```
{
  "type": "",
  "title": "Invalid field name",
  "errorCode": "INVALID_FIELD_NAME",
  "detail": "The following field names [invalid_field_names_list] are invalid",
  "errorDetails": []
}
```

Verify a Callback

Use this interface to verify a callback. A callback must be verified before it can be subscribed to event notifications. A successful response returns a message indicating the verification was successful, and you can subscribe the callback to subscriptions.

Service URL:

/rest/api/v1.3/notifications/callbacks/{callbackName}/actions/verify

Path Parameters:

callbackName – Name of the callback to verify.

Request Method:

GET

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Request Body - Required Properties:

Not applicable

Sample Request Body

Not applicable

Sample Response Body - Success

```
{
  "status": "Successfully verified [<callback_name>] "
}
```

Sample Response Body - Failure

400 Bad Request

Callback already exists: Requests fail if a callback with the same name specified already exists. The error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "Callback with name [<callback_name>] already exists",
  "errorDetails": []
}
```

Invalid callback URL: Requests fail if the system cannot connect to the callback URL.

Error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "Could not connect to callback url [<callback_url> ]",
  "errorDetails": []
}
```

Retrieve a Callback

Use this endpoint to retrieve an existing callback. A successful response returns the callback's name, URL, and batch size.

Service URL:

`/rest/api/v1.3/notifications/callbacks/{callbackName}`

Path Parameters:

`callbackName` – Name of the callback to retrieve.

Request Method:

GET

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Request Body - Required Properties:

Not applicable

Sample Request Body

Not applicable

Sample Response Body - Success

```
{
  "name": "<callback_name>",
  "url": "<callback_url>",
  "batchSize": <batch_size>,
  "links": [
    {
      "rel": "self",
      "href": "<callback_url>",
      "method": "GET"
    },
    {
      "rel": "deleteCallBack",
      "href": "<callback_url>",
      "method": "DELETE"
    },
    {
      "rel": "createCallBack",
      "href": "<callback_url>",
      "method": "POST"
    },
    {
      "rel": "updateCallBack",
      "href": "<callback_url>",
```

```
"method": "PUT"
}
]
```

Sample Response Body - Failure

404 Not Found

Callback not found: Requests fail if the callback name specified is not found. The error resembles:

```
{
  "type": "",
  "title": "Object not found",
  "errorCode": "OBJECT_NOT_FOUND",
  "detail": "CallBack [<callback_name>] not found",
  "errorDetails": []
}
```

Update a Callback

Use this endpoint to update an existing callback. You must specify the callback name, URL, and authentication details in the request body. Using this endpoint, you can update an existing callback's URL or batch size.

A successful response returns the updated details for the callback.

Service URL:

`/rest/api/v1.3/notifications/callbacks/{callbackName}`

Path Parameters:

`callbackName` – Name of the callback to update.

Request Method:

PUT

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Request Body - Required Properties:

`name` – Name of the callback to update.

`authentication` – Authentication credentials to connect to the callback URL. Basic authentication (B) using a username and password is the only authentication method currently supported.

One of:

`url` – New URL location for the callback.

`batchSize` – New callback batch size.

Request Body - Optional Properties:

- `backupUrl` – A back up callback URL for Responsys to call, in the event the `url` is unreachable. After 3 unsuccessful attempts at the `url`, Responsys will call the back up URL. The back up URL can be on the same domain as the primary URL, but cannot be the exact location.

Sample Request Body

```
{  
  "url": "<callback_url>",
```

```
"backupUrl ": "<backupCallbackUrl>",
"batchSize": <batch_size>,
"authentication": {
  "type": "B",
  "data": {
    "userName": "<username>",
    "password": "<password>"
  }
}
}
```

Sample Response Body - Success

```
{
  "name": "<callback_name>",
  "url": "<callback_url>",
  "backupUrl ": "<backupCallbackUrl>",
  "batchSize": 1,
  "links": [
    {
      "rel": "self",
      "href": "<callback_url>",
      "method": "PUT"
    },
    {
      "rel": "getCallBack",
      "href": "<callback_url>",
      "method": "GET"
    },
    {
      "rel": "deleteCallBack",
      "href": "<callback_url>",
      "method": "DELETE"
    }
  ]
}
```

Sample Response Body - Failure

404 Not Found

Callback not found: Requests fail if the callback name specified is not found. The error resembles:

```
{
  "type": "",
  "title": "Object not found",
  "errorCode": "OBJECT_NOT_FOUND",
  "detail": "CallBack [<callback_name>] not found",
  "errorDetails": []
}
```

400 Bad Request

URL already exists: Requests fail if a callback with the URL specified already exists.

The error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "Callback url [<callback_url> ] already exists",
  "errorDetails": []
}
```

Invalid credentials: Requests fail if the credentials specified in the request cannot be used to connect to the URL specified. The error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "Could not connect to callback url [<callback_url>]",
  "errorDetails": []
}
```

Invalid authentication: Requests fail if the authentication method specified is invalid.

Only Basic Authentication (B) is supported. The error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "Only Basic Authentication is supported currently and value should be 'B' ",
  "errorDetails": []
}
```

Empty batch size: Requests fail if the batch size specified is less than 0. The error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "Batch Size cannot be less than 0",
  "errorDetails": []
}
```

Empty callback name: Requests fail if a callback name is not specified. The error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "callback name cannot be empty or null",
  "errorDetails": []
}
```

Empty URL: Requests fail if a callback URL is not specified. The error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "callback url cannot be empty or null",
  "errorDetails": []
}
```

Empty authentication: Requests fail if an authentication type is not specified. The error resembles:

```
{
  "type": "",
  "title": "Invalid field name",
  "errorCode": "INVALID_FIELD_NAME",
  "detail": "The following field names [invalid_field_names_list] are invalid",
  "errorDetails": []
}
```

Delete a Callback

Use this endpoint to delete an existing callback. A callback can only be deleted if there are no active subscriptions.

A successful response returns a 200 response code and a response body indicating the callback was deleted.

Service URL:

`/rest/api/v1.3/notifications/callbacks/{callbackName}`

Path Parameters:

`callbackName` – Name of the callback to delete.

Request Method:

DELETE

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Request Body - Required Properties:

Not applicable

Sample Request Body

Not applicable

Sample Response Body - Success

```
{
  "message": "CallBack [<callback_name> ] deleted successfully",
  "status": true
}
```

Sample Response Body - Failure

404 Not Found

Callback not found: Requests fail if the callback name specified is not found. The error resembles:

```
{
  "type": "",
  "title": "Object not found",
  "errorCode": "OBJECT_NOT_FOUND",
  "detail": "CallBack [<callback_name>] not found",
}
```

```
"errorDetails": []  
}
```

400 Bad Request

Subscriptions exist for the callback: Requests fail if subscriptions exist for the callback. Delete all subscriptions for the callback before deleting the callback. The error resembles:

```
{  
  "type": "",  
  "title": "Invalid request parameters",  
  "errorCode": "INVALID_PARAMETER",  
  "detail": "Subscription with same name already exists",  
  "errorDetails": []  
}
```

Subscriptions

Subscriptions are used to subscribe a callback to event notifications.

To create a subscription, you must specify the events for which you want to receive notifications. After a subscription is created, Responsys will then notify the callback URL when the event types subscribed to happen in Responsys. This enables marketers to take action when they are notified of the event.

The [Retrieve a List of Supported Events](#) API enables you to retrieve a list of supported event types in Responsys.

Create Subscription

Use this endpoint to create a Subscription. Subscriptions are created for existing callbacks. In the request body, you must specify the name of the new subscription, the name of the callback you are creating the subscription for, and the event types for the subscription.

A successful response returns information about the subscription.

Service URL:

`/rest/api/v1.3/notifications/subscriptions`

Request Method:

POST

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Request Body - Required Properties:

name – The name of the subscription you want to create. This name must be unique.

callbackName – The name of the existing callback for which you want to associate with the subscription.

eventNames – Comma-separated list of fully qualified event names for which you want to receive event notifications for. You can retrieve a list of supported events using the [Retrieve a List of Supported Events API](#).

filters - Optional. This is an array of filters that ensures a specific event is only triggered for certain campaigns. Contains an array of eventTypes and an array, values, of criteria to matched. As on 20D, the only supported criteria is a campaignNames array.

Sample Request Body

```
{
  "name": "<subscription_name>",
  "callBackName": "<callback_name>",
  "eventTypes": [
    "EMAIL_BOUNCED",
    "EMAIL_FAILED",
    "EMAIL_SKIPPED",
    "SMS_FAILED",
    "SMS_SKIPPED",
    "SMS_MO_FW_FAILED"
  ]
}
```

```

]
"filters": [
  {
    "eventTypes": [
      "EMAIL_FAILED"
    ],
    "values": [
      {
        "campaignNames": [
          "vk_campaign_failed"
        ]
      }
    ]
  }
]
}
]
}

```

Sample Response Body - Success

```

{
  "subscriptionName": "<subscription_name>",
  "callBackUrl": "<callback_url>",
  "callBackName": "<callback_name>",
  "batchSize": <batchsize>,
  "eventTypes": [
    "EMAIL_BOUNCED",
    "EMAIL_FAILED",
    "EMAIL_SKIPPED",
    "SMS_FAILED",
    "SMS_SKIPPED",
    "SMS_MO_FW_FAILED"
  ],
  "status": "ACTIVE"
  "filters": [
    {
      "eventTypes": [
        "EMAIL_FAILED"
      ],
      "values": [

```

```

    {
      "campaignNames": [
        "vk_campaign_failed"
      ]
    }
  ]
}
]
}

```

Sample Response Body - Failure

404 Not Found

Event type not supported: Requests fail if any events specified are not supported.

The error resembles:

```

{
  "type": "",
  "title": "Object not found",
  "errorCode": "OBJECT_NOT_FOUND",
  "detail": "Event Type [ WEBPUSH_ ] not found or is not supported",
  "errorDetails": []
}

```

Invalid callback URL: Requests fail if the system cannot connect to the callback URL.

Error resembles:

```

{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "Could not connect to callback url [<callback_url> ]",
  "errorDetails": []
}

```

Empty callback URL: Requests fail if a callback with the URL specified already exists.

The error resembles:

```
{
  "type": "",
  "title": "Object not found",
  "errorCode": "OBJECT_NOT_FOUND",
  "detail": "Callback [<callback_name> ] not found ",
  "errorDetails": []
}
```

400 Bad Request

No events specified in the request: Requests fail if any events specified are not supported. The error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "Atleast one event type should be defined in the request",
  "errorDetails": []
}
```

Empty subscription name specified: Requests fail if any events specified are not supported. The error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "Subscription name is empty on null",
  "errorDetails": []
}
```

Subscription already active: Requests fail if any event types specified in the request are already active for the specified callback. If an active subscription already exists with any event types specified, update the current subscription instead. The error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "There's an active subscription for event [ EMAIL_FAILED ]",
  "errorDetails": []
}
```

Get a Subscription

Use this endpoint to retrieve information about an existing subscription. A successful response returns the subscription name, callback URL, batch size, status of the subscription, and the event types associated with the subscription.

Service URL:

`/rest/api/v1.3/notifications/subscriptions/{subscriptionName}`

Path Parameters:

`subscriptionName` – Name of the subscription to retrieve.

Request Method:

GET

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Request Body - Required Properties:

Not applicable

Sample Request Body

Not applicable

Sample Response Body - Success

```
{
  "subscriptionName": "<subscription_name>",
  "callbackUrl": "<callback_url>",
  "callbackName": "<callback_name>",
  "batchSize": 2,
  "eventTypes": [
    "EMAIL_BOUNCED"
  ],
  "status": "ACTIVE"
  "filters": [
    {
      "eventTypes": [
        "EMAIL_FAILED"
      ],
      "values": [
        {
          "campaignNames": [
            "vk_campaign_failed"
          ]
        }
      ]
    }
  ]
}
```

Sample Response Body - Failure

404 Not Found

Subscription not available: Requests fail if the subscription is not found. Verify the subscription exists. The error resembles:

```
{
  "type": "",
  "title": "Object not found",
  "errorCode": "OBJECT_NOT_FOUND",
  "detail": "Subscription with name [<subscription_name>] not found",
  "errorDetails": [ ]
}
```

Get all Subscriptions

Use this endpoint to retrieve all active Subscriptions within your Responsys account. A successful response returns a list of all active subscriptions, the associated callback url, callback name, batch size, subscription status, and subscription event types.

Service URL:

/rest/api/v1.3/notifications/subscriptions

Request Method:

GET

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Request Body - Required Properties:

Not applicable

Sample Request Body

Not applicable

Sample Response Body - Success

```
[
  {
    "subscriptionName": "<subscription_name>",
    "callbackUrl": ""<callback_url>",
    "callbackName": "<callback_name>",
    "batchSize": 200,
    "eventTypes": [
      "EMAIL_SKIPPED"
    ],
    "status": "ACTIVE",
    "filters": [
      {
        "eventTypes": [
          "EMAIL_FAILED"
        ],
        "values": [
          {
            "campaignNames": [
              "vk_campaign_failed"
            ]
          }
        ]
      }
    ],
    {
      "eventTypes": [
        "EMAIL_SKIPPED"
      ],
      "values": [
        {
          "campaignNames": [
            "vk_campaign_skip"
          ]
        }
      ]
    }
  ]
}
```

```

    }
  ],
  "links": [
    {
      "rel": "self",
      "href": "/rest/api/v1.3/notifications/subscriptions",
      "method": "GET"
    },
    {
      "rel": "createSubscription",
      "href": "/rest/api/v1.3/notifications/subscriptions",
      "method": "POST"
    },
    {
      "rel": "deleteSubscription",
      "href": "/rest/api/v1.3/notifications/subscriptions/<subscription_name>",
      "method": "DELETE"
    },
    {
      "rel": "updateSubscription",
      "href": "/rest/api/v1.3/notifications/subscriptions/<subscription_name>",
      "method": "PUT"
    },
    {
      "rel": "getSubscription",
      "href": "/rest/api/v1.3/notifications/subscriptions/<subscription_name>",
      "method": "GET"
    }
  ]
},
{
  "subscriptionName": "<subscription_name>",
  "callbackUrl": "<callback_url>",
  "callbackName": "<callback_name>",
  "batchSize": 200,
  "eventTypes": [
    "SMS_MO_FW_FAILED"
  ],
  "status": "ACTIVE",
  "links": [

```

```

{
  "rel": "self",
  "href": "/rest/api/v1.3/notifications/subscriptions",
  "method": "GET"
},
{
  "rel": "createSubscription",
  "href": "/rest/api/v1.3/notifications/subscriptions",
  "method": "POST"
},
{
  "rel": "deleteSubscription",
  "href": "/rest/api/v1.3/notifications/subscriptions/<subscription_name>",
  "method": "DELETE"
},
{
  "rel": "updateSubscription",
  "href": "/rest/api/v1.3/notifications/subscriptions/<subscription_name>",
  "method": "PUT"
},
{
  "rel": "getSubscription",
  "href": "/rest/api/v1.3/notifications/subscriptions/<subscription_name>",
  "method": "GET"
}
]
}
]

```

Sample Response Body - Failure

404 Not Found

No Subscriptions: Requests fail if no subscriptions are found within the account. The error resembles:

```

{
  "type": "",

```

```
"title": "No Subscriptions found in this account",  
"errorCode": "NO_SUBSCRIPTION_FOUND",  
"detail": "No subscription found in this account",  
"errorDetails": []  
}
```

Get all Subscriptions for a Callback

Use this endpoint to retrieve all subscriptions associated with a callback.

A successful response returns the callback name, callback URL, and batch size. For the specified callback, all subscriptions associated with the callback are returned.

Subscription details returned include the subscription name, subscription status, and subscription event types.

Service URL:

`/rest/api/v1.3/notifications/subscriptions/{subscriptionName}`

Path Parameters:

`callbackName` – Name of the callback to retrieve subscriptions for.

Request Method:

GET

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Request Body - Required Properties:

Not applicable

Sample Request Body

Not applicable

Sample Response Body - Success

```
{
  "callbackName": "<callback_name>",
  "callbackUrl": "<callback_url>",
  "batchSize": 200,
  "subscriptions": [
    {
      "name": "Subscription_MMS",
      "status": "ACTIVE",
      "eventTypes": [
        "MMS_SKIPPED",
        "MMS_FAILED"
      ]
    },
    {
      "filters": [
        {
          "eventTypes": [
            "EMAIL_FAILED"
          ],
          "values": [
            {
              "campaignNames": [
                "vk_campaign_failed"
              ]
            }
          ]
        }
      ]
    }
  ],
  {
    "name": "Subscription_EmailSMS",
```

```
"status": "ACTIVE",
"eventTypes": [
  "EMAIL_BOUNCED",
  "EMAIL_FAILED",
  "EMAIL_SKIPPED",
  "SMS_FAILED",
  "SMS_SKIPPED",
  "SMS_MO_FW_FAILED"
],
{
  "name": "Subscription_WebPush",
  "status": "ACTIVE",
  "eventTypes": [
    "WEBPUSH_FAILED",
    "WEBPUSH_SKIPPED",
    "WEBPUSH_BOUNCED",
    "WEBPUSH_CLOSED"
  ],
},
{
  "name": "Subscription_Push",
  "status": "ACTIVE",
  "eventTypes": [
    "PUSH_SKIPPED",
    "PUSH_FAILED",
    "PUSH_BOUNCED"
  ],
},
],
"links": [
  {
    "rel": "self",
    "href": "<callback_url>",
    "method": "GET"
  },
  {
    "rel": "deleteCallBack",
    "href": "<callback_url>",
    "method": "DELETE"
  }
]
```

```

    },
    {
      "rel": "getCallBack",
      "href": "<callback_url>",
      "method": "GET"
    },
    {
      "rel": "verifyCallBack",
      "href": "<callback_url>",
      "method": "GET"
    },
    {
      "rel": "createCallBack",
      "href": "<callback_url>",
      "method": "POST"
    },
    {
      "rel": "updateCallBack",
      "href": "<callback_url>",
      "method": "PUT"
    },
    {
      "rel": "getAllCallbacks",
      "href": "<callback_url>",
      "method": "GET"
    }
  ]
}

```

For callbacks with subscriptions with no event types, or callbacks with no subscriptions:

```

{
  "callBackName": "<callback_name>",
  "callBackUrl": "<callback_url>",
  "batchSize": 100,
  "subscriptions": []
}

```

Sample Response Body - Failure

400 Bad Request

Invalid callback: Requests fail if the specified callback name cannot be found. The error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "CallBack [<callback_name> ] does not exist",
  "errorDetails": []
}
```

Update a Subscription

Use this interface to update an existing subscription. You might want to use this endpoint to update the event types for an existing subscription.

A successful response returns details about the updated subscription.

Service URL:

/rest/api/v1.3/notifications/subscriptions/{subscriptionName}

Path Parameters:

subscriptionName – Name of the subscription to update.

Request Method:

GET

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Request Body - Required Properties:

callbackName – Required. Unique name of the callback.

eventTypes – Required. Array of events to update the subscription. Comma-separated list of event types.

status – Optional. Indicates the status of the callback subscription. Possible values include: ACTIVE or PAUSED.

statusReason – Optional. Describes a reason for changing the status.

filters - Optional. This is an array of filters that ensures a specific event is only triggered for certain campaigns. Contains an array of eventTypes and an array, values, of criteria to matched. As on 20D, the only supported criteria is a campaignNames array.

Sample Request Body

```
{
  "callbackName": "<callback_name>",
  "eventTypes": ["EMAIL_BOUNCED"],
  "status": "PAUSED"
}
```

Sample Response Body - Success

```
{
  "subscriptionName": "<subscription_name>",
  "callbackUrl": "<callback_url>",
  "callbackName": "<callback_name>",
  "batchSize": 100,
  "eventTypes": [
    "EMAIL_BOUNCED"
  ],
  "status": "PAUSED"
  "filters": [
    {
      "eventTypes": [
        "EMAIL_FAILED"
      ],
      "values": [
        {
          "campaignNames": [
            "vk_campaign_failed"
          ]
        }
      ]
    }
  ]
}
```

Sample Response Body - Failure

404 Not Found

Invalid subscription: Requests fail if the specified subscription cannot be found. The error resembles:

```
{
  "type": "",
  "title": "Object not found",
  "errorCode": "OBJECT_NOT_FOUND",
```

```
"detail": "Subscription with name [<subscription_name> ] not found",
"errorDetails": []
}
```

Callback not found: Requests fail if the callback cannot be found. The error resembles:

```
{
  "type": "",
  "title": "Object not found",
  "errorCode": "OBJECT_NOT_FOUND",
  "detail": "Subscription [<subscription_name> ] does not exists",
  "errorDetails": []
}
```

Event not supported or invalid: Requests fail if any events specified are not found or are not supported. The error resembles:

```
{
  "type": "",
  "title": "Object not found",
  "errorCode": "OBJECT_NOT_FOUND",
  "detail": "Event Type [ SMS_FAILED1 ] not found or is not supported",
  "errorDetails": []
}
```

No events specified: Requests fail if no events are specified in the request. The error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "Event type is empty or null",
}
```

```
"errorDetails": []  
}
```

Status is not specified: Requests fail if no subscription status is specified in the request. The error resembles:

```
{  
  "type": "",  
  "title": "Invalid request parameters",  
  "errorCode": "INVALID_PARAMETER",  
  "detail": "Subscription status cannot be empty or null",  
  "errorDetails": []  
}
```

Invalid status: Requests fail if the subscription status specified in the request is invalid. Subscription status must either be ACTIVE or PAUSED. The error resembles:

```
{  
  "type": "",  
  "title": "Invalid request parameters",  
  "errorCode": "INVALID_PARAMETER",  
  "detail": "Subscription status can only be ACTIVE or PAUSED",  
  "errorDetails": []  
}
```

Duplicate events: Requests fail if duplicate event types are specified in the request. The error resembles:

```
{  
  "type": "",  
  "title": "Invalid request parameters",  
  "errorCode": "INVALID_PARAMETER",  
  "detail": "Duplicate event types found in the request",  
  "errorDetails": []  
}
```

```
}
```

Event already subscribed: Requests fail if any specified events are already being used in other active subscriptions. The error resembles:

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": "There's an active subscription for event [ SMS_MO_FW_FAILED ] with other
subscriptions",
  "errorDetails": []
}
```

Event List

This endpoint retrieves supported event types in Responsys. A successful response returns a list of all supported event types. The events returned are the event types for which you can create subscriptions.

Retrieve a List of Supported Events

This endpoint retrieves supported event types in Responsys. A successful response returns a list of all supported event types. The events returned are the event types for which you can create subscriptions.

Service URL:

`/rest/api/v1.3/notifications/eventList`

Request Method:

GET

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Request Body - Required Properties:

Not applicable

Sample Request Body

Not applicable

Sample Response Body - Success

```
[  
  "EMAIL_BOUNCED",  
  "EMAIL_FAILED",  
  "EMAIL_SKIPPED",  
  "EMAIL_CLICKED",  
  "SMS_FAILED",  
  "SMS_SKIPPED",  
  "SMS_MO_FW_FAILED",  
  "MMS_SKIPPED",  
  "MMS_FAILED",  
  "PUSH_SKIPPED",  
  "PUSH_FAILED",  
  "PUSH_BOUNCED",  
  "WEBPUSH_FAILED",  
  "WEBPUSH_SKIPPED",  
  "WEBPUSH_BOUNCED",  
  "WEBPUSH_CLOSED",  
  "EMAIL_OPTOUT",  
  "EMAIL_OPTIN",  
  "EMAIL_COMPLAINT",  
  "PUSH_OPT_IN",  
  "PUSH_OPT_OUT",  
  "SMS_OPT_IN",  
  "SMS_OPT_OUT",  
  "WEBPUSH_OPTIN",  
  "WEBPUSH_OPTOUT",  
  "WEBPUSH_FAILED",  
  "SMS_RECEIPT"  
]
```

The payloads Responsys will send for each event type are outlined [here](#).

Event Responses

When a subscribed event is triggered in Responsys, Responsys will call your callback URL. The payloads Responsys will send for each event type are outlined below.

Email Bounced

```
{
  "campaignId": "193897761",
  "riid": "3738409741",
  "failureReason": "MAILBOX_FULL",
  "customerId": "10",
  "recipient": "example@oracle.com",
  "failureType": "EMAIL_BOUNCED",
  "channelType": "Email",
  "campaignName": "Summer Promotions"
},
{
  "campaignId": "193897761",
  "riid": "3738409741",
  "failureReason": "USER_NOT_FOUND",
  "recipient": "example@oracle.com",
  "failureType": "EMAIL_BOUNCED",
  "channelType": "Email",
  "campaignName": "Summer Promotions"
}
```

Email Clicked

```
{
  "campaignId": "194527041",
  "riid": "3780835341",
  "customerId": "123",
  "failureType": "EMAIL_CLICKED",
```

```
"channelType": "Email",
"campaignName": "Summer_Campaign",
"optionalData": {
  "props": {
    "prop": {
      "name": "EMAIL_ADDRESS_",
      "value": "john.doe@oracle.com"
    }
  }
}
```

Email Complaint

```
{
  "campaignId": "76686941",
  "riid": "95628461",
  "customerId": "1141676",
  "failureType": "EMAIL_COMPLAINT",
  "channelType": "Email",
  "campaignName": "Summer Campaign",
  "optionalData": null
}
```

Email Failed

```
{
  "campaignId": "193897821",
  "riid": "3738409721",
  "failureReason": "Failing Reason",
  "recipient": "example@oracle.com",
  "failureType": "EMAIL_FAILED",
  "channelType": "Email",
  "campaignName": "Summer Promotions",
  "optionalData": {
```

```
"props": {
  "prop": [
    {
      "name": "CUSTOM_COLUMN1",
      "value": "John 1"
    },
    {
      "name": "EMAIL_ADDRESS_",
      "value": "example@oracle.com"
    }
  ]
}
```

Email Opt In

```
{
  "riid": "25667",
  "customerId": "100AB",
  "failureType": "EMAIL_OPTIN",
  "channelType": "Email",
  "optionalData": null
}
```

Email Opt Out

```
{
  "riid": "25667",
  "customerId": "100AB",
  "failureType": "EMAIL_OPTOUT",
  "channelType": "Email",
  "optionalData": null
}
```

Email Skipped

```
{
  "campaignId": "193897761",
  "riid": "3738409721",
  "failureReason": "Expression resulted in an empty value",
  "recipient": "oracle@example.com",
  "failureType": "EMAIL_SKIPPED",
  "channelType": "Email",
  "campaignName": "Summer Promotions",
  "optionalData": {
    "props": {
      "prop": [
        {
          "name": "CUSTOM_COLUMN1",
          "value": "John"
        },
        {
          "name": "EMAIL_ADDRESS_",
          "value": "oracle@example.com"
        }
      ]
    }
  }
}
```

MMS Failed

```
{
  "campaignId": "193898101",
  "riid": "3658253801",
  "failureReason": "failing",
  "recipient": "US - 16502229405",
  "failureType": "MMS_FAILED",
  "channelType": "Mms",
  "campaignName": "Summer Promotions"
}
```

```
}
```

MMS Skipped

```
{
  "campaignId": "193898281",
  "riid": "3658253801",
  "failureReason": "Error personalizing field: EXAMPLE.CUSTOMER_ID_: empty field value;",
  "recipient": "US - 16502229405",
  "failureType": "MMS_SKIPPED",
  "channelType": "Mms",
  "campaignName": "Summer Promotions"
}
```

Push Bounced

```
{
  "campaignId": "193558821",
  "riid": "3721112141",
  "failureReason": "InvalidParameters",
  "failureType": "PUSH_BOUNCED",
  "channelType": "Push",
  "campaignName": "Summer Promotions"
}
```

Push Failed

```
{
  "campaignId": "193293541",
  "riid": "3732406721",
  "failureReason": "Message Expiry time has passed. Message Expiry time must be sometime in the future.",
  "failureType": "PUSH_FAILED",
  "channelType": "Push",
}
```

```
"campaignName": "Summer Promotions"
}
```

Push Opt In

```
{
  "riid": "138435587",
  "customerId": null,
  "failureType": "PUSH_OPT_IN",
  "channelType": "Push",
  "optionalData": null
}
```

Push Opt Out

```
{
  "riid": "138435587",
  "customerId": null,
  "failureType": "PUSH_OPT_OUT",
  "channelType": "Push",
  "optionalData": null
}
```

Push Skipped

```
{
  "campaignId": "193561021",
  "riid": "3728082761",
  "failureReason": "Error personalizing field: NEW_PUSH_LIST_APP.ADVERTISER_ID_: empty field value;",
  "failureType": "PUSH_SKIPPED",
  "channelType": "Push",
  "campaignName": "Summer Promotions"
}
```

SMS Failed

```
{
  "campaignId": "193897941",
  "riid": "3658253801",
  "failureReason": "failing",
  "recipient": "US - 16502229405",
  "failureType": "SMS_FAILED",
  "channelType": "Sms",
  "campaignName": "SMS_Fail",
  "optionalData": {
    "props": {
      "prop": [
        {
          "name": "CUSTOM_COLUMN1",
          "value": "CUSTOM_COLUMN1"
        },
        {
          "name": "MOBILE_NUMBER_",
          "value": "16502229405"
        }
      ]
    }
  }
}

{
  "campaignId": "883221",
  "riid": "3558989401",
  "failureReason": "java.lang.NullPointerException",
  "recipient": "-1 - -1",
  "failureType": "SMS_FAILED",
  "channelType": "Sms",
  "campaignName": "Summer Promotions"
}
```

SMS MO FW

```
{
  "campaignId": "218604367",
  "riid": "287229527",
  "recipient": "US - 16502882154",
  "failureType": "SMS_MO_FW_FAILED",
  "channelType": "Sms",
  "campaignName": "SMS_MO_FW"
}
```

SMS Opt In

```
{
  "campaignId": "0",
  "riid": "2774307",
  "customerId": "CUST_2774307",
  "failureType": "SMS_OPT_IN",
  "channelType": "Sms",
  "campaignName": "N/A",
  "optionalData": null
}
```

SMS Opt Out

```
{
  "campaignId": "0",
  "riid": "2774307",
  "customerId": "CUST_2774307",
  "failureType": "SMS_OPT_OUT",
  "channelType": "Sms",
  "campaignName": "N/A",
  "optionalData": null
}
```

SMS Receipt

```
{
  "campaignId": "15648030",
  "riid": "29074598",
  "customerId": null,
  "failureType": "SMS_DELIVERY_RECEIPT",
  "channelType": "Sms",
  "campaignName": "Summer Promotions",
  "optionalData": null
}
```

SMS Skipped

```
{
  "campaignId": "193897881",
  "riid": "3738409701",
  "failureReason": "Skipped Record - skipped because the country associated with this record is not served by the Short/Long code used in this campaign.;",
  "recipient": "IN - 919632155746",
  "failureType": "SMS_SKIPPED",
  "channelType": "Sms",
  "campaignName": "SMS_SKipped",
  "optionalData": {
    "props": {
      "prop": [
        {
          "name": "CUSTOM_COLUMN1",
          "value": "John"
        },
        {
          "name": "MOBILE_NUMBER_",
          "value": "9632155746"
        }
      ]
    }
  }
}
```

```
}  
}
```

Web Push Bounced

```
{  
  "campaignId": "151332521",  
  "riid": "23309",  
  "failureReason": "Subscription is expired/Endpoint Not Found",  
  "failureType": "WEBPUSH_BOUNCED",  
  "channelType": "WebPush",  
  "campaignName": "Summer Promotions"  
}
```

Web Push Closed

```
{  
  "campaignId": "151367801",  
  "riid": "835122601",  
  "failureType": "WEBPUSH_CLOSED",  
  "channelType": "WebPush",  
  "campaignName": "Summer Promotions"  
}
```

Web Push Failed

```
{  
  "campaignId": "151354081",  
  "riid": "836412861",  
  "failureReason": "Fail Launch",  
  "failureType": "WEBPUSH_FAILED",  
  "channelType": "WebPush",  
  "campaignName": "Summer Promotions"
```

```
}
```

Web Push Opt In

```
{  
  "campaignId": "-1",  
  "riid": "334235767",  
  "customerId": null,  
  "failureType": "WEBPUSH_OPTIN",  
  "channelType": "WebPush",  
  "campaignName": "N/A",  
  "optionalData": null  
}
```

Web Push Opt Out

```
{  
  "campaignId": "-1",  
  "riid": "334235767",  
  "customerId": null,  
  "failureType": "WEBPUSH_OPTOUT",  
  "channelType": "WebPush",  
  "campaignName": "N/A",  
  "optionalData": null  
}
```

Web Push Skipped

```
{  
  "campaignId": "151354081",  
  "riid": "836212461",  
  "failureReason": "Skip Launch;",  
  "failureType": "WEBPUSH_SKIPPED",  
}
```

```
"channelType": "WebPush",  
"campaignName": "Summer Promotions"  
}
```

Frequently Asked Questions

How are custom event properties sent from Responsys to my callback URL?

Responsys will include all custom event properties in the optionalData array. Let's say for example an email failed to send, and Responsys sent the following response to your callback URL.

```
{
  "campaignId": "193897821",
  "riid": "3738409721",
  "failureReason": "Failing Reason",
  "recipient": "example@oracle.com",
  "failureType": "EMAIL_FAILED",
  "channelType": "Email",
  "campaignName": "Summer Promotions",
  "optionalData": {
    "props": {
      "prop": [
        {
          "name": "CUSTOM_COLUMN1",
          "value": "John 1"
        },
        {
          "name": "EMAIL_ADDRESS_",
          "value": "example@oracle.com"
        }
      ]
    }
  }
}
```

The optionalData includes your custom event properties.

What event types are supported?

Currently in Responsys 20B, there are 15 event types supported. You can always use the [Retrieve a List of Supported Events](#) API endpoint to see what event types you can use.

Are there any plans to support other events in the future?

We plan to add more events to this list based on our product roadmap, customer requests, and new event availability within Responsys. Speak to your customer support representative to submit requests for new event types to support your marketing activities.

Is Basic Authentication the only authentication method supported?

Yes. Basic Authentication is the only authentication method currently supported.

How can I whitelist the callback service?

Add *.responsys.net to your whitelist.

Do callbacks expire?

There is no callback expiry.

How does the callback service retry connections?

If a customer's callback service is unavailable, Responsys will retry in 3000 milliseconds, up to 3 times. After 3 times, Responsys will move the payload from our callback service to an event archive on the Responsys platform. You must contact Support to get this data extracted.