Genesys Multicloud CX Web-based API Reference

Third-Party Messaging Overview

7/14/2025

Contents

The Third-Party Messaging API delivers messages between external messaging services (such as SMS or email) and Digital Channels.

API Overview

When a consumer sends a message, the third-party system must send a POST request to deliver the message to the agent in Digital Channels chat. In return, the agent's response is delivered to the third-party system through a webhook that Digital Channels sends as a POST request. The third-party system may also send notifications about errors and different types of receipts.

> **Important**
> Contact your Genesys representative to configure webhooks for Digital Channels.

Digital Channels requires the third-party system to respond to webhooks as follows:

- Send a response to the webhook within 10 seconds.
- The response for processed webhooks must have a 2XX HTTP code. We recommend **200 OK**. The **201 Accepted** code is reserved for when Digital Channels supports multiple messages per webhook.
- The response body should contain the `messageId` field with the unique identifier of the message. Digital Channels uses this ID when it processes delivery receipts to find the message in the history and possibly update its status.
- The response for webhooks that cannot be processed right now but should be redelivered must have a 5XX HTTP code. Digital Channels tries to redeliver the webhook according to the configuration for retry timeout and retry attempts. Contact your Genesys representative for details about this configuration.

The webhook POST request contains an X-Hub-Signature header with the SHA1 signature of the post payload. The signature is calculated using the keyed-hash message authentication code (HMAC), where the key is the app secret. The signature is then prefixed with `sha1=`. The third-party system should verify this signature to validate the integrity and origin of the payload.

API Details

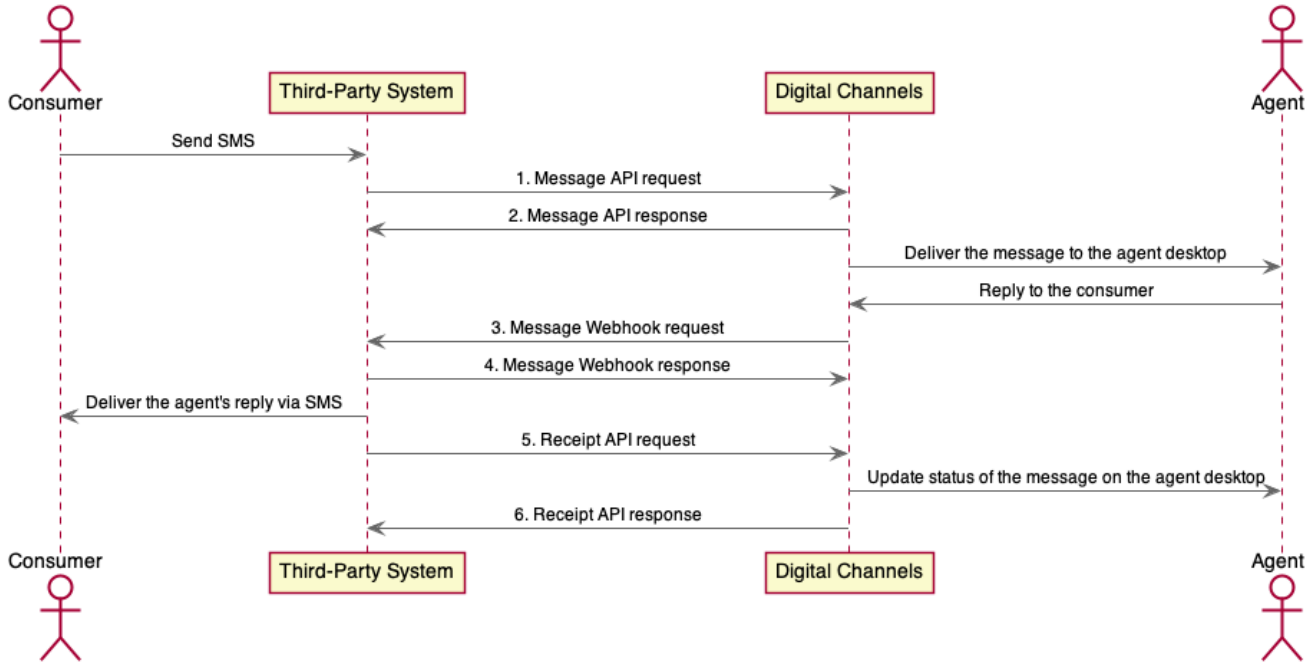
Find the API requests, responses, and details here:

- Third-Party Messaging API
- Third-Party Messaging Webhook

Getting Started

Make sure to check out the Introduction to Engage Cloud APIs page for more information about core concepts in the Engage Cloud APIs, such as requests and responses.

Example: SMS data flow



1. Message API request

```
POST /channels/0123456789/messages HTTP/1.1
 x-ccid: 0ff723ff-6c6f-44e0-9ce3-ce579337eadd
 X-Hub-Signature: sha1=9577d7a3922516701c2e14ca043145eb2057b803

 {
   "messages": [
     {
       "channel": {
         "from": {
           "id": "16504661149"
         },
         "to": {
           "id": "0123456789"
         }
       },
       "type": "Text",
       "text": "Consumer message text"
     }
   ]
 }
```

2. Message API response

```
{
   "status": {
     "code": 0
   },
   "data": {
     "ids": [
       "1613473893265-0"
     ]
   }
 }
```

3. Message Webhook request

```
POST / HTTP/1.1
 X-Hub-Signature: sha1=20a00a8efbe06a43e403e1975873930a027a8f67
 X-B3-TraceId: 05bb858a-3468-4c82-a8ce-3b1a9c4e4b67

 {
   "channel": {
     "id": "0123456789",
     "type": "sms",
     "from": {
       "id": "0123456789"
     },
     "to": {
       "id": "16504661149"
     },
     "messageId": "1613474523265-0"
   },
   "type": "Text",
   "text": "Agent message text"
 }
```

4. Message Webhook response

```
{
   "messageId": "1613474523265-0"
 }
```

5. Receipt API request

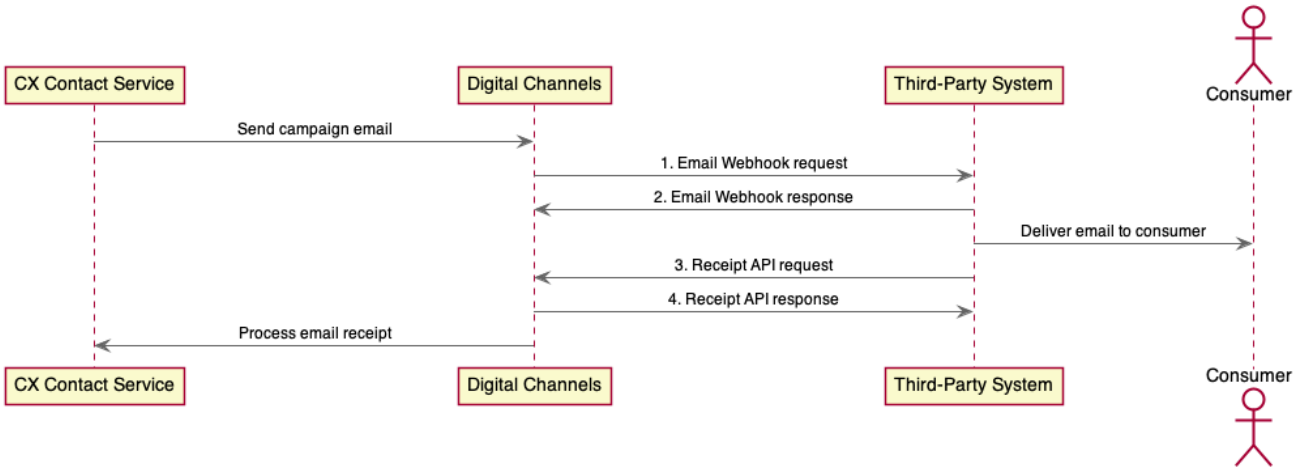
```
POST /channels/0123456789/messages HTTP/1.1
 x-ccid: 0ff723ff-6c6f-44e0-9ce3-ce579337eadd
 X-Hub-Signature: sha1=8d9652cea2d2289ae76fc8b870b96598c6d94b05

 {
   "messages": [
     {
       "channel": {
         "from": {
           "id": "0123456789"
         },
         "to": {
           "id": "16504661149"
         },
         "messageId": "1613474523265-0"
       },
       "type": "Receipt",
       "status": "Delivered"
```

```
    }
  ]
}
```

6. Receipt API response

```
{
  "status": {
    "code": 0
  }
}
```

Example: Email data flow



1. Email Webhook request

```
POST / HTTP/1.1
X-Hub-Signature: sha1=0959a2a00f3f57075d06094e5d9cdc32128f9ec8
X-B3-TraceId: 5767ea58-1511-4a98-bc56-7f34c43e3a51

{
  "channel": {
    "id": "company.com",
    "type": "email",
    "from": {
      "id": "no-reply@company.com",
      "name": "Sender display name"
    },
    "to": [
      {
        "id": "recipient-1@mail.com",
        "name": "First Recipient"
      },
      {
        "id": "recipient-2@mail.com",
        "name": "Second Recipient"
      }
    ],
```

```
    "messageId": "f7acbac6-68db-4031-aec2-4bbd02627c4b",
    "context": {
      "replyTo": {
        "id": "reply-to@company.com",
        "name": "Responder display name"
      }
    }
  },
  "type": "Email",
  "subject": "Subject",
  "content": [
    {
      "type": "text",
      "text": "Plain text email body"
    },
    {
      "type": "html",
      "text": "Html email body"
    }
  ]
}
```

2. Email Webhook response

```
{
   "messageId": "f7acbac6-68db-4031-aec2-4bbd02627c4b"
}
```

3. Receipt API request

```
POST /channels/company.com/messages HTTP/1.1
 x-ccid: 0ff723ff-6c6f-44e0-9ce3-ce579337eadd
 X-Hub-Signature: sha1=f6d8c023c909270eb2d50abdf519151dc80e128a

 {
   "messages": [
     {
       "channel": {
         "from": {
           "id": "no-reply@company.com"
         },
         "to": {
           "id": "recipient-1@mail.com"
         },
         "messageId": "f7acbac6-68db-4031-aec2-4bbd02627c4b"
       },
       "type": "Receipt",
       "status": "Failed",
       "reasons": [
         {
           "code": 350,
           "message": "Message not allowed. Your account has been suspended."
         }
       ]
     }
   ]
 }
```

4. Receipt API response

```
{
```

```
  "status": {
    "code": 0
  }
}
```

Delivery receipts

The third-party system can use delivery receipts to notify Digital Channels about delivery status or errors sending content messages. For usage examples, see the requests payload above for messages with `type="Receipt"`.

The following table describes possible values of the `status` field.

Status	Description
Delivered	The message was delivered to the recipient.
Removed	A previously published message was removed.
Failed	The message was not delivered.

If `status="Failed"`, the third-party system can attach a reason code and an optional message to the `reasons` field of the message.

The following table describes possible values of failure reason codes.

Reason code	Description
305	The message has expired.
310	Rate limit reached.
350	Message not allowed.
510	Unsupported message.
512	Unknown message.
520	Invalid message structure.
530	Invalid destination.
500	General error.