

Marketing Cloud Data Management and Analytics

Marketing Cloud APIs and SDKs

▼ Marketing Cloud APIs and SDKs

Introduction

Using the Marketing Cloud platform, you can call the public Marketing Cloud API, create a mobile app that integrates with Marketing Cloud, or build custom apps and components on top of Marketing Cloud.

Marketing Cloud Packages

A package is your starting point for all developer activity. Create packages to connect to the API. You can also extend Marketing Cloud with packages containing custom apps or custom Journey Builder activities and events. Your organization or business partners build these customizations, or they are installed from AppExchange.

APIs

The Marketing Cloud offers a REST and SOAP API that share a common authentication mechanism based on OAuth 2. Use these APIs to interact with the following Marketing Cloud functionality.

- Tracking
- Subscribers and lists
- Programs
- Triggered sends
- Content
- Contacts
- Journey Builder
- Mobile Connect
- MobilePush
- Campaigns

SDKs

The Marketing Cloud SDKs provide a cross-functional framework around the REST and SOAP APIs that allow developers to integrate APIs using native language code libraries. We also provide SDKs for iOS and Android to use with MobilePush.

Programmatic Languages

The Marketing Cloud offers custom scripting languages to personalize landing pages, create applications, construct cross-channel templates or layouts, and work with messaging functions on the Marketing Cloud platform.

▼ Set Up Your Development Environment for Legacy Packages

Set Up Your Development Environment for Legacy Packages

1. [Install a Package](#) Create an installed package in your Marketing Cloud account to create API integrations, install custom apps, or add custom Journey Builder components.
2. [Get a Client ID and Secret](#) To use the Marketing Cloud API, create an API integration. Creating an API integration gets you a client ID and secret.

3. [Get an Access Token](#) Use your client ID and secret credentials to get an OAuth access token directly from the requestToken API authentication service. Use the access token to authenticate subsequent calls.

▼ Set Up Your Development Environment for Enhanced Packages

Set Up Your Development Environment for Enhanced Packages

1. [Create a package](#). Create an enhanced installed package in your Marketing Cloud account to create API integrations, install custom apps, or add custom Journey Builder components.
2. [Get credentials](#). To use Marketing Cloud APIs, add an API integration to your installed package, and obtain credentials for authentication.
3. Get an access token. Use the access token to authenticate API requests.
 - a. For web and public apps, [set up authorization code grant type](#).
 - b. For server-to-server integrations, [set up client credentials grant type](#).

▼ Get an Access Token for Legacy Packages

Get an Access Token for Legacy Packages

This document applies only to API integrations in legacy packages. To get an access token for OAuth 2.0 API integrations, review Set Up Your Development Environment for Enhanced Packages.

For your authentication requests, we recommend using our tenant-specific Marketing Cloud-branded endpoint structure, which includes your tenant's subdomain. Find your authentication endpoint in Installed Packages.

The first step in any API-based integration is getting an OAuth access token to authenticate your calls.

Using the client ID and secret from the installed package, call the REST Auth service to obtain an OAuth access token for authenticating your API calls. This sample call demonstrates how to POST to that resource to acquire an access token.

```
POST https://YOUR_SUBDOMAIN.auth.marketingcloudapis.com/v1/requestToken
Content-Type: application/json
{
  "clientId": "YOUR_CLIENT_ID",
  "clientSecret": "YOUR_CLIENT_SECRET"
}

200 OK
{
  "accessToken": "YOUR_ACCESS_TOKEN"
  "expiresIn": 3600
}
```

The access token is returned in the **accessToken** property. Specify this access token to authenticate all other REST API calls via the **Authorization** header field with the **Bearer** HTTP authorization scheme.

```
GET https://www.YOUR_SUBDOMAIN.rest.marketingcloudapis.com/platform/v1/endpoints
Accept: application/json
Authorization: Bearer YOUR_ACCESS_TOKEN
```

The following example uses the same access token to authenticate with the SOAP API.

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <h:fueloauth xmlns="http://exacttarget.com"
      xmlns:h="http://exacttarget.com">
      YOUR_ACCESS_TOKEN
    </h:fueloauth>
  </s:Header>
  [...]
</s:Envelope>
```

Refresh an Access Token

Note the **expiresIn** property in the HTTP response to the **requestToken** API call. Access tokens expire one hour after they are issued. If you attempt to use an expired token, you receive a **401 Unauthorized** HTTP response. If this happens, refresh your access token by calling **requestToken** again.

Important Considerations

Follow these considerations when using the API directly and doing your own OAuth token management.

1. Do not request a new access token for every API call you make—each access token is good for an hour and is reusable. Making two API calls for every one operation is inefficient and causes throttling.
2. Be careful where you store your client ID and secret. Never expose this information on the client side via JavaScript or store it in a mobile application. Ensure that these credentials are stored securely in your

▼ requestToken Reference for Legacy Packages

requestToken Reference for Legacy Packages

This document applies only to API integrations in legacy packages. To get an access token for OAuth 2.0 API integrations, review Set Up Your Development Environment for Enhanced Packages.

For your authentication requests, we recommend using our tenant-specific Marketing Cloud-branded endpoint structure, which includes your tenant's subdomain. Find your endpoints in Installed Packages. New and existing customers on S1 through S10 can continue to use <https://auth.exacttargetapis.com>, which does not include your subdomain.

To obtain an OAuth token, perform an HTTP POST specifying your **clientId** and **clientSecret** in the request body.

Query Parameters

Name	Type	Required	Description
<u>legacy</u>	boolean	<input type="checkbox"/>	Set to 1 to return a legacy token

JSON Parameters

Name	Type	Required	Description
<u>clientId</u>	string	Required	First part of the Client Credentials App Key pair generated when creating an API Integration
<u>clientSecret</u>	string	Required	Second part of the Client Credentials App Key pair generated when creating an API Integration
<u>accessType</u>	string	no	Set to offline to return a RefreshToken that is not session-based for later use
<u>refreshToken</u>	string	no	Use the refreshToken obtained through SSO for Marketing Cloud Apps along with the accessType option to return an accessToken that is contextualized for your app. The refreshToken is valid for up to 700 days until it is used and can only be used once. It has a 5-minute revocation period after use, allowing for more attempts in case the service doesn't respond immediately.

Example Request for Production

```
//Request token for a Marketing Cloud Production environment https://mc.exacttarget.com.
```

```
POST https://YOUR_SUBDOMAIN.auth.marketingcloudapis.com/v1/requestToken
Content-Type: application/json
{
  "clientId": "gyjzvytv7ukqtf3x2qdyfsn",
  "clientSecret": "*****"
}
```

Example Request for Testing

```
//Request token for a Marketing Cloud Production Support (sandbox) environment https://mc.test.exacttarget.com.  
  
POST https://auth-test.exacttargetapis.com/v1/requestToken  
Content-Type: application/json  
{  
  "clientId": "gyjzvytv7ukqtfn3x2qdyfsn",  
  "clientSecret": "*****"  
}
```

Example Response

The HTTP POST with valid values for clientId and clientSecret returns two values:

1. **accessToken** - An OAuth token passed to subsequent API requests.
2. **expiresIn** - The expiration period of the OAuth token in seconds. The default value of 3600 seconds cannot be changed.

```
{  
  "accessToken": "*****",  
  "expiresIn": 3600  
}
```

Example Request for SSO

```
//Request token for a Marketing Cloud app.  
  
POST https://YOUR_SUBDOMAIN.auth.marketingcloudapis.com/v1/requestToken  
Content-Type: application/json  
{  
  "clientId": "gyjzvytv7ukqtfn3x2qdyfsn",  
  "clientSecret": "*****",  
  "refreshToken": "*****",  
  "accessToken": "offline"  
}
```

Example Response for SSO

```
{  
  "accessToken": "*****",  
  "expiresIn": 3600  
  "refreshToken": "*****"  
}
```

▼ Access Token Reference

Access Token Reference

You can request an access token for the two different grant types. For server-to-server API integrations, review [Access Token for Server-to-Server Integrations](#). For web or public app integrations, review [Access Token for Web and Public App Integrations](#).

▼ Intro to Marketing Cloud APIs

Intro to Marketing Cloud APIs

The Marketing Cloud offers two APIs that share a common authentication mechanism based on OAuth 2:

- The REST API exposes broader access to Marketing Cloud capabilities.
- The SOAP API provides comprehensive access to most email functionality.

To use either API, you need a client ID and secret, obtained from Marketing Cloud | Installed Packages. The APIs don't have full parity, and you may need to use both SOAP and REST to accomplish your business goals.

Note

You do not need a Marketing Cloud user to call the APIs, but you do need a Marketing Cloud user when creating an API integration in Installed Packages. The Marketing Cloud user must have the Installed Package | Administer permission.

REST API

The REST API uses JSON request and response bodies and resource endpoints to support multi-channel use. All new Marketing Cloud technologies implement REST API. REST calls are synchronous, with timeout values of 120 for non-tracking operations and 300 seconds for tracking and data retrieve operations. The maximum payload of any call is four megabytes.

Use the REST API for this functionality:

- Contacts
- Content Builder
- Journey Builder
- Mobile Connect
- MobilePush
- Campaigns
- Triggered Sends (may also use SOAP)

Web Services SOAP API

The SOAP API uses SOAP envelopes to pass information between you and Marketing Cloud. We recommend a limit of no more than 2k per minute for SOAP calls. Support may request your SOAP envelope to troubleshoot issues.

Use the SOAP API for this functionality:

- Tracking
- Subscribers and lists
- Automations
- Triggered sends (may also use REST)
- Content
- Most other email activities

▼ API Integration Security

API Integration Security

Keep the following security considerations in mind when integrating your Salesforce apps with the Marketing Cloud API. For general guidelines around web application penetration testing for your composite app, review the [OWASP Top Ten checklist](#).

Enforce Least Privilege

Request minimum required scope for the OAuth token for your app API token. This follows the principle of least privilege and reduces risk associated with the API token.

Secure Storage

Store only the refresh token on your external web server. Keep the access token in memory only and request a new access token when needed. Follow industry best practices to securely store the refresh token on an external platform. Treat the refresh token like a Salesforce credential.

Secure in Transit

Always enforce TLS when making calls to the Marketing Cloud APIs. Provide the access token only as the authorization header and never as query parameters. Maintain up-to-date TLS configurations on your external web server.

Authentication and Session Management

Implement authentication and session management correctly by using secure procedures to create, manage, and end a session for each authorized user. Rotate session IDs out and set them with the correct cookie flags. We recommend you use your framework's session management features as they're thoroughly tested and more frequently updated.

Access Control

Your app must verify the user's session and permission levels before giving access to any restricted data or function. For example, a standard user should not be able to access admin-only pages, or user A should not view user B's purchase history.

Sensitive Information in Errors

Your app must handle errors and responses correctly to avoid most of the fingerprinting and enumeration process by a possible attacker. Hide common error responses from the user, such as stack traces and debug logs, because an attacker can use these to gain more information about the server or app.

Cross Site Request Forgery (CSRF)

Attackers use this vulnerability to trick an authenticated user into performing an unwanted action on the vulnerable server. The attacker crafts a URL or a FORM inside a malicious page and tricks the user to access to it. CSRF attacks typically target state-changing requests, as there is no way for the attacker to see the response of the forged request.

HTML Injection and Cross Site Scripting (XSS)

HTML injection vulnerabilities occur when an attacker injects HTML into a vulnerable website and makes it appear as if it was originally there. For example, the attacker injects an iframe and displays a completely different page. With Cross Site Scripting, an attacker injects Javascript that executes in the context of a vulnerable domain. The attacker then crafts a payload and tricks a user into visiting the link. The attacker's Javascript then executes on the user's browser.

Arbitrary Redirects

A vulnerable server may perform a redirect function using user-controlled data in some URL. This allows an attacker to use a normal-looking server URL to redirect a user to a malicious site. In addition, if your site uses one page to perform forwards to other resources, an attacker could modify the input parameter and bypass permission checks.

Remote Code Execution

A web app or server may run some code that is vulnerable to specially crafted input data. That input data executes commands in the target machine. Attackers typically execute code remotely from one of three sources:

1. Web server runs some vulnerable service that listens to open ports on the internet. Resolution: Check each service that is listening at a port on your web server for vulnerabilities or public exploits.
2. Web app uses vulnerable components. Resolution: Double check every software package in your app, such as gems, nodes, or libraries, that support process documents, process images, open connections like external URI, or parse XML.
3. Application is processing user input as serialized data. Resolution: Execute deserialized user data cautiously as it could lead to remote code execution.

Insecure Software

Most applications use some kind of third-party components, such as Javascript libraries, server side frameworks, and application servers. Ensure you are using the latest available version with no known security vulnerabilities. Vulnerable versions are easy to identify and can open up your application to a broader attack surface.

SQL Injection

A SQL injection attack inserts a SQL query via the input data from the client to the application. In SQL injection attacks, SQL commands are injected as a part of user supplied input in order to effect the execution of predefined SQL commands.

Storage of Sensitive Data

Store sensitive data, such as passwords, credit card information, Social Security Numbers, and other PII, securely on servers using your platform's industry best practices for secure storage.

▼ Your Subdomain and Your Tenant's Endpoints

Your Subdomain and Your Tenant's Endpoints

As part of our ongoing commitment to customer trust and security, integrations into Marketing Cloud are now required to utilize tenant-specific endpoints by September 30, 2022.

Marketing Cloud automatically assigns a unique, system-generated subdomain to each of its tenants. A tenant represents your top-level Enterprise account and its business units, your Core account, your top-level Agency account, or your Client account, depending on your tenant type.

Your subdomain is represented by a 28-character string starting with the letters "mc". When your subdomain is appended to Marketing Cloud APIs, it creates endpoints that are unique to your tenant. Use these endpoints to improve the performance of your API requests.

For example, Northern Trail Outfitters' tenant has this subdomain: mc563885gzs27c5t9-63k636ttgm. These are their endpoints:

- REST API endpoint: mc563885gzs27c5t9-63k636ttgm.rest.marketingcloudapis.com
- SOAP API endpoint: mc563885gzs27c5t9-63k636ttgm.soap.marketingcloudapis.com

Locate Your Subdomain and Endpoints

Locate your tenant's endpoints, which contain your subdomain, in Marketing Cloud and use them in your API requests. Only your tenant can use your endpoints. No other Marketing Cloud customer can use them for their API requests.

You can use your subdomain only with the marketingcloudapis.com endpoint structure. You can't use exacttargetapis.com or exacttarget.com with your subdomain to create your tenant's endpoints.

API Integrations with an Access Token

For SOAP and REST API integrations that use access tokens for authorization, locate your tenant's SOAP, REST, and Authentication Base URIs on the API Integration component of packages created in Installed Packages.

API Integrations with a Username and Password

For SOAP API integrations that use a username and password for authentication, locate your tenant's SOAP WSDL and SOAP Service Endpoint either under Account Settings in Setup or under User Settings in Cloud Preferences. API User must be enabled.

▼ REST API

REST API

The REST API uses JSON request and response bodies and resource endpoints to support multi-channel use. All new Marketing Cloud technologies implement REST API. REST calls are synchronous, with timeout values of 120 for non-tracking operations and 300 seconds for tracking and data retrieve operations. The maximum payload of any call is four megabytes.

Use the REST API for this functionality:

- Contacts
- Content Builder
- Journey Builder
- Mobile Connect

- MobilePush
- Campaigns
- Triggered Sends (may also use SOAP)

▼ Automation Studio API

Automation Studio API

Overview

Marketers and developers use Automation Studio in Marketing Cloud to execute multistep marketing and data management activities on an immediate, triggered, or scheduled basis.

Considerations for the Trigger API

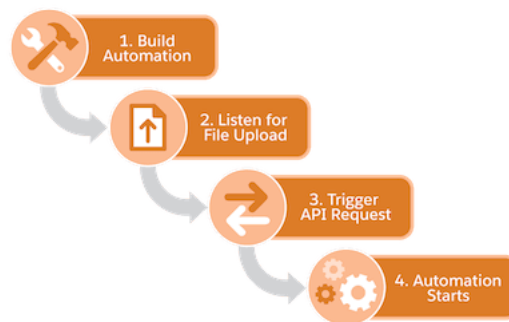
This feature is a Beta Service. Customer may opt to try such Beta Service in its sole discretion. Any use of the Beta Service is subject to the applicable Beta Services Terms provided at [Agreements and Terms](#).

Before you begin to use the Trigger API, you must [opt in to the beta](#).

Trigger API

Marketing developers, service providers, and third-party developers use the Automation Studio Trigger REST API to initiate marketing automations.

Marketing Cloud listens for the trigger API and runs the automation when a file that matches the pattern is uploaded.



API Specification

This table shows the resources in Automation Studio REST API, with links to each page.

HTTP Method	Resource	Description
POST	/automation/v1/automations/trigger	Starts an automation by posting a file's details for evaluation by Automation Studio.
GET	/automation/v1/automations/trigger/status/{requestID}	Get the status of a specified Trigger API requestID.

▼ Create a Lambda Function for MC Data APIs

Create a Lambda Function for MC Data APIs

Overview

There are many ways to fire a Trigger API (beta) request when uploading a file to an external location such as Amazon S3. This tutorial shows how to use Amazon Web Services (AWS) lambda functionality to authenticate with Marketing Cloud. Then, how to fire a trigger API request after you upload a file to an S3 bucket.

Considerations for the Trigger API

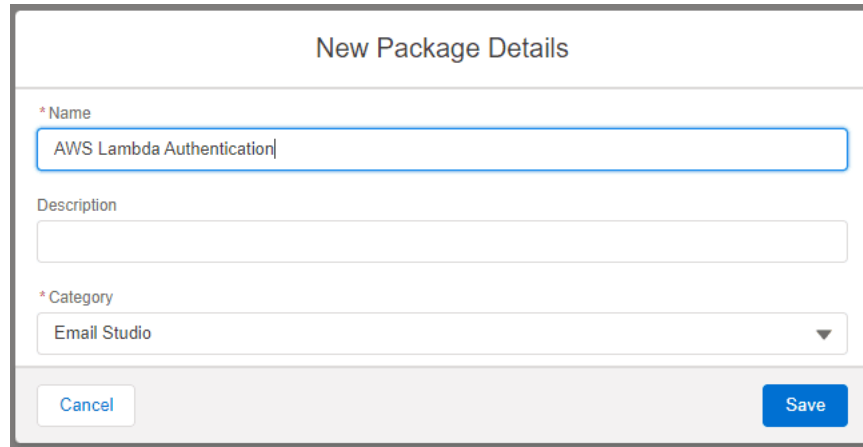
This feature is a Beta Service. Customer may opt to try such Beta Service in its sole discretion. Any use of the Beta Service is subject to the applicable Beta Services Terms provided at [Agreements and Terms](#).

Before you begin to use the Trigger API, you must [opt in to the beta](#).

Setup Installed Package task topic 1

Your new AWS lambda function can access your Marketing Cloud data after you create an installed package in Salesforce Marketing Cloud.

1. From Setup, find and select Platform Tools.
2. From Platform Tools, find and select Apps.
3. From Apps, find and select Installed Packages.
4. Create a new installed package.



Note

Enter the **Name** and **Description** values that you want.

1. Add a component under the newly created Installed Package.
 - a. The component is an API integration type.
 - b. The component is a Server-to-Server integration type.
 - c. Give the component the permissions for your lambda task: **automation_write**, and **automation_execute**.

Set Up Secrets in AWS task topic 2

To encrypt your SFMC installed package information at rest, store the information in AWS Secrets Manager. To further restrict values, authenticate using identity and access management (IAM). Store the Marketing Cloud Client ID, Client Secret, Authentication Base URI, and REST Base URI in AWS Secrets Manager.

1. In the AWS Web Console, navigate to Secrets Manager.
2. Select Store a new secret.
3. Select Other type of secret as the Secret type.

4. Add your Client ID, Client Secret, Authentication Base URI, and REST Base URI from the end of [Setup Installed Package \[task topic 1\]](#). The sample lambda code uses the following keys mapped to their associated values:
 - a. MC_Client_ID.
 - b. MC_Client_Secret.
 - c. MC_Auth_URI.
 - d. MC_Rest_URI.

Secret type [Info](#)

☐ Credentials for Amazon RDS database
 ☐ Credentials for Amazon DocumentDB database
 ☐ Credentials for Amazon Redshift cluster
 ☒ Other type of secret
 API key, OAuth token, other.

Key/value pairs [Info](#)

Key/value | Plaintext

MC_Client_ID	<YOUR_MC_CLIENT_ID>	Remove
MC_Client_Secret	<YOUR_MC_CLIENT_SECRET>	Remove
MC_Auth_URI	<YOUR_MC_AUTH_URI>	Remove
MC_Rest_URI	<YOUR_MC_REST_URI>	Remove
+ Add row		

Encryption key [Info](#)
 You can encrypt using the KMS key that Secrets Manager creates or a customer managed KMS key that you create.

▼

[Add new key](#) [↗](#)

Note

You can select any encryption key for this operation. The names of the keys can be whatever you need.

1. To navigate to the Configure secret step, select Next
2. Configure the name, description, tags, resource permissions, and replication as necessary for your use case. You're using SFMC_Installed_Package for this example.
3. To navigate to the Configure rotation step, select Next.
4. Configure rotation on this page as applicable to your own setup.
5. Select Next and review the summary. When you're satisfied with it, select Store button to create your secret.

Create an IAM User task topic 3

To allow your AWS lambda function to access your newly created secret, create an IAM user with access to Secrets Manager.

1. Navigate to IAM in the AWS Web Console.

2. To create a user, select Add users.
3. Enter a username. This example uses SFMC_API_Client.
4. To enable the IAM user to authenticate in your AWS Lambda function, select Access key - Programmatic access.

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

[+ Add another user](#)

Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

- Select AWS credential type*
- ☒ **Access key - Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.
 - ☐ **Password - AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

5. To configure the user's permissions, click Next: Permissions.
6. Enter a Group name for your AWS Secrets Manager. This example uses SecretsAccess.
7. To give access to AWS Secrets Manager in that group, add SecretsManagerReadWrite policy.
8. To move to the user's tags, select Create group and Next: Tags.
9. Add any applicable tags to your new IAM users, then select Next: Review.
10. Select Create user.

Create a Lambda Function task topic 4

Make an AWS Lambda function that creates a data extension in your SFMC account. This example uses Python 3.7 for setting up the lambda function. You need the Access Key ID and Secret Access Key from your IAM user for subsequent steps in this document.

1. In the AWS Web Console navigate to Lambda.
2. Select Create function.
3. Select the Lambda type Author from scratch.
4. Enter a function name. In this example, the lambda function is named FILE_TRIGGER_AUTOMATION.
5. Select Runtime Python 3.7.

Author from scratch
Start with a simple Hello World example.

Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

Container image
Select a container image to deploy for your function.

Browse serverless app repository
Deploy a sample Lambda application from the AWS Serverless Application Repository.

Basic information

Function name
Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.7

Architecture Info
Choose the instruction set architecture you want for your function code.

☒ x86_64
☐ arm64

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

► Change default execution role

6. Select Create function.

Note

Selecting a different runtime can require importing more dependencies. The sample lambda code shown requires Python 3.7.

Configure the Lambda Function task topic 5

Tie all the previous steps together, set up environment variables and configure the lambda function to access the SFMC secrets and fire the trigger API request.

1. In the AWS Web Console, navigate to the lambda function.
2. Copy the Sample Lambda Code to the Code tab, and save your changes.

Sample Lambda Code

```
import requests
import uuid
import json
import boto3
import base64
import os
from botocore.exceptions import ClientError

def get_aws_secret():
    print('Retrieving AWS Secret...')
    secret_name = os.environ['AWS_SECRET_NAME']
    region_name = "us-east-1"

    session = boto3.session.Session(
        aws_access_key_id=os.environ['AWS_ACCESS'],
        aws_secret_access_key=os.environ['AWS_SECRET']
    )
    client = session.client(
        service_name='secretsmanager',
        region_name=region_name
    )

    try:
        get_secret_value_response = client.get_secret_value(
            SecretId=secret_name
        )
    except ClientError as e:
        if e.response['Error']['Code'] == 'DecryptionFailureException':
            raise e
        elif e.response['Error']['Code'] == 'InternalServiceErrorException':
            raise e
        elif e.response['Error']['Code'] == 'InvalidParameterException':
            raise e
        elif e.response['Error']['Code'] == 'InvalidRequestException':
```

```

        raise e
    elif e.response['Error']['Code'] == 'ResourceNotFoundException':
        raise e
    else:
        if 'SecretString' in get_secret_value_response:
            secret = get_secret_value_response['SecretString']
            print('Retrieved AWS Secret!')
            return json.loads(secret)
        return None

def get_auth_token(aws_secrets):
    headers = {
        'content-type': 'application/json',
        'user-agent': 'Python'
    }

    payload = {
        'client_id': aws_secrets['MC_Client_ID'],
        'client_secret': aws_secrets['MC_Client_Secret'],
        'grant_type': 'client_credentials',
    }

    token_endpoint = aws_secrets['MC_Auth_URI'].strip() + '/v2/token'

    response = requests.post(token_endpoint, data=json.dumps(payload), headers=headers)
    tokenResponse = response.json()

    if 'access_token' not in tokenResponse:
        raise Exception('Unable to authenticate using provided Client ID/Secret provided.')

    return tokenResponse['access_token']

def build_trigger_params(file_path, etag):
    return {
        'fileTransferLocationKey': os.environ['SFMC_FILE_LOCATION_EXTERNAL_KEY'],
        'filename': file_path,
        'deduplicateKey': etag
    }

def get_s3_data_from_event(event_data):
    try:
        file_path = event_data['Records'][0]['s3']['object']['key']
    except Exception:
        file_path = None

    try:
        eTag = event_data['Records'][0]['s3']['object']['eTag']
    except Exception:
        eTag = None

    return (file_path, eTag)

def create_trigger_request(auth_token, rest_uri, event_data):
    file_path, eTag = get_s3_data_from_event(event_data)

    if file_path is None:
        raise Exception('No file path provided in event data.')

    try:
        trigger_uri = 'automation/v1/automations/trigger'

        response = requests.post(
            f'{rest_uri}{trigger_uri}',
            data = json.dumps(build_trigger_params(file_path, eTag)),
            headers = {
                'Accept': 'application/json',
                'authorization': f'Bearer {auth_token}'
            }
        )

        if response:
            return response.json()
        else:
            return None
    except Exception:
        return None

def lambda_handler(event, context):
    secrets = get_aws_secret()
    auth_token = get_auth_token(secrets)
    create_response = create_trigger_request(auth_token, secrets['MC_Rest_URI'], event)

```

```

if create_response:
    return {
        'statusCode': 200,
        'body': json.dumps(create_response)
    }
else:
    return {
        'statusCode': 400
    }

```

1. In the Configuration tab, select Environment variables.
2. Select Add environment variable for each key value pair.
 - a. Enter the key AWS_ACCESS and the value <SECRET_ARN>.
 - b. Enter key AWS_SECRET with value <IAM_ACCESS_KEY>.
 - c. Enter key AWS_SECRET_NAME with value <IAM_ACCESS_SECRET>.
 - d. Enter key SFMC_FILE_LOCATION_EXTERNAL_KEY with value <SFMC_FILE_LOCATION_EXTERNAL_KEY> This is the external key used in Marketing Cloud's File Location configuration.

Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#)

Key	Value	
<input type="text" value="AWS_ACCESS"/>	<input type="text" value="<SECRET_ARN>"/>	<input type="button" value="Remove"/>
<input type="text" value="AWS_SECRET"/>	<input type="text" value="<IAM_ACCESS_KEY>"/>	<input type="button" value="Remove"/>
<input type="text" value="AWS_SECRET_NAME"/>	<input type="text" value="<IAM_ACCESS_SECRET>"/>	<input type="button" value="Remove"/>
<input type="text" value="SFMC_FILE_LOCATION_EXTERNAL_KEY"/>	<input type="text" value="<SFMC_FILE_LOCATION_EXTERNAL_KEY>"/>	<input type="button" value="Remove"/>
<input type="button" value="Add environment variable"/>		

► Encryption configuration

3. Select the General configuration section.
4. Select Edit, and then change the Timeout value to at least 30 seconds. That interval is the minimum time required for the lambda function to authenticate to Marketing Cloud and fire the Trigger API request.

Configure the S3 Bucket task topic 6

1. On the S3 bucket where files are uploaded, navigate to the Properties tab, and select Create event notification.
2. Enter an Event name and any optional configuration parameters.
3. Select the Event types that fire the lambda function.
4. Choose the lambda function (File_Trigger_Automation) previously created, and then save your changes.

Destination

Before Amazon S3 can publish messages to a destination, you must grant the Amazon S3 principal the necessary permissions to call the relevant API to publish messages to an SNS topic, an SQS queue, or a Lambda function. [Learn more](#)

Destination

Choose a destination to publish the event. [Learn more](#)

☒ **Lambda function**
Run a Lambda function script based on S3 events.

☐ **SNS topic**
Send notifications to email, SMS, or an HTTP endpoint.

☐ **SQS queue**
Send notifications to an SQS queue to be read by a server.

Specify Lambda function

☒ Choose from your Lambda functions
 ☐ Enter Lambda function ARN

Lambda function

File_Trigger_Automation

Cancel
Save changes

Test With a File task topic 7

1. To test your new lambda function create a trigger automation in Automation Studio.
2. Upload a qualifying file to the S3 bucket location.
3. Confirm that your automation started.

▼ Content Builder API

Content Builder API

Overview




Marketers use Content Builder in Marketing Cloud as a single cross-channel repository for marketing content, such as emails, images, text, and other documents.

Marketing developers, service providers, and third-party developers can create and manipulate marketing content using the Content Builder REST API.

The API uses the term **asset** to refer to all types of marketing content. Assets are hierarchical; an asset can be a message that contains a template, which is itself an asset. The template can contain slots, which are also assets. Assets are stored in both Elasticsearch and in SQL Server; Elasticsearch provides a fast document retrieval engine, while SQL Server provides the reliability Salesforce is famous for. Use the asset model described in this documentation to create, update, delete, query for, and publish assets.

API Specification

These tables include the most commonly used REST resources that make up the Content Builder API.

 HTTP Method	 Resource	 Description
<u>POST</u>	/asset/v1/content/assets/query	Gets an asset collection by advanced query.
<u>GET</u>	/asset/v1/content/assets/{id}	Gets an asset by ID.
<u>POST</u>	/asset/v1/content/assets	Inserts an asset.
<u>PUT</u>	/asset/v1/content/assets/{id}	Updates a full asset.
<u>PATCH</u>	/asset/v1/content/assets/{id}	Updates part of an asset.
<u>DELETE</u>	/asset/v1/content/assets/{id}	Deletes an asset.
<u>GET</u>	/asset/v1/content/assets/{id}/file	Gets the binary file for an asset.
<u>GET</u>	/asset/v1/content/assets/salutations	Gets the default header and footer for an account.
<u>GET</u>	/asset/v1/content/assets/{id}/salutations	Gets the header and footer for a message.
<u>GET</u>	/asset/v1/content/assets/{id}/channelviews/{viewname}	Returns the requested channel view's compiled HTML for the asset.
<u>POST</u>	/asset/v1/content/categories	Inserts a category.
<u>GET</u>	/asset/v1/content/categories	Gets a collection of categories.
<u>GET</u>	/asset/v1/content/categories/{id}	Gets a category by ID.
<u>PUT</u>	/asset/v1/content/categories/{id}	Updates a category by ID.
<u>DELETE</u>	/asset/v1/content/categories/{id}	Deletes a category by ID.
<u>GET</u>	/asset/v1/content/deletedAssets	Returns a collection of all assets deleted in the last 30 days.
<u>PATCH</u>	/asset/v1/content/deletedAssets/AssetID	Updates part of a deleted asset.

▼ Asset Model Reference

Asset Model Reference

The asset model offers a comprehensive yet simple representation of marketing content across Marketing Cloud. This model makes it easy for other clouds, partners, and clients to build their own content representation. The model supports all types of assets, such as emails, HTML, and images. Assets are hierarchical and can contain assets within assets, within assets.

The examples in this documentation are in the context of a partner company that uses the CMS for its email business.

- [Asset Type](#)
- [File Upload](#)
- [Customer Key](#)
- [Categories](#)
- [Channels](#)
- [Views](#)
- [Content](#)
- [Slots](#)
- [Blocks](#)
- [Design and Super Content](#)
- [Data](#)
- [Meta](#)
- [Sharing](#)
- [Template](#)

▼ Asset Type

Asset Type

Every asset has a type, such as email, template, block, cloud page, jpg image, or pdf document. The asset model parses, stores, and searches all assets in the same way, regardless of type, but the type is important to distinguish assets in queries and in the Content Builder UI.

Add an asset type to the asset as an object, using the `assetType` attribute. When creating an asset, only an `assetType.id` is required. If the `assetType.name` is omitted or does not match the ID, the API automatically sets the name.

Creating a webpage (AssetTypeID 205) does not create the asset in the CloudPages UI.

Sample Empty Asset

```
{
  "name": "NTO Welcome Series Email",
  "assetType": {
    "name": "templatebasedemail",
    "id": 207
  }
}
```

▼ File Upload

File Upload

Some asset types are files, such as gif, pdf, and png, and can be created using the asset model. Types are named based on the file extension they support.

Add a file to the asset as an encoded base64 string using the `file` attribute. The maximum size per file is 5 MB.

Sample File Upload

```
{
  "name": "ipx transparent gif asset",
  "assetType": {
    "name": "gif",
    "id": 20
  },
  "file": "R0lGODlhAQABAIAAAAAAAP///yH5BAEAAAAALAAAAABAAEAAIBRAA7",
  "modelVersion": 2,
  "fileProperties": {
    "fileName": "ipx transparent gif asset.gif"
  }
}
```

▼ Customer Key

Customer Key

Customer keys allow you to identify and search for an asset by a data value known only by you. A customer key is a unique value across your enterprise that is typically stored in your database. The customer key is not a required field, and if you don't provide a value, a unique GUID will be created for you.

Add a customer key to the asset as a string with a 36-character limit, using the `customerKey` attribute.

Sample Asset with Customer Key

```
{
  "name": "NTO Welcome Series Email",
  "assetType": {
    "id": 207,
    "name": "templatebasedemail"
  },
  "customerKey": "090c69e7-59ac-4286-8733-f8f7613ab470"
}
```

▼ Categories (Folders)

Categories (Folders)

You can create assets in any folder known by your users, or move an existing asset into a folder. The asset stores the folder information in an attribute called category.

To create or move an asset into a folder, add a category to the asset as an object, using the `category` attribute. Add the ID of the category.

Sample Asset with Category

```
{
  "name": "NTO Welcome Series Email",
  "assetType": {
    "id": 207,
    "name": "templatebasedemail"
  },
  "category": { "id": 1234567 }
}
```

▼ Channels

Channels

Channels are virtual separations of use cases for your marketing content. They typically correspond to a different internal app, or a different partner, even though partners with separate use cases could have multiple channels. For example, Email Studio has the "email" channel and a web app could have the "web" channel.

An asset can be restricted to one or some channels and can hold different content for different channels using views.

Add channel restrictions to the asset as an object, using the `channels` attribute.

▼ Views

Views

For multi-channel assets, use a channel view to hold the different content for the different channels. This allows the asset to parse and render its content differently based on the channel in which it is used. Use a generic `html` view as a fallback in case the channel view used to render the asset is not found in the views.

Add views to the asset as an object under the `views` attribute. Each key under views represents a channel view and contains an asset for this channel.

Sample Asset with Views

```
{
  "name": "NTO Welcome Series Email",
  "channels": {
    "email": true,
    "web": false
  },
  "views": {
    "html": {},
    "text": {},
    "subjectline": {},
    "preheader": {},
    "viewAsAWebPage": {},
    "subscriptioncenter": {},
    "forwardHTML": {},
    "forwardText": {}
  },
  "assetType": {
    "name": "templatebasedemail",
    "id": 207
  }
}
```

▼ Content

Content

The content attribute contains the asset's actual content that will be sent to customers. The content attribute is used at all levels of the asset model to hold each asset's content and identify where to sub-content should go.

When pasting HTML into the content attribute, HTML contains certain reserved chars that need to be escaped in JSON. Run the HTML through a JSON escape tool first to ensure that the HTML is valid in the JSON payload.

Add content to the asset as a string, using the `content` attribute.

Sample Asset with Content

```
{
  "name": "NTO Welcome Series Email",
  "channels": {
    "email": true,
    "web": false
  },
  "views": {
    "html": {
      "content": "<!DOCTYPE html><body>This is a simple message.</body></html>"
    },
    "text": {},
    "subjectline": {},
    "preheader": {}
  },
  "assetType": {
    "name": "templatebasedemail",
    "id": 207
  }
}
```

▼ Slots

Slots

Slots allow you to control, manipulate, and restrict content blocks. You can create slots, move slots to different blocks, and enforce block number and type restrictions within slots. Blocks within slots are limited to 2x recursion, with a maximum depth of 4 objects.

Add slots to the asset as an object under `slots` and to the `content` as a placeholder, where the object keys match the placeholder's data-key attribute. In this example, the partner uses slots with one initial block per slot and no number or type restrictions.

Sample Asset with Slots

```
{
  "name": "NTO Welcome Series Email",
  "channels": {
    "email": true,
    "web": false
  },
  "views": {
    "html": {
      "content": "<!DOCTYPE html><body><div><div>Some header content</div><div data-type=\"slot\" data-key=\"firstslot\"></div><br /><div data-type=\"slot\" data-key=\"secondslot\"></div></div></body></html>",
      "slots": {
        "firstslot": {},
        "secondslot": {}
      }
    },
    "text": {},
    "subjectline": {},
    "preheader": {}
  },
  "assetType": {
    "name": "templatebasedemail",
    "id": 207
  }
}
```

```

"assetType": {
  "name": "templatebasedemail",
  "id": 207
}
}

```

▼ Blocks

Blocks

Blocks are the simplest form of sub-content in an asset. Blocks have their own asset type, can be nested inside an asset, and have their own content and design. They can also be saved or loaded independently in Content Builder. Each block's assetType matches the individual block editing experience.

Add blocks to the asset as an object under `blocks` and to the `content` as a placeholder, where the object keys match the placeholder's `data-key` attribute. All blocks must be loaded inside of slots.

Min, max, allowed blocks

An asset can specify the minimum and maximum number of blocks, as well as the type of blocks allowed to live under it.

Add block attributes to the asset under as number, number, and array of strings under the `minBlocks`, `maxBlocks`, and `allowedBlocks` attributes, respectively. The array holds assetType names for the allowed blocks. This example requires exactly one follow-up link of the `textblock` type and at least one author.

Sample Asset with Blocks

```

{
  "name": "NTO Welcome Series Email",
  "channels": {
    "email": true,
    "web": false
  },
  "views": {
    "html": {
      "content": "<!DOCTYPE html><body><div><div>Some header content</div><div data-type=\"slot\" data-key=\"firstslot\"></div><br /><div data-type=\"slot\" data-key=\"secondslot\"></div></div></body></html>",
      "slots": {
        "firstslot": {
          "content": "<div data-type=\"block\" data-key=\"block1\"></div><div data-type=\"block\" data-key=\"block2\"></div>",
          "blocks": {
            "block1": {
              "content": "123",
              "assetType": {
                "name": "freeformblock",
                "id": 195
              }
            },
            "block2": {
              "content": "234",
              "assetType": {
                "name": "htmlblock",
                "id": 197
              }
            }
          }
        },
        "secondslot": {
          "content": "<div data-type=\"block\" data-key=\"block3\"></div><div data-type=\"block\" data-key=\"block4\"></div>",
          "minBlocks": 1,
          "maxBlocks": 1,
          "allowedBlocks": ["textblock"],
          "blocks": {
            "block3": {
              "content": "345",
              "assetType": {
                "name": "textblock",
                "id": 196
              }
            }
          }
        }
      }
    }
  }
}

```

```

    "text": {},
    "subjectline": {},
    "preheader": {}
  },
  "assetType": {
    "name": "templatebasedemail",
    "id": 207
  }
}

```

▼ Design and Super Content

Design and Super Content

Design content is a default for the content attribute that displays when editing an asset in Marketing Cloud or as content thumbnails or previews when there is no actual content. Once the asset is published, design content is ignored. Design is only supported when used with blocks and slots.

Super content is realistic content that displays when editing an asset. Like design content, super content is only used when editing and is ignored once the asset is published. Super content can be used to show human readable content instead of code, valid URLs and images instead of broken ones, markup that will be resolved at publishing time, or images that don't exist yet. Super content is only supported when used with slots.

Add design content to the asset as a string, using the `design` attribute.

Add super content to the asset as a string, using the `superContent` attribute. In this example, the partner uses `!!!attribute` to identify placeholders they later replace using their API.

Sample Asset with Design and Super Content

```

{
  "name": "NTO Welcome Series Email",
  "channels": {
    "email": true,
    "web": false
  },
  "views": {
    "html": {
      "content": "<!DOCTYPE html><body><div><div>Some header content</div><div data-type=\"slot\" data-key=\"firstslot\"></div><br /><div data-type=\"slot\" data-key=\"secondslot\"></div></div></body></html>",
      "slots": {
        "firstslot": {
          "content": "<div data-type=\"block\" data-key=\"block1\"></div><div data-type=\"block\" data-key=\"block2\"></div>",
          "design": "<div>Drag blocks here to start editing them!</div>",
          "blocks": {
            "block1": {
              "content": "123",
              "design": "click here to start editing this block",
              "assetType": {
                "name": "freeformblock",
                "id": 195
              }
            },
            "block2": {
              "content": "234",
              "design": "click here to start editing this block",
              "assetType": {
                "name": "htmlblock",
                "id": 197
              }
            }
          }
        },
        "secondslot": {
          "content": "<div data-type=\"block\" data-key=\"block3\"></div><div data-type=\"block\" data-key=\"block4\"></div>",
          "design": "<div>Drag blocks here to start editing them!</div>",
          "minBlocks": 1,
          "maxBlocks": 1,
          "allowedBlocks": ["textblock"],
          "blocks": {
            "block3": {
              "superContent": "Some ampscript or custom markup or http://www.valid.url/likeThis",
              "content": "Some %%ampscript%% or !!!customMarkup or http://www.invalid.url/%%likeThis%%",
              "design": "click here to start editing this block",
              "assetType": {

```

```

      "name": "textblock",
      "id": 196
    }
  },
  "text": {},
  "subjectline": {},
  "preheader": {}
},
"assetType": {
  "name": "templatebasedemail",
  "id": 207
}
}

```

▼ Data

Data

Data lets you store additional information for a channel in the asset, to be used by the channel's parser, or some UI that manipulates the asset for the channel. If using data, store properly namespaced custom fields under the channel name. If data is returned, be sure to pass it through the API.

Add data to the asset as an object, under the `data` attribute. Store channel-specific attributes under their channel name inside of `data`. For example, the `email` channel needs additional information.

Sample Asset with Data

```

{
  "name": "NTO Welcome Series Email",
  "channels": {
    "email": true,
    "web": false
  },
  "assetType": {
    "name": "templatebasedemail",
    "id": 207
  },
  "data": {
    "email": {
      "attributes": [
        {
          "DisplayName": "sample_attribute",
          "Name": "__AdditionalEmailAttribute1",
          "Value": "your_tracking",
          "Order": 1,
          "Channel": "email",
          "AttributeType": "AdditionalEmailAttribute"
        },
        {
          "DisplayName": "utm_medium",
          "Name": "__AdditionalEmailAttribute2",
          "Value": "salesforce",
          "Order": 2,
          "Channel": "email",
          "AttributeType": "AdditionalEmailAttribute"
        }
      ]
    }
  }
}

```

▼ Meta

Meta

Meta is used much like data but for internal functionality in Content Builder. Use channel-specific attributes instead of meta to leverage the data attribute. If meta is returned, be sure to pass it through the API.

▼ Sharing

Sharing

You can share assets with one or more business units that have Content Builder Sharing enabled.

Add sharing business units to the asset as an array, using the `sharingProperties` object. To share an asset, include the enterprise business unit's MID in the asset's `sharedWith` property.

Sample Asset with Sharing Properties

```
{
  "name": "NTO Welcome Series Email",
  "assetType": {
    "id": 207,
    "name": "templatebasedemail"
  },
  "sharingProperties": {
    "sharedWith": [333, 555],
    "sharingType": "view"
  }
}
```

sharedWith

An individual asset is shared with the same sharing settings and permissions in all business units. `SharedWith` is a list of up to 100 business unit MIDs that an asset is shared with.

- In Enterprise 1.0 accounts, assets can only be shared down to child business units.
- In Enterprise 2.0 accounts, assets can be shared to any business unit.
- Sharing in agency accounts is not supported.
- To share an asset with your entire enterprise, set `sharedWith` to 0.

┆ Changes to the `sharedWith` property are asynchronous.

sharingType

`SharingType` indicates the permission that you are granting to the list of MIDs in `sharedWith`. Asset sharing has three valid types.

- `view` - Recipients can view the shared asset and select it to be sent in a message, but they cannot edit or move the content.
- `edit` - Recipients can edit the shared asset, but they cannot move it. Any edit made in one business unit appears in the other business units that have access to the content.

┆ Edit is the only sharing type allowed for categories.

- `local` - This option is available for emails only. Recipients can create a local edit of the email, which appears only in their business unit as a separate asset with its own Customer Key and ID. This edit does not overwrite the shared email in other business units. When you create a local edit, the original shared email no longer appears in the recipient's business unit. However, if you delete the local edit, the original shared email appears again. If the business unit that shared the original email deletes or unshares the original, local edits are deleted.

▼ Create Local Copy of a Shared Asset

Create Local Copy of a Shared Asset

To edit an existing asset locally, share the asset, then create a local copy. The original asset and local copy retain sharing properties that associate them with each other.

1. Share an asset to a set of MIDs. In this example, the asset is shared with business units 333 and 555.

```
{
  ...
  "sharingProperties": {
    "sharedWith": [ 333, 555 ],
    "sharingType": "local"
  }
}
```

```
}  
}
```

2. Retrieve the original asset using `GET /asset/v1/content/assets/{id}`.
3. Create a new asset with the original asset's payload plus the sharingProperties using `POST /asset/v1/content/assets`.

- `sharedFrom` is the asset ID of the original asset.
- `sharedFromMID` is MID the asset came from.

```
{  
  ...  
  "sharingProperties" : {  
    "sharedFrom" : 12345,  
    "sharedFromMID" : 333  
  }  
}
```

4. If you retrieve the original asset again, notice that the `sharingProperties` are updated with the local edit created by business unit 333. The read-only `localAssets` property stores the MID and asset ID for the local edit in the recipient MID.

```
{  
  "sharingProperties" : {  
    "sharedWith" : [ 333, 555 ],  
    "sharingType" : "local",  
    "localAssets" : {  
      "333" : "98765"  
    }  
  }  
}
```

▼ Shared Categories (Folders)

Shared Categories (Folders)

You can share categories with one or more business units that have Content Builder Sharing enabled.

- Create shared categories under the root category called Shared Content. Categories under the Content Builder root category cannot be shared. Root categories in Content Builder are read-only.
- All shared categories live in and are owned by the enterprise business unit. Shared categories can be shared selectively to specific MIDs or globally, if your account has access to Content Builder Across Enterprise Sharing.
- Shared categories can hold up to 100 subcategories and 5,000 assets.

Note

The Shared Content root category can hold 5,000 assets, but it is limited to 100 subcategories.

- When creating or moving assets into a shared category, share them with the same MIDs that the category is shared with.

Shared Category Scenarios

Each of the following examples describes a scenario where you cannot create or update a shared category because the `sharedWith` properties of the categories aren't compatible.

Scenario 1

If you create shared category **Indianapolis** under the Shared Category root category and only share it to business units A and B, you cannot create a category under **Indianapolis** that is shared globally.

Scenario 2

If you create shared category **Indianapolis** under the Shared Category root category and only share it to business units A and B, you cannot create a category under **Indianapolis** that is shared to business unit C.

Scenario 3

If shared category **Indy 500** is shared with business units A and B, and a shared category under **Indy 500** named **SubFolder** is shared with business unit B, you cannot remove business unit B from **Indy 500's** `sharedWith` properties.

Scenario 4

If shared category **Salesforce Tower** is shared with business units A and B, and you have several assets in the shared category shared in the same fashion, you cannot update the sharing settings of **Salesforce Tower** to share the category with only business unit C. The updated `sharedWith` property would conflict with the properties of the assets in that shared category.

Differences Between SharingProperties on Assets and Categories

Here is the `sharingProperties` object from a sample shared category:

```
{
  ...
  "sharingProperties": {
    "sharedWith": [ 333, 555 ],
    "sharingType": "edit"
  }
}
```

The `sharingProperties` object on assets and categories is structured the same, but it is populated differently. The only valid properties in the `sharingProperties` object for categories are `sharedWith` and `sharingType`. `sharedWith` is an array of business unit MIDs that you are sharing this category with. `sharingType` indicates the permission that you are granting to the MIDs. The only possible value for categories is `edit`, meaning shared categories can always be viewed and edited. If a shared category's `sharedWith` property is updated to add or remove MIDs, all assets in that shared category are asynchronously updated to have the same sharing properties, as long as the update to the category's `sharedWith` property does not fall into Scenario 4.

▼ Template

Template

Template designates another asset as a starting point for the current asset. The template acts as a fallback if blocks or slots in the asset are not found in their corresponding attribute. For example, if the asset contains a slots placeholder but no slots attribute, and that asset has a template, the given slot in template.slots will be used, if found. Templates can also provide locking mechanism to prevent assets from using them as templates to modify specific sections of content.

You can only use a template within the `html` view inside a top level attribute views `object.views.html.template`.

Add a template to the asset as an object, under the `template` attribute. For example, this asset uses a template for the `html` view, with some changes to the content slots.

The template is pulled from an existing template asset and therefore has an ID attribute. When blocks are imported as existing blocks or when they are saved separately, they will also have an ID attribute, like `block1` in this example.

Sample Asset with Template

```
{
  "name": "NTO Welcome Series Email",
  "channels": {
    "email": true,
    "web": false
  },
  "views": {
    "html": {
      "template": {
        "id": 108494,
        "content": "<!DOCTYPE html><body><div><div>Some header content</div><div data-type=\"slot\" data-key=\"firstslot\"></div><br /><div data-type=\"slot\" data-key=\"secondslot\"></div></div></body></html>",
        "slots": {
          "firstslot": {
            "content": "<div data-type=\"block\" data-key=\"block1\"></div><div data-type=\"block\" data-key=\"block2\"></div>",
            "design": "<div>Drag blocks here to start editing them!</div>",
            "blocks": {
              "block1": {
                "content": "123",
                "design": "click here to start editing this block",
                "assetType": {
                  "name": "freeformblock",
                  "id": 195
                }
              }
            }
          }
        }
      }
    }
  }
}
```

```

        "block2": {
          "content": "234",
          "design": "click here to start editing this block",
          "assetType": {
            "name": "htmlblock",
            "id": 197
          }
        }
      },
      "secondslot": {
        "content": "<div data-type=\"block\" data-key=\"block3\"></div><div data-type=\"block\" data-key=\"block4\"></div>",
        "design": "<div>Drag blocks here to start editing them!</div>",
        "minBlocks": 1,
        "maxBlocks": 1,
        "allowedBlocks": ["textblock"],
        "blocks": {
          "block3": {
            "superContent": "Some ampscript or custom markup or http://www.valid.url/likeThis",
            "content": "Some %%ampscript%% or !!!customMarkup or http://www.invalid.url/%%likeThis%%",
            "design": "click here to start editing this block",
            "assetType": {
              "name": "textblock",
              "id": 196
            }
          }
        }
      },
      "assetType": {
        "name": "template",
        "id": 4
      }
    },
    "content": "<!DOCTYPE html><body><div><div>Some CHANGED header content</div><div data-type=\"slot\" data-key=\"firstslot\"></div><br /><div data-type=\"slot\" data-key=\"secondslot\"></div></div></body></html>",
    "slots": {
      "firstslot": {
        "content": "<div data-type=\"block\" data-key=\"block1\"></div><div data-type=\"block\" data-key=\"block2\"></div>",
        "design": "<div>Drag blocks here to start editing them!</div>",
        "blocks": {
          "block1": {
            "id": 456,
            "content": "CHANGED",
            "design": "click here to start editing this block",
            "assetType": {
              "name": "freeformblock",
              "id": 195
            }
          },
          "block2": {
            "content": "CHANGED",
            "design": "click here to start editing this block",
            "assetType": {
              "name": "htmlblock",
              "id": 197
            }
          }
        }
      },
      "secondslot": {
        "content": "<div data-type=\"block\" data-key=\"block3\"></div><div data-type=\"block\" data-key=\"block4\"></div>",
        "design": "<div>Drag blocks here to start editing them!</div>",
        "minBlocks": 1,
        "maxBlocks": 1,
        "allowedBlocks": ["textblock"],
        "blocks": {
          "block3": {
            "superContent": "Some CHANGED ampscript or custom markup or http://www.valid.url/likeThis",
            "content": "Some CHANGED %%ampscript%% or !!!customMarkup or http://www.invalid.url/%%likeThis%%",
            "design": "click here to start editing this block",
            "assetType": {
              "name": "textblock",
              "id": 196
            }
          }
        }
      }
    }
  },
  "text": {}
}

```

```

    "subjectline": {},
    "preheader": {}
  },
  "assetType": {
    "name": "templatebasedemail",
    "id": 207
  },
  "data": {
    "email": {
      "someEmailSpecificAttribute": 5
    }
  }
}

```

▼ Asset Model Examples

Asset Model Examples

Use these examples to create basic HTML paste and template-based emails.

Note

When pasting HTML into the content attribute, HTML contains certain reserved chars that need to be escaped in JSON. Run the HTML through a JSON escape tool first to ensure that the HTML is valid in the JSON payload.

- [Asset Model Examples](#)
 - [HTML Paste Email Example](#)
 - [Simple Template Based Email Example](#)
 - [Complex Template Based Email Example](#)
 - [Related Items](#)

HTML Paste Email Example

```

{
  "name": "my first message",
  "channels": {
    "email": true,
    "web": false
  },
  "views": {
    "html": {
      "content": "<!DOCTYPE html><body>This is a simple html paste email.</body></html>"
    },
    "text": {},
    "subjectline": {},
    "preheader": {}
  },
  "assetType": {
    "name": "htmlemail",
    "id": 208
  }
}

```

Simple Template Based Email Example

Source

```

{
  "assetType": {
    "id": 207
  },
  "name": "NTO Welcome Series Email",
  "views": {
    "subjectline": {
      "content": "Welcome %%First_Name%% to Northern Trail Outfitters"
    },
    "preheader": {},
    "html": {
      "content": "<html><head></head><body><div><div data-type=\"slot\" data-key=\"myslot\"></div></div></body></html>",
      "slots": {
        "myslot": {
          "content": ""
        }
      }
    }
  }
}

```

```
"design": "<p style=\"font-family: arial; color: #CCCCC; font-size: 12px; font-weight: bold; text-align: center; display: flex; flex-direction: column; justify-content: center; height: 150px; padding: 10px; margin: 0; border: 1px dashed #CCCCC;\">Drop blocks or content here</p>"
},
"template": {
  "id": 108494
}
}
},
"data": {
  "email": {
    "options": {
      "characterEncoding": "utf-8"
    }
  }
}
```

Complex Template Based Email Example

Source

```
{
  "id": 108495,
  "customerKey": "090c69e7-59ac-4286-8733-f8f7613ab470",
  "objectId": "1cd57450-3f4a-46a8-9177-068f031ff56f",
  "contentType": "application/vnd.etmc.email.Message; kind=template",
  "assetType": {
    "id": 207,
    "name": "templatebasedemail",
    "displayName": "Template-Based Email"
  },
  "name": "NTO Welcome Series Email",
  "description": "",
  "owner": {
    "id": 10817573,
    "email": "user@example.com",
    "name": "User"
  },
  "createdDate": "2016-12-05T21:28:43.317-06:00",
  "createdBy": {
    "id": 10817573,
    "email": "user@example.com",
    "name": "User"
  },
  "modifiedDate": "2016-12-07T07:43:15.17-06:00",
  "modifiedBy": {
    "id": 10817573,
    "email": "user@example.com",
    "name": "User"
  },
  "enterpriseId": 10806781,
  "memberId": 10806781,
  "status": {
    "id": 1,
    "name": "Draft"
  },
  "sharingProperties": {
    "sharedWith": [
      333,
      555
    ],
    "sharingType": "view"
  },
  "category": {
    "id": 1467802,
    "parentId": 0,
    "name": "Content Builder"
  },
  "views": {
    "subjectline": {
      "contentType": "application/vnd.etmc.email.View; kind=subjectline",
      "thumbnail": {
        "thumbnailHtml": "Welcome %%First_Name%% to Northern Trail Outfitters"
      },
      "content": "Welcome %%First_Name%% to Northern Trail Outfitters",
      "meta": {},
      "data": {
        "email": {
```

```

        "options": {
            "generateFrom": ""
        }
    },
    "modelVersion": 2
},
"preheader": {
    "thumbnail": {
        "thumbnailHtml": ""
    },
    "data": {
        "email": {
            "options": {
                "generateFrom": ""
            }
        }
    },
    "modelVersion": 2
},
"text": {
    "data": {
        "email": {
            "options": {
                "generateFrom": "html"
            }
        }
    },
    "generateFrom": "html",
    "modelVersion": 2
},
"viewAsWebPage": {
    "thumbnail": {
        "thumbnailHtml": ""
    },
    "data": {
        "email": {
            "options": {
                "generateFrom": ""
            }
        }
    },
    "modelVersion": 2
},
"subscriptioncenter": {
    "thumbnail": {
        "thumbnailHtml": ""
    },
    "data": {
        "email": {
            "options": {
                "generateFrom": ""
            }
        }
    },
    "modelVersion": 2
},
"forwardHTML": {
    "thumbnail": {
        "thumbnailHtml": ""
    },
    "data": {
        "email": {
            "options": {
                "generateFrom": ""
            }
        }
    },
    "modelVersion": 2
},
"forwardText": {
    "thumbnail": {
        "thumbnailHtml": ""
    },
    "data": {
        "email": {
            "options": {
                "generateFrom": ""
            }
        }
    },
    "modelVersion": 2
},

```

```
"modelVersion": 2
},
"html": {
  "content": "<!DOCTYPE html> \n<html> \n <head> \n <meta name=\"viewport\" content=\"width=device-width, initial-scale=1, maximum-scale=1, user-scalable=0\" /> \n <meta http-equiv=\"Content-Type\" content=\"text/html; charset=UTF-8\" /> \n <style type=\"text/css\"> .ReadMsgBody { \n width: 100%; \n } \n .ExternalClass { \n width: 100%; \n } \n div, p, a, li, td { \n -webkit-text-size-adjust:none; \n } \n body { \n -webkit-text-size-adjust:100%; \n -ms-text-size-adjust:100%; \n margin:0 !important; \n padding:0 !important; \n } \n table td { \n border-collapse: collapse; \n } \n img { \n outline:0; \n } \n a img { \n border:none; \n } \n @media screen and (max-width: 480px) { \n td[class=\"headercell-phone\"] { \n display:none; \n } \n table[class=\"container\"] { \n width:360px !important; \n } \n table[class=\"hero\"] { \n width:100% !important; \n } \n table[class=\"full-width\"] { \n width:100% !important; \n float:left !important; \n padding:10px 15px 10px 15px !important; \n } \n table[class=\"full-width-hero-content\"], td[class=\"full-width-hero-content\"] { \n width:90% !important; \n float:left !important; \n padding: 0 5% 0 5% !important; \n display: block !important; \n } \n table[class=\"full-width-hero-image\"], td[class=\"full-width-hero-image\"] { \n float:left !important; \n display: block !important; \n } \n img[class=\"photo\"] { \n width:100% !important; \n height:auto !important; \n display: block; \n } \n td[id=\"preheader\"] { \n width:auto !important; \n } \n td[class=\"row\"] { \n width:100% !important; \n } \n td[class=\"hide\"] { \n display:none !important; \n } \n td[class=\"headercell-phone\"] { \n display:inline !important; \n } \n td[class=\"headercell-desktop\"] img { \n display: none !important; \n } \n span[class=\"center-text\"] { \n text-align:center !important; \n } \n } \n </style> \n </head> \n <body bgcolor=\"#e8e8e8\"> \n <!-- ===== --> \n <!-- Flexible Branding Template --> \n <!-- ===== --> \n <table width=\"100%\" border=\"0\" cellspacing=\"0\" cellpadding=\"0\" bgcolor=\"#e8e8e8\"> \n <tr> \n <td> \n <!-- ===== --> \n <!-- White background wrapper --> \n <!-- ===== --> \n <table class=\"container\" bgcolor=\"#ffffff\" width=\"600\" align=\"center\" border=\"0\" cellspacing=\"0\" cellpadding=\"0\" style=\"background-color:#ffffff !important;\"> \n <tr> \n <td> \n <!-- ===== --> \n <!-- Logo and Preheader --> \n <!-- ===== --> \n <table class=\"container\" bgcolor=\"#6ccdf0\" style=\"padding: 20px 0 0 0;\" width=\"600\" align=\"center\" border=\"0\" cellspacing=\"0\" cellpadding=\"0\"> \n <tr> \n <td width=\"195\" style=\"padding:0 15px 15px 0px;> \n <center> \n <a href=\"http://ntoutfitters.com\" alias=\"\"NTO Logo\" style=\"outline:none\"><img src=\"https://s3-us-west-2.amazonaws.com/rd-industry-emails/retail/nto/brand-assets/email-templates/branding-nto-logo-white.png\"></a> \n </center> \n </td> \n <td class=\"hide\" width=\"225\" style=\"font-family:Gotham, Helvetica, Arial, sans-serif; color:#ffffff; font-size:13px; line-height:12px; text-align:right; padding:0 25px 0 0;\"> \n <!-- CONTENT AREA --> \n <div data-type=\"slot\" data-key=\"preheader\" data-allowed-blocks=\"\" data-max-blocks=\"\" data-label=\"\"> \n </div> \n <!-- END CONTENT AREA --> \n </td> \n </tr> \n </table> \n <!-- ===== --> \n <table class=\"container\" width=\"600\" align=\"center\" border=\"0\" cellspacing=\"0\" cellpadding=\"0\"> \n <tr> \n <td style=\"font-family:Gotham, Helvetica, Arial, sans-serif; color:#4f4f4f; font-size:16px; line-height:22px; padding:0 0 25px 0;\"> \n <!-- CONTENT AREA --> \n <div data-type=\"slot\" data-key=\"heroimage\" data-allowed-blocks=\"\" data-max-blocks=\"\" data-label=\"\"> \n </div> \n <!-- END CONTENT AREA --> \n </td> \n </tr> \n </table> \n <!-- ===== --> \n <!-- Section --> \n <!-- ===== --> \n <table class=\"container\" width=\"600\" align=\"center\" border=\"0\" cellspacing=\"0\" cellpadding=\"0\"> \n <tr> \n <td style=\"font-family:Gotham, Helvetica, Arial, sans-serif; color:#4f4f4f; font-size:16px; line-height:22px; padding:0 25px 0 25px;\"> \n <!-- CONTENT AREA --> \n <div data-type=\"slot\" data-key=\"cell01\" data-allowed-blocks=\"\" data-max-blocks=\"\" data-label=\"\"> \n </div> \n <!-- END CONTENT AREA --> \n </td> \n </tr> \n </table> \n <!-- ===== --> \n <!-- Section --> \n <!-- ===== --> \n <table class=\"container\" width=\"600\" align=\"center\" border=\"0\" cellspacing=\"0\" cellpadding=\"0\"> \n <tr> \n <td style=\"font-family:Gotham, Helvetica, Arial, sans-serif; color:#4f4f4f; font-size:16px; line-height:22px; padding:0 25px 0 25px;\"> \n <!-- CONTENT AREA --> \n <div data-type=\"slot\" data-key=\"cell02\" data-allowed-blocks=\"\" data-max-blocks=\"\" data-label=\"\"> \n </div> \n <!-- END CONTENT AREA --> \n </td> \n </tr> \n </table> \n <!-- ===== --> \n <!-- Section --> \n <!-- ===== --> \n <table class=\"container\" width=\"600\" align=\"center\" border=\"0\" cellspacing=\"0\" cellpadding=\"0\"> \n <tr> \n <td style=\"font-family:Gotham, Helvetica, Arial, sans-serif; color:#4f4f4f; font-size:16px; line-height:22px; padding:0 25px 0 25px;\"> \n <!-- CONTENT AREA --> \n <div data-type=\"slot\" data-key=\"cell03\" data-allowed-blocks=\"\" data-max-blocks=\"\" data-label=\"\"> \n </div> \n <!-- END CONTENT AREA --> \n </td> \n </tr> \n </table> \n <!-- ===== --> \n <!-- Section --> \n <!-- ===== --> \n <table class=\"container\" width=\"600\" align=\"center\" border=\"0\" cellspacing=\"0\" cellpadding=\"0\"> \n <tr> \n <td style=\"font-family:Gotham, Helvetica, Arial, sans-serif; color:#4f4f4f; font-size:16px; line-height:22px; padding:0 25px 0 25px;\"> \n <!-- CONTENT AREA --> \n <div data-type=\"slot\" data-key=\"cell04\" data-allowed-blocks=\"\" data-max-blocks=\"\" data-label=\"\"> \n </div> \n <!-- END CONTENT AREA --> \n </td> \n </tr> \n </table> \n <!-- ===== --> \n <!-- Footer --> \n <!-- ===== --> \n <table width=\"100%\" border=\"0\" cellspacing=\"0\" cellpadding=\"0\" bgcolor=\"#393939\"> \n <tr> \n <td height=\"7px\" bgcolor=\"#6ccdf0\"> \n </td> \n <td> \n <tr> \n <td style=\"font-family:Gotham, Helvetica, Arial, sans-serif; text-align:center; color:#a5a5a5; font-size:9px; line-height:10px; padding:0 50px 35px 50px;\"> \n <a href=\"http://ntoutfitters.com\" target=\"_blank\" alias=\"\"NTO Logo\" style=\"outline:none\"><img src=\"https://s3-us-west-2.amazonaws.com/rd-industry-emails/retail/nto/brand-assets/email-templates/branding-nto-logo-white.png\"></a> \n <br> \n <span style=\"display:block;font-weight:bold;font-size:14px;\">Northern Trail Outfitters</span> \n <br> \n <span style=\"display:block;font-weight:bold;font-size:14px;\">Member_BusName% registered address: \n <br> \n <span style=\"display:block;font-weight:bold;font-size:14px;\">Member_State%, %Member_PostalCode%, %Member_Country%. \n <br> \n </td> \n <td> \n <tr> \n <td style=\"font-family:Gotham, Helvetica, Arial, sans-serif; text-align:center; color:#a5a5a5; font-size:12px; font-weight:bold; line-height:16px; padding:0 55px 25px 55px;\"> \n <a href=\"http://www.ntoutfitters.com\" alias=\"\"Contact Us\" target=\"_blank\" style=\"color:#a5a5a5;text-decoration:none;\">Contact Us</a> \n <a href=\"http://www.ntoutfitters.com\" alias=\"\"Store Locator\" target=\"_blank\" style=\"color:#a5a5a5;text-decoration:none;\">Store Locator</a> \n <a href=\"%subscription_center_url%\" alias=\"\"Manage Subscriptions\" target=\"_blank\" style=\"color:#a5a5a5;text-
```

```

decoration:none;">Manage Subscriptions</a> | \n      <a href="%profile_center_url%" target="_blank" alias="Profile Center"
style="color:#a5a5a5;text-decoration:none;">Update Profile</a> | \n      <a class="mod_click" target="_blank"
href="%unsubscribe_center_url%" alias="Unsubscribe" title="Unsubscribe" style="color:#a5a5a5;text-decoration:none;">Unsubscribe</a>
\n      <br> \n      </td> \n      </tr> \n      <tr> \n      <td style="font-family:Gotham, Helvetica, Arial, sans-serif;
text-align:center; color:#a5a5a5; font-size:12px; line-height:14px; padding:0 100px 25px 100px; text-align:center;"> \n      <a
href="https://www.facebook.com/NTOutfitters" alias="NT0 Facebook" target="_blank"></a> \n      <a href="https://twitter.com/NTOutfitters" alias="NT0 Twitter" target="_blank"></a> \n      <a href="https://www.pinterest.com/ntoutfitters/" alias="NT0 Pinterest" target="_blank"></a> \n      <a href="http://www.ntoutfitters.com/" alias="NT0 Email" target="_blank">
</a> \n      </td> \n      </tr> \n      </table> \n      <div id="tracking_pixel"> \n      <custom name="opencounter"
type="tracking"> \n      </div> \n      <div id="campaign_pixel"> \n      </div> \n      </body> \n      </html>";

"slots": {
  "preheader": {
    "content": "<div data-type=\"block\" data-key=\"szcm0z6zm01hp015izi2akyb9\" data-original-key=\"none\"></div>",
    "design": "<p style=\"font-family: arial; color: #CCCCC; font-size: 12px; font-weight: bold; text-align: center;
display: flex; flex-direction: column; justify-content: center; height: 150px; padding: 10px; margin: 0; border: 1px dashed
#CCCCC;\>Drop blocks or content here</p>",
    "blocks": {
      "szcm0z6zm01hp015izi2akyb9": {
        "assetType": {
          "id": 197,
          "name": "htmlblock"
        },
        "content": "<table cellpadding=\"0\" cellspacing=\"0\" width=\"100%\" class=\"stylingblock-content-wrapper\"
style=\"\"><tr><td class=\"stylingblock-content-wrapper camarker-inner\"><table width=\"100%\" cellspacing=\"0\" cellpadding=\"0\"
border=\"0\"> <tr> <td style=\"font-family: Gotham,Helvetica,Arial,sans-serif; color: #ffffff; font-size: 13px; line-height: 12px;
text-align: right;\> Welcome aboard!<br> <a alias=\"View as a Web Page\" class=\"view_link\" style=\"color: #393939; text-decoration:
none; font-size: 11px; line-height: 24px;\> target=\"_blank\" href=\"%view_email_url%\">WEB VERSION</a> </td> </tr> </table></td>
</tr></table>",
        "design": "<div class=\"default-design\"><div style=\"width:100%;height:150px;\></div></div>",
        "data": {
          "email": {
            "options": {
              "generateFrom": ""
            }
          }
        },
        "modelVersion": 2
      }
    },
    "maxBlocks": 0,
    "allowedBlocks": [],
    "data": {
      "email": {
        "options": {
          "generateFrom": ""
        }
      }
    },
    "modelVersion": 2
  },
  "heroimage": {
    "content": "<div data-type=\"block\" data-key=\"j0amkj3nxauswhdnyr4b73nmi\" data-original-key=\"none\"></div>",
    "design": "<p style=\"font-family: arial; color: #CCCCC; font-size: 12px; font-weight: bold; text-align: center;
display: flex; flex-direction: column; justify-content: center; height: 150px; padding: 10px; margin: 0; border: 1px dashed
#CCCCC;\>Drop blocks or content here</p>",
    "blocks": {
      "j0amkj3nxauswhdnyr4b73nmi": {
        "assetType": {
          "id": 199,
          "name": "imageblock"
        },
        "content": "<table cellpadding=\"0\" cellspacing=\"0\" width=\"100%\" class=\"stylingblock-content-wrapper\"
style=\"\"><tr><td class=\"stylingblock-content-wrapper camarker-inner\"><table width=\"100%\" cellspacing=\"0\" cellpadding=\"0\">
<tr><td><img src=\"https://s3-us-west-2.amazonaws.com/rd-industry-emails/retail/nto/campaign-toolkits/welcome-series/welcome-step-1/welcome-
step1-hero.png\" height=\"auto\" width=\"100%\" class=\"photo\" style=\"height:auto;width:100%;display: block;padding-top: 0px; padding-
right: 0px; padding-bottom: 0px; padding-left: 0px;\></td></tr></table></td></tr></table>",
        "design": "<div class=\"default-design\"><div style=\"width:100%;height:150px;\></div></div>",
        "data": {
          "email": {
            "options": {
              "generateFrom": ""
            }
          }
        },
        "modelVersion": 2
      }
    }
  }
}

```

```

},
"maxBlocks": 0,
"allowedBlocks": [],
"data": {
  "email": {
    "options": {
      "generateFrom": ""
    }
  }
},
},
"modelVersion": 2
},
"cell01": {
  "content": "<div data-type=\"block\" data-key=\"2p58k7y5gk7g0kcnb6jgeklmni\" data-original-key=\"none\"></div>",
  "design": "<p style=\"font-family: arial; color: #CCCCCC; font-size: 12px; font-weight: bold; text-align: center; display: flex; flex-direction: column; justify-content: center; height: 150px; padding: 10px; margin: 0; border: 1px dashed #CCCCCC;\>Drop blocks or content here</p>",
  "blocks": {
    "2p58k7y5gk7g0kcnb6jgeklmni": {
      "assetType": {
        "id": 197,
        "name": "htmlblock"
      },
      "content": "<table cellpadding=\"0\" cellspacing=\"0\" width=\"100%\" class=\"stylingblock-content-wrapper\" style=\"\"><tr><td class=\"stylingblock-content-wrapper camarker-inner\"><table width=\"100%\" cellspacing=\"0\" cellpadding=\"0\" border=\"0\"> <tr> <td style=\"font-family: Gotham,Helvetica,Arial,sans-serif; color: #6d6d6d; text-align: left; font-size: 16px; line-height: 22px; padding: 0px 0px 15px;\> %%First_Name%%,<br> <br> Hello and welcome! Thanks for signing up by %%Opt_in_Source%% - you're officially registered with NTO. Expect to hear from us about the latest outdoor news and all of our best offers and discounts. In the meantime, here's a few things to get you started... </td> </tr> </table></td></tr></table>",
      "design": "<div class=\"default-design\"><div style=\"width:100%;height:150px;\></div></div>",
      "data": {
        "email": {
          "options": {
            "generateFrom": ""
          }
        }
      },
      "modelVersion": 2
    },
    "maxBlocks": 0,
    "allowedBlocks": [],
    "data": {
      "email": {
        "options": {
          "generateFrom": ""
        }
      }
    },
    "modelVersion": 2
  },
  "cell02": {
    "content": "<div data-type=\"block\" data-key=\"bxfltwledxksb8ozl95n91wcdi\" data-original-key=\"none\"></div>",
    "design": "<p style=\"font-family: arial; color: #CCCCCC; font-size: 12px; font-weight: bold; text-align: center; display: flex; flex-direction: column; justify-content: center; height: 150px; padding: 10px; margin: 0; border: 1px dashed #CCCCCC;\>Drop blocks or content here</p>",
    "blocks": {
      "bxfltwledxksb8ozl95n91wcdi": {
        "assetType": {
          "id": 197,
          "name": "htmlblock"
        },
        "content": "<table cellpadding=\"0\" cellspacing=\"0\" width=\"100%\" class=\"stylingblock-content-wrapper\" style=\"\"><tr><td class=\"stylingblock-content-wrapper camarker-inner\"><table width=\"100%\" cellspacing=\"0\" cellpadding=\"0\" border=\"0\" style=\"margin: 10px 0px;\> <tr> <td width=\"15%\" style=\"font-family: Gotham,Helvetica,Arial,sans-serif; color: #6d6d6d; font-size: 16px; line-height: 22px; padding: 0px; text-align: center;\> <img src=\"https://s3-us-west-2.amazonaws.com/rd-industry-emails/retail/nto/campaign-toolkits/welcome-series/welcome-step-1/welcome-step1-icon1.png\"> </td> <td width=\"85%\" style=\"font-family: Gotham,Helvetica,Arial,sans-serif; color: #6d6d6d; font-size: 14px; line-height: 22px; padding: 5px;\> <div class=\"full-width\"> <b>Let's get to know each other.</b><br> Take a minute to fill out your profile and communications preferences so we can send you the most relevant and timely offers and discounts. <!-- ===== --> <!-- Blue CTA BTN --> <!-- ===== --> <table width=\"40%\" class=\"full-width\"> <tr> <td style=\"padding: 10px 0px 0px;\> <a style=\"color: #ffffff; text-decoration: none;\> target=\"_blank\" class=\"mod_click\" href=\"%profile_center_url%\"><span style=\"font-family: Gotham,Helvetica,Arial,sans-serif; display: block; color: #ffffff; text-align: center; font-size: 12px; padding: 7px 0px; border-radius: 20px; background-color: #6ccd0;\>Complete Profile</span></a> </td> </tr> </table> </td> </tr> </table></td></tr></table></div>",
        "design": "<div class=\"default-design\"><div style=\"width:100%;height:150px;\></div></div>",
        "data": {
          "email": {
            "options": {
              "generateFrom": ""
            }
          }
        }
      },
      "modelVersion": 2
    },
    "maxBlocks": 0,
    "allowedBlocks": [],
    "data": {
      "email": {
        "options": {
          "generateFrom": ""
        }
      }
    },
    "modelVersion": 2
  },
  "cell03": {
    "content": "<div data-type=\"block\" data-key=\"bxfltwledxksb8ozl95n91wcdi\" data-original-key=\"none\"></div>",
    "design": "<p style=\"font-family: arial; color: #CCCCCC; font-size: 12px; font-weight: bold; text-align: center; display: flex; flex-direction: column; justify-content: center; height: 150px; padding: 10px; margin: 0; border: 1px dashed #CCCCCC;\>Drop blocks or content here</p>",
    "blocks": {
      "bxfltwledxksb8ozl95n91wcdi": {
        "assetType": {
          "id": 197,
          "name": "htmlblock"
        },
        "content": "<table cellpadding=\"0\" cellspacing=\"0\" width=\"100%\" class=\"stylingblock-content-wrapper\" style=\"\"><tr><td class=\"stylingblock-content-wrapper camarker-inner\"><table width=\"100%\" cellspacing=\"0\" cellpadding=\"0\" border=\"0\" style=\"margin: 10px 0px;\> <tr> <td width=\"15%\" style=\"font-family: Gotham,Helvetica,Arial,sans-serif; color: #6d6d6d; font-size: 16px; line-height: 22px; padding: 0px; text-align: center;\> <img src=\"https://s3-us-west-2.amazonaws.com/rd-industry-emails/retail/nto/campaign-toolkits/welcome-series/welcome-step-1/welcome-step1-icon1.png\"> </td> <td width=\"85%\" style=\"font-family: Gotham,Helvetica,Arial,sans-serif; color: #6d6d6d; font-size: 14px; line-height: 22px; padding: 5px;\> <div class=\"full-width\"> <b>Let's get to know each other.</b><br> Take a minute to fill out your profile and communications preferences so we can send you the most relevant and timely offers and discounts. <!-- ===== --> <!-- Blue CTA BTN --> <!-- ===== --> <table width=\"40%\" class=\"full-width\"> <tr> <td style=\"padding: 10px 0px 0px;\> <a style=\"color: #ffffff; text-decoration: none;\> target=\"_blank\" class=\"mod_click\" href=\"%profile_center_url%\"><span style=\"font-family: Gotham,Helvetica,Arial,sans-serif; display: block; color: #ffffff; text-align: center; font-size: 12px; padding: 7px 0px; border-radius: 20px; background-color: #6ccd0;\>Complete Profile</span></a> </td> </tr> </table> </td> </tr> </table></td></tr></table></div>",
        "design": "<div class=\"default-design\"><div style=\"width:100%;height:150px;\></div></div>",
        "data": {
          "email": {
            "options": {
              "generateFrom": ""
            }
          }
        }
      },
      "modelVersion": 2
    },
    "maxBlocks": 0,
    "allowedBlocks": [],
    "data": {
      "email": {
        "options": {
          "generateFrom": ""
        }
      }
    },
    "modelVersion": 2
  },
  "cell04": {
    "content": "<div data-type=\"block\" data-key=\"bxfltwledxksb8ozl95n91wcdi\" data-original-key=\"none\"></div>",
    "design": "<p style=\"font-family: arial; color: #CCCCCC; font-size: 12px; font-weight: bold; text-align: center; display: flex; flex-direction: column; justify-content: center; height: 150px; padding: 10px; margin: 0; border: 1px dashed #CCCCCC;\>Drop blocks or content here</p>",
    "blocks": {
      "bxfltwledxksb8ozl95n91wcdi": {
        "assetType": {
          "id": 197,
          "name": "htmlblock"
        },
        "content": "<table cellpadding=\"0\" cellspacing=\"0\" width=\"100%\" class=\"stylingblock-content-wrapper\" style=\"\"><tr><td class=\"stylingblock-content-wrapper camarker-inner\"><table width=\"100%\" cellspacing=\"0\" cellpadding=\"0\" border=\"0\" style=\"margin: 10px 0px;\> <tr> <td width=\"15%\" style=\"font-family: Gotham,Helvetica,Arial,sans-serif; color: #6d6d6d; font-size: 16px; line-height: 22px; padding: 0px; text-align: center;\> <img src=\"https://s3-us-west-2.amazonaws.com/rd-industry-emails/retail/nto/campaign-toolkits/welcome-series/welcome-step-1/welcome-step1-icon1.png\"> </td> <td width=\"85%\" style=\"font-family: Gotham,Helvetica,Arial,sans-serif; color: #6d6d6d; font-size: 14px; line-height: 22px; padding: 5px;\> <div class=\"full-width\"> <b>Let's get to know each other.</b><br> Take a minute to fill out your profile and communications preferences so we can send you the most relevant and timely offers and discounts. <!-- ===== --> <!-- Blue CTA BTN --> <!-- ===== --> <table width=\"40%\" class=\"full-width\"> <tr> <td style=\"padding: 10px 0px 0px;\> <a style=\"color: #ffffff; text-decoration: none;\> target=\"_blank\" class=\"mod_click\" href=\"%profile_center_url%\"><span style=\"font-family: Gotham,Helvetica,Arial,sans-serif; display: block; color: #ffffff; text-align: center; font-size: 12px; padding: 7px 0px; border-radius: 20px; background-color: #6ccd0;\>Complete Profile</span></a> </td> </tr> </table> </td> </tr> </table></td></tr></table></div>",
        "design": "<div class=\"default-design\"><div style=\"width:100%;height:150px;\></div></div>",
        "data": {
          "email": {
            "options": {
              "generateFrom": ""
            }
          }
        }
      },
      "modelVersion": 2
    },
    "maxBlocks": 0,
    "allowedBlocks": [],
    "data": {
      "email": {
        "options": {
          "generateFrom": ""
        }
      }
    },
    "modelVersion": 2
  },
  "cell05": {
    "content": "<div data-type=\"block\" data-key=\"bxfltwledxksb8ozl95n91wcdi\" data-original-key=\"none\"></div>",
    "design": "<p style=\"font-family: arial; color: #CCCCCC; font-size: 12px; font-weight: bold; text-align: center; display: flex; flex-direction: column; justify-content: center; height: 150px; padding: 10px; margin: 0; border: 1px dashed #CCCCCC;\>Drop blocks or content here</p>",
    "blocks": {
      "bxfltwledxksb8ozl95n91wcdi": {
        "assetType": {
          "id": 197,
          "name": "htmlblock"
        },
        "content": "<table cellpadding=\"0\" cellspacing=\"0\" width=\"100%\" class=\"stylingblock-content-wrapper\" style=\"\"><tr><td class=\"stylingblock-content-wrapper camarker-inner\"><table width=\"100%\" cellspacing=\"0\" cellpadding=\"0\" border=\"0\" style=\"margin: 10px 0px;\> <tr> <td width=\"15%\" style=\"font-family: Gotham,Helvetica,Arial,sans-serif; color: #6d6d6d; font-size: 16px; line-height: 22px; padding: 0px; text-align: center;\> <img src=\"https://s3-us-west-2.amazonaws.com/rd-industry-emails/retail/nto/campaign-toolkits/welcome-series/welcome-step-1/welcome-step1-icon1.png\"> </td> <td width=\"85%\" style=\"font-family: Gotham,Helvetica,Arial,sans-serif; color: #6d6d6d; font-size: 14px; line-height: 22px; padding: 5px;\> <div class=\"full-width\"> <b>Let's get to know each other.</b><br> Take a minute to fill out your profile and communications preferences so
```



```

    },
    "modelVersion": 2
  },
  "maxBlocks": 0,
  "allowedBlocks": [],
  "data": {
    "email": {
      "options": {
        "generateFrom": ""
      }
    }
  },
  "modelVersion": 2
},
"cell03": {
  "content": "<div data-type=\"block\" data-key=\"nnrzsge987d6b1jqk2n4b1emi\" data-original-key=\"none\"></div>",
  "design": "<p style=\"font-family: arial; color: #CCCCC; font-size: 12px; font-weight: bold; text-align: center; display: flex; flex-direction: column; justify-content: center; height: 150px; padding: 10px; margin: 0; border: 1px dashed #CCCCC;\>Drop blocks or content here</p>",
  "blocks": {
    "nnrzsge987d6b1jqk2n4b1emi": {
      "assetType": {
        "id": 201,
        "name": "dynamicblock"
      },
      "content": "<table cellpadding=\"0\" cellspacing=\"0\" width=\"100%\" class=\"stylingblock-content-wrapper\" style=\"><tr><td class=\"stylingblock-content-wrapper camarker-inner\">{#if ([[isNTOMember] == true])}}\n{= BeginImpressionRegion(\"NTO Start Earning Rewards\")}}<table cellpadding=\"0\" cellspacing=\"0\" width=\"100%\" class=\"stylingblock-content-wrapper\"><tr><td class=\"stylingblock-content-wrapper camarker-inner\"><table width=\"100%\" cellspacing=\"0\" cellpadding=\"0\" border=\"0\" style=\"margin: 10px 0px;\> \n \n <tr> \n <td width=\"15%\" style=\"font-family: Gotham,Helvetica,Arial,sans-serif; color: #6d6d6d; font-size: 16px; line-height: 22px; padding: 0px; text-align: center;\> \n <td width=\"85%\" style=\"font-family: Gotham,Helvetica,Arial,sans-serif; color: #6d6d6d; font-size: 14px; line-height: 22px; padding: 5px;\> \n <td class=\"full-width\"> \n <b>Start earning rewards with myNTO.\n </b>\n <br> As an exclusive member of myNTO, you're on your way to earning points on every NTO purchase. Keep it up! \n <!-- ===== --> \n <!-- Blue CTA BTN --> \n <!-- ===== --> \n <table width=\"40%\" class=\"full-width\"> \n <tr> \n <td width=\"175\" style=\"padding: 10px 0px 0px;\> \n <a style=\"color: #ffffff; text-decoration: none;\> \n <span style=\"font-family: Gotham,Helvetica,Arial,sans-serif; display: block; color: #ffffff; text-align: center; font-size: 12px; padding: 7px 0px; border-radius: 20px; background-color: #6ccdf0;\>Learn More</span></a> \n </td> \n </tr> \n </table></td></tr></table>{= EndImpressionRegion()}}\n{.else}}\n{= BeginImpressionRegion(\"NTO Join for Discounts\")}}<table cellpadding=\"0\" cellspacing=\"0\" width=\"100%\" class=\"stylingblock-content-wrapper\"><tr><td class=\"stylingblock-content-wrapper camarker-inner\"><table width=\"100%\" cellspacing=\"0\" cellpadding=\"0\" border=\"0\" style=\"margin: 10px 0px;\> \n \n <tr> \n <td width=\"15%\" style=\"font-family: Gotham,Helvetica,Arial,sans-serif; color: #6d6d6d; font-size: 16px; line-height: 22px; padding: 0px; text-align: center;\> \n <td class=\"full-width\"> \n <img src=\"https://s3-us-west-2.amazonaws.com/rd-industry-emails/retail/nto/campaign-toolkits/welcome-series/welcome-step-1/welcome-step1-icon2.png\"> \n </td> \n <td width=\"85%\" style=\"font-family: Gotham,Helvetica,Arial,sans-serif; color: #6d6d6d; font-size: 14px; line-height: 22px; padding: 5px;\> \n <td class=\"full-width\"> \n <b>Join myNTO for exclusive discounts.\n </b>\n <br> Exclusive offers and rewards have never been so easy! Join myNTO and start earning points on every purchase. \n <!-- ===== --> \n <!-- Blue CTA BTN --> \n <!-- ===== --> \n <table width=\"40%\" class=\"full-width\"> \n <tr> \n <td style=\"padding: 10px 0px 0px;\> \n <a style=\"color: #ffffff; text-decoration: none;\> \n <span style=\"font-family: Gotham,Helvetica,Arial,sans-serif; display: block; color: #ffffff; text-align: center; font-size: 12px; padding: 7px 0px; border-radius: 20px; background-color: #6ccdf0;\>Learn More</span></a> \n </td> \n </tr> \n </table> \n </td> \n </tr> \n </table></td></tr></table>{= EndImpressionRegion()}}\n{.if}}</td></tr></table>",
    "superContent": "<table cellpadding=\"0\" cellspacing=\"0\" width=\"100%\" class=\"stylingblock-content-wrapper\"><tr><td class=\"stylingblock-content-wrapper camarker-inner\"><table width=\"100%\" cellspacing=\"0\" cellpadding=\"0\" border=\"0\" style=\"margin: 10px 0px;\> \n \n <tbody><tr> \n <td width=\"15%\" style=\"font-family: Gotham,Helvetica,Arial,sans-serif; color: #6d6d6d; font-size: 16px; line-height: 22px; padding: 0px; text-align: center;\> \n <td class=\"full-width\"> \n <img src=\"https://s3-us-west-2.amazonaws.com/rd-industry-emails/retail/nto/campaign-toolkits/welcome-series/welcome-step-1/welcome-step1-icon2.png\"> \n </td> \n <td width=\"85%\" style=\"font-family: Gotham,Helvetica,Arial,sans-serif; color: #6d6d6d; font-size: 14px; line-height: 22px; padding: 5px;\> \n <td class=\"full-width\"> \n <b>Join myNTO for exclusive discounts.\n </b>\n <br> Exclusive offers and rewards have never been so easy! Join myNTO and start earning points on every purchase. \n <!-- ===== --> \n <!-- Blue CTA BTN --> \n <!-- ===== --> \n <table width=\"40%\" class=\"full-width\"> \n <tbody><tr> \n <td style=\"padding: 10px 0px 0px;\> \n <a style=\"color: #ffffff; text-decoration: none;\> \n <span style=\"font-family: Gotham,Helvetica,Arial,sans-serif; display: block; color: #ffffff; text-align: center; font-size: 12px; padding: 7px 0px; border-radius: 20px; background-color: #6ccdf0;\>Learn More</span></a> \n </td> \n </tr> \n </tbody></table> \n </td> \n </tr> \n </table></td></tr></table>",
    "meta": {
      "options": {
        "dynamicContent": {
          "views": {
            "html": {
              "name": "",
              "defaultContent": {
                "id": 108497,
                "name": "NTO Join for Discounts",
                "type": {

```

```

      "id": 197,
      "name": "htmlblock",
      "displayName": "HTML Block"
    },
    "textContent": "<div data-marker=\"wrapper\" class=\"stylingblock-content-wrapper\" style=\"\"><table width=\"100%\" cellspacing=\"0\" cellpadding=\"0\" border=\"0\" style=\"margin: 10px 0px;\"> \n \n <tr> \n <td width=\"15%\" style=\"font-family: Gotham,Helvetica,Arial,sans-serif; color: #6d6d6d; font-size: 16px; line-height: 22px; padding: 0px; text-align: center;\" class=\"full-width\"> \n <img src=\"https://s3-us-west-2.amazonaws.com/rd-industry-emails/retail/nto/campaign-toolkits/welcome-series/welcome-step-1/welcome-step1-icon2.png\"> \n </td> \n <td width=\"85%\" style=\"font-family: Gotham,Helvetica,Arial,sans-serif; color: #6d6d6d; font-size: 14px; line-height: 22px; padding: 5px;\" class=\"full-width\"> \n <b>Join myNTO for exclusive discounts.\n </b>\n <br> Exclusive offers and rewards have never been so easy! Join myNTO and start earning points on every purchase. \n <!-- ===== --> \n <!-- Blue CTA BTN --> \n <!-- ===== --> \n <table width=\"40%\" class=\"full-width\"> \n \n <tr> \n <td style=\"padding: 10px 0px 0px;\"> \n <a style=\"color: #ffffff; text-decoration: none;\" target=\"_blank\" class=\"mod_click\" href=\"http://ntoretail.com\"><span style=\"font-family: Gotham,Helvetica,Arial,sans-serif; display: block; color: #ffffff; text-align: center; font-size: 12px; padding: 7px 0px; border-radius: 20px; background-color: #6ccdf0;\">Learn More</span></a> \n </td> \n </tr> \n \n </table> \n </td> \n </tr> \n \n</table></div>\",
      "thumbnail": "http://image.example.com"
    },
    "dataSource": {
      "type": "profileattribute",
      "name": "Profile Attributes"
    },
    "variants": [
      {
        "name": "Rule 1",
        "content": {
          "id": 108496,
          "name": "NTO Start Earning Rewards",
          "asset": {
            "id": 108496,
            "customerKey": "391fa72d-d3a5-47c8-8cb0-9aa18d5e805d",
            "objectID": "1e91f142-b6c7-45b0-acfc-8df0925ddb9f",
            "assetType": {
              "id": 197,
              "name": "htmlblock",
              "displayName": "HTML Block"
            }
          },
          "name": "NTO Start Earning Rewards",
          "owner": {
            "id": 10817573,
            "email": "user@example.com",
            "name": "User"
          },
          "createdDate": "2016-12-05T21:34:41-06:00",
          "createdBy": {
            "id": 10817573,
            "email": "user@example.com",
            "name": "User"
          },
          "modifiedDate": "2016-12-05T21:34:41-06:00",
          "modifiedBy": {
            "id": 10817573,
            "email": "user@example.com",
            "name": "User"
          },
          "enterpriseId": 10806781,
          "memberId": 10806781,
          "status": {
            "id": 1,
            "name": "Draft"
          },
          "thumbnailUrl": "/v1/assets/108496/thumbnail",
          "category": {
            "id": 1467802,
            "parentId": 0,
            "name": "Content Builder"
          },
          "options": {
            "generateFrom": null
          }
        },
        "content": "<div data-marker=\"wrapper\" class=\"stylingblock-content-wrapper\" style=\"\"><table width=\"100%\" cellspacing=\"0\" cellpadding=\"0\" border=\"0\" style=\"margin: 10px 0px;\"> \n \n <tr> \n \n <td width=\"15%\" style=\"font-family: Gotham,Helvetica,Arial,sans-serif; color: #6d6d6d; font-size: 16px; line-height: 22px; padding: 0px; text-align: center;\" class=\"full-width\"> \n <img src=\"http://image.s4.exct.net/lib/fe901570726d037d72/m/1/welcome-step1-icon2.png\"> \n </td> \n <td width=\"85%\" style=\"font-family: Gotham,Helvetica,Arial,sans-serif; color: #6d6d6d; font-size: 14px; line-height: 22px; padding: 5px;\" class=\"full-width\"> \n <b>Start earning rewards with myNTO.\n </b>\n <br> As an exclusive member of myNTO, you're on your way to earning points on every NTO purchase. Keep it up! \n <!-- ===== --> \n <!-- Blue CTA BTN --> \n <!-- ===== --> \n <table width=\"40%\" class=\"full-width\"> \n \n

```

```
|  |
| --- |
|> \n
  |

```

```

    },
    "modelVersion": 2
  },
  "template": {
    "id": 108494,
    "customerKey": "9422321d-1ec4-452e-a407-9dc6657025d9",
    "objectID": "83db8cbc-5cbd-49b4-b0ce-7993e03fff94",
    "assetType": {
      "id": 4,
      "name": "template",
      "displayName": "Template"
    },
    "name": "NTO Example Template",
    "owner": {
      "id": 10817573,
      "email": "user@example.com",
      "name": "User"
    },
    "createdDate": "2016-12-05T21:28:17-06:00",
    "createdBy": {
      "id": 10817573,
      "email": "user@example.com",
      "name": "User"
    },
    "modifiedDate": "2016-12-05T21:38:07-06:00",
    "modifiedBy": {
      "id": 10817573,
      "email": "user@example.com",
      "name": "User"
    },
    "enterpriseId": 10806781,
    "memberId": 10806781,
    "status": {
      "id": 1,
      "name": "Draft"
    },
    "category": {
      "id": 1467802,
      "parentId": 0,
      "name": "Content Builder"
    },
    "content": "<!DOCTYPE html> \n<html> \n <head> \n <meta name=\\"viewport\\" content=\\"width=device-width, initial-
scale=1, maximum-scale=1, user-scalable=0\\" /> \n <meta http-equiv=\\"Content-Type\\" content=\\"text/html; charset=UTF-8\\" /> \n
<style type=\\"text/css\\"> .ReadMsgBody {\n width: 100%;\n }\n .ExternalClass {\n width: 100%;\n }\n div,
p, a, li, td {\n -webkit-text-size-adjust:none;\n }\n body {\n -webkit-text-size-adjust:100%;\n -ms-text-
size-adjust:100%;\n margin:0 !important;\n padding:0 !important;\n }\n table td {\n border-collapse:
collapse;\n }\n img {\n outline:0;\n }\n a img {\n border:none;\n }\n @media screen and
(max-width: 480px) {\n td[class=\\"headercell-phone\\"] {\n display:none;\n }\n table[class=\\"container\\"]
{\n width:360px !important;\n }\n table[class=\\"hero\\"] {\n width:100% !important;\n }\n
table[class=\\"full-width\\"], td[class=\\"full-width\\"] {\n width:100% !important;\n float:left !important;\n
padding:10px 15px 10px 15px !important;\n }\n table[class=\\"full-width-hero-content\\"], td[class=\\"full-width-hero-
content\\"] {\n width:90% !important;\n float:left !important;\n padding: 0 5% 0 5% !important;\n
display: block !important;\n }\n table[class=\\"full-width-hero-image\\"], td[class=\\"full-width-hero-image\\"] {\n
float:left !important;\n display: block !important;\n }\n img[class=\\"photo\\"] {\n width:100%
!important;\n height:auto !important;\n display: block;\n }\n td[id=\\"preheader\\"]{\n
width:auto !important;\n }\n td[class=\\"row\\"]{\n width:100% !important;\n }\n td[class=\\"hide\\"]
{\n display:none !important;\n }\n td[class=\\"headercell-phone\\"] {\n display:inline !important;\n
}\n td[class=\\"headercell-desktop\\"] img {\n display: none !important;\n }\n span[class=\\"center-text\\"]
{\n text-align:center !important;\n }\n }</style> \n </head> \n <body bgcolor=\\"#e8e8e8\\"> \n <!--
===== --> \n <!-- Flexible Branding Template --> \n <!-- ===== --> \n <table
width=\\"100\\" border=\\"0\\" cellspacing=\\"0\\" cellpadding=\\"0\\" bgcolor=\\"#e8e8e8\\"> \n <tr> \n <td> \n <!--
===== --> \n <!-- White background wrapper --> \n <!-- ===== --> \n
<table class=\\"container\\" bgcolor=\\"#ffffff\\" width=\\"600\\" align=\\"center\\" border=\\"0\\" cellspacing=\\"0\\" cellpadding=\\"0\\"
style=\\"background-color:\#ffffff !important;\> \n <tr> \n <td> \n <!--
===== --> \n <!-- Logo and Preheader --> \n <!-- ===== --> -
-> \n <table class=\\"container\\" bgcolor=\\"#6ccdf0\\" style=\\"padding: 20px 0 0 0;\> \n <tr> \n <td width=\\"195\\" style=\\"padding:0
15px 15px 0px;\> \n <center>\n <a href=\\"http://ntoutfitters.com\\" alias=\\"NTO Logo\\"
style=\\"outline:none\\><img src=\\"https://s3-us-west-2.amazonaws.com/rd-industry-emails/retail/nto/brand-assets/email-
templates/branding-nto-logo-white.png\\"></a>\n </center> \n <td> \n <td
class=\\"hide\\" width=\\"225\\" style=\\"font-family:Gotham, Helvetica, Arial, sans-serif; color:\#ffffff; font-size:13px; line-height:12px;
text-align:right; padding:0 25px 0 0;\> \n <!-- CONTENT AREA --> \n <div data-type=\\"slot\\"
data-key=\\"preheader\\" data-allowed-blocks=\\"\\\" data-max-blocks=\\"\\\" data-label=\\"\\\">\n </div> \n
<!-- END CONTENT AREA --> \n </td> \n </tr> \n </table> \n <!--
===== --> \n <!-- Hero Image --> \n <!-- ===== --> \n
<table class=\\"container\\" width=\\"600\\" align=\\"center\\" border=\\"0\\" cellspacing=\\"0\\" cellpadding=\\"0\\"> \n <tr> \n
<td style=\\"font-family:Gotham, Helvetica, Arial, sans-serif; color:\#4f4f4f; font-size:16px; line-height:22px; padding:0 0 25px 0;\> \n
<!-- CONTENT AREA --> \n <div data-type=\\"slot\\" data-key=\\"heroimage\\" data-allowed-blocks=\\"\\\" data-max-
blocks=\\"\\\" data-label=\\"\\\">\n </div> \n <!-- END CONTENT AREA --> \n </td>
\n <tr> \n </table> \n <!-- ===== --> \n <!--

```

```

Section --> \n
<!-- ===== --> \n
<table class="container" width="600"
align="center" border="0" cellpadding="0" cellspacing="0"> \n
<tr> \n
<td style="font-
family:Gotham, Helvetica, Arial, sans-serif; color:#4f4f4f; font-size:16px; line-height:22px; padding:0 25px 0 25px;"> \n
<!-- CONTENT AREA --> \n
<div data-type="slot" data-key="cell01" data-allowed-blocks="" data-max-blocks=""
data-label="">\n
</div> \n
<!-- END CONTENT AREA --> \n
</td> \n
</tr> \n
</table> \n
<!-- ===== --> \n
<!-- Section --> \n
<!-- ===== --> \n
<table class="container" width="600" align="center" border="0"
cellspacing="0" cellpadding="0" cellspacing="0"> \n
<tr> \n
<td style="font-family:Gotham, Helvetica, Arial,
sans-serif; color:#4f4f4f; font-size:16px; line-height:22px; padding:0 25px 0 25px;"> \n
<!-- CONTENT AREA --> \n
<div data-type="slot" data-key="cell02" data-allowed-blocks="" data-max-blocks="" data-label="">\n
</div> \n
<!-- END CONTENT AREA --> \n
</td> \n
</tr> \n
</table> \n
<!-- ===== --> \n
<!-- Section --> \n
<!-- ===== --> \n
<table class="container" width="600" align="center" border="0" cellpadding="0" cellspacing="0"> \n
<tr> \n
<td style="font-family:Gotham, Helvetica, Arial, sans-serif; color:#4f4f4f; font-size:16px; line-height:22px; padding:0 25px 0 25px;"> \n
<!-- CONTENT AREA --> \n
<div data-type="slot" data-key="cell03" data-allowed-
blocks="" data-max-blocks="" data-label="">\n
</div> \n
<!-- END CONTENT AREA --> \n
</td> \n
</tr> \n
</table> \n
<!-- ===== --> \n
<!-- Section --> \n
<!-- ===== --> \n
<table class="container" width="600"
align="center" border="0" cellpadding="0" cellspacing="0"> \n
<tr> \n
<td style="font-
family:Gotham, Helvetica, Arial, sans-serif; color:#4f4f4f; font-size:16px; line-height:22px; padding:0 25px 0 25px;"> \n
</td> \n
</tr> \n
</table> \n
<!-- ===== --> \n
<!-- Section --> \n
<!-- ===== --> \n
<table class="container" width="600"
align="center" border="0" cellpadding="0" cellspacing="0"> \n
<tr> \n
<td style="font-
family:Gotham, Helvetica, Arial, sans-serif; color:#4f4f4f; font-size:16px; line-height:22px; padding:0 25px 0 25px;"> \n
</td> \n
</tr> \n
</table> \n
<!-- ===== --> \n
<!-- Footer --> \n
<!-- ===== --> \n
<table width="100%" border="0" cellpadding="0" cellspacing="0" bgcolor="#393939"> \n
<tr> \n
<td height="7px" bgcolor="#6ccdf0"> \n
</td> \n
</tr> \n
<tr> \n
<td style="font-
family:Gotham, Helvetica, Arial, sans-serif; text-align:center; color:#a5a5a5; font-size:9px; line-height:10px; padding:0 50px 35px
50px;"> \n
<a href="http://ntoutfitters.com/" target="_blank" style="outline:none;"></a>\n
<br> \n
<span style="display:block;font-weight:bold; font-
size:14px;">Northern Trail Outfitters</span>\n
<br> %%Member_Busname%% registered address: \n
<br> %%Member_Adr%%
%%Member_City%%, %%Member_State%%, %%Member_PostalCode%%, %%Member_Country%%.\n
<br> \n
</td> \n
</tr> \n
<tr> \n
<td style="font-family:Gotham, Helvetica, Arial, sans-serif; text-align:center; color:#a5a5a5; font-size:12px; font-
weight:bold; line-height:16px; padding:0 55px 25px 55px;"> \n
<a href="http://www.ntoutfitters.com/" alias="Contact Us"
target="_blank" style="color:#a5a5a5;text-decoration:none;">Contact Us</a> | \n
<a href="http://www.ntoutfitters.com/"
alias="Store Locator" target="_blank" style="color:#a5a5a5;text-decoration:none;">Store Locator</a> | \n
<a
href="%%subscription_center_url%%" alias="Manage Subscriptions" target="_blank" style="color:#a5a5a5;text-
decoration:none;">Manage Subscriptions</a> | \n
<a href="%%profile_center_url%" target="_blank" alias="Profile Center"
style="color:#a5a5a5;text-decoration:none;">Update Profile</a> | \n
<a class="mod_click" target="_blank"
href="%%unsubscribe_center_url%" alias="Unsubscribe" title="Unsubscribe" style="color:#a5a5a5;text-decoration:none;">Unsubscribe</a>
</td> \n
</tr> \n
<tr> \n
<td style="font-family:Gotham, Helvetica, Arial, sans-serif;
text-align:center; color:#a5a5a5; font-size:12px; line-height:14px; padding:0 100px 25px 100px; text-align:center;"> \n
<a
href="https://www.facebook.com/NTOutfitters" alias="NTO Facebook" target="_blank"></a> \n
<a href="https://twitter.com/NTOutfitters" alias="NTO Twitter" target="_blank"></a> \n
<a href="https://www.pinterest.com/ntoutfitters/" alias="NTO Pinterest" target="_blank"></a> \n
<a href="http://www.ntoutfitters.com/" alias="NTO Email" target="_blank">
</a> \n
</td> \n
</tr> \n
</table> \n
<div id="tracking_pixel"> \n
<custom name="opencounter"
type="tracking"> \n
</div> \n
<div id="campaign_pixel"> \n
</div> \n
</body> \n
</html>,"

"slots": {
"preheader": {
"content": "",
"design": "<p style='font-family: arial; color: #CCCCCC; font-size: 12px; font-weight: bold; text-align:
center; display: flex; flex-direction: column; justify-content: center; height: 150px; padding: 10px; margin: 0; border: 1px dashed
#CCCCCC;'>Drop blocks or content here</p>",
"maxBlocks": 0,
"allowedBlocks": [],
"data": {
"email": {
"options": {
"generateFrom": ""
}
}
},
"modelVersion": 2
},
"heroimage": {
"content": "",
"design": "<p style='font-family: arial; color: #CCCCCC; font-size: 12px; font-weight: bold; text-align:
center; display: flex; flex-direction: column; justify-content: center; height: 150px; padding: 10px; margin: 0; border: 1px dashed
#CCCCCC;'>Drop blocks or content here</p>",
"maxBlocks": 0,
"allowedBlocks": [],
"data": {
"email": {
"options": {

```

```

        "generateFrom": ""
    }
},
    "modelVersion": 2
},
    "cell01": {
        "content": "",
        "design": "<p style=\"font-family: arial; color: #CCCCCC; font-size: 12px; font-weight: bold; text-align: center; display: flex; flex-direction: column; justify-content: center; height: 150px; padding: 10px; margin: 0; border: 1px dashed #CCCCCC;\>Drop blocks or content here</p>",
        "maxBlocks": 0,
        "allowedBlocks": [],
        "data": {
            "email": {
                "options": {
                    "generateFrom": ""
                }
            }
        }
    },
    "modelVersion": 2
},
    "cell02": {
        "content": "",
        "design": "<p style=\"font-family: arial; color: #CCCCCC; font-size: 12px; font-weight: bold; text-align: center; display: flex; flex-direction: column; justify-content: center; height: 150px; padding: 10px; margin: 0; border: 1px dashed #CCCCCC;\>Drop blocks or content here</p>",
        "maxBlocks": 0,
        "allowedBlocks": [],
        "data": {
            "email": {
                "options": {
                    "generateFrom": ""
                }
            }
        }
    },
    "modelVersion": 2
},
    "cell03": {
        "content": "",
        "design": "<p style=\"font-family: arial; color: #CCCCCC; font-size: 12px; font-weight: bold; text-align: center; display: flex; flex-direction: column; justify-content: center; height: 150px; padding: 10px; margin: 0; border: 1px dashed #CCCCCC;\>Drop blocks or content here</p>",
        "maxBlocks": 0,
        "allowedBlocks": [],
        "data": {
            "email": {
                "options": {
                    "generateFrom": ""
                }
            }
        }
    },
    "modelVersion": 2
},
    "data": {
        "email": {
            "options": {
                "generateFrom": ""
            }
        }
    },
    "modelVersion": 2
},
    "data": {
        "email": {
            "options": {
                "generateFrom": ""
            }
        }
    },
    "modelVersion": 2
},
    "data": {
        "campaigns": {
            "campaigns": [
                {
                    "CampaignId": 14263,
                    "CampaignAssociationId": 39882
                }
            ]
        }
    }
}

```

```

    }
  ],
  "email": {
    "options": {
      "characterEncoding": "utf-8"
    },
    "attributes": [
      {
        "DisplayName": "sample_attribute",
        "Name": "__AdditionalEmailAttribute1",
        "Value": "your_tracking",
        "Order": 1,
        "Channel": "email",
        "AttributeType": "AdditionalEmailAttribute"
      },
      {
        "DisplayName": "utm_medium",
        "Name": "__AdditionalEmailAttribute2",
        "Value": "salesforce",
        "Order": 2,
        "Channel": "email",
        "AttributeType": "AdditionalEmailAttribute"
      },
      {
        "DisplayName": "utm_campaign",
        "Name": "__AdditionalEmailAttribute3",
        "Value": "newsletter",
        "Order": 3,
        "Channel": "email",
        "AttributeType": "AdditionalEmailAttribute"
      },
      {
        "DisplayName": "utm_source",
        "Name": "__AdditionalEmailAttribute4",
        "Value": "email",
        "Order": 4,
        "Channel": "email",
        "AttributeType": "AdditionalEmailAttribute"
      },
      {
        "DisplayName": "__AdditionalEmailAttributes5",
        "Name": "__AdditionalEmailAttributes5",
        "Value": "50_percent_off",
        "Order": 5,
        "Channel": "email",
        "AttributeType": "AdditionalEmailAttribute"
      }
    ],
    "legacy": {
      "legacyId": 3272283,
      "legacyKey": "090c69e7-59ac-4286-8733-f8f7613ab470",
      "legacyType": "email",
      "legacyCategoryId": 1609187
    }
  },
  "modelVersion": 2
}

```

Character encoding in `data.email.options.characterEncoding` can be one of the following:



- utf-8
- us-ascii
- shift_jis
- EUC-KR
- iso-8859-1
- iso-8859-2
- iso-8859-6
- iso-8859-11
- big5

- koi8-r
- iso-2022-jp
- GB2312

▼ List of Asset Attributes

List of Asset Attributes

This table lists possible attributes used when creating or manipulating an asset. Each asset will likely need no more than a few of these attributes.

 Key	 Type	 Required for	 Read-only	 Searchable
<u>id</u>	number		✓	✓
<u>customerKey</u>	string			✓
<u>objectID</u>	string		✓	✓
<u>name</u>	string			✓
<u>description</u>	string			✓
<u>owner</u>	object		✓	✓
<u>createdDate</u>	string		✓	✓
<u>createdBy</u>	object		✓	✓
<u>modifiedDate</u>	string		✓	✓
<u>modifiedBy</u>	object		✓	✓
<u>enterpriseID</u>	number		✓	✓
<u>memberID</u>	number		✓	✓
<u>activeDate</u>	string			✓
<u>expirationDate</u>	string			✓
<u>contentType</u>	string			✓
<u>assetType</u>	object			✓
<u>category</u>	object			✓
<u>content</u>	string			✓
<u>design</u>	string			✓
<u>superContent</u>	string			✓
<u>file</u>	object			✓
<u>views</u>	object			✓
<u>generateFrom</u>	string			✓
<u>slots</u>	object			✓
<u>blocks</u>	object			✓
<u>minBlocks</u>	number			✓
<u>maxBlocks</u>	number			✓

Aa Key	Type	Required for	Read-only	Searchable
<u>allowedBlocks</u>	array			✓
<u>template</u>	object			✓
<u>meta</u>	object			
<u>customFields</u>	object			✓
<u>data</u>	object			✓
<u>channels</u>	object			✓
<u>version</u>	number		✓	✓
<u>locked</u>	boolean			✓
<u>status</u>	string			✓
<u>tags</u>	array			✓
<u>businessUnitAvailability</u>	object			✓

▼ List of Asset Types

List of Asset Types

Asset types are the file or content types supported by Content Builder. Base asset types don't represent actual file types that can be used in the asset model. All other asset types represent files and inherit from a base asset type.

Base Asset Types

# AssetTypeID	Aa Name	# Inherits
1	<u>asset</u>	0
2	<u>file</u>	1
3	<u>block</u>	1
4	<u>template</u>	1
5	<u>message</u>	1
6	<u>custom</u>	1
7	<u>default</u>	1
8	<u>image</u>	2
9	<u>rawimage</u>	2
10	<u>video</u>	2
11	<u>document</u>	2
12	<u>audio</u>	2
13	<u>archive</u>	2
14	<u>code</u>	2
15	<u>textfile</u>	2

Asset Types

# AssetTypeID	Aa Name	# Inherits
16	<u>ai</u>	8
17	<u>psd</u>	8

# AssetTypeID	Aa Name	# Inherits
18	<u>pdd</u>	8
19	<u>eps</u>	8
20	<u>gif</u>	8
21	<u>jpe</u>	8
22	<u>jpeg</u>	8
23	<u>jpg</u>	8
24	<u>jp2</u>	8
25	<u>jpg</u>	8
26	<u>pict</u>	8
27	<u>pct</u>	8
28	<u>png</u>	8
29	<u>tif</u>	8
30	<u>tiff</u>	8
31	<u>tga</u>	8
32	<u>bmp</u>	8
33	<u>wmf</u>	8
34	<u>vsd</u>	8
35	<u>pnm</u>	8
36	<u>pgm</u>	8
37	<u>pbm</u>	8
38	<u>ppm</u>	8
39	<u>svg</u>	8
40	<u>3fr</u>	9
41	<u>ari</u>	9
42	<u>arw</u>	9
43	<u>bay.</u>	9
44	<u>cap</u>	9
45	<u>crw</u>	9
46	<u>cr2</u>	9
47	<u>dcr</u>	9
48	<u>dcs</u>	9
49	<u>dng</u>	9
50	<u>drf</u>	9
51	<u>eip</u>	9
52	<u>erf</u>	9
53	<u>fff</u>	9
54	<u>iig</u>	9
55	<u>k25</u>	9
56	<u>kdc</u>	9
57	<u>mef</u>	9
58	<u>mos</u>	9
59	<u>mrw</u>	9
60	<u>nef</u>	9
61	<u>nrw</u>	9
62	<u>orf</u>	9

# AssetTypeID	Aa Name	# Inherits
63	<u>pef</u>	9
64	<u>ptx</u>	9
65	<u>pxn</u>	9
66	<u>raf</u>	9
67	<u>raw</u>	9
68	<u>rw2</u>	9
69	<u>rwI</u>	9
70	<u>rwz</u>	9
71	<u>srf</u>	9
72	<u>sr2</u>	9
73	<u>srw</u>	9
74	<u>x3f</u>	9
75	<u>3gp</u>	10
76	<u>3gp2</u>	10
77	<u>3g2</u>	10
78	<u>3gp2</u>	10
79	<u>asf</u>	10
80	<u>avi</u>	10
81	<u>m2ts</u>	10
82	<u>mts</u>	10
83	<u>dif</u>	10
84	<u>dv</u>	10
85	<u>mkv</u>	10
86	<u>mpg</u>	10
87	<u>f4v</u>	10
88	<u>flv</u>	10
89	<u>mjpg</u>	10
90	<u>mjpeg</u>	10
91	<u>mxl</u>	10
92	<u>mpeg</u>	10
93	<u>mp4</u>	10
94	<u>m4v</u>	10
95	<u>mp4v</u>	10
96	<u>mov</u>	10
97	<u>swf</u>	10
98	<u>wmv</u>	10
99	<u>rm</u>	10
100	<u>ogv</u>	10
101	<u>indd</u>	11
102	<u>indt</u>	11
103	<u>incx</u>	11
104	<u>wwcx</u>	11
105	<u>doc</u>	11
106	<u>docx</u>	11
107	<u>dot</u>	11

# AssetTypeID	Aa Name	# Inherits
108	<u>dotx</u>	11
109	<u>mdb</u>	11
110	<u>mpp</u>	11
111	<u>ics</u>	11
112	<u>xls</u>	11
113	<u>xlsx</u>	11
114	<u>xlk</u>	11
115	<u>xlsm</u>	11
116	<u>xlt</u>	11
117	<u>xltm</u>	11
118	<u>csv</u>	11
119	<u>tsv</u>	11
120	<u>tab</u>	11
121	<u>pps</u>	11
122	<u>ppsx</u>	11
123	<u>ppt</u>	11
124	<u>pptx</u>	11
125	<u>pot</u>	11
126	<u>thmx</u>	11
127	<u>pdf</u>	11
128	<u>ps</u>	11
129	<u>qxd</u>	11
130	<u>rtf</u>	11
131	<u>sxc</u>	11
132	<u>sxi</u>	11
133	<u>sxw</u>	11
134	<u>odt</u>	11
135	<u>ods</u>	11
136	<u>ots</u>	11
137	<u>odp</u>	11
138	<u>otp</u>	11
139	<u>epub</u>	11
140	<u>dvi</u>	11
141	<u>key</u>	11
142	<u>keynote</u>	11
143	<u>pez</u>	11
144	<u>aac</u>	12
145	<u>m4a</u>	12
146	<u>au</u>	12
147	<u>aif</u>	12
148	<u>aiff</u>	12
149	<u>aifc</u>	12
150	<u>mp3</u>	12
151	<u>wav</u>	12
152	<u>wma</u>	12

# AssetTypeID	Aa Name	# Inherits
153	<u>midi</u>	12
154	<u>oga</u>	12
155	<u>ogg</u>	12
156	<u>ra</u>	12
157	<u>vox</u>	12
158	<u>voc</u>	12
159	<u>Zz</u>	13
160	<u>arj</u>	13
161	<u>bz2</u>	13
162	<u>cab</u>	13
163	<u>gz</u>	13
164	<u>gzip</u>	13
165	<u>iso</u>	13
166	<u>lha</u>	13
167	<u>sit</u>	13
168	<u>tgz</u>	13
169	<u>jar</u>	13
170	<u>rar</u>	13
171	<u>tar</u>	13
172	<u>zip</u>	13
173	<u>ppg</u>	13
174	<u>htm</u>	14
175	<u>html</u>	14
176	<u>xhtml</u>	14
177	<u>xht</u>	14
178	<u>css</u>	14
179	<u>less</u>	14
180	<u>sass</u>	14
181	<u>js</u>	14
182	<u>json</u>	14
183	<u>atom</u>	14
184	<u>rss</u>	14
185	<u>xml</u>	14
186	<u>xsl</u>	14
187	<u>xslt</u>	14
188	<u>md</u>	14
189	<u>markdown</u>	14
190	<u>as</u>	14
191	<u>fla</u>	14
192	<u>eml</u>	14
193	<u>text</u>	15
194	<u>txt</u>	15
195	<u>freeformblock</u>	3
196	<u>textblock</u>	3
197	<u>htmlblock</u>	3

# AssetTypeID	Aa Name	# Inherits
198	textplusimageblock	3
199	imageblock	3
200	abtestblock	3
201	dynamicblock	3
202	stylingblock	3
203	einsteincontentblock	3
205	webpage	1
206	webtemplate	1
207	templatebasedemail	5
208	htmlemail	5
209	textonlyemail	5
210	socialshareblock	3
211	socialfollowblock	3
212	buttonblock	3
213	layoutblock	3
214	defaulttemplate	4
215	smartcaptureblock	0
216	smartcaptureformfieldblock	0
217	smartcapturesubmitoptionsblock	0
218	slotpropertiesblock	0
219	externalcontentblock	0
220	codesnippetblock	0
221	rssfeedblock	0
222	formstylingblock	0
223	referenceblock	0
224	imagecarouselblock	0
225	customblock	0
226	liveimageblock	0
227	livesettingblock	0
228	contentmap	0
230	jsonmessage	1
232	icemailformblock	1

▼ Considerations for Compiling and Rendering the Asset Model

Considerations for Compiling and Rendering the Asset Model

Compile an asset in the context of a channel to return a flattened string of the asset content. You can use two compiling types:

- standard compiling
- rendering

Rendering accounts for superContent and design. Use rendering only for user representation and manipulation of content, as opposed to sending and publishing content.

Standard Compiling

- IF an asset contains content:

- FOR EACH placeholder block or slot in the content:
 - REPLACE the placeholder with the compiled string from the block or slot
- RETURN replaced content
- ELSE:
 - IF an asset contains views:
 - IF an asset contains a view for the specified channel:
 - RETURN the compiled channel view
 - ELSE IF an asset contains an HTML view:
 - RETURN the compiled HTML view
 - ELSE:
 - RETURN empty string

Rendering

- IF an asset contains content, design, or superContent:
 - IF an asset contains superContent and superContent is not empty:
 - usedContent = superContent
 - ELSE IF an asset contains content and content is not empty:
 - usedContent = content
 - ELSE IF an asset contains design and design is not empty:
 - usedContent = design
 - ELSE:
 - RETURN empty string
 - FOR EACH placeholder block or slot in the content:
 - REPLACE the placeholder with the rendered string from the block or slot
 - RETURN replaced content
- ELSE:
 - IF an asset contains views:
 - IF an asset contains a view for the specified channel:
 - RETURN the rendered channel view
 - ELSE IF there is an HTML view:
 - RETURN the rendered HTML view
 - ELSE:
 - RETURN empty string

Sample Full Compiler in JavaScript

```
function getReferences(content, type) {
  var typeMarker = '<div data-type="' + type + '" data-key="';
  var splitContent = content.split(typeMarker);
  var results = [];
  if (splitContent.length > 1) {
    for (var i = 1; i < splitContent.length; i++) {
      var endTagMatches = splitContent[i].match(/(\>)|(>[^\<]*<\/div>)/i);
      var match = endTagMatches[0] || ">";
      results.push(typeMarker + splitContent[i].split(match)[0] + match);
    }
  }
}
```

```

    }
    return results;
}

function compile(asset, channel) {
    asset = asset || {};
    var content = asset.superContent || asset.content || asset.design;
    if (content) {
        ["slot", "block"].forEach(function (type) {
            var references = getReferences(content, type);
            var types = type + "s";
            references.forEach(function (reference) {
                var refKey = reference.split('data-key="')[1].split('"')[0];
                if (asset[types] && asset[types][refKey]) {
                    content = content.replace(reference, compile(asset[types][refKey]));
                } else {
                    console.error(
                        "Bad Asset: referenced " + type + " does not exist: " + refKey
                    );
                }
            });
        });
        return content;
    } else if (asset.views) {
        if (asset.views[channel]) {
            return compile(asset.views[channel], channel);
        } else if (asset.views.html) {
            return compile(asset.views.html, channel);
        }
    }
    return "";
}

```

▼ Event Notification Service

Event Notification Service

Use the Event Notification Service (ENS) API to receive notifications when certain events occur in Marketing Cloud. You can be notified when customers request password resets, get order confirmations, log in using two-factor authentication, and other events.

Example: Northern Trail Outfitters, or NTO, requests that Marketing Cloud send an online order confirmation email to a customer. The customer's email address isn't valid, so the ENS notifies NTO almost immediately that the order confirmation bounced. Using that feedback, NTO's system sends the customer the order confirmation via SMS instead.

▼ Get Started with Event Notification Service

Get Started with Event Notification Service

Prerequisites

- [Set up your development environment](#) to use Marketing Cloud APIs.
- In Installed Packages, [create an API integration](#) to get a client ID and client secret. Use the following permissions settings for the API integration.
 - Event Notifications: Callbacks - Read, Create, Update, Delete
 - Event Notifications: Subscriptions - Read, Create, Update, Delete

Ensure that your client ID and secret are stored securely. Never expose this information on the client side via JavaScript or store it in a mobile application.

Set Up the Event Notification Service API

1. Supply an HTTPS callback URL, also known as a webhook, for your application. HTTP is not supported for callback URLs.
2. [Register](#) and [verify](#) your callback URL.

2. [Create a subscription](#) that requests notification events. A subscription indicates which event types to receive notifications for and which callback, or webhook, to receive them on.
3. Prepare your system for the possibility of receiving a notification event more than once because the ENS is an "at least once" delivery system.

To stop receiving notifications, you can pause a subscription any time. When you unpause the subscription, only new event notifications are delivered. Notification events that occurred during the pause are dropped.

Related Items

- [Use ENS in Marketing Cloud](#)

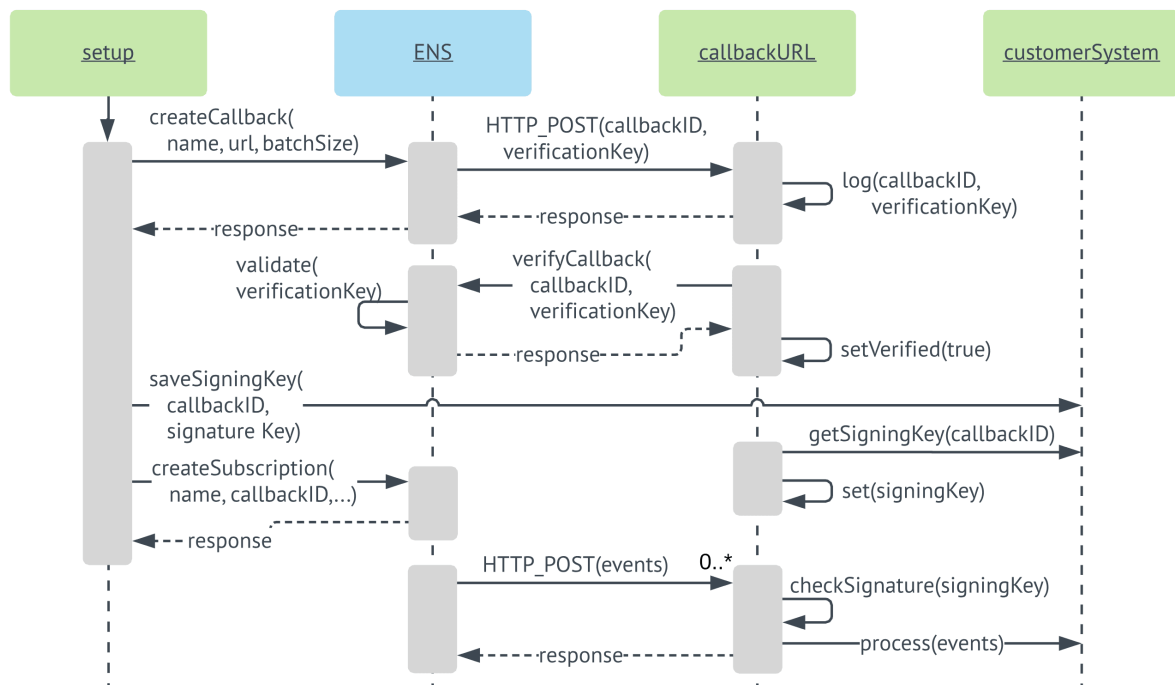
Contents [Prerequisites](#) [Set Up the Event Notification Service API](#) [Re](#)

▼ Event Notification Service Activities

Event Notification Service Activities

For the Event Notification Service to establish an active subscription that receives and processes notification events involves a sequence of activities. You are responsible for creating a callback (setup), for providing a callback URL endpoint (callbackURL), and for processing posted notifications (customerSystem).

The diagram shows the sequence of events and the required components.



1. To register your callback URL, your system calls the [Create Callback API](#).
2. The ENS syntactically validates your callback URL and issues an HTTPS POST to that URL. The [response body](#) contains a unique callback ID and verification key.
3. To [complete callback creation](#), your callback captures the callback ID and verification key and returns an HTTP 200 status.
4. Your system [verifies the callback](#). Ensure that your callback completes verification within four hours.
5. The ENS compares the verification key supplied and confirms that it is the same as the verification key provided during callback validation. Your callback prepares to receive event notification payloads for the [events you subscribe to](#).

6. Your setup uses the [Create Subscription API](#) to register interest in event notifications and provides the callback ID returned during callback validation.
7. The ENS sends batches of event notifications to your callback via an HTTPS POST up to the maximum batch size you indicated when you created the callback.
8. Your callback receives a batch of notification events and has a maximum of two seconds to [process them and return](#) an HTTP 200, 201, 202, 203, or 204, indicating successful processing.

▼ Supported Notification Events

Supported Notification Events




The Event Notification Service supports these event notification types and their corresponding payloads.

Notification Event Category	Notification Event Type	Description
TransactionalSendEvents	EmailSent	Indicates that the email was sent to the email provider.
TransactionalSendEvents	EmailNotSent	Indicates that the email was not sent and includes the reason.
TransactionalSendEvents	EmailBounced	Indicates that the email bounced and includes the reason.
EngagementEvents	EmailOpen	Indicates that the email was opened.
EngagementEvents	EmailClick	Indicates that the recipient clicked a link in the email.
EngagementEvents	EmailUnsubscribe	Indicates that the recipient clicked the unsubscribe link in the email.
TransactionalSendEvents	SmsSent	Indicates that the SMS message was sent to the aggregator.
TransactionalSendEvents	SmsNotSent	Indicates that the SMS message was not sent and includes the reason.
TransactionalSendEvents	SmsTransient	Indicates the most recent transmission status between Salesforce and the mobile device. Not the final disposition. This information is provided by our delivery partners and mobile carriers but is not available in all locales. No actions to take.
TransactionalSendEvents	SmsBounced	Indicates that the SMS message bounced and includes the reason.
TransactionalSendEvents	SmsDelivered	Indicates that the SMS message was delivered.

Common Event Attributes

All or some of these attributes are shared for all event data payloads.




Attribute	Type	Description
eventCategoryType	string	The taxonomy of the event
timestampUTC	number	UTC Epoch time
compositeId	string	Internal tracking ID (deprecated)
composite	object	Object containing broken down composite ID
composite.jobId	string	Marketing Cloud Job ID
composite.batchId	string	Marketing Cloud Batch ID
composite.listId	string	Marketing Cloud List ID
definitionKey	string	Send Definition Customer Key (Transactional Sent Events Only)

 Attribute	 Type	 Description
<u>definitionId</u>	string	Send Definition Id (Transactional Sent Events Only)
<u>mid</u>	number	Tenant Business Unit Id event was produced from
<u>eid</u>	number	Tenant Enterprise Id event was produced from
<u>info</u>	object	Object containing event-specific details

Transactional Email Sent Event

This event notification payload is an example of what your callback receives for a TransactionalSendEvents.EmailSent event.

Email Sent Info Attributes

 Attribute	 Type	 Description
<u>to</u>	string	Tenant Business Id event was produced from
<u>subscriberKey</u>	string	Tenant Enterprise Id event was produced from
<u>messageKey</u>	string	Object containing event-specific details
<u>status</u>	string	Result of the Sent Event
<u>renderedSubject</u>	string	The subject line after personalization




Transactional Email Sent Payload Example

```
{
  "eventCategoryType": "TransactionalSendEvents.EmailSent",
  "timestampUTC": 1600698608530,
  "compositeId": "466651f2-9c28-e911-a261-78e3b50b4f00.2001070.44609.4153.1251508322",
  "composite": {
    "jobId": "2001070",
    "batchId": "4153",
    "listId": "44609",
    "subscriberId": "1251508322",
    "emailId": "182583"
  },
  "definitionKey": "makana-appt",
  "mid": 1447640,
  "eid": 1447640,
  "sendClassificationType": "Operational",
  "info": {
    "to": "mduarte.10166.0749@salesforce.exacttargettest.com",
    "subscriberKey": "mduarte.10166.0749@sf.exacttargettest.com",
    "messageKey": "f2b21aa0-fc16-11ea-9bc1-fbf804a5a66e",
    "status": "Sent",
    "renderedSubject": "Don't Forget Your Appointment"
  },
  "definitionId": "466651f2-9c28-e911-a261-78e3b50b4f00"
}
```

Transactional Email Not Sent Event

This event notification payload is an example of what your callback receives for a TransactionalSendEvents.EmailNotSent event.

Email Not Sent Info Attributes

 Attribute	 Type	 Description
<u>to</u>	string	Tenant Business Id event was produced from
<u>subscriberKey</u>	string	Tenant Enterprise Id event was produced from

Aa Attribute	Type	Description
<u>messageKey</u>	string	Object containing event-specific details
<u>reason</u>	string	The descriptive reason
<u>statusCode</u>	string	The subscriber error code
<u>statusMessage</u>	string	The subscriber error message

TransactionalSendEvents.EmailNotSent Payload Example

```
[
{
  "eventCategoryType": "TransactionalSendEvents.EmailNotSent",
  "timestampUTC": 1600630238373,
  "compositeId": "3d9ee7e3-57f1-e911-a2d6-1402ec8faa09.1722202.69399.0.0",
  "definitionKey": "makana-appt",
  "mid": 1476456,
  "eid": 1476266,
  "info": {
    "to": "mduarte.10166.0749@sf.exacttargettest.com",
    "subscriberKey": "101660749",
    "messageKey": "c3896725-6847-40bd-9217-dda17ee6cba2",
    "reason": "The subscriber ExactTarget system status is unsubscribed",
    "statusCode": "1",
    "statusMessage": "Unsubscribed"
  },
  "definitionId": "3d9ee7e3-57f1-e911-a2d6-1402ec8faa09",
  "composite": {
    "jobId": "2001070",
    "batchId": "4153",
    "listId": "44609",
    "subscriberId": "1251508322",
    "emailId": "182583"
  }
}
]
```

Transactional Email Bounced Event

This event notification payload is an example of what your callback receives for a TransactionalSendEvents.EmailBounced event.

Email Bounce Info Attributes

Aa Attribute	Type	Description
<u>to</u>	string	Tenant Business Id event was produced from
<u>subscriberKey</u>	string	Tenant Enterprise Id event was produced from
<u>messageKey</u>	string	Object containing event-specific details
<u>bounceCode</u>	string	The type of bounce - "HARD" or "SOFT"
<u>bounceMessage</u>	string	The string from the recipient email send the provider
<u>smtpReason</u>	string	The error value of the SMTP reason

TransactionalSendEvents.EmailBounced Payload Example

```
[
{
  "eventCategoryType": "TransactionalSendEvents.EmailBounced",
  "timestampUTC": 1583849944000,
  "compositeId": "9bcfe8e8-5a3b-e911-a2c7-1402ec83d7c0.1888699.69238.29.207500672",
  "definitionKey": "makana-appt",
  "mid": 1476456,
  "eid": 1476266,
  "info": {
    "to": "mduarte.10166.0749@sf.exacttargettest.com",
    "subscriberKey": "5bed12d54c74fd93ee1de599",

```

```

      "messageKey": "hJlMt8-fckKT4sillivusA",
      "bounceCode": "HARD",
      "bounceMessage": "5.1.1 (bad destination mailbox address) User Unknown",
      "smtpReason": "5.1.1"
    },
    "definitionId": "9bcfe8e8-5a3b-e911-a2c7-1402ec83d7c0",
    "composite": {
      "jobId": "2001070",
      "batchId": "4153",
      "listId": "44609",
      "subscriberId": "1251508322",
      "emailId": "297967"
    }
  }
}
]

```

Engagement Event Email Open

This event notification payload is an example of what your callback receives for an EngagementEvents.EmailOpen event.

Email Open Info Attributes

Attribute	Type	Description
<u>ipAddress</u>	string	IP address detected from click source
<u>userAgent</u>	string	User agent from click source
<u>location</u>	object	Contains information about location based on IP address. If the location was unable to be detected, this object will be empty

EngagementEvents.EmailOpen Payload Example

```

{
  "eventCategoryType": "EngagementEvents.EmailOpen",
  "timestampUTC": 1600722902000,
  "compositeId": "2001071.44609.4119.135804581",
  "definitionKey": "NA",
  "definitionId": "NA",
  "channel": "email",
  "mid": 1447640,
  "eid": 1447640,
  "composite": {
    "jobId": "2001071",
    "batchId": "4119",
    "listId": "44609",
    "subscriberId": "135804581"
  },
  "info": {
    "ipAddress": "52.5.174.125",
    "userAgent": "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1",
    "location": {
    }
  }
}

```

Engagement Event Email Click

This event notification payload is an example of what your callback receives for an EngagementEvents.EmailClick event.

Email Click Info Attributes

Attribute	Type	Description
<u>ipAddress</u>	string	IP address detected from click source
<u>jobUrlId</u>	string	Marketing Cloud internal job location
<u>contentLink</u>	string	Object containing event-specific details

Attribute	Type	Description
<u>impressionRegion</u>	string	Result of the Sent Event
<u>userAgent</u>	string	User agent from click source
<u>location</u>	object	Contains information about location based on IP address. If location was unable to be detected, this object will be empty
<u>location.country</u>	string	Country for the Location object.
<u>location.region</u>	string	Region for the Location object.
<u>location.city</u>	string	City for the Location object.
<u>location.postalCode</u>	string	Postal code for the Location object.
<u>location.latitude</u>	string	Latitude coordinates for the Location object.
<u>location.longitude</u>	string	Longitude coordinates for the Location object.

EngagementEvents.EmailClick Payload Example

```
{
  "eventCategoryType": "EngagementEvents.EmailClick",
  "timestampUTC": 1591407815000,
  "compositeId": "1722201.69238.15596.177361590",
  "definitionKey": "NA",
  "definitionId": "NA",
  "channel": "email",
  "mid": 1476456,
  "eid": 1476266,
  "composite": {
    "jobId": "1722201",
    "batchId": "15596",
    "listId": "69238",
    "subscriberId": "177361590"
  },
  "info": {
    "ipAddress": "52.5.174.125",
    "jobUrlId": "74165311",
    "contentLink": "https://portal.makanahealth.com/memberID\u003d?18937918018\u0026cm_ven\u003dExactTarget\u0026cm_cat\u003dMakana+Member+Notifications\u0026cm_pla\u003dAll+Subscribers\u0026cm_ite\u003dh"
  },
  "impressionRegion": "0",
  "userAgent": "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.2; AskTB5.5; MSOffice 12)",
  "location": {
    "country": "US",
    "region": "VA",
    "city": "ASHBURN",
    "postalCode": "20146-20149",
    "latitude": "39.0438",
    "longitude": "-77.4879"
  }
}
```

Engagement Event Email Unsubscribe

This event notification payload is an example of what your callback receives for an EngagementEvents.EmailUnsubscribe event.

Email Unsubscribe Info Attributes

Attribute	Type	Description
<u>to</u>	string	Tenant Business Id event was produced from
<u>domain</u>	string	email domain of event
<u>unsubscribeDate</u>	number	The date the subscriber unsubscribed

Aa Attribute	Type	Description
<u>unsubscribeMethod</u>	string	How the unsubscribe occurred; click,reply
<u>renderedSubject</u>	string	The subject line after personalization

EngagementEvents.EmailUnsubscribe Payload Example

```
{
  "eventCategoryType": "EngagementEvents.EmailUnsubscribe",
  "timestampUTC": 1591407815000,
  "compositeId": "1722201.69238.15607.178363503",
  "definitionKey": "NA",
  "definitionId": "NA",
  "mid": 1476456,
  "eid": 1476266,
  "composite": {
    "jobId": "1722201",
    "batchId": "15607",
    "listId": "69238",
    "subscriberId": "178363503"
  },
  "info": {
    "to": "pventura.10047.0953@gmail.exacttargettest.com",
    "domain": "gmail.exacttargettest.com",
    "unsubscribeDate": 1591386215000,
    "unsubscribeMethod": "Click"
  }
}
```

Transactional SMS Sent Event Payload Example

This event notification payload is an example of what your callback receives for a TransactionalSendEvents.SmsSent event.

```
[
  {
    "eventCategoryType": "TransactionalSendEvents.SmsSent",
    "timestampUTC": <timestamp>,
    "compositeId": "<compositeIDString>",
    "definitionKey": "<definitionKeyString>",
    "mid": <midNumber>,
    "eid": <eidNumber>,
    "info": {
      "to": "<mobileNumberString>",
      "subscriberKey": "<subscriberKeyString>",
      "messageKey": "<messageKeyString>"
    }
  }
]
```

Transactional SMS Not Sent Event Payload Example

This event notification payload is an example of what your callback receives for a TransactionalSendEvents.SmsNotSent event.

```
[
  {
    "eventCategoryType": "TransactionalSendEvents.SmsNotSent",
    "timestampUTC": <timestamp>,
    "compositeId": "<compositeIDString>",
    "definitionKey": "<definitionKeyString>",
    "mid": <midNumber>,
    "eid": <eidNumber>,
    "info": {
      "to": "<mobileNumberString>",
      "subscriberKey": "<subscriberKeyString>",
      "messageKey": "<messageKeyString>",
      "reason": "Mobile Address not found",
      "statusCode": "47"
    }
  }
]
```

Transactional SMS Transient Event Payload Example

This event notification payload is an example of what your callback receives for a TransactionalSendEvents.SmsTransient event.

```
[
  {
    "eventCategoryType": "TransactionalSendEvents.SmsTransient",
    "timestampUTC": 1612564224759,
    "compositeId": "api.8345b8f9-8122-4fda-b03f-1b4da9ba7a50.21602c12-ad90-e911-a2cc-1402ec936a31.888.14408292926.0",
    "composite": {
      "tsid": "api.d7f4cbbf-03cd-47f8-b391-4602b0931071",
      "jobId": "0d64fa93-d636-e911-80f3-1402ec6b9425",
      "batchId": "7021",
      "mobileNumber": "13175554040",
      "subscriberId": "0"},
    "definitionKey": "makana-appt",
    "mid": "<midNumber>, 1476456",
    "eid": "<eidNumber>, 1476266",
    "Info": {
      "to": "14405552926",
      "subscriberKey": "14405552926",
      "messageKey": "bc196910-6801-11eb-ac0c-ab0c84c819e8",
      "ordinal": "<segmentOrdinal>, \"0\",
      "statusCode": "<SmsStandardStatus>, \"3000\",
      "statusMessage": "Enroute",
      "fromAddress": "12345",
      "countryCode": "US",
    }
  }
]
```

Transactional SMS Delivered Event Payload Example

This event notification payload is an example of what your callback receives for a TransactionalSendEvents.SmsDelivered event.

```
[
  {
    "eventCategoryType": "TransactionalSendEvents.SmsTransient",
    "timestampUTC": 1612564224759,
    "compositeId": "api.8345b8f9-8122-4fda-b03f-1b4da9ba7a50.21602c12-ad90-e911-a2cc-1402ec936a31.888.14408292926.0",
    "composite": {
      "tsid": "api.d7f4cbbf-03cd-47f8-b391-4602b0931071",
      "jobId": "0d64fa93-d636-e911-80f3-1402ec6b9425",
      "batchId": "7021",
      "mobileNumber": "13175554040",
      "subscriberId": "0"},
    "definitionKey": "makana-appt",
    "mid": "<midNumber>, 1476456",
    "eid": "<eidNumber>, 1476266",
    "Info": {
      "to": "14405552926",
      "subscriberKey": "14405552926",
      "messageKey": "bc196910-6801-11eb-ac0c-ab0c84c819e8",
      "ordinal": "<segmentOrdinal>, \"0\",
      "statusCode": "<SmsStandardStatus>, \"4000\",
      "statusMessage": "Delivered",
      "fromAddress": "12345",
      "countryCode": "US",
    }
  }
]
```

Transactional SMS Bounced Event Payload Example

This event notification payload is an example of what your callback receives for a TransactionalSendEvents.SmsBounced event.

```
[
  {
    "eventCategoryType": "TransactionalSendEvents.SmsTransient",
    "timestampUTC": 1612564224759,
    "compositeId": "api.8345b8f9-8122-4fda-b03f-1b4da9ba7a50.21602c12-ad90-e911-a2cc-1402ec936a31.888.14408292926.0",
    "composite": {
      "tsid": "api.d7f4cbbf-03cd-47f8-b391-4602b0931071",
      "jobId": "0d64fa93-d636-e911-80f3-1402ec6b9425",
      "batchId": "7021",
      "mobileNumber": "13175554040",
      "subscriberId": "0"},
    "definitionKey": "makana-appt",
    "mid": "<midNumber>, 1476456",
    "eid": "<eidNumber>, 1476266",
    "Info": {
      "to": "14405552926",
```



```

    "subscriberKey": "14405552926",
    "messageKey": "bc196910-6801-11eb-ac0c-ab0c84c819e8",
    "ordinal": "<segmentOrdinal>", "0",
    "statusCode": "<SmsStandardStatus>", "4501",
    "statusMessage": "Expired",
    "fromAddress": "12345",
    "countryCode": "US",
  },
}
]

```

▼ Relate ENS Sent Events to Engagement Events

Relate ENS Sent Events to Engagement Events

A key objective of the Event Notification Service is to normalize data from sent events by correlating the transactional API request messageKey (the ID you use to specify a unique message) to an engagement event. For this example, we will be using SQL Server, but the way you implement this in your own system will depend on your event storage design.

1. Create a table to store sent event
2. Create a table to store engagement events
3. With both send events and engagement events, extract the composite data and concatenate the jobId, batchId and subscriberId.

For example, if your composite object looks like this "composite":{ "jobId": "2001070", "batchId": "4153", "listId": "44609", "subscriberId": "1251508322"}, the resultant concatenation could look like 2001070.4153.1251508322.

4. Store the concatenated value in its own column on each table, such as "SFMC_CompositeKey".
5. Join your data on the concatenated values to normalize the send events to your engagement events. For example, if Open engagement events are stored in a table named "EmailOpen" and Sent events are stored in a table named "EmailSent", you can find the related send and engagement events by using the following pseudo-query.

```

SELECT
  EmailSent.messageKey as 'LocalIdentifier',
  EmailSent.SFMC_CompositeKey as 'SFMCIdentifier',
  EmailOpen.timestampUTC as 'OpenDate_UTC'
FROM
  EmailSent
JOIN
  EmailOpen
ON EmailSent.SFMC_CompositeKey = EmailOpen.SFMC_CompositeKey

```

▼ Subscription Filters

Subscription Filters

Your Event Notification Service subscription can filter the notifications that are sent to the associated callback. Filters are expressed as a list of one or more name=value pairs on the subscription.

To deliver the notification event to the callback, the name=value pairs must all be true.

Subscription Filtering Data Items

The following data items are available for subscription filtering for each transactional send event type.

Notification Event Category	Notification Event Type	Data Items Available for Filtering
TransactionalSendEvents	EmailSent	definitionKey
TransactionalSendEvents	EmailNotSent	definitionKey, statusCode
TransactionalSendEvents	EmailBounced	definitionKey, bounceCode, smtpReason

Example

To filter TransactionalSendEvents.EmailSent events for a specific transactional send definition, include the following subscription filter when you create a subscription.

```
definitionKey=12345
```

If you apply this filter to a subscription for TransactionalSendEvents.EmailSent events, the callback receives TransactionalSendEvents.EmailSent notification events only for the transactional send definition of 12345. All other TransactionalSendEvents.EmailSent notification events are filtered out and not delivered to the callback.

▼ Notification Signing

Notification Signing

Marketing Cloud Event Notification Service signs every notification delivered to your callback with an HMAC-SHA256 signature. The signature is produced by signing the entire notification payload using a signature key unique to the callback.

Note

Because the signature is a hash of the payload, the x-sfmc-ens-signature will be different with each request.

You receive the signature key when you create the callback.

Note

You can retrieve a callback's signature key only during callback creation. Be sure to save the signature key for future use.

The x-sfmc-ens-signature arrives base64 encoded. The signature must first be decoded to provide the actual HMAC-SHA256 hash.

Here's a Java pseudo-code example of decoding:

```
/*
documented algorithm to verify - use java
import java.util.Base64;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
*/
Mac sha256_HMAC = Mac.getInstance(ALGORITHM);
SecretKeySpec secret_key = new SecretKeySpec(Base64.getDecoder().decode(TEST_BASE64_SIGKEY), ALGORITHM);
sha256_HMAC.init(secret_key);
String actual = Base64.getEncoder().encodeToString(sha256_HMAC.doFinal(TEST_PAYLOAD.getBytes(CHARSET)));
```

To validate the authenticity of a notification, use the callback signature key to create an HMAC-SHA256 signature of the notification payload received. Then compare the HMAC-SHA256 signature to the value in `x-sfmc-ens-signature`. If the values match, you know that the source of the notification is Marketing Cloud.

▼ Retries and Callback Suspensions

Retries and Callback Suspensions

Your Event Notification Service (ENS) callback has a maximum of two seconds to process an event batch and return an HTTP 200, 201, 202, 203, or 204 response, indicating successful processing.

The ENS retries up to three times to deliver the batch of notification events on these occasions.

- Your callback doesn't respond within two seconds.
- Your callback is unavailable or unreachable.
- Your callback responds with any HTTP code other than 200, 201, 202, 203, or 204.

If the notification batch can't be delivered after three retries, the batch is dropped as undeliverable.

When two or more consecutive delivery failures occur for a callback in a 5-minute window, the ENS suspends your callback for 10 minutes. During this suspension period, the ENS doesn't attempt to deliver notification events, and they are dropped.

After the initial 10-minute suspension period, the ENS resumes delivery attempts to your callback. Any additional delivery failures put your callback into suspension for increasing periods of time, up to one hour.

▼ Journey Builder API

Journey Builder API

Marketers use Journey Builder to create event-driven, responsive campaigns to distribute across any channel (online and offline), at any time, at any frequency. Journey Builder provides an interface for designing individualized customer communication plans on top of a robust marketing engine. Built on a secure extensibility framework that allows marketing developers, Journey Builder allows service providers, and third-party developers to customize journeys for specific business purposes. The API allows you to programmatically access and interact with Journey Builder functionality. See Marketing Cloud Help to learn how to use Journey Builder to create advanced, enterprise-scale, marketing automations.

Note

The Contact and Journey Builder APIs share some of the same functionality, but the Journey Builder API is the preferred API. Use the Contact API only if your business cases require you to use data encryption, account expansion, or batching.

Extend Journey Builder's Capabilities

When the out-of-box events and activities don't meet your business needs, you can extend the native capabilities of Journey Builder by developing custom events and activities. Marketers use these events and activities when creating journeys on the Journey Builder Canvas.

▼ Get Started with the Journey Builder API

Get Started with the Journey Builder API

Prerequisites

To use the Journey Builder APIs, complete the Marketing Cloud API authentication process.

1. To get a client ID and secret, create an installed package with an API Integration component.
2. Using the above credentials, get an OAuth access token for authenticating your API calls.
3. Use the access token in the header to authenticate each subsequent call.

Use the Journey Builder API

1. [Create your sendable data extension in Marketing Cloud.](#)
2. [Create your event definition in Marketing Cloud.](#) The event definition determines entry settings for the journey, controlling who gets in and when. Use the sendable data extension created in step 1 as the entry source. If using a custom application for Journey Builder functionality, you can also define an entry event via the API using the POST /eventDefinitions resource.
3. [Build the journey.](#) Use the Journey Specification to build out the JSON of your journey.
 - Define the activities. Activities determine the pace and content of contact messaging and data updates as the journey executes. Marketers create a chain of activities by dragging and dropping activities onto the Canvas.
 - Define a goal (optional). A Goal measures the journey's effectiveness. The Marketing Cloud continually evaluates journeys. The Journey Builder dashboard displays their performance toward meeting goal targets.
4. [Publish the journey.](#) Once a journey is active, Marketing Cloud continually evaluates Contacts using the entry event criteria to determine if a journey should be invoked.
5. [Fire an entry event.](#) A journey begins when the event it is configured to detect occurs. Use the event definition key created for the event definition in step 2.

▼ Journey Builder Glossary

Journey Builder Glossary

- Activity

An Activity is the message action, decision, or data update (or combination of these); it is a component in a journey which allows a marketer's goals to be met. Activities can be provided by Marketing Cloud to communicate with other Marketing Cloud applications/data, or they can be custom activities created by developers and hosted on non-Salesforce systems. Every journey must have at least one activity before it can be activated.

- Administrator

A Marketing Cloud user whose account is provisioned with administration privileges. Administrators have access to the Event Administration page.

- App Switcher

The navigation found in the header after logging in to Marketing Cloud.

- Audit Log

A journey audit log contains a record of a journey and all of its versions. Marketers use this information to monitor ongoing campaigns, identify opportunities for improvement, and troubleshoot issues. Retrieve the log via the API to see history about creating, modifying, activating, stopping, and deleting a journey.

- Canvas View

The graphical user interface (GUI) which a marketer uses to create and configure journeys within Journey Builder.

- Channel Address Priority

For each journey (version), you can define a primary and secondary email for a contact instead of the default email. The `defaults` property on the Journey object contains an ordered list of email expressions that determine which email address to use as the default, starting with the first expression.

- Contact

A proprietary data format and its corresponding application, created by the Marketing Cloud which represents a person (also known as a Subscriber or Customer). The concept of a Contact has evolved from the Marketing Cloud's Subscriber concept. This person is typically the source of the Entry Event that invokes a journey.

- Custom Activity

A type of Activity in Journey Builder which has been developed by non-Salesforce engineers and is configured to operate within the Journey Builder Engine and be added via the Journey Builder Canvas.

- Custom Entry Event

A type of Entry Event in Journey Builder developed by non-Salesforce engineers and configured to operate within the Journey Builder Engine. Once implemented, marketers configure Entry Events using the Journey Builder Canvas.

- Data Binding

A proprietary language construct that marketers and developers can use to associate I/O for each component of the journey to provide it context and to provide subsequent components with information they require to operate successfully. Data Binding for Journey Builder uses the Mustache template language syntax. In the Journey Builder UI, data bound to a journey is called event data.

- Data Extension

A table within the application database used to contain app data. In most cases, data stored in a data extension relates to a Contact/Subscriber, but it does not belong in the subscriber profile and preference attributes.

- Entry Event

An entry event is a marketer-defined set of conditions that prompt a journey to run, marking the starting point for a journey. Marketers place Entry Events on the Journey Builder Canvas. Entry Events operate by detecting a user-initiated or administrator-defined Event from inside or outside the Marketing Cloud environment. When the Event they are configured to detect is broadcast and all filter criteria are satisfied, the journey starts.

- Event API

The Event API is a RESTful API resource used by developers to provide contacts to Entry Events for Journey Builder.

- Event Definition Source (also known as Event Data Source)

An Event Definition Source is a Data Extension that has specific fields added to it as a log of events. These Data Extensions are then associated with an Entry Event in Journey Builder to define the event source. Their secondary purpose is to provide historical analytic data.

- Event Properties

Event Properties are information, such as ContactID, associated with an Event and passed on to a journey.

- Filter (also known as Contact Filter)

A Filter is a set of configurable criteria that defines an expression to validate whether or not the associated journey(s) should occur. These criteria are then tested against the account's contact configuration for boolean evaluation.

- Fire Event Activity

The Fire Event Activity is a custom Automation Studio Activity that allows marketers to inject large batches of Contacts into Journey Builder. Fire Event is found in the Automation Studio application, which is available under the Journey Builder node in the App Switcher.

- Goal

A Goal is marketer-configurable outcome that is designed within a journey on the Journey Builder canvas.

- Interaction Service

The Interaction Service is a REST API resource used by developers to create, update, read, delete, publish, and stop journeys in Journey Builder.

- Journey

A journey (sometimes called interaction) is the foundation for a campaign that is defined by a marketer using the Journey Builder canvas. A journey is composed of an entry event, a filter, activities, and, optionally, a goal.

- Journey Specification

This is a JSON (JavaScript Object Notation) representation of a journey. The Journey Builder engineering team uses this format to manage journeys while a marketer is authoring, and we expose this format for use by developers in the Interaction Service.

- Legacy Platform

The Legacy Platform refers to the custom component framework for creating custom entry events and custom activities in Journey Builder prior to the Journey Specification and Interaction Service introduced in August 2014.




- Platform V1

Platform V1 refers to the custom component framework for creating custom events and custom activities in Journey Builder that uses the Journey Specification and Interaction Service. Platform V1 was introduced in August 2014.

▼ Journey Builder API Specification

Journey Builder API Specification

The following table includes the most commonly used REST resources that make up the Interaction Service of the Journey Builder API.

 Logical Name	 HTTP Method	 Resource
Get Event Definitions	GET	/eventDefinitions
Get Event Definition	GET	/eventDefinitions/key:{key} or /eventDefinitions/{id}
Create Event Definition	POST	/eventDefinitions
Update Event Definition	PUT	/eventDefinitions/key:{key} or /eventDefinitions/{id}
Delete Event Definition	DELETE	/eventDefinitions/key:{key} or /eventDefinitions/{id}

Logical Name	HTTP Method	Resource
Fire Entry Event	POST	/events
Get Discovery Document	GET	/rest
Get Journey by ID	GET	/interactions/{id}
Get Collection of Journeys	GET	/interactions
Create or Save Journey	POST	/interactions
Update Journey Version	PUT	/interactions
Delete Journey by ID	DELETE	/interactions/{id}
Publish Journey Version Async	POST	/interactions/publishAsync/{id}?versionNumber={versionNumber}
Get Publish Status	GET	/interactions/publishStatus/{statusId}
Stop Journey	POST	/interactions/stop/{id}?versionNumber={versionNumber}
Get Journey Audit Log	GET	/interactions/{id}/audit/{action} or /interactions/{key}/audit/{action}

▼ Event Definition Overview

Event Definition Overview

Event definitions define the name and a data schema for an event. When an event definition is created, the API defines the event definition key. The API then uses the event definition key to route events to the appropriate journeys. For example, an event definition has the key "Acme-MovieRented". Two or more journeys should be invoked when a contact rents a movie. Use the same event definition key to invoke both journeys when a contact rents a movie.

The Journey Specification requires an event (`trigger`) object to have an event definition key set in order for the journey to be published.

Example Event Object

```
{
  "key": "a-simple-journey",
  "name": "My Workflow",
  "description": "This is a description of my workflow.",
  "version": 1,
  "workflowApiVersion": 1.0,
  "triggers": [
    {
      "key": "move-rented-event-activity-key",
      "name": "Move Rented",
      "type": "Event",
      "arguments": [],
      "eventDefinitionKey": "Acme-MovieRented"
    }
  ]
}
```

▼ Event Definition Schema

Event Definition Schema

When creating an event definition, the API uses the information contained in the `schema` object to create a data extension associated with the event definition. The schema defines the name, nullability, and default values of the fields to be included in the data extension.

Example JSON Schema Definition

```
{
  "schema": {
    "sendableCustomObjectField": "contactKey",
    "sendableSubscriberField": "contactKey",
    "fields": [
      {
        "name": "firstName",
        "dataType": "text",

```

```

      "maxLength": "100",
      "isNullable": false,
      "defaultValue": ""
    },
    {
      "name": "contactKey",
      "dataType": "text",
      "maxLength": "100",
      "isNullable": false,
      "defaultValue": ""
    }
  ]
}

```

Schema Fields

The event definition schema must specify a subscriber key field. This can be a subscriber key or email address, depending on the account type. The field specified will be used as the contact key when invoking journeys.

<u>Aa</u> Field	Description
<u>SendableCustomObjectField</u>	This field maps to the subscriber key (or subscriber id/email address). This field should hold, in most cases, the contact key. Journey Builder uses this field to populate the contact key when invoking journeys. Journeys require a contact key.
<u>SendableSubscriberField</u>	Valid options are SubscriberKey, SubscriberId or EmailAddress. Most API implementations use the SubscriberKey value.

▼ Event Definition Schedule

Event Definition Schedule

An automation that executes a fire event activity will automatically be created if a schedule is defined in the Event Definition. The sample json below should be inserted into the Event Definition request body.

Sample JSON

```

{
  "schedule": {
    "startDateTime": "02/12/2015T14:30",
    "endDateTime": "11/23/2029T19:45",
    "timezone": "EasternStandardTime",
    "occurrences": 52,
    "endType": "endTime",
    "frequency": "daily",
    "recurrencePattern": "interval",
    "interval": 1,
    "sunday": false,
    "monday": false,
    "tuesday": false,
    "wednesday": false,
    "thursday": false,
    "friday": false,
    "saturday": false,
    "scheduledDay": 17,
    "scheduledDayOfWeek": "Monday",
    "scheduledWeek": "Fourth",
    "scheduledMonth": "February"
  }
}

```

<u>Aa</u> Field	<input checked="" type="checkbox"/> Required (Read Only)	Description
<u>StartDateTime</u>	<input checked="" type="checkbox"/>	The first time the scheduled automation should run.
<u>EndDateTime</u>	<input type="checkbox"/>	The last time the scheduled automation should run. Required if EndType = EndDate

Aa Field	<input checked="" type="checkbox"/> Required (Read Only)	Description
<u>Occurrences</u>	<input type="checkbox"/>	Integer indicating how many times the scheduled automation should run. Required if EndType = Occurrences.
<u>EndType</u>	<input checked="" type="checkbox"/>	EndDate or Occurrences, indicates if automation schedule should stop after a specified date or a specified number of runs.
<u>Frequency</u>	<input checked="" type="checkbox"/>	Minutely, Hourly, Daily, Weekly, Monthly, Yearly
<u>RecurrencePattern</u>	<input checked="" type="checkbox"/>	Interval - used by Minutely, Hourly, DailyEveryWeekDay - used by DailyByDay - used by Weekly, Monthly, YearlyByWeek - used by Monthly, Yearly
<u>Interval</u>	<input type="checkbox"/>	Integer, used for Minutely, Hourly, Daily, Weekly, and Monthly schedules (not used for Yearly). Required if RecurrencePattern = Interval.
<u>Sunday</u>	<input type="checkbox"/>	Boolean, may be null, only used for weekly schedules.
<u>Monday</u>	<input type="checkbox"/>	Boolean, may be null, only used for weekly schedules.
<u>Tuesday</u>	<input type="checkbox"/>	Boolean, may be null, only used for weekly schedules.
<u>Wednesday</u>	<input type="checkbox"/>	Boolean, may be null, only used for weekly schedules.
<u>Thursday</u>	<input type="checkbox"/>	Boolean, may be null, only used for weekly schedules.
<u>Friday</u>	<input type="checkbox"/>	Boolean, may be null, only used for weekly schedules.
<u>Saturday</u>	<input type="checkbox"/>	Boolean, may be null, only used for weekly schedules.
<u>ScheduledDay</u>	<input type="checkbox"/>	Day of month (1 to 31), used for Monthly and Yearly schedules.
<u>ScheduledDayOfWeek</u>	<input type="checkbox"/>	Name of day of week (Sunday), used for Monthly and Yearly schedules.
<u>ScheduledWeek</u>	<input type="checkbox"/>	First, Second, Third, Fourth, Last, used for Monthly and Yearly schedules.

Build Journey

▼ Get Started with the Journey Specification

Get Started with the Journey Specification

What is the Journey Specification?

The **Journey Specification** is a structured JSON representation of a single journey in Journey Builder. It allows Journey Builder to scale to meet the massive enterprise demands that true, event-driven, one-to-one cloud applications offer.

The Journey Spec is used via REST API requests to the Marketing Cloud Interaction Service.

What benefits does it offer developers?

The largest benefit of the Journey Spec is that it provides developers with a structural syntax to programmatically compose journeys. Some other benefits include:

- Makes the Marketing Cloud Interaction Service available for customers and developers to create journeys completely via REST API requests.
- Provides constructs for composing extremely complex journeys in Journey Builder without the need of a browser.
- Designed to be somewhat readable by humans (assuming a basic understanding of JSON formatting), with textual references for statuses and options instead of cryptic database enumerations.
- Allows developers to quickly use CRUD (Create, Read, Update, and Delete) operations in conjunction with journeys.
- Allows for the Journey Builder APIs to have a constant, portable definition format for journeys, which can easily be shared in support and social.
- Allows Journey Builder engineers to rapidly create a robust ecosystem of Journey Builder functionality.
- Guarantees that all changes to a journey are atomic, preventing partial data corruption.

Structure

The Journey Spec is represented using a JSON serialized string. Journey, event, activity, and goals are the four (4) primary object properties the specification makes available:

- Journey: (sometimes called interaction) The **core** object, without this object you are unable to create any other properties of a journey using the spec.
- Event: This is called a **trigger** in the Journey Spec and determines entry settings for the journey.
- Activity: Instead of separating Waits, Activities, Exits, and so on, the Journey Spec calls all of these activities. Outcomes are used to map one activity to the next activity in the flow of a journey, once the current activity has completed.
- Goal: This object is how Journey Builder can provide information on the success of a journey.

Using the primary objects of the Journey Spec, you can create extremely complex journeys using just a text editor and a REST client.

Important considerations about the Journey Spec hierarchy

- The trigger contains the eventDefinitionKey of the entry event that will trigger this journey. There can only be a single entry event.
- There is no start activity. The first activity is the activity which has no preceding activity. It is generally the first one in the list of activities.
- There is no end activity. Endings or exits are denoted by an outcome with a **next** value of null. Contacts will be ejected from the journey when they reach that outcome.
- Goal is unique, and is not an activity (primarily due to the fact that it may or may not cause a person to exit the journey).
- Journey keys must be unique within a MID.
- Activity keys must be unique within a journey.
- Arguments may contain expressions to be processed by the engine using Data Binding. Use the Mustache template language to write these expressions. These expressions can access data that are available in the journey context (e.g. contact data) or data that are the output of a previous activity.
- Outcomes will be an array of outcome objects on the activity. The outcome object(s) will denote the next activity/activities.

Journey Object

The journey object has the following properties:

- id [read-only] - A unique identifier for this journey that is generated and assigned by the journey API during creation.
- version [read-only] - This number denotes the iteration of this particular journey.
- key [required] - The customer key (also a unique identifier within the MID) for this journey. This is defined by the developer and can be used to address this journey in most API calls.
- name [required] - The display name used in the Journey Builder UI for this journey, this will be visible to everyone who logs into your Marketing Cloud account. This can also be used as a filter with the nameOrDescription Query param.
- description - The human readable description of this journey that informs others of purpose. This can also be used as a filter with the nameOrDescription Query param.
- workflowApiVersion [required] - This number represents the current release of the Journey Spec that the Journey Builder APIs will expect in order to understand how to parse and properly respond to your API requests. This specification defines version "1.0".
- goals - This is an array of **goals** containing a single object. (We currently only support a single goal per journey.)
- triggers - This is an array of **triggers** containing a single object. (We currently only support a single trigger per journey.)
- defaults - This object contains an ordered list of email expressions used to determine which email address to use as the default, starting with the first expression.
- activities - This is an array that includes all the activities of the journey.

Example Journey Object

// If you are just building the JSON to use in an API request, use the following syntax. This object contains the required properties to create any journey in Journey Builder.

```
{
  "key": "a-key-that-is-unique-for-MID",
  "name": "This is my first journey",
  "description": "This is a description of my journey.",
  "defaults": {
    "email": [
      "{{Event.event-key.EmailAddress}}",
      "{{Contact.Default.Email}}"
    ]
  },
  "workflowApiVersion": 1.0
}
```

// To reference the journey later, perhaps for an AJAX request, the following format for creating a journey Object might be more useful.

// In JavaScript, this is a variable named 'myInteraction' whose value is an **empty object**.

```
var myInteraction = {};
```

// This is an alternative syntax for adding attributes to an object along with their respective properties (just for your reference).

```
myInteraction.key = "a-key-that-is-unique-for-MID";
myInteraction.name = "This is our first journey";
myInteraction.description = "This is a description of my journey.";
myInteraction.workflowApiVersion = 1.0;
myInteraction.version = 1;
myInteraction.triggers = [];
myInteraction.defaults = [];
myInteraction.activities = [];
myInteraction.goal = [];
```

Event Object

An event (called **trigger** in the Journey Spec) is an object on the journey Object that determines the entry settings for the journey. The recommended **type** for events fired via the API and for custom events is **Event**, which uses the contact model and an optional expression to determine when contacts should enter the journey.

The Event object has the following properties:

- **key** - The MID-unique key used to identify this entry Event.
- **name** - The display name for this Entry Event which will be shown in the Journey Builder UI.
- **type** - The type of Entry Event this is (only value currently supported is event).
- **eventDefinitionKey** - This is the key used to filter appropriate events into filter for evaluation on entry into the journey. Do not include a period in the event definition key.
- **arguments** - This object represents the arguments this Event expects to be passed **for use at runtime**.
- **configurationArguments** - This object represents the arguments this particular event expects to be passed **for use both at publish and runtime**.
 - **criteria** - This is an XML string that represents the contact filter criteria.

// This is the format of an Event (trigger) object (outside of the journey Object).

```
{
  "key": "event-key",
  "name": "Starting point for the journey",
  "type": "Event",
  "eventDefinitionKey": "my-entry-event-key",
  "arguments": {},
  "configurationArguments": {}
}
```

// The Event object can be plugged in to the Journey Spec like this:

```
{
  "key": "a-key-that-is-unique-for-MID",
  "name": "This is our first journey",
  "description": "We are using this journey to learn how to use the Journey Spec in Journey Builder",
  "workflowApiVersion": 1.0,
```

```

    "triggers": [
      {
        "key": "event-key",
        "name": "Starting point for the journey",
        "type": "Event",
        "eventDefinitionKey": "my-entry-event-key",
        "arguments": {},
        "configurationArguments": {},
      }
    ]
  }
}

```

Activity Objects

Activity Objects represent *almost* every object type in the Journey Spec. The only objects that are not technically considered an activity are goals and events.

Supported Activity types:

- [Email](#)
- [MultiCriteriaDecision](#)
- [RandomSplit](#)
- [EngagementSplit](#)
- [DataExtensionUpdate](#)
- [Rest](#)
- [Wait](#)

The Activity object has the following properties:

- id - This is the Marketing Cloud-provided unique ID for this particular activity.
- key - This is the customer key (journey-unique) for this particular activity.
- name - This is the display name for this activity, it will be shown in the Journey Builder UI.
- type - This property defines what type of activity this is. The expected input for each activity must be passed as an argument to operate correctly.
- outcomes - This is an array of 'outcome' objects (the following represents a single, generic outcome object).
 - key - This is the customer key (journey-Unique) for this particular outcome.
 - next - This value is a string which must map to a valid, journey-unique activity key.
- arguments - This is an object which represents the arguments this particular activity expects to be passed **for use at runtime**. Each activity type has its own unique argument parameters which are expected (see below).
- configurationArguments - This is an object which represents the arguments this particular activity expects to be passed **for use both at publish and runtime**. Each activity type has its own unique argument parameters which are expected (see below).

```

// This is the format of an activity object (outside of the array and journey object).
{
  "key": "activity-key",
  "name": "The display name for the activity",
  "type": "[one of the supported activity types]",
  "outcomes": [],
  "configurationArguments": {},
  "arguments": {},
  "metadata": {}
}

```

// Outcomes and configurationArguments are addressed in the Activity Outcome Objects section.

// The activity object can be plugged in to the Journey Spec like this:

```

{
  "key": "a-key-that-is-unique-for-MID",
  "name": "This is our first journey",
  "description": "We are using this journey to learn how to use the Journey Spec in Journey Builder",

```

```

"workflowApiVersion": 1.0,
"activities": [
  {
    "key": "activity-key",
    "name": "The display name for the activity",
    "type": "[one of the supported activity types]",
    "outcomes": [],
    "configurationArguments": {},
    "arguments": {},
    "metaData": {}
  }
]
}

```

Activity Outcome

The outcome objects provide some context for decision activities in determining which outcome they should select from the array. For example, on a `RandomSplit` the arguments are used to determine the likelihood that each contact has of traversing each outcome. Here is an example of an activity outcome:

```

// Please note that the arguments attribute is [optional] on some activities and required by others.
{
  "next": "the-key-of-the-next-activity-to-start-when-this-activity-is-done",
  "arguments": {
    "someKey": "someValue"
  }
}

```

Goal Object

The goal object is used to measure effectiveness of an associated journey. Goals are only evaluated via their criteria / Filter XML.

The goal object has the following properties:

- name - The display name for this goal.
- key - The customer key for this goal.
- type - The type of goal this is (only option currently is ContactEvent).
- description - The description for this goal, will be displayed in the Journey Builder user interface.
- metaData - A set of properties which are not specific to the definition or execution of this Event, but are related to it.
 - conversionUnit - This value is used for deterministic evaluations of the goal, can be either "percentage" or "wholenumber".
 - conversionValue - Based on the conversionUnit this is the metric Journey Builder uses to determine if the goal has been satisfied.

```

// This is the format of a Goal object (outside of the journey object).
{
  "key": "goal-key",
  "name": "Our success metric",
  "description": "This goal determines the success of the journey",
  "type": "ContactEvent",
  "arguments": {},
  "configurationArguments": {
    "criteria": "<Filter XML goes here>"
  },
  "metaData": {}
}

```

// The Goal object can be plugged in to the Journey Spec like this:

```

{
  "key": "a-key-that-is-unique-for-MID",
  "name": "This is our first journey",
  "description": "We are using this journey to learn how to use the Journey Spec in Journey Builder",
  "workflowApiVersion": 1.0,
  "goals": [
    {
      "key": "goal-key",
      "name": "Our success metric",
      "description": "This goal determines the success of the journey",
      "type": "ContactEvent",

```

```

    "arguments": {},
    "configurationArguments": {},
    "metaData": {}
  }
}

```

Using the above JSON, we could make a PUT request to Journey Builder and update our original journey with this new goal.

▼ Journey Specification Example

Journey Specification Example

The following JSON example represents an entire journey:

```

{
  "id": "unique-UUID-provided-by-SFMC",
  "key": "a-key-that-is-unique-for-MID",
  "version": 1,
  "name": "My first journey",
  "description": "This is a description of my journey.",
  "workflowApiVersion": 1.0,
  "createdDate": "2015-02-18T14:56:13.423",
  "modifiedDate": "2015-03-10T13:49:05.763",
  "triggers": [
    {
      "key": "event-key",
      "name": "Starting point for the journey",
      "type": "Event",
      "eventDefinitionKey": "my-entry-event-key",
      "arguments": {},
      "configurationArguments": {},
      "metaData": {}
    }
  ],
  "defaults": {
    "email": [
      "{{Event.event-key.EmailAddress}}",
      "{{Contact.Default.Email}}"
    ]
  },
  "activities": [
    {
      "key": "call-web-service",
      "name": "Call web service to get email",
      "type": "Rest",
      "metaData": {
        "flowDisplayName": "CallWebService"
      },
      "outcomes": [
        {
          "key": "call-web-service-then-send-welcome-email",
          "next": "send-welcome-email"
        }
      ],
      "configurationArguments": {
        "save": {
          "url": "https://www.example.com/endpoint",
          "useJwt": false,
          "body": ""
        },
        "validate": {
          "url": "https://www.example.com/endpoint",
          "useJwt": false,
          "body": ""
        },
        "publish": {
          "url": "https://www.example.com/endpoint",
          "useJwt": false,
          "headers": "https://www.example.com/endpoint",
          "body": ""
        }
      },
      "arguments": {
        "execute": {
          "url": "https://www.example.com/endpoint",
          "inArguments": [{
            "myArgument": ""
          }]
        }
      }
    }
  ]
}

```

```

        "body": "",
        "useJwt": false
    }
},
{
    "key": "send-welcome-email",
    "name": "Welcome_email",
    "type": "EMAILV2",
    "outcomes": [
        {
            "key": "sent-welcome-email-then-random-split",
            "next": "random-split"
        }
    ],
    "metaData": {},
    "configurationArguments": {
        "triggeredSend": {
            "emailId": "12345678",
        }
    }
},
{
    "key": "random-split",
    "name": "Random_split",
    "type": "RandomSplit",
    "outcomes": [
        {
            "key": "random-split-then-send-sms",
            "next": "send-sms",
            "arguments": {
                "percentage": 90
            }
        },
        {
            "key": "random-split-then-10-percent-end",
            "next": "send-sms2",
            "arguments": {
                "percentage": 10
            }
        }
    ]
},
{
    "key": "send-sms",
    "name": "Send_SMS",
    "type": "SMS",
    "arguments": {
        "smsToSend": "ef47e4c0-5def-11e3-949a-0000200c9a66",
        "phoneNumber": ""
    }
},
{
    "key": "send-sms2",
    "name": "Send_SMS_2",
    "type": "SMS",
    "arguments": {
        "smsToSend": "ab47e4c0-5def-11e3-949a-0000200c9a66",
        "phoneNumber": ""
    }
}
],
"goals": [
    {
        "key": "goal-key",
        "name": "Our success metric",
        "description": "This goal determines the success of the journey",
        "type": "Event",
        "arguments": {
            "criteria": "filterXML_or_JSON_String"
        },
        "metaData": {
            "isExitCriteria": true,
            "conversionUnit": "percentage",
            "conversionValue": "50",
            "eventDefinitionId": "unique-UUID-generated-by-SFMC",
            "eventDefinitionKey": "Event-unique-key-generated-by-SFMC",
            "configurationDescription": "PurchaseDate is on or after Today minus 1 day and SubTotal is not null ",
            "chainType": "none"
        }
    }
]
}

```



▼ Validation Checks to Publish a Journey

Validation Checks to Publish a Journey

A journey cannot become active and receive incoming contacts via the POST /events resource or any other source until all validation tests are completed and the journey is in a running, published state.

Journey Operating States

- **Draft:** Author-time state of a journey, when a marketer is building the journey. Journeys in this state cannot receive incoming events. This is the only state in which a journey's structure and configuration may be modified.
- **Running/Published:** Once a marketer is pleased with the journey draft, he or she can activate the journey. This confirms that all components of the journey are in a valid state to publish. Once all components have successfully validated, the journey can receive incoming events.
- **Running/Unpublished:** When a new version of a journey is published, the previously published version becomes unpublished. This means that the version is still running (contacts are still flowing through the journey), but no new contacts are being accepted into this version.
- **Stopped:** A marketer can stop a journey to prevent it from accepting any further incoming events. When stopped, all contacts currently in the journey are ejected, as if they immediately reached an end of the journey, regardless of where they are currently in the journey.

Journey Activation Checks

In order to move into the **Running/Published** state, the journey must meet each of the following requirements:

- The entry event must be completely configured and saved.
- The journey must contain at least one activity.
- Each activity must be completely configured and saved.
- The journey activity graph must be acyclic, with the exception of the wait-until activity.
- The journey must be named.
- If the journey contains a goal, it must be completely configured and saved.

If all the above items are satisfied, the journey goes through a validation pass. During this phase, the entry event, activities, and the goal (if defined) are validated. Additionally, any Marketing Cloud channel application activities are internally validated. For example, if a send email message activity is part of a journey and configured, the email associated with the activity must pass Automation Studio's validations, such as containing a Profile Center link.

Causes Failure during the Activation Phase

The send email activity is used in the journey and:

- The email does not contain a Profile Center link
- There are errors in the email itself
- If the email contains personalization string which isn't in the profile attributes A custom activity is used in the journey and:
 - The save method of the activity does not return a 200 HTTP response
 - The validate method of the activity does not return a 200 HTTP response
 - The publish method of the activity does not return a 200 HTTP response

▼ How Data Binding Works

How Data Binding Works

Data Binding allows you to bind an Activity created via the Journey Builder API or a Custom Activity to data in a running journey. This process greatly simplifies how a marketer or developer dynamically connects data for each contact flowing through a journey.

The Marketing Cloud built the syntax for Data Binding on the Mustache template language library.

Data Binding lets you write expressions evaluated by the Journey Builder Engine at runtime, when an Activity executes for a contact. The inArguments and outArguments include these expressions. The system does not evaluate these expressions when saving, validating, or publishing a journey.

The standard POST body to an Activity without Data Binding leaves the inArguments and outArguments empty:

```
{
  "contents": {
    "inArguments": [],
    "outArguments": [],
    "journeyId": "1234abcd-56ef-78gh-90ij-9876klmn5432",
    "activityId": "1234abcd-56ef-78gh-90ij-9876klmn5432",
    "definitionInstanceId": "1234abcd-56ef-78gh-90ij-9876klmn5432",
    "keyValue": "somekeyvalue@domain.tld"
  }
}
```

While this call contains some useful information, it provides a limited approach. InArguments provide your Activity with information needed to handle internal decisions, update your data, or reach out to services your Custom Activity depends on. OutArguments contain the key and value pair for each field expected on the response body of the request. The following example uses Data Binding within the config.json:

```
{
  "workflowApiVersion": "1.0",
  "type": "MOBILE",
  "lang": {
    "en-US": {
      "name": "My Custom Activity",
      "description": "This activity will make the world a better place"
    }
  },
  "arguments": {
    "execute": {
      "inArguments": [
        {
          "contactIdentifier": "{{Contact.Key}}"
        },
        {
          "emailAddress": "{{InteractionDefaults.Email}}"
        },
        {
          "twitterHandle": "{{Contact.Default.Twitter}}"
        },
        {
          "lastLogin": "{{Contact.Attribute.Engagement.LastLogin}}"
        }
      ],
      "outArguments": [
        {
          "foundSignupDate": ""
        }
      ],
      "url": "https://some-endpoint.com/execute"
    }
  },
  "configurationArguments": {
    "save": {},
    "publish": {},
    "validate": {},
    "edit": {
      "uri": "https://my.endpoint.tld/path/to/activity/index.html",
      "height": 350,
      "width": 550
    }
  }
}
```

Data Binding Syntax Basics

Using Mustache syntax provides a similar experience to using personalization strings in Marketing Cloud. However, Marketing Cloud limits Data Binding to Contacts and Event Data for a journey for the supplied Contact.

To illustrate how Mustache syntax works, assume you use the following JSON data:

```
{
  "Contact": [
    {
      "ID": 1234567890,
      "Key": "1234abcd-56ef-78gh-90ij-9876klmn5432",
      "FirstName": "John", //Profile Attribute
      "LastName": "Smith", //Profile Attribute
      "Nickname": "Johnny", //Profile Attribute
      "PhysicalAddresses": [ //Data Extension registered w/ Contact Model
        { // A row in the Data Extension
          "type": "work",
          "Address1": "9876 Money Maker Lane",
          "Address2": "Suite 100",
          "City": "Palo Alto",
          "State": "CA",
          "PostalCode": "94301"
        }
      ]
    }
  ]
}
```

To bind the full name of the Contact:

```
{{Contact.Attribute.Persona.FirstName}} {{Contact.Attribute.Persona.LastName}}
// Would output "John Smith"
```

Supported Methods of Data Binding

Contact Builder Context

Using Data Binding with Contact Builder provides the most powerful set of data to Marketing Cloud customers.

- `Contact.Key` - The Journey Builder Engine requires this contact key value to provide the appropriate context.
- `InteractionDefaults.Email` - Use this pattern to retrieve the default address of the given type for the specified Contact.
- `Contact.Default.[Twitter|Facebook|SMS]` - Use this pattern to retrieve the default address (non-email) of the given type for the specified Contact.
- `Contact.Attribute.[FullyQualifiedAttributeName]` - Use this pattern to retrieve the specified attribute from the Contact Builder. The fully qualified attribute names generally appear in the form of `[AttributeSet Name].[AttributeName]`. By default, the attribute set name from Contact Builder's Data Designer is the same as the data extension name. The attribute set name of a system-defined set may be different from the data extension name, so use the attribute set name from Contact Builder. (The attribute set name is different from the attribute group name.) Attribute sets and names can use special characters like spaces, but you **must** surround these values with double quotation marks. Example: `Contact.Attribute."Product Orders"."Product Name"`

Here are some examples based on the following data assumptions:

- The contact opted-in for subscriptions
- The contact subscribed with Email, Twitter and SMS channels as defaults using the following values:
 - `john.smith@example.com`
 - `@johnnyballgame`
 - `555-555-5555`
- Contact includes related data on an AttributeSet named **Person** (using the JSON data from above)

```
// To fetch the Default Email Address for the Contact
{{InteractionDefaults.Email}}
```

```
// To fetch the Default Twitter Address for the Contact
{{Contact.Default.Twitter}}
```

```
// To fetch the Default SMS Address for the Contact
```

```
{{Contact.Default.SMS}}
```

```
// To fetch the Home Postal Code for the Contact  
{{Contact.Attribute.Person.Work.PostalCode}}
```

```
// To fetch full name for the Contact  
{{Contact.Attribute.Person.FirstName}} {{Contact.Attribute.Person.LastName}}
```

Journey Context

You may need to retrieve information from the running journey's context. An expression can also reference the outArguments from a prior activity in the current journey context using the following placeholder patterns:

- `Interaction.[ActivityCustomerKey].[OutArgumentName]` - Uses a journey-unique Activity Customer Key and the Activity specifies the OutArgumentName value.
- `Context.IsTest` - Returns **true** for a journey in Test Mode. Use this parameter when the Activity needs to act differently when running the journey as a test.
- `Context.PublicationId` - Use the publication ID to associate the currently executing activity to a particular publication (version) of the journey.
- `Context.DefinitionId` - Use the definition ID to associate the currently executing activity to a particular definition of the journey.
- `Context.DefinitionInstanceId` - This key identifies the actual instance of a journey. Each time a contact runs through the engine, the engine creates a unique ID. Use this information to match logs.
- `Context.StartActivityKey` - This key contains the starting key for the session associated with the activity. The key includes a blank value when starting a new activity. Coming off of a wait, this key contains the key of the wait activity.
- `Context.VersionNumber` - This key retrieves the entire journey response body, from which you can pull the journey version number. See the Journey Spec for an example of the response body format.

Event Context

You can reference an event within an expression (the incoming data associated with the entry event). Adhere to the following pattern when using data binding with events:

```
// Capture a field "ProductId" from the EventDefinitionKey "my-custom-product-entry-event-key"  
{{Event.my-custom-product-entry-event-key.ProductId}}
```

Evaluating Expressions

An activity can include many different data bound fields. The fields include Mustache expressions specified at journey design time. For instance, for an activity with a databound field expecting an email, the customer can specify `{{InteractionDefaults.Email}}`.

- When processing a contact in a journey, the engine evaluates all expressions.
- Once a journey publishes, and every subsequent hour, the system compiles each expression to ensure evaluation at runtime.
- When the system cannot compile an expression, publish fails and the system disables the journey. These types of failures could happen if a customer renames or deletes a field or does anything to invalidate the expression.
- The system compiles all expressions in memory during this hourly validation. This process allows the engine to cache all information needed to retrieve for a given journey, and it speeds up the replacement process of the Mustache placeholders. For example, you can combine all contact calls that can happen within this session of the journey into one contact service call.
- The `defaults` property on the journey Object contains an ordered list of email expressions that determine which email address to use as the default, starting with the first expression.

Best Practices

Mustache and Handlebars (Handlebars extend Mustache functionality) allows for complex ways of using placeholders. Use the fully qualified name approach for specifying placeholders to cut down on the change of ambiguity issues. The system checks inner-scope before the outer-scope. Review the following code for examples.

If there is non-alpha character, such as a space or a colon, in the attribute set or field name, use double quotation marks with Mustache and Handlebar syntax.

```
{{#with Contact.FilterId}}
  {{#with Contact.AttributeId}}
    {{1234abcd-56ef-78gh-90ij-9876klmn5432}}-{{1234abcd-56ef-78gh-90ij-9876klmn5432}}
  {{/with}}
{{/with}}
```

▼ Tag your Journeys

Tag your Journeys

You can add tags to journeys to help identify and categorize them. A journey can have multiple tags, and a tag can be associated with multiple journeys. To add tags to journeys via the API, you need the object ID, or original definition ID, for each journey, and the unique tag names.

1. Relate tags to journeys.

The Create Tags resource creates an association for each tag/object pair. For example, if the payload includes 10 tags and 20 journeys, the API call creates 200 associations.

2. Filter journeys by tag.

When retrieving journeys, specify a single tag to filter results to only include journeys associated with that tag.

3. Delete tag associations.

Delete one or more tags associated with a journey at the same time.

Build Journey

▼ Contact Event

Contact Event

The event is used as the default entry event to provide a boolean check against a contact's attributes, that determines if a contact should be injected into a journey. This is the same check used in the decision split activity.

Journey Specification Format

The activity must be formatted as specified below. The following JSON object will be included in the `triggers` array of a journey, as defined in the Journey Specification.

Helpful Hints

The `schemaVersionId` comes from the GET /schema resource.

Sample Request

```
{
  "arguments": {},
  "configurationArguments": {
    "criteria": "<filterXML>",
    "schemaVersionId": 125
  },
  "description": "The description for the entry event",
  "key": "<user-defined-key-for-this-entry-event-OR-app-extension-key>",
  "metadata": {
    "configurationDescription": "",
    "eventDefinitionId": "<GUID>",
    "eventDefinitionKey": "<event-key-from-entry-event-admin>"
  },
  "name": "The name of this entry event",
}
```

```

    "outcomes": [],
    "type": "Event"
  }

```

Sample Response

```

{
  "arguments": {},
  "configurationArguments": {
    "criteria": "<filterXML>",
    "schemaVersionId": 125
  },
  "description": "The description for the entry event",
  "id": "<unique-GUID-provided-by-SFMC>",
  "key": "<user-defined-key-for-this-entry-event>",
  "metaData": {
    "configurationDescription": "",
    "eventDefinitionId": "<GUID>",
    "eventDefinitionKey": "<event-key-from-entry-event-admin>"
  },
  "eventDefinitionId": "<GUID>",
  "eventDefinitionKey": "<event-key-from-entry-event-admin>",
  "name": "The name of this entry event",
  "outcomes": [],
  "type": "Event"
}

```

▼ Goals Event Formats

Goals

The goal object is used to measure the effectiveness of an associated journey. Goals are only evaluated via their criteria or filter XML.

Journey Specification Format

The activity must be formatted as specified below. The following JSON object will be included in the `goals` array of a journey, as defined in the Journey Specification.

Helpful Hints

- When a journey is saved, each goal is saved.
- Goals are evaluated every 24 hours and upon exiting each wait activity.
- The `metaData.conversionUnit` property is a string that can be either "percentage" or "wholenumber".
- The `metaData.conversionValue` property is a number as a string.
- The `configurationArguments.criteria` property is an XML string that defines a filter that is used as a data-gate to determine if a contact qualifies for this goal.

Sample Request

```

{
  "goals": [
    {
      "key": "user-defined-key",
      "name": "name of your goal",
      "description": "The description of this goal",
      "type": "Event",
      "outcomes": [],
      "arguments": {},
      "configurationArguments": {
        "criteria": "<FilterDefinition><ConditionSet Operator=\"AND\" ConditionSetName=\"Individual Filter Grouping\"><Condition Key=\"bulkevent_de.Locale\" Operator=\"Equal\"><Value><![CDATA[asdf]]></Value></Condition></ConditionSet></FilterDefinition>",
        "schemaVersionId": 122
      },
      "metaData": {
        "conversionValue": "1",
        "conversionUnit": "wholenumber",
        "conversionTypeId": 1,

```

```

        "isExitCriteria": true,
        "eventDefinitionKey": "My-Event-Key",
        "eventDefinitionId": "<guid>"
    }
}
]
}

```

Sample Response

```

{
  "goals": [
    {
      "id": "SFM-provided-GUID",
      "key": "user-defined-key",
      "name": "name of your goal",
      "description": "The description of this goal",
      "type": "ContactEvent",
      "outcomes": [],
      "arguments": {},
      "configurationArguments": {
        "criteria": "<FilterDefinition><ConditionSet Operator=\"AND\" ConditionSetName=\"Individual Filter Grouping\"><Condition  
Key=\"bulkevent_de.Locale\" Operator=\"Equal\"><Value><![CDATA[asdf]]></Value></Condition></ConditionSet></FilterDefinition>",
        "schemaVersionId": 122
      },
      "metaData": {
        "conversionValue": "1",
        "conversionUnit": "wholenumber",
        "conversionTypeId": 1,
        "isExitCriteria": true,
        "eventDefinitionKey": "My-Event-Key",
        "eventDefinitionId": "<guid>"
      }
    }
  ]
}

```

▼ Multi Criteria Decision (DecisionSplit) Format

Multi Criteria Decision (DecisionSplit) Format

The multi-criteria decision activity, also called the decision split activity, decides which branch a contact follows in a workflow based on contact attribute values at runtime. This activity type cannot be used in a custom activity.

Journey Specification Format

The activity must be formatted as specified below. The following JSON object is included in the `activities` array of a journey, as defined in the Journey Specification.

Helpful Hints

- The outcome of this decision affects goal statistics.
- The `schemaVersionId` comes from the GET `/schema` resource.
- Use entry data or activity output data as configurationArguments in an outcome branch.
- Use the `schema` object on a custom activity to make activity output data visible to downstream activities.

Sample Request

```

{
  "type": "MultiCriteriaDecision",
  "key": "MY-ACTIVITY-KEY-1",
  "name": "My Split",
  "configurationArguments": {
    "criteria": {
      "OUTCOME-1": "<Filter XML HERE>...",
      "OUTCOME-2": "<Filter XML HERE>...",
      "OUTCOME-3": "<Filter XML HERE>..."
    },
    "schemaVersionId": 47
  }
}

```

```

      "outcomes": [
        {
          "key": "OUTCOME-2",
          "metaData": {
            "label": "Branch A"
          },
          "next": "<next activity key>"
        },
        {
          "key": "OUTCOME-1",
          "metaData": {
            "label": "Branch B"
          },
          "next": "<next activity key>"
        },
        {
          "key": "OUTCOME-3",
          "metaData": {
            "label": "Branch C"
          },
          "next": "<next activity key>"
        }
      ]
    }
  ]
}

```

Sample Response

```

{
  "id": "<unique-GUID-provided-by-SFMC>",
  "type": "MultiCriteriaDecision",
  "key": "MY-ACTIVITY-KEY-1",
  "name": "My Split",
  "configurationArguments": {
    "criteria": {
      "OUTCOME-1": "<Filter XML HERE>...",
      "OUTCOME-2": "<Filter XML HERE>...",
      "OUTCOME-3": "<Filter XML HERE>..."
    },
    "schemaVersionId": 47
  },
  "outcomes": [
    {
      "key": "OUTCOME-2",
      "metaData": {
        "label": "Branch A"
      },
      "next": "<next activity key>"
    },
    {
      "key": "OUTCOME-1",
      "metaData": {
        "label": "Branch B"
      },
      "next": "<next activity key>"
    },
    {
      "key": "OUTCOME-3",
      "metaData": {
        "label": "Branch C"
      },
      "next": "<next activity key>"
    }
  ]
}

```

▼ Data Extension Update Activity Format

Data Extension Update Activity Format

The data extension update activity is used to modify the contact model attribute and data extension values at runtime. This activity type cannot be used in a custom activity.

Journey Specification Format

The activity must be formatted as specified below. The following JSON object is included in the `activities` array of a journey, as defined in the Journey Specification.

Sample Request

```
{
  "key": "data-extension-update-1",
  "name": "My Data Extension Update 1",
  "type": "DataExtensionUpdate",
  "outcomes": [
    {
      "key": "dataUpdate-outcome-1",
      "next": "<key of next activity>"
    }
  ],
  "arguments": {
    "value": "some value"
  },
  "configurationArguments": {
    "dataExtensionId": "<data extension id>",
    "field": "<field id>"
  }
}
```

Sample Response

```
{
  "id": "<Marketing Cloud provided GUID>",
  "key": "data-extension-update-1",
  "name": "My Data Extension Update 1",
  "type": "DataExtensionUpdate",
  "outcomes": [
    {
      "key": "dataUpdate-outcome-1",
      "next": "<key of next activity>"
    }
  ],
  "arguments": {
    "value": "some value"
  },
  "configurationArguments": {
    "dataExtensionId": "<data extension id>",
    "field": "<field id>"
  }
}
```

▼ Send Email Format

Email Activity Format

The version 2 email activity, also known as send email activity, allows you to send email messages from within your journeys. This activity type cannot be used in a custom activity.

Warning

Existing version 1 email activities remain in the legacy format and function as normal until they are edited. When a marketer edits and saves a send email activity, Journey Builder automatically upgrades the activity by mapping the data to the version 2 format.

Journey Specification Format

The activity must be formatted as specified below. The following JSON object is included in the `activities` array of a journey, as defined in the Journey Specification.

Helpful Hints

- The email activity creates a triggered send which it uses during runtime to deliver the email message.
 - When a journey is unpublished, the triggered send is set to inactive.
 - If the activity is republished, the triggered send is reactivated to keep reporting numbers for accurate versions.

- You can obtain the sender profile ID and delivery profile ID by going to Email Studio > Admin > Delivery Settings.
- The email you configure for use with this activity must meet the following Marketing Cloud delivery standards:
 - Contain a Profile Center link
 - Pass validations
 - Contain only valid AmpScript and SSJS
 - Must not contain any errors
- If an exception occurs while publishing the journey containing an invalid email, the errors will display in the Journey Builder UI.

Sample Request

```
{
  "type": "EMAILV2",
  "key": "<activity key>",
  "name": "<activity name>",
  "applicationId": "<Marketing Cloud provided GUID>",
  "outcomes": [
    {
      "key": "<outcome key>",
      "next": "<key of next activity>"
    }
  ],
  "metaData": {
    "icon": "/img/email-icon.svg",
    "iconSmall": "/img/email-icon.svg",
    "category": "message",
    "version": "1.0",
    "isConfigured": true
  },
  "configurationArguments": {
    "triggeredSend": {
      "emailId": "<email id>",
      "emailSubject": "<subject>",
      "preHeader": "<preheader>",
      "description": "<description>",
      "campaigns": [
        {
          "id": "<campaign id>",
          "name": "<campaign name>",
          "color": "<campaign color>"
        }
      ],
      "sendClassificationId": "<send classification id>",
      "senderProfileId": "<sender profile id>",
      "deliveryProfileId": "<delivery profile id>",
      "publicationListId": "<publication list id>",
      "suppressionLists": [
        {
          "name": "<suppression list name>",
          "id": "<suppression list id>"
        }
      ],
      "domainExclusions": [
        {
          "name": "<domain exclusion name>",
          "id": "<domain exclusion id>"
        }
      ],
      "exclusionFilter": "<exclusion script>",
      "isTrackingClicks": true,
      "isMultipart": true,
      "isSendLogging": true,
      "suppressTracking": true,
      "ccEmail": "<cc email>",
      "bccEmail": "<bcc email>",
      "keyword": "<keyword>",
      "throttleLimit": 500,
      "throttleOpens": "12:00",
      "throttleCloses": "12:30",
      "isSalesforceTracking": true
    }
  }
}
```



```
}
}
```

Sample Response

```
{
  "id": "<Marketing Cloud provided GUID>",
  "key": "<activity key>",
  "name": "<activity name>",
  "type": "EMAILV2",
  "outcomes": [
    {
      "next": "<key of next activity>",
      "arguments": {},
      "metaData": {}
    }
  ],
  "arguments": {
    },
    "configurationArguments": {
      "triggeredSend": {
        "emailId": "<email id>",
        "emailSubject": "<subject>",
        "preHeader": "<preheader>",
        "description": "<description>",
        "campaigns": [
          {
            "id": "<campaign id>",
            "name": "<campaign name>",
            "color": "<campaign color>"
          }
        ],
        "sendClassificationId": "<send classification id>",
        "senderProfileId": "<sender profile id>",
        "deliveryProfileId": "<delivery profile id>",
        "publicationListId": "<publication list id>",
        "suppressionLists": [
          {
            "name": "<suppression list name>",
            "id": "<suppression list id>"
          }
        ],
        "domainExclusions": [
          {
            "name": "<domain exclusion name>",
            "id": "<domain exclusion id>"
          }
        ],
        "exclusionFilter": "<exclusion script>",
        "isTrackingClicks": true,
        "isMultipart": true,
        "isSendLogging": true,
        "suppressTracking": true,
        "ccEmail": "<cc email>",
        "bccEmail": "<bcc email>",
        "keyword": "<keyword>",
        "throttleLimit": 500,
        "throttleOpens": "12:00",
        "throttleCloses": "12:30",
        "isSalesforceTracking": true
      }
    },
    "metaData": {
      "icon": "/img/email-icon.svg",
      "iconSmall": "/img/email-icon.svg",
      "category": "message",
      "version": "1.0",
      "isConfigured": true
    }
  }
}
```

▼ Engagement Split Format

Engagement Split Format

The engagement split activity is used to change the path for a contact based on how they did or did not interact with an email. It allows you to define logical flows for your journey based on clicks and opens. This activity type cannot be used in a custom activity.

Helpful Hints

- This activity works only with the send email types for splitting the journey path.
- The outcome of this decision affects goal statistics.

Journey Specification Format

The activity must be formatted as specified below. The following JSON object is included in the `activities` array of a journey, as defined in the Journey Specification.

Sample Request

```
{
  "key": "ENGAGEMENTDECISION-1", // The UI adds an incremented number each time a new activity of this type is brought onto the stage.
  "name": "Engagement Decision", // "Last Modified Today"
  "description": "Description for this Activity",
  "type": "EngagementDecision",
  "outcomes": [
    {
      "next": "WAIT-[#]",
      "arguments": {
        "when": true
      }
    },
    {
      "next": "WAIT-[#]",
      "arguments": {
        "when": false
      }
    }
  ],
  "configurationArguments": {
    "refActivityCustomerKey": "EMAIL-1",
    "statsTypeId": 2, // 2 = Email opened, 3 = Email clicked, 6 = Email bounced
    "statusUrlId": "0"
  },
  "metaData": {
    "refActivityName": "<nameOfEmailForReference",
  }
}
```

Sample Response

```
{
  "id": "<guid>",
  "key": "ENGAGEMENTDECISION-1", // The UI adds an incremented number each time a new activity of this type is brought onto the stage.
  "name": "Engagement Decision", // "Last Modified Today"
  "description": "Description for this Activity",
  "type": "EngagementDecision",
  "outcomes": [
    {
      "key": "<guid>",
      "next": "WAIT-[#]",
      "arguments": {
        "when": true
      }
    },
    {
      "key": "<guid>",
      "next": "WAIT-[#]",
      "arguments": {
        "when": false
      }
    }
  ],
  "configurationArguments": {
    "refActivityCustomerKey": "EMAIL-1",
    "statsTypeId": 2,
    "statusUrlId": "0"
  }
}
```

```

    },
    "metaData": {
      "refActivityName": "<nameOfEmailForReference",
    }
  }
}

```

▼ Random Split Format

Random Split Format

The random split activity allows a marketer to create a multiple outcomes that contacts are randomly siphoned within a journey. This activity type cannot be used in a custom activity.

Helpful Hints

- The percentages must add up to 100 and are represented as only positive integers greater than zero and less than 100.
- The outcome of this decision affects goal statistics.

Journey Specification Format

The activity must be formatted as specified below. The following JSON object is included in the `activities` array of a journey, as defined in the Journey Specification.

Sample Request

```

{
  "key": "Random-Split-1",
  "type": "randomSplit",
  "outcomes": [
    {
      "next": "send-sms",
      "arguments": {
        "percentage": 90
      }
    },
    {
      "next": null,
      "arguments": {
        "percentage": 10
      }
    }
  ]
}

```

Sample Response

```

{
  "id": "<Marketing Cloud provided GUID>",
  "key": "Random-Split-1",
  "type": "randomSplit",
  "outcomes": [
    {
      "next": "send-sms",
      "arguments": {
        "percentage": 90
      }
    },
    {
      "next": null,
      "arguments": {
        "percentage": 10
      }
    }
  ]
}

```

▼ Join Activity Format

Join Activity Format

Join is not actually an activity. Joins are a convention of outcomes, but are represented in the journey as activities to make them easier to edit. Multiple outcomes with the same `next` value, pointing to the same activity, constitutes a join. Joins are used by marketers to to easily join branches back together. This activity type cannot be used in a custom activity.

Journey Specification Format

The activity must be formatted as specified below. The following JSON object is included in the `activities` array of a journey, as defined in the Journey Specification.

Sample Request

```
{
  "key": "Wait4",
  "type": "Wait",
  "configurationArguments": {
    "waitDuration": 2,
    "waitUnit": "DAYS"
  },
  "outcomes": [
    {
      "next": "Email6"
    }
  ]
},
{
  "key": "Email5",
  "type": "EMAILV2",
  "outcomes": [
    {
      "next": "Email6"
    }
  ]
}
```

Sample Response

```
{
  "key": "Wait4",
  "type": "Wait",
  "configurationArguments": {
    "waitDuration": 2,
    "waitUnit": "DAYS"
  },
  "outcomes": [
    {
      "next": "Email6"
    }
  ]
},
{
  "key": "Email5",
  "type": "EMAILV2",
  "outcomes": [
    {
      "next": "Email6"
    }
  ]
}
```

▼ REST (Custom) Activity Format

REST (Custom) Activity Format

The REST activity can be used to extend Journey Builder functionality. It works by posting request bodies to external endpoints.

Journey Specification Format

The activity must be formatted as specified below. The following JSON object is included in the `activities` array of a journey, as defined in the Journey Specification.

Helpful Hints

- The outcome of this decision affects goal statistics.
- If the `useJwt` property is set to true for any of the REST activity's methods, then a JWT for the account is generated, encoded, and sent as the POST body.
- If the `execute` method timeout is greater than the maximum allowed for the account, then use the maximum timeout value as configured in the account settings.
- The maximum timeout for the custom activity is 100,000 milliseconds. See [Custom Activity Configuration](#) for more details.

Compatibility Information

This information is provided for developers upgrading their activities from the legacy format to the Journey Specification.

0.5 SerializedObject	1.0 Serialized Object
<code>execute.uri</code>	<code>configurationArguments.url</code>
<code>execute.verb</code>	<code>configurationArguments.method</code>
<code>execute.body</code>	<code>arguments.execute-body</code>
<code>execute.format</code>	not supported (default is JSON)
<code>execute.useJwt</code>	<code>configurationArguments.useJwt</code>
<code>execute.timeout</code>	<code>configurationArguments.timeout</code>
<code>execute.inArguments</code>	<code>arguments.executeInArguments</code>
<code>execute.headers</code>	<code>arguments.header</code>

Sample Activity

```
{
  "key": "rest-1",
  "name": "My Rest Activity",
  "type": "Rest",
  "metaData": {
    "flowDisplayName": "myFlow" // Displayed in the UI
  },
  "outcomes": [
    {
      "key": "rest-1-outcome-1",
      "next": "<key of next activity>"
    }
  ],
  "configurationArguments": {
    "save": {
      "url": "https://www.example.com/endpoint",
      "useJwt": false,
      "timeout": 3000,
      "headers": "https://www.example.com/endpoint",
      "body": "Hello"
    },
    "validate": {
      "url": "https://www.example.com/endpoint",
      "useJwt": false,
      "timeout": 3000,
      "headers": "https://www.example.com/endpoint",
      "body": "Hello"
    },
    "publish": {
      "url": "https://www.example.com/endpoint",
      "useJwt": false,
      "timeout": 3000,
      "headers": "https://www.example.com/endpoint",
      "body": "Hello"
    }
  },
  "arguments": {
    "execute": {
      "url": "https://www.example.com/endpoint",
      "headers": "",
      "inArguments": [
        {
          "myArgument": ""
        }
      ]
    }
  }
}
```

```

    },
    "body": "Hello . This field kept for backward compatibility.", // notice the Data Binding
    "method": "POST",
    "useJwt": true,
    "timeout": 3000
  }
}

```

▼ Wait Format

Wait Format

The wait activity is used to indicate a period of time that a journey should pause before continuing on to process the subsequent outcome activity. This activity type cannot be used in a custom activity.

Journey Specification Format

The activity must be formatted as specified below. The following JSON object is included in the `activities` array of a journey, as defined in the Journey Specification.

Helpful Hints

- Valid values for the `configurationArguments.waitUnit` property are minutes, days, hours, weeks, or months.

Sample Request

```

{
  "key": "wait-1",
  "name": "My Wait Activity",
  "type": "Wait",
  "outcomes": [{
    "key": "wait-1-outcome-1",
    "next": "<key of next activity>"
  }],
  "configurationArguments": {
    "waitDuration": "1",
    "waitUnit": "weeks",
    "specifiedTime": "00:00",
    "timeZoneId": "<one of the valid timezones listed below>"
  }
}

```

Sample Response

```



{
  "id": "<Marketing Cloud provided GUID>",
  "key": "wait-1",
  "name": "My Wait Activity",
  "type": "Wait",
  "outcomes": [{
    "key": "wait-1-outcome-1",
    "next": "<key of next activity>"
  }],
  "configurationArguments": {
    "waitDuration": "1",
    "waitUnit": "weeks",
  }
}

```

Compatibility Information

This information is provided for developers upgrading their activities from the legacy format to the Journey Spec.

Aa 0.5 Value	≡ 1.0 Value
<u>duration</u>	waitDuration
<u>durationUnits</u>	waitUnit

 0.5 Value	 1.0 Value
<u>specifiedTime</u>	specificTime
<u>timeZone</u>	timeZoneld

Valid Timezones for Use with Wait Activities

- Afghanistan Standard Time
- Alaskan Standard Time
- Arab Standard Time
- Arabian Standard Time
- Arabic Standard Time
- Argentina Standard Time
- Atlantic Standard Time
- AUS Central Standard Time
- AUS Eastern Standard Time
- Azerbaijan Standard Time
- Azores Standard Time
- Canada Central Standard Time
- Cape Verde Standard Time
- Caucasus Standard Time
- Cen. Australia Standard Time
- Central America Standard Time
- Central Asia Standard Time
- Central Brazilian Standard Time
- Central Europe Standard Time
- Central European Standard Time
- Central Pacific Standard Time
- Central Standard Time
- Central Standard Time (Mexico)
- China Standard Time
- Dateline Standard Time
- E. Africa Standard Time
- E. Australia Standard Time
- E. Europe Standard Time
- E. South America Standard Time
- Eastern Standard Time
- Egypt Standard Time
- Ekaterinburg Standard Time
- Fiji Standard Time
- FLE Standard Time

- Georgian Standard Time
- GMT Standard Time
- Greenland Standard Time
- Greenwich Standard Time
- GTB Standard Time
- Hawaiian Standard Time
- India Standard Time
- Iran Standard Time
- Israel Standard Time
- Jordan Standard Time
- Korea Standard Time
- Mauritius Standard Time
- Mid-Atlantic Standard Time
- Middle East Standard Time
- Montevideo Standard Time
- Morocco Standard Time
- Mountain Standard Time
- Mountain Standard Time (Mexico)
- Myanmar Standard Time
- N. Central Asia Standard Time
- Namibia Standard Time
- Nepal Standard Time
- New Zealand Standard Time
- Newfoundland Standard Time
- North Asia East Standard Time
- North Asia Standard Time
- Pacific SA Standard Time
- Pacific Standard Time
- Pacific Standard Time (Mexico)
- Pakistan Standard Time
- Romance Standard Time
- Russian Standard Time
- SA Pacific Standard Time
- SA Western Standard Time
- Samoa Standard Time
- SE Asia Standard Time
- Singapore Standard Time
- South Africa Standard Time

- Sri Lanka Standard Time
- Taipei Standard Time
- Tasmania Standard Time
- Tokyo Standard Time
- Tonga Standard Time
- US Eastern Standard Time
- US Mountain Standard Time
- Venezuela Standard Time
- Vladivostok Standard Time
- W. Australia Standard Time
- W. Central Africa Standard Time
- W. Europe Standard Time
- West Asia Standard Time
- West Pacific Standard Time
- Yakutsk Standard Time

▼ Fire an Entry Event

Fire an Entry Event

There are two ways to send events to Journey Builder:

- Using a Fire Event Activity in Automation Studio
- By POSTing an event to the Journey Builder API

To send events using the API:

1. Define the event and its properties.
 - If defining the event in Marketing Cloud, make note of the Event Definition Key for API.
 - If defining the event via the API, use the POST /eventDefinition resource.
2. Use a valid OAuth access token to authorize requests to the Marketing Cloud REST APIs.
3. Using the POST /events resource, POST the required fields and event properties to the API.

Results

When an event is posted to the API, the data schema of the associated data extension validates that all fields from the API call exist in the data extension. The data extension associated with an event is defined in the DataExtensionID field in the EventDefinition table.

- If a field is missing, a validation exception is thrown.
- If the request is successful, an eventInstanceId is returned. This ID is a unique identifier for this specific request.

▼ Single-Send Journeys

Single-Send Journeys

There are two types of Journey Builder sends:

- Multi-step journeys
- Single-send journeys

Most use cases for interacting with journeys via API involve multi-step journeys. Multi-step journeys enable you to create and manage customer journeys through a series of interactions with your brand. These journeys can include multiple channels and customized journey paths based on customer interactions or attributes. Multi-step journeys can also have different activities and versions. The Journey Builder developer documentation is based on multi-step journeys.

You can employ single-send journeys for simple, single-use journeys that marketers create in Journey Builder. However, you can interact with single-send journeys via API, if required, by modifying the journey specification. Goals, versions, and recurring scheduling are not supported for single-send journeys. Email, SMS, and push notifications are the only activities available for single-send journeys.

Specification for Single-Send Journeys

To set up a single-send journey via API, follow the [journey specification](#), with these key modifications:

- In the journey object, use `Quicksend` for the `definitionType` parameter.
- In the journey object, for the `channel` parameter, use one of the following values: `email` for email, `push` for push notifications, or `sms` for SMS messages.
- In the activity object, for the `type` parameter in the activities list, use one of the following values: `emailv2` for email, `pushnotificationactivity` for push notifications, or `smssync` for SMS messages. You can define only one activity.
- In the trigger portion of the event object, use `EmailAudience` as the `type` parameter for email messages and SMS messages, and use `ContactAudience` as the `type` parameter for push notifications. `EmailAudience` is the same as the data extension entry source in Journey Builder.
- Don't include goals, versioning, or a recurrent schedule.

▼ GroupConnect Chat Messaging API

GroupConnect Chat Messaging API

Send real-time event-based messages to your users in Facebook Messenger and LINE.

Use Cases

- Send Account Updates - Send a message when changes occur in the user account.
 - Profile
 - Preferences
 - Settings
 - Membership status
 - Password
- Send Payment Updates
 - Send a receipt
 - Send an out-of-stock notification
 - Notify an auction has ended
 - Send a status change on a payment transaction
- Send Shipping Updates
 - Product is shipped
 - Status changes to in-transit
 - Product is delivered
 - Shipment is delayed

- Send Reservation Updates
 - Itinerary changes
 - Location changes
 - Confirm cancellation
 - Cancel hotel booking
 - Car rental pick-up time changes
 - Confirm upgrade
- Send Transportation Updates
 - Flight status changes
 - Ride is canceled
 - Trip is started
 - Ferry has arrived
- Send Ticket Updates
 - Concert start time changes
 - Event location changes
 - Show is cancelled
 - Refund opportunity is available

Known Limitations

- This API cannot automatically create contacts.
- This API cannot use content templates from Content Builder.
- API send reports are not available.
- This API addresses only install package permissions.

▼ Set Up Your GroupConnect Chat Messaging API

Set Up Your GroupConnect Chat Messaging API

Prerequisites

Your account must be provisioned for GroupConnect before you can use the GroupConnect Chat Messaging API.

Follow these steps to set up the GroupConnect Chat Messaging API in your account.

1. [Create an API integration](#) to use Marketing Cloud API.
2. [Get an API integration key](#).
3. For the installed packages scope, [provide these permissions](#).
 - Channels
 - OTT
 - Read
 - Send
 - Provisioning
 - OTT Channels

- Read
 - Update
4. Register new Facebook pages or LINE channels with Marketing Cloud's Chat Messaging API. Use the [POST /ott/v1/registration](#) endpoint. Skip this step if you already have channels and pages registered with Marketing Cloud.
 5. Send outbound messages to Facebook Messenger and LINE via Chat Messaging API. Use the POST /ott/v1/send

▼ Use the MobileConnect API

Use the MobileConnect API

Subscribe Mobile Numbers to a Short Code

Delivery of an outbound SMS message through the MobileConnect application requires an active short code and keyword subscription. MobileConnect assists marketers and developers with the creation of message subscriptions through the SMS Opt-In template.

MobileConnect supports three variations of the SMS Opt-In:

1. Single Opt-In: Contact texts JOIN (or another specified keyword) to your short code to subscribe to a specified keyword on that short code.
2. Double Opt-In: Contact texts JOIN (or another specified keyword) to your short code and receives a text message asking them to reply Y or YES to confirm their subscription. Upon replying Y or YES, MobileConnect subscribes the contact to the specified keyword on that short code.
3. Double Opt-In with Age Confirmation: Contact texts JOIN (or another specified keyword) to your short code, the contact reply to the initial confirmation message and confirm with their birth date before subscribing to the specified keyword on that short code.

Warning

CTIA and MMA best practices state that SMS subscriptions originating from a web form or mobile app must use the Double Opt-In variation. Use the MobileConnect QueueMO REST API to replicate the original inbound text of JOIN. This way, the contact will only receive one text message that requires replying with Y or YES to finalize creation of the subscription.

Create New Message

1 Select Template | **2 Define Content** | **3 Select Audience** | **4 Schedule/Activate** | **Cancel**

Outbound Send an SMS message to your subscribers	Text Response Create an automatic response to an inbound SMS message	Vote/Survey Send an SMS message or activate a published SMS message inviting recipients to vote or respond to a survey	Mobile Opt-In Invite people to subscribe to your SMS and MMS messages	Info Capture Prompt recipients to provide information to be stored as Contact attributes
Outbound Media Send an MMS message to your subscribers	Media Response Create an automatic MMS response to inbound messages	Media Share Capture and store inbound MMS messages	Email Opt-In Invite people to subscribe to your email messages	Send Email Invite people to receive a triggered email message

✓ Select Template
Mobile Opt-In

2 Define Content

3 Schedule/Activate

BackNext

Setup

Message Name *

Hackathon Demo

Short/Long Code *

86288US

Keyword*

Select Existing KeywordTEST

Keyword in use: test June 03, 2012 To: ongoing

Create Keyword

Create

Call to Action

SMS Send and External Source

A subscriber receives an outbound message or a person responds to a call to action from an external source.

External Source Message (Optional)

Text 'TEST' to 86288.

Messages

Subscription Options

Double Opt-In

Subscriber will receive4 messages everyMonth

☐ Allow a single opt-in per mobile number.

Initial Message

Thanks for subscribing! Please respond with y to confirm your subscription.

75 characters

Acceptance Response

y x yes x

Confirmation Message

Thanks for subscribing! You will receive approx. 4 messages per month.

70 characters

Opt-in Error Message

Sorry, there was an error. Please try again.

44 characters

Example QueueMO API Request

```
POST `https://YOUR_SUBDOMAIN.rest.marketingcloudapis.com/sms/v1/queueMO`
Content-Type: application/json
Authorization: Bearer YOUR_OAUTH_ACCESS_TOKEN

{
  "mobileNumbers": [
    "15555551212"
  ],
  "shortCode": "Code",
  "messageText": "KEYWORD"
}
```

Key:

- `mobileNumbers` includes the mobile number to subscribe
- `shortCode` includes the SMS short code used in the SMS Opt-In template
- `messageText` contains the keyword used in the SMS Opt-In template

Confirm Subscription Status in MobileConnect

Once you create an SMS Opt-In message, ensure that MobileConnect creates the subscriptions correctly.

1. Navigate to the MobileConnect overview screen.
2. Open the Contacts application.
3. Click **Manage** under Contacts.

4. Search for the specific mobile number.
5. Click the mobile number.
6. Click **Subscriptions**.
7. Confirm the subscription status of **Subscribed**. MobileConnect will only deliver outbound SMS messages to mobile numbers that maintain a **Subscribed** status for the specified short code. Mobile numbers with an **In Progress** status did not fully complete the Double Opt-in process and will not receive SMS messages.

Addresses

Subscriptions

MOBILE

12014035908

Create Subscription

KEYWORD	CODE	SOURCE	OPT-IN METHOD	OPT-IN DATE	OPT-OUT DATE	OPT-OUT SOURCE	STATUS
DRIVER	88769	Application	Manually Added	6/4/2013 7:04 PM			Subscribed

Insider Tip: You built a mobile app, now you need users. The expectation that contacts will see your sign promoting the app, take out their phone, open the App Store or Google Play, search for your app, and finally download your app represents only part of your overall plan. Allow consumers to request a link to download your app via SMS. For example, "Text SMCAPP to (your short code) to download the mobile app." When a contact texts SMCAPP to your short code, that contact will receive a text message containing a link to your app in the App Store or Google Play. Create this message with the MobileConnect Text Response template.

Trigger SMS Messages from a Mobile App

Once you build a mobile app and subscribe contacts to your short code and keyword, you can deliver SMS messages. Use the MobileConnect Outbound message template to define the content for each message. Follow these steps to create your message:

1. Name the message.
2. Select the short code to be used for delivery. Use the same code on which contacts subscribed to a specific keyword.
3. Select the **API Trigger** send method.
4. Define message content. Both the MessageContact and MessageList APIs allow for override of defined message content.
5. Activate your message. Both the messageContact and messageList request bodies use the API key received during this step.

1 Select Template
2 Define Content
3 Select Audience
4 Schedule/Activate
Cancel

Outbound

Send a SMS message to your subscribers

Text Response

Create an automatic response to an inbound SMS message

Vote/Survey

Send an SMS message or activate a published SMS message inviting recipients to vote or respond to a survey

SMS Opt-In

Invite people to subscribe to your SMS messages

Info Capture

Prompt recipients to provide information to be stored as Contact attributes

Email Opt-In

Invite people to subscribe to your email messages

Send Email

Invite people to receive a triggered email message

Back to Overview
Cancel
Save

Create New Message

Select Template
Outbound - API Triggered
2 Define Content/Activate
Back
Activate

Setup

Message Name *
Hackathon Test Message

Short/Long Code *
80565
US

Messages

Send Method ?
☐ Schedule
☒ API Trigger
☐ Automation
☐ Interaction

Outbound Message
This is an SMS message from MobileConnect.
42 characters

Campaign
Add Campaign Association
Create Campaign

Back to Overview
Cancel
Save

Create New Message

Select Template
Outbound - API Triggered
2 Define Content/Activate
Back
Activate

Setup

Message Name *
Hackathon Test Message

Short/Long Code *
80565
US

Messages

Send Method ?
☐ Schedule
☒ API Trigger
☐ Automation
☐ Interaction

Outbound Message
This is an SMS message from MobileConnect.
42 characters

Campaign
Add Campaign Association
Create Campaign

Activate
Click Confirm to immediately activate your message.
API Key for this message: MJ41MT030Dow
Cancel
Confirm

Example messageContact API Request

```
POST https://YOUR_SUBDOMAIN.rest.marketingcloudapis.com/sms/v1/messageContact/MzA6Nzg6MA/send
Content-Type: application/json
Authorization: Bearer YOUR_OAUTH_ACCESS_TOKEN
{
  "mobileNumbers": [
    "13175551212"
  ],
  "Subscribe": false,
  "Resubscribe": false,
  "Override": true,
  "messageText": "Welcome to Code@"
}
```

If you wish to ensure that the contact exists for the specified mobile number and that the mobile number subscribed to the specified keyword for your API message, set the `Subscribe` and `Resubscribe` values to `true`, as shown in the example below.

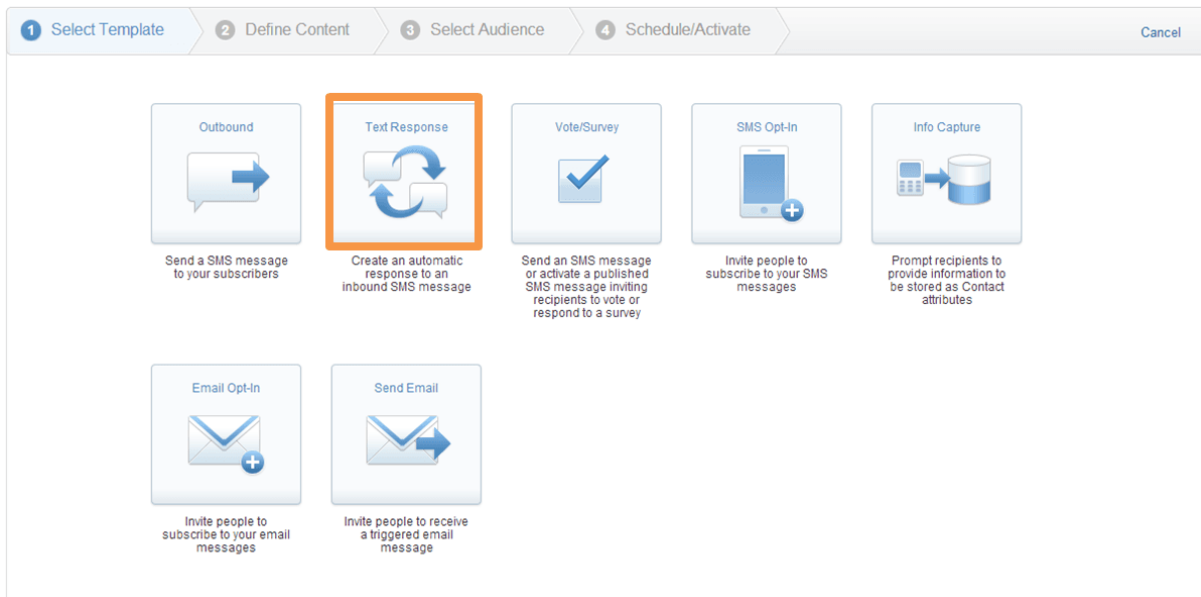
```
POST https://YOUR_SUBDOMAIN.rest.marketingcloudapis.com/sms/v1/messageContact/MzA6Nzg6MA/send
Content-Type: application/json
Authorization: Bearer YOUR_OAUTH_ACCESS_TOKEN
{
  "mobileNumbers": [
    "13175551212"
  ],
  "Subscribe": true,
  "Resubscribe": true,
  "Override": true,
  "messageText": "Welcome to Code@"
}
```

Key:

- `Override` allows for the override of message content defined within the MobileConnect UI
- `messageText` defines message content if `Override` = true

Deliver On-Demand SMS Content with Text Response

SMS allows for two conversations initiated by consumers. Use the MobileConnect Text Response template to create content sent to consumers in response to an SMS message to your short code. For example: "Text STOCK to (your short code) to get a real-time quote of Brand ABC stock price."



Select Template Text Response **2 Define Content** **3 Schedule/Activate** Back Next

Setup

Message Name * Hackathon Stock Demo

Short/Long Code * 83767 GB

Keyword* ☐ Select Existing Keyword STOCK ☒ Create Keyword STOCK X

Call to Action ? SMS Send and External Source

A subscriber receives an outbound message or a person responds to a call to action from an external source.

External Source Message (Optional)
Text 'STOCK' to 83767.

Example Message Content for STOCK Use Case

```
%%[
Set @mobile = v([MOBILE_NUMBER])
Set @mobile1 = Substring(@mobile,2,10)
Set @keyword = v([MSG(0).VERB])
set @noun = lowercase([MSG(0).NOUN(0)])
set @url = Concat("http://finance.example.com/d/quotes.csv?s=CRM&f=snd11yr",@mobile1,"&alert_keyword=",@keyword)
set @msg = [MSG(0)]set @message = HttpGet(@url)]%%
example.com Stock is: %%=ProperCase(@message)=%% Thanks for Responding!
```

Messages

Response Message *

%%[Set @mobile = v([MOBILE_NUMBER]) Set @mobile1 = Substring(@mobile,2,10) Set @keyword = v([MSG(0).VERB]) set @noun = lowercase([MSG(0).NOUN(0)]) set @url = Concat("http://finance.yahoo.com/d/quotes.csv?s=CRM&f=snd11yr",@mobile1,"&alert_keyword=",@keyword)]%%

Character Count Unavailable.

[Add Alternate Response ?](#)

Error Message

Sorry, there was an error. Please try again.

45 characters

▼ SMS Status Codes

SMS Status Codes

This table includes status codes returned from MobileConnect API calls for all locations. Use these codes to evaluate and troubleshoot your SMS sends.

# Code	Status	Definition
1000	QueuedToSfmcSendService	Message queued to the internal send service.
1500	QueueFailureToSfmcSendService	Message failed to queue to the internal send service. Retry your send.
1501	ValidationError	Status indicates an internal validation error. Retry your send.
2000	DeliveredToAggregator	Message delivered to aggregator. Status is updated when delivery confirmation comes from carrier or mobile device. For shared codes, this status is the final one.
2500	FailedToAggregator	Message not delivered to aggregator. Retry your send.
2501	UnknownToAggregator	Unknown aggregator error.
2502	FailedToAggregatorDueToInvalidDestinationAddress	Invalid Destination Address.
2600	ThrottledToAggregator	Message not accepted by aggregator due to capacity issues. Salesforce exhausted the retry process.

# Code	Status	Definition
2601	SocketExceptionToAggregator	Socket connection error to aggregator. Can retry. If this status is logged, we exhausted our retries.
3000	Enroute	Message is en route to carrier. Waiting on carrier confirmation.
3001	SentToCarrier	Message sent to carrier. Waiting to be accepted by carrier.
3002	AcceptedByCarrier	Message accepted by carrier. Waiting for delivery confirmation.
3400	Unknown	Unknown error.
4000	Delivered	Message delivered to mobile device.
4500	Undeliverable	Message not delivered to mobile device.
4501	Expired	Message expired. Message exhausted the carrier retry process. Mobile device is out of carrier range.
4502	Deleted	Message deleted by the carrier.
4503	Rejected	Message rejected. Carrier detected a loop or assumed that message is spam. This status can indicate an administrative or financial problem between the operator and the end users.
4504	FailedDueToUnknownSubscriber	Unknown Subscriber.
4505	FailedDueToInvalidDestinationAddress	Invalid Destination Address.

▼ Einstein Recommendations API

Einstein Recommendations API

The Einstein Recommendations engine empowers marketers to deliver millions of messages across key digital channels that are personalized for each individual's needs, interests, and preferences.

Developers use the Einstein Recommendations API to update the data used to provide recommendations, manage consumer privacy requests, and to easily download large reports.

Set Up Your Development Environment

1. Log in to Marketing Cloud.
2. Go to **Einstein | Email or Web Recommendations | Reporting | Admin Tools | Manage API Keys**.
3. Copy the **Crimson Kraken** API key value. This key is created when your account is provisioned.
4. To authenticate your calls, pass the API key in the header like this:

```
Host: https://app.igodigital.com
POST /api/v2/organization/{customer_mid}/...
Content-Type: application/json
Authorization: Bearer YOUR_API_KEY
```

▼ Privacy Batch API

Privacy Batch API

Change privacy settings for a Einstein Recommendations profile using the privacy batch API. The batch identifies the profiles and specifies which privacy actions to perform on each profile.

The privacy batch API includes these resources:

- [Create a Batch](#). This API resource creates a batch of privacy setting changes. The call returns a batch_id you then use to GET the status of the batch changes.
- [Get Batch Status](#). This API resource checks the status of a batch of privacy settings changes. The call returns the status of each change being made to the profiles.

▼ Create a Batch

Create a Batch

Overview

This API resource creates a batch of privacy setting changes. The call returns a `batch_id` you then use to GET the status of the batch changes.

Method and Endpoint

POST https://app.igodigital.com/api/v2/organization/{customer_mid}/privacy

URL Parameters

Name	Type	Description
<code>customer_mid</code>	string	ID for the account.

JSON Parameters

Name	Type	Required	Description
<code>batch</code>	array	<input checked="" type="checkbox"/>	Contains key value pairs where the key represents a profile ID and the value is an array of strings representing the intended state of the profile's privacy settings after the record is processed. To remove any existing privacy settings for a profile, pass an empty array, such as <code>[]</code> .

Privacy Values to Restrict

To apply the privacy states to the profile, include one or more of these values in the array.

Value	Description
<code>do_not_track</code>	Prevents data received via collect from being saved by Personalization Builder.
<code>do_not_profile</code>	Prevents using the profile in generating recommendations.
<code>do_not_process</code>	Prevents processing of the profile's data.

Privacy Values for Right to be Forgotten

These values represent exclusive actions for a single profile. Don't use them with any other processing strings in that profile's array.

Value	Description
<code>rtbf_delete</code>	Immediately deletes the profile and its associated data.
<code>rtbf_suppress</code>	Deletes the profile and its associated data after a suppression period of up to 30 days. You can change the suppression period privacy setting in Marketing Cloud. Fourteen days is the default.
<code>remove_rtbf_suppress</code>	Stops a previously scheduled profile deletion.

Example Request

```
Host: https://app.igodigital.com
POST /api/v2/organization/12345678/privacy
Content-Type: application/json
Authorization: Bearer YOUR_API_KEY
```

```
{
  "batch": {
    "hashed-id-01": ["do_not_track"], // Adds "do_not_track" privacy setting, removes "do_not_profile" or "do_not_process" if they
    were previously set
  }
}
```

```

        "hashed-id-02": ["do_not_profile"], // Adds "do_not_profile" privacy setting, removes "do_not_track" or "do_not_process" if they
were previously set
        "hashed-id-03": ["do_not_process"], // Adds "do_not_process" privacy setting, removes "do_not_profile" or "do_not_track" if they
were previously set
        "hashed-id-04": ["do_not_track", "do_not_profile"], // Adds "do_not_track" and "do_not_profile" privacy settings, removes
"do_not_process" if it was previously set
        "hashed-id-05": ["do_not_track", "do_not_process"], // Adds "do_not_track" and "do_not_process" privacy settings, removes
"do_not_profile" if it was previously set
        "hashed-id-06": ["do_not_profile", "do_not_process"], // Adds "do_not_profile" and "do_not_process" privacy settings, removes
"do_not_track" if it was previously set
        "hashed-id-07": ["do_not_track", "do_not_profile", "do_not_process"], // Adds "do_not_track", "do_not_profile" and
"do_not_process" privacy settings
        "hashed-id-08": [], // Removes any existing privacy settings for profile
        "hashed-id-09": ["rtbf_delete"], // Deletes profile and associated data
        "hashed-id-10": ["rtbf_suppress"] // Deletes profile and associated data after suppression period
    }
}

```

Example Response

```

{
  "batch_id": "af417962-c188-4110-adc2-7ea67590e254"
}

```

▼ Get Batch Status

Get Batch Status

Overview

This API resource checks the status of a batch of privacy settings changes. The call returns the status of each change made to the profiles.

Method and Endpoint

GET https://app.igodigital.com/api/v2/organization/{customer_mid}/privacy/{batch_id}

URL Parameters

Name	Type	Description
<u>customer_mid</u>	string	ID for the account.
<u>batch_id</u>	string	ID generated when creating a batch of privacy setting changes.

Example Request

```

Host: https://app.igodigital.com
GET /api/v2/organization/12345678/privacy/af417962-c188-4110-adc2-7ea67590e254
Content-Type: application/json
Authorization: Bearer YOUR_API_KEY

```

Example Response

Status values for a change can be `initiated`, `success`, `failed` or `cancelled`.

```

[
  {
    "date_initiated": "2018-03-12 18:45:17 UTC",
    "profile_id": "hashed-id-01",
    "action": "add_do_not_track",
    "source": "api",
    "status": "success",
    "batch_id": "af417962-c188-4110-adc2-7ea67590e254",
    "date_completed": "2018-03-12 18:45:17 UTC"
  },
  {
    "date_initiated": "2018-03-12 18:45:17 UTC",
    "profile_id": "hashed-id-08",
    "action": "remove_do_not_profile",

```

```

    "source": "api",
    "status": "initiated",
    "batch_id": "af417962-c188-4110-adc2-7ea67590e254",
    "date_completed": "2018-03-12 18:45:17 UTC"
  }
]

```

▼ Download Email Recommendation Log

Download Email Recommendation Log

Marketers can retrieve a report listing the count of email recommendations by product code. For a given date range, this report lists each product code and the number of the times that product was recommended for each email recommendation configuration, and, optionally, job id. To know which job id a recommendation is associated with, the email recommendation image must have a URL parameter named `sfmc_j`. The email job id is the parameter's value.

Before attempting to run or download the report, turn it on in Personalization Builder.

Parameters

To download an email recommendation log, you need these values.

Aa Value	Description
<code>MID</code>	The account MID that contains the data.
<code>api_key</code>	A valid API Key for that MID.
<code>start_date</code>	Start date for the report.
<code>end_date</code>	End date for the report.
<code>aggregation_method</code>	The only option available for this report is <code>day</code> . Personalization Builder rolls up counts per day.

Example

```

curl --header 'Content-Type: application/json' -XPOST https://
app.igodigital.com/api/v2/reporting/report/displays_by_item.csv -d '{
  "mid": "1314420",
  "api_key": "<redacted>",
  "aggregation_method": "day",
  "start_date": "2017-10-29",
  "end_date": "2017-11-29"
}' > report.csv

```

▼ Einstein Recommendation Collect Tracking API

Einstein Recommendation Collect Tracking API

Overview

Einstein Recommendation Collect Tracking calls are available to be implemented with a server side option. To use these implementation methods, make an HTTP POST call to the appropriate URL, with the corresponding payload parameter values.

In order to manage session and user values you'll must send `user_id` and `session_id` on the same node level as the payload. On the first call, you don't must pass these values and they're returned in the JSON response.

See the parameter definitions and example URLs below.

Note

All calls must be sent with Content-Type: application/json.

Below is a sample of the minimum requirement to implement an observation script, also referred to as "Collect".

```

http://MID.collect.igodigital.com/c2/MID/track_page_view.json
{
  "payload": {

```

```

    "title": "INSERT_PAGE_TITLE",
    "url": "INSERT_PAGE_URL",
    "referrer": "INSERT_REFERRER_URL",
    "user_info": {
      "email": "INSERT_EMAIL_OR_SUBSCRIBER_ID"
    }
  },
  "user_id": "",
  "session_id": ""
}

```

Parameter Definitions

MID

A unique client identifier, typically the MID associated with the Marketing Cloud Business Unit.

title:

The title of the page the user is viewing. This is pulled from the page title with web applications.

url:

The url or path to the application page.

referrer:

The previous page or referring url for the application.

email:

Any page where the user's email address (or other identifier) is known to the site, this value should be sent. The users email address should be sent on the following pages and any others where the users email address is known: Newsletter signup, login, account overview, cart, checkout, purchase confirmation, order history.

user_id:

The user_id value returned from the first Collect API call for an individual user. This value should be persisted for all future API calls for that individual.

Note

If you are unable to persist this value for future sessions, a new profile may will be created along with the new session. Duplicate records sharing the same email will be merged at the end of the session.

session_id:

The session_id value returned from the first Collect API call for the user's current session. This value should be persisted for all API calls in an individual's session.

Item Detail View Collect Example

The following sample implements an observation script to track a specific item detail page view.

```

http://MID.collect.igodigital.com/c2/MID/track_page_view.json
{
  "payload": {
    "item": "INSERT_ITEM",
    "user_info": {
      "email": "INSERT_EMAIL_OR_SUBSCRIBER_ID"
    }
  },
  "user_id": "",
  "session_id": ""
}

```

Additional Parameter Definitions

item:

Represents the item that is being displayed to the customer, such as on an item detail page.

Real-Time Rating Collect Example

The following sample implements an observation script to track when a user submits a rating for a specific item, using the minimum requirements.

```
http://MID.collect.igodigital.com/c2/MID/track_rating.json?
{
  "payload": {
    "item": "INSERT_ITEM",
    "rating": "INSERT_ITEM_RATING",
    "user_info": {
      "email": "INSERT_EMAIL_OR_SUBSCRIBER_ID"
    }
  },
  "user_id": "",
  "session_id": ""
}
```

Additional Parameter Definitions

rating:

This variable should be populated with a numerical representation of the rating submitted by the user, only when the user has actually rated the item. The item value must be passed in within the same call in order for the rating to be properly recorded.

Cart Collect Example

The following sample implements an observation script to track a snapshot of all items in the user's cart.

```
http://MID.collect.igodigital.com/c2/MID/track_cart.json?
{
  "payload": {
    "cart": [
      {
        "item": "INSERT_ITEM",
        "quantity": "INSERT_QUANTITY",
        "price": "INSERT_ITEM_PRICE",
        "unique_id": "INSERT_UNIQUE_ID"
      },
      {
        "item": "INSERT_ITEM",
        "quantity": "INSERT_QUANTITY",
        "price": "INSERT_ITEM_PRICE",
        "unique_id": "INSERT_UNIQUE_ID"
      }
    ],
    "user_info": {
      "email": "INSERT_EMAIL_OR_SUBSCRIBER_ID"
    }
  },
  "user_id": "",
  "session_id": ""
}
```

Additional Parameter Definitions

item:

This variable should be sent with an identifier of the item that is in the user's cart.

quantity:

This variable should contain the quantities of each corresponding item in the.

price:

This variable should contain the unit price for each corresponding cart item. Even if the customer is purchasing multiples of an item this variable should still be sent with the unit price.

Clear Cart Collect Example

The following sample implements an observation script to indicate that the user has emptied their shopping cart.

```
http://MID.collect.igodigital.com/c2/MID/track_cart.json?
{
  "payload": {
    "cart": {"clear_cart": true},
    "user_info": {
      "email": "INSERT_EMAIL_OR_SUBSCRIBER_ID"
    }
  },
  "user_id": "",
  "session_id": ""
}
```

Additional Parameter Definitions

clear_cart:

When true, indicates that the user has removed all items from their shopping cart.

Conversion Collect Example

The following sample implements an observation script to track a conversion event.

```
http://MID.collect.igodigital.com/c2/MID/track_conversion.json?
{
  "payload": {
    "cart": [
      {
        "item": "INSERT_ITEM",
        "quantity": "INSERT_QUANTITY",
        "price": "INSERT_ITEM_PRICE"
      },
      {
        "item": "INSERT_ITEM",
        "quantity": "INSERT_QUANTITY",
        "price": "INSERT_ITEM_PRICE"
      }
    ],
    "user_info": {
      "email": "INSERT_EMAIL_OR_SUBSCRIBER_ID"
    },
    "order_number": "INSERT_ORDER_NUMBER",
    "shipping": "INSERT_SHIPPING_CHARGE",
    "discount": "INSERT_DISCOUNT_AMOUNT"
  },
  "user_id": "",
  "session_id": ""
}
```

Additional Parameter Definitions

order_number:

This is a value that should contain the customer's order number once a conversion event occurs.

shipping:

This optional variable may be sent with the shipping charge associated with the user's order.

discount:

This optional variable may be sent with the discount amount that was applied to the user's order.

Search Collect Example

The following sample implements an observation script to track a specific item detail page view.

```
http://MID.collect.igodigital.com/c2/MID/track_page_view.json?
{
  "payload": {
    "search": "INSERT_SEARCH_TERM",
  }
}
```



```

    "title": "INSERT_PAGE_TITLE",
    "url": "INSERT_PAGE_URL",
    "referrer": "INSERT_REFERRER_URL",
    "user_info": {
      "email": "INSERT_EMAIL_OR_SUBSCRIBER_ID"
    }
  },
  "user_id": "",
  "session_id": ""
}

```

Additional Parameter Definitions

search:

This variable is used to hold search terms. This variable should be populated with internal and external search terms (for example, from Google) when available. Example: search=Blue Sky

Category Page Collect Example

The following sample implements an observation script to track a specific item detail page view.

```

http://MID.collect.igodigital.com/c2/MID/track_page_view.json?
{
  "payload": {
    "category": "INSERT_CATEGORY_NAME",
    "title": "INSERT_PAGE_TITLE",
    "url": "INSERT_PAGE_URL",
    "referrer": "INSERT_REFERRER_URL",
    "user_info": {
      "email": "INSERT_EMAIL_OR_SUBSCRIBER_ID"
    }
  },
  "user_id": "",
  "session_id": ""
}

```

Additional Parameter Definitions

category:

This variable should be sent on each category and subcategory page.

Example Response

```

{
  user_id: "5fb6fd46-7fd6-11e4-93db-123139337486"
  session_id: "5fb70c14-7fd6-11e4-93db-123139337486"
}

```

Create or Update a Single Item Example

This sample implements streaming updates to a single item in your catalog.

Replace the INSERT_ values with dynamic code to get the appropriate value for the item. Acceptable values for item_type are content, product, or banner. If the value for item_type is not specified, or the value provided is not recognized, it defaults to content.

```

https://MID.collect.igodigital.com/c2/MID/update_item_secure.json?api_key=<api_key>
{"payload":
{
  "item": "INSERT_ITEM",
  "unique_id": "INSERT_UNIQUE_ITEM_ID",
  "item_type": "INSERT_ITEM_TYPE",
  "name": "INSERT_ITEM_NAME_OR_TITLE",
  "INSERT_ATTRIBUTE_NAME": "INSERT_ATTRIBUTE_VALUE"
}
}

```

Create or Update Multiple Items Example

This sample updates multiple items in your catalog.

Replace the INSERT_ values with dynamic code to get the appropriate value for the item. Acceptable values for item_type are content, product, or banner. If the value for item_type is not specified, or the value provided is not recognized, it defaults to content.

```
https://MID.collect.igodigital.com/c2/MID/update_item_secure.json?api_key=<api_key>
{"payload":
[
  {
    "item": "INSERT_ITEM",
    "item_type": "INSERT_ITEM_TYPE",
    "name": "INSERT_ITEM_NAME_OR_TITLE",
    "INSERT_ATTRIBUTE_NAME": "INSERT_ATTRIBUTE_VALUE"
  },
  {
    "item": "INSERT_ITEM",
    "item_type": "INSERT_ITEM_TYPE",
    "name": "INSERT_ITEM_NAME_OR_TITLE",
    "INSERT_ATTRIBUTE_NAME": "INSERT_ATTRIBUTE_VALUE"
  }
]
}
```

Use this sample when an attribute has multiple values.

```
{
  "payload":
  {
    "item": "INSERT_ITEM",
    "item_type": "INSERT_ITEM_TYPE",
    "name": "INSERT_ITEM_NAME_OR_TITLE",
    "INSERT_ATTRIBUTE_NAME": ["INSERT_ATTRIBUTE_VALUE1", "INSERT_ATTRIBUTE_VALUE2", "INSERT_ATTRIBUTE_VALUE3"]
  }
}
```

Update Item Availability Example

This sample updates the availability of an item in your catalog. Possible values are: Y, or N

```
https://MID.collect.igodigital.com/api/update_product?key=<API_KEY>&sku=INSERT_UNIQUE_ITEM_ID&availability=N
```

[Contents](#)[Overview](#)[Item Detail View Collect Example](#)[Real-Time Rating Collect Example](#)[Cart Collect Example](#)[Clear Cart Collect Example](#)[Conversion Collect Example](#)[Search Collect Example](#)[Category Page Collect Example](#)[Create or Update a Single Item Example](#)[Create or Update Multiple Items Example](#)[Update](#)

▼ Transactional Messaging API

Transactional Messaging API

This documentation applies only to the Transactional Messaging API. For triggered sends that are managed in Email Studio, review the Triggered Email Scenario Guide.

Transactional messaging is immediate, automated, non-promotional messaging. Use-cases include order confirmation messages, password reset emails, and bank balance inquiry messages. Use the Marketing Cloud Transactional Messaging REST API to send personalized transactional email and SMS messages to your customers. To receive immediate notifications about whether your message was sent and other events, use the Marketing Cloud Event Notification Service.

These features distinguish the Transactional Messaging API from triggered sends in Email Studio and from MobileConnect outbound SMS messages.

- The API runs on an updated messaging platform, which improves scale and send speed.
- All messages are sent as quickly as possible. For email, there's no need to select high, medium, or low priority in a send definition.
- You can track each message through Marketing Cloud from the client's host system, including the Event Notification Service, using the messageKey attribute.

Considerations

- Singleton send requests, which use the recipient object attribute rather than the recipients array attribute, must provide a unique messageKey as an ID.
- The API doesn't support suppression lists or exclusion scripts.
- The API supports send logging only if your account is configured for auto send logging. To maintain maximum performance, ensure that your send log is configured with a data retention policy.
- To deduplicate at send time, use messageKey. Don't use a primary key on the triggered send data extension.
- You can create up to 500 total email transactional send definitions plus email triggered send definitions in a seven-day period for one business unit. This limit applies to send definitions created in the Marketing Cloud application and via API. It doesn't apply to the Journey Builder email activity triggered send definition.

▼ Get Started with Transactional Messaging API

Get Started with Transactional Messaging API

- For transactional emails:
 - In Content Builder, [create the email message content to send](#).
 - In Email Studio, either [create a list](#) to manage recipient data or [apply the recipient details to all subscribers](#).
- For transactional SMS:
 - In MobileConnect, [set up a keyword for your transactional short code](#).
- In Installed Packages, [create an API integration](#) to get a client ID and client secret. Use the following email permissions settings for the API integration: write, read, and send. Use the following SMS permissions settings for the API integration: write, read, and send.

Ensure that your client ID and secret are stored securely. Never expose this information on the client side via JavaScript or store it in a mobile application.

Tip: To save the attributes sent in the API request, [create a triggered send data extension](#) in Email Studio.

Send a Transactional Email Message

1. Get an access token to generate API requests: [Server-to-server integrations](#) and [web and public app integrations](#). To authenticate all your API requests, use an access token. You can reuse the access token for multiple requests before it expires.
2. [Set up an email send definition via the API](#). The send definition contains the message template and delivery configuration. You must reference a send definition with each message API request.
3. [Send a transactional email](#). Send your email to one or more recipients when triggered by an API.
4. [Create an optional subscription](#). A subscription uses a webhook to send you analytics in real time about whether emails are sent, not sent, or bounced.

Send a Transactional SMS Message

1. Get an access token to generate API requests: [Server-to-server integrations](#) and [web and public app integrations](#). To authenticate all your API requests, use an access token. You can reuse the access token for multiple requests before it expires.
2. [Set up a transactional SMS definition via the API](#). The send definition contains the message template and delivery configuration. We recommend that you set up a definition for each messaging use case, such as one for two-factor

authentication and one for order confirmations. You must reference a send definition with each outbound SMS send API request.

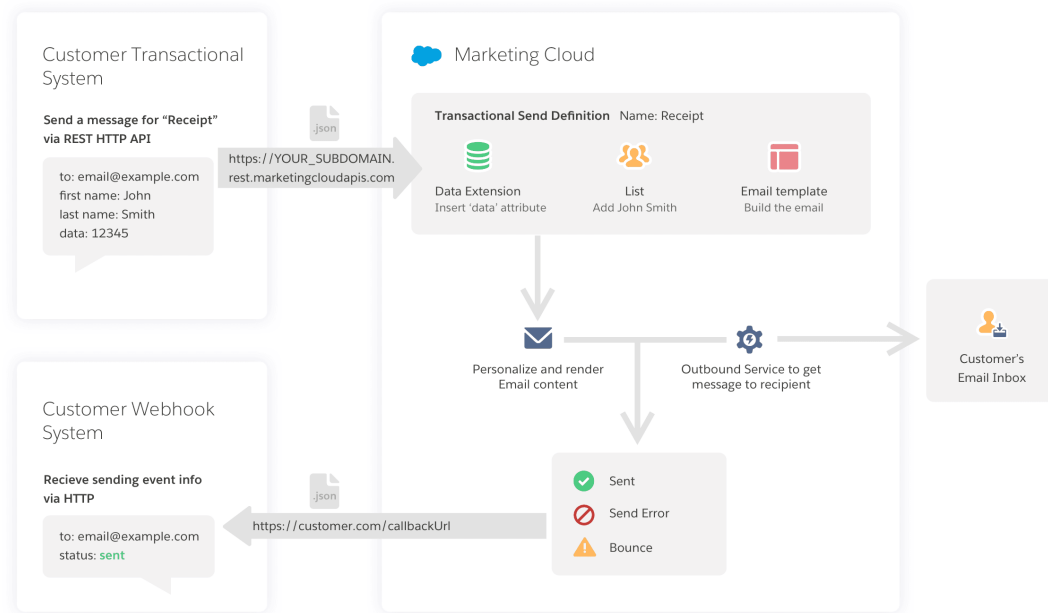
3. Send a transactional SMS. Send your SMS to one or more recipients when triggered by the API.
4. Create an optional subscription. A subscription uses a webhook to send you analytics in real time about whether SMS messages are sent, not sent, or bounced.

▼ Transactional Messaging API Activities

Transactional Messaging API Activities

When you send a message via the Transactional Messaging API, it goes through these steps.

1. Your system, outside of Marketing Cloud, sends a transactional message using the Marketing Cloud Transactional Messaging API.
2. Information from the API request is applied to the transactional message definition. This step personalizes the message per recipient, assigns recipients to a contact list, and saves request parameters to a data extension, if used.
3. The Event Notification Service sends the status of the message via a webhook to another system outside of Marketing Cloud.



▼ Transactional Send Response Level Error Codes

Transactional Send Response Level Error Codes

A response level error code occurs when there is a problem with syntax or other synchronous processing error, such as reusing a duplicate messageKey or malformed email address.

Response-Level Error Codes

# Error Code	Aa Error Message	Message-specific or definition (Resource)	Channel
10000	<u>MCMS_UTM_Validation</u>	/definitions	email sms
10001	<u>MCMS_UTM_Validation_InvalidDataType</u>	/definitions	email sms
10002	<u>MCMS_UTM_Validation_MissingRequiredField</u>	/definitions	email sms
10003	<u>MCMS_UTM_Validation_IncorrectFormat</u>	/definitions	email sms
10005	<u>MCMS_UTM_Validation_InvalidValue</u>	/definitions	email sms
30000	<u>MCMS_UTM_Runtime</u>	/definitions	email sms
30001	<u>MCMS_UTM_Runtime_InvalidOperation</u>	/definitions	email sms
30003	<u>MCMS_UTM_Runtime_ObjectNotFound</u>	/definitions	email sms
30004	<u>MCMS_UTM_Runtime_ObjectAlreadyExists</u>	/definitions	email sms
40000	<u>MCMS_UTM_Authentication</u>	/definitions	email sms
40001	<u>MCMS_UTM_Authentication_NotAuthenticated</u>	/definitions	email sms
109001	<u>MCMS_UTM_ResponsesHasErrors</u>	/messages	email sms
109101	<u>MCMS_UTM_Validation_TooManySubscribers</u>	/messages	email sms
109102	<u>MCMS_UTM_Validation_NoSubscribers</u>	/messages	email sms
109103	<u>MCMS_UTM_Validation_NoValidSubscribers</u>	/messages	email sms
109104	<u>MCMS_UTM_Validation_MissingRequiredField_messageKey.</u>	/messages	email sms
109105	<u>MCMS_UTM_Validation_MissingRequiredField_subscriberKey.</u>	/messages	email sms

# Error Code	Aa Error Message	Message-specific or definition (Resource)	Channel
109106	<u>MCMS_UTM_Validation_MissingRequiredField_to</u>	/messages	email sms
109107	<u>MCMS_UTM_Validation_MissingRequiredField_definitionKey</u>	/definitions	email sms
109108	<u>MCMS_UTM_Validation_InvalidValue_messageKey</u>	/messages	email sms
109109	<u>MCMS_UTM_Validation_InvalidValue_subscriberKey</u>	/messages	email sms
109110	<u>MCMS_UTM_Validation_DuplicateMessageKey</u>	/messages	email sms
109111	<u>MCMS_UTM_Validation_InvalidValue_definitionKey</u>	/messages	email sms
109112	<u>MCMS_UTM_Validation_InvalidValue_options</u>	/definitions	email sms
109113	<u>MCMS_UTM_Validation_InvalidValue_autoAddSubscriber</u>	/definitions	email sms
109114	<u>MCMS_UTM_Validation_InvalidValue_resubscribe</u>	/definitions	email sms
109115	<u>MCMS_UTM_Validation_InvalidValue_keyword</u>	/definitions	sms
109116	<u>MCMS_UTM_Validation_InvalidValue_resubscribe_true_without_keyword_specified</u>	/definitions	sms
109120	<u>MCMS_UTM_Validation_MissingRequiredField_keyword</u>	/definitions	sms
109121	<u>MCMS_UTM_Validation_InvalidValue_to</u>	/messages	email sms
109122	<u>MCMS_UTM_Validation_UnknownFieldName</u>	/messages	email sms
109123	<u>MCMS_UTM_Validation_InvalidValue_classification_IncorrectType</u>	/definitions	email
109124	<u>MCMS_UTM_Validation_email_does_not_pass_validation</u>	/definitions	email
109125	<u>MCMS_UTM_Validation_InvalidValue_updateSubscriber</u>	/definitions	email sms

Contents

▼ Transactional Send Subscriber Error Codes



Transactional Send Subscriber Error Codes



A subscriber error code occurs when a message can't be delivered due to an incorrect email address or other incorrect user setup.

Subscriber Error Codes

# Error Code	 Error Message	 SYSTEM ERROR	 Explanation
1	<u>Unsubscribed</u>	The subscriber ExactTarget system status is unsubscribed	Email subscriber's status in Marketing Cloud is unsubscribed
2	<u>Held</u>	The subscriber ExactTarget system status is held	Email subscriber's status in Marketing Cloud is Held
3	<u>PartnerUnsubscribed</u>	The subscriber partner system status is unsubscribed	Email subscriber's status in the partner system is unsubscribed
4	<u>MissingEmailAddress</u>	The subscriber has no email address	Email subscriber doesn't have an email address
5	<u>InvalidEmailAddress</u>	The subscriber email address is invalid	Subscriber's email address isn't valid
6	<u>DuplicateEmail</u>	The subscriber email is the same as that of another job subscriber	Subscriber's email address matches another subscriber's
7	<u>InvalidSubscriberIDProvided</u>	The subscriber ID provided is invalid	Invalid subscriber ID
8	<u>MissingSubscriberID</u>	The subscriber key and subscriber ID are missing	Missing the subscriber key and subscriber ID
9	<u>MissingOwnerID</u>	This is an enterprise account and the ID for the child client account is missing	Missing the ID for the child client account in this enterprise account
10	<u>MissingRequiredAttributes</u>	There are required attribute values missing for the subscriber	Missing required attributes for the subscriber
11	<u>NoAllSubscribersListEntry</u>	The subscriber is not found on the AllSubscribers list for the account	Subscriber isn't on the AllSubscribers list for the account
12	<u>InvalidOwnerIDProvided</u>	The child client account is not a member of the enterprise account	Child client account isn't a member of the enterprise account
13	<u>SubscriberKeyMismatch</u>	The subscriber key in the ExactTarget system does not match the subscriber key provided	Marketing Cloud subscriber key doesn't match the subscriber key provided. Check the subscriber key you provided against the Marketing Cloud subscriber key and try again.
14	<u>EmailAddressMismatch</u>	The email address in the ExactTarget system does not match the email address provided	Email address in Marketing Cloud doesn't match the email address provided. Check the email address you provided against the one in Marketing Cloud and try again.
15	<u>UnspecifiedError</u>	The subscriber did not pass validation. Please contact your customer service representative.	Subscriber didn't pass validation. Contact your Marketing Cloud account representative.
16	<u>InvalidAttributeValue</u>	The value specified for a subscriber attribute does not match the attribute type	Subscriber attribute value doesn't match the attribute type
17	<u>AttributeValueMaxLengthErr</u>	The value specified for a subscriber attribute is over the maximum length allowed for the attribute	Subscriber attribute is too long for the given attribute size limit
18	<u>InvalidAttributeValueCount</u>	The number of subscriber attribute values does not match the expected count	Number of subscriber attributes doesn't match the expected count
19	<u>MissingRequiredFields</u>	There are required data extension fields missing for the subscriber	Missing required data extension fields for the subscriber
20	<u>InvalidFieldValue</u>	The value specified for a data extension field does not match the field type	Data extension field value doesn't match the field type
21	<u>DuplicateDataExtensionRow</u>	A duplicate row cannot be inserted into the Triggered Send data extension	You can't insert a duplicate row into the triggered send data extension
22	<u>DataExtensionInsertFailed</u>	Failed to insert a row into the Triggered Send data extension	Failed to insert a row into the triggered send data extension

# Error Code	 Error Message	 SYSTEM ERROR	 Explanation
23	<u>DomainExclusion</u>	The subscriber was excluded by a domain exclusion list	Subscriber excluded by a domain exclusion list
24	<u>ListDetectiveExclusion</u>	The subscriber was excluded by List Detective	Subscriber excluded by List Detective
25	<u>SubscriberBlackedOut</u>	The subscriber is currently blacked out and the Triggered Send will be rescheduled	Subscriber is not available on their email calendar. The triggered send will be rescheduled.
26	<u>BuildEmailError</u>	An error occurred when attempting to build an email for the subscriber	An error occurred when attempting to build an email for the subscriber
27	<u>SuppressionListExclusion</u>	The subscriber was excluded by a suppression list	Subscriber excluded by a suppression list
28	<u>OptOutListExclusion</u>	The subscriber was excluded by an opt out list	Subscriber excluded by an opt out list
29	<u>MissingSubscriberKeyValue</u>	The subscriber key value is null or empty	Subscriber key value is null or empty
30	<u>SubscriberKeyTooLong</u>	The subscriber key value exceeds the maximum length	Subscriber key value is too long. Enter a key that is X or fewer characters.
31	<u>ListLevelOptOut</u>	The subscriber has opted out of the list	Subscriber opted out of the list.
32	<u>MissingSendGroupID</u>	The required SendGroupID was not specified for the subscriber	Missing required SendGroupID for the subscriber
33	<u>AccountLevelOptOut</u>	The subscriber is master unsubscribed at the account, enterprise or business unit level	Subscriber is master unsubscribed at the account, enterprise or business unit level
34	<u>MissingMessageFileName</u>	The SMTP subscriber message file name is missing	Missing SMTP subscriber message file name
35	<u>InvalidEmailAddressID</u>	The EmailAddressID for the subscriber supporting data privacy is missing or invalid	Missing or invalid EmailAddressID for the subscriber supporting data privacy
36	<u>InvalidBusinessUnit</u>	The subscriber is not a member of the enterprise business unit and has been excluded by the business unit filter	Subscriber isn't a member of the enterprise business unit and is excluded by the business unit filter
37	<u>QueuedTransactionDeleted</u>	Queued Transaction for a Message has been deleted	Message's queued transaction was deleted
38	<u>InvalidSalesforceID</u>	The subscriber has a subscriber key that appears to be an invalid SalesforceID. Case sensitive SFID values cannot be used as Subscriber Key values since these are not case sensitive.	Invalid SalesforceID for the subscriber key. Don't use case-sensitive SFID values because SalesforceIDs aren't case sensitive.
39	<u>SubscriberQueueCleared</u>	Queued unsent subscribers were cleared from the queue.	Queued unsent subscribers are cleared from the queue.
40	<u>SubscriberDeleted</u>	The subscriber or contact has been deleted and cannot be added back to the system.	Subscriber or contact is deleted and can't be added to Marketing Cloud.
41	<u>DuplicateSubscriber</u>	The subscriber is a duplicate of a recent send.	Subscriber is a duplicate of a recent send.
42	<u>RestrictedForProcessing</u>	The subscriber or contact has been restricted for processing	Subscriber or contact is restricted for processing
43	<u>SubscriberRequestExpired</u>	The triggered send request to send to the subscriber or contact is expired	Expired triggered send request to the subscriber or contact
100	<u>Error</u>	Error when building email for subscriber	An error occurred when building the email for the subscriber.
101	<u>Failure</u>	Failure when sending email to subscriber	Failed to send email to the subscriber.
102	<u>Excluded</u>	Excluded by send time filter.	Excluded by the send-time filter.

# Error Code	 Error Message	 SYSTEM ERROR	 Explanation
103	<u>MessageBuildError</u>	An error occurred when building the subscriber message.	An error occurred when building the subscriber message.
104	<u>RecursiveScriptError</u>	A submitted script contains a self-reference that could lead to an infinite recursion.	A submitted script contains a self-reference that can lead to an infinite recursion.
105	<u>InvalidSendTimeDataError</u>	The send time data XML is invalid.	Invalid send-time data XML file.
106	<u>MissingSendDataExtensionSourceRowError</u>	The source row for the subscriber in the data extension send source is missing.	Missing source row for the subscriber in the data extension send source.
107	<u>MissingSubscriberDataError</u>	The Members <i>source row is missing</i> .	Missing members source row.
108	<u>MissingListSubscriberDataError</u>	The tblListSub source row is missing.	Missing tblListSub source row.
109	<u>MissingSubscriberKeyError</u>	The subscriber key is missing.	Missing subscriber key.
110	<u>InvalidEmailAddressError</u>	The email address for the subscriber is invalid.	Invalid subscriber email address.
111	<u>ScriptRaiseError</u>	The subscriber has been excluded by a RaiseError function call.	Subscriber is excluded by a RaiseError function call.
112	<u>EmptyHTTPGetReturnError</u>	An HTTPGet call for the subscriber returned an empty result.	An HTTPGet request for the subscriber returned an empty result.
113	<u>EmptyHTTPGetFunctionReturnError</u>	An HTTPGet function call for the subscriber returned an empty result.	An HTTPGet function request for the subscriber returned an empty result.
114	<u>MissingSFIDError</u>	The Salesforce ID for the subscriber is missing.	Missing subscriber's Salesforce ID.
115	<u>MissingSFDataError</u>	The Salesforce data for the subscriber is missing.	Missing subscriber's Salesforce data.
116	<u>MissingSFSubscriberDataError</u>	The SF Subscriber row for the subscriber is missing.	Missing subscriber's SF Subscriber row.
117	<u>CouldNotResolveSecureEmailToken</u>	The account is configured for secure email addresses, but the secure email address token for this subscriber could not be resolved.	The account is configured for secure email addresses, but the secure email address token for the subscriber could not be resolved.
118	<u>GlobalUnsub</u>	The email address is on the Global Unsub list.	Email address is on the Global Unsub list.
119	<u>PartnerUnsub</u>	The email address is on the Partner Unsub list.	Email address is on the Partner Unsub list.
120	<u>ListDetectiveExclusion</u>	The email address was excluded by List Detective.	Email address was excluded by List Detective.
121	<u>ProcessedByPairedMember</u>	The alternate job in the highly available job pair processed the subscriber.	The alternate job in the highly available job pair processed the subscriber.
122	<u>SMTPFileNotFound</u>	The file containing the SMTP message for this subscriber was not found.	Can't find file containing the subscriber's SMTP message.
123	<u>SMTPFileIOError</u>	The file containing the SMTP message for this subscriber could not be read.	Can't read file containing the subscriber's SMTP message.
124	<u>SMTPFileFormatError</u>	The file containing the SMTP message for this subscriber is in an invalid format.	Invalid format for the file containing the subscriber's SMTP message.
125	<u>DecryptionFailed</u>	An encrypted field on the sendable data extension could not be decrypted.	Can't decrypt an encrypted field on the sendable data extension.
126	<u>MessageVolumeLimitExceeded</u>	Account email volume limit has been exceeded and subscriber will not be processed.	Exceeded account email volume limit. Can't process subscriber.
127	<u>EmptySubject</u>	The resolved email subject for this subscriber is empty.	The resolved email subject for the subscriber is empty.

# Error Code	 Error Message	 SYSTEM ERROR	 Explanation
128	ResolvedEmailBodyTooShort	The resolved email body for this subscriber is too short.	The resolved email body for the subscriber is too short.
129	PhoneNumberValidationFailed	The phone number failed validation at send time.	Phone number failed validation at send time.
130	PayloadExceedsMaximum	The payload exceeds maximum.	Payload is too large. Ensure that the payload is less than the maximum data amount specified by your admin.
131	LinkDataExceedsMaximumSize	The compressed job subscriber link data exceeds the Link database storage capacity of 256k.	Compressed job subscriber link data is too large. Link database storage capacity is 256,000.
132	InvalidCCEmailAddressError	The resolved CC email address for the subscriber is invalid.	Invalid resolved CC email address for the subscriber.
133	InvalidBCCEmailAddressError	The resolved BCC email address for the subscriber is invalid.	Invalid resolved BCC email address for the subscriber.
134	SubscriberDeleted	The subscriber or contact is being processed for deletion and cannot be sent to.	Can't send because subscriber or contact is being processed for deletion.
135	RestrictedForProcessing	The subscriber or contact is restricted for processing and cannot be sent to.	Can't send because subscriber or contact is restricted for processing.
136	SubscriberKeyMismatch	The subscriber or contact subscriber key no longer matches the send data extension subscriber key value.	Subscriber key or contact key doesn't match the send data extension subscriber key.
137	SubscriberIDMismatch	The subscriber or contact subscriber id no longer matches the send data extension subscriber id value.	Subscriber ID or contact ID doesn't match the send data extension subscriber ID.
138	ExceededExpirationPolicy	The triggered send request to send to the subscriber or contact is expired.	Expired triggered send request to the subscriber or contact.
139	TimedoutInQueue	The triggered send request to send to the subscriber or contact is timed out waiting in the queue.	Triggered send request to the subscriber or contact has timed out while waiting in the queue.
140	UnverifiedFromEmailAddress	The from email address is unverified.	Unverified From email address.
141	V2TriggeredSendDeletedFromQueue	The subscriber was deleted by the CLEAR_TS_QUEUE slot worker.	Subscriber deleted by the CLEAR_TS_QUEUE slot worker.
1000	UnsubscribedList	Subscriber excluded from send because unsubscribed from list.	Subscriber excluded from the send because subscriber is unsubscribed from the list.
1010	UnsubscribedMaster	Subscriber excluded from send because unsubscribed from client all subscriber list.	Subscriber excluded from the send because subscriber is unsubscribed from the client All Subscribers list.
1020	UnsubscribedGlobal	Subscriber excluded from send because unsubscribed from ExactTarget globally.	Subscriber excluded from the send because subscriber is globally unsubscribed from Marketing Cloud.
1030	Held	Subscriber excluded from send because held.	Subscriber excluded from send because subscriber is in held status.
1040	Deleted	Subscriber excluded from send because deleted.	Subscriber excluded from send because subscriber was deleted.

▼ REST API v1 Reference

REST API v1 Reference

Address

- [Address: Validate Email](#)

Assets

- [Assets: Simple Query](#)
- [Assets: Advanced Query](#)
- [Assets: Use Case Examples for Enhanced Content Search](#)
- [Assets: Create Category](#)
- [Assets: Get Categories](#)
- [Assets: Delete Asset](#)
- [Assets: Get Category by ID](#)
- [Assets: Update Category by ID](#)
- [Assets: Delete Category by ID](#)

Audit

- [Audit: Get Audit Events](#)
- [Audit: Get Security Events](#)

Auth

- [Auth: Get Base URLs](#)
- [Auth: Get User Info](#)

Automation Studio

- [Automation Studio: Create Trigger \(Beta\)](#)
- [Automation Studio: Check Triggers Status \(Beta\)](#)

Campaigns

- [Campaigns: Create Campaign](#)
- [Campaigns: Get Campaign Collection](#)
- [Campaigns: Get Campaign](#)
- [Campaigns: Delete Campaign](#)
- [Campaigns: Associate Asset to Campaign](#)
- [Campaigns: Get Collection of Campaign Assets](#)
- [Campaigns: Get Campaign Asset](#)
- [Campaigns: Unassociate Asset from Campaign](#)

Chat Messaging

- [Chat Messaging: Send Message to Recipient](#)
- [Chat Messaging: Retrieve Sent Message by MessageId](#)
- [Chat Messaging: Register OTT Property](#)
- [Chat Messaging: Retrieve Registration Property by Platform and Id](#)

- [Chat Messaging: Retrieve Registration Collection by Platform](#)

Contacts

- [Contacts: Get Schemas Collection](#)
- [Contacts: Create Contacts](#)
- [Contacts: Update Contacts](#)
- [Contacts: Insert Attribute Values by ID](#)
- [Contacts: Update Attribute Values by ID](#)
- [Contacts: Get Contact Key for Email Addresses](#)
- [Contacts: Search Attribute Sets by Name](#)
- [Contacts: Search Attribute Groups by Schema](#)
- [Contacts: Get Custom Object Info](#)
- [Contacts: Search Attribute Group ID by Schema](#)
- [Contacts: Search Attribute Set Definitions](#)
- [Contacts: Get or Create Contacts by Key and Type](#)
- [Contacts: Configure Settings for Deleting Contacts](#)
- [Contacts: Delete Contacts by ID](#)
- [Contacts: Delete Contacts by Key](#)
- [Contacts: Delete Contacts by List Reference](#)
- [Contacts: Get Status of the Contact Delete Operation](#)
- [Contacts: Get Contact Delete Requests Details](#)
- [Contacts: Get Contact Delete Requests Summary](#)
- [Contacts: Retry Failed Delete Request](#)
- [Contacts: Restrict Contacts by ID](#)
- [Contacts: Restrict Contacts by Key](#)
- [Contacts: Restrict Contacts by List Reference](#)
- [Contacts: Get Status of the Contact Restrict Operation](#)
- [Contacts: Retry Failed Restrict Request](#)
- [Contacts: Remove Contact from a Journey](#)
- [Contacts: Get Contact's Exit Status from a Journey](#)
- [Contacts: Get List of Journeys the Contact is in](#)
- [Contacts: Add Contact Preferences](#)
- [Contacts: Add Contact Preferences by Contact ID](#)
- [Contacts: Get Contact Preferences by Contact ID](#)
- [Contacts: Search Contact Preferences by Reference Type](#)
- [Contacts: Get Contact Preferences by Contact Key](#)

Data Events

- [Data Events: When to Use Asynchronous Processing](#)

Domain Verification

- [Domain Verification: Domain Verification Bulk Insert](#)
- [Domain Verification: Retrieve List of Verified Domains](#)
- [Domain Verification: Create Record in an Authenticated Domain](#)
- [Domain Verification: Update Record in an Authenticated Domain](#)
- [Domain Verification: Delete User Domain From Addresses](#)
- [Domain Verification: Check if Domain Exists](#)
- [Domain Verification: Register Domain for Validation](#)
- [Domain Verification: Registered Domain Inheritance](#)

Event Notification

- [Event Notification: Create Callback](#)
- [Event Notification: Verify Callback](#)
- [Event Notification: Get Callback](#)
- [Event Notification: Update Callback](#)
- [Event Notification: Delete Callback](#)
- [Event Notification: Get All Callbacks](#)
- [Event Notification: Create Subscription](#)
- [Event Notification: Delete Subscription](#)
- [Event Notification: Get Subscription](#)
- [Event Notification: Get All Subscriptions for a Callback](#)
- [Event Notification: Update Subscription](#)
- [Event Notification: Regenerate Signature Key](#)

Interaction

- [Interaction: Retrieve REST Discovery Document](#)
- [Interaction: Insert Journey](#)
- [Interaction: Search Journeys](#)
- [Interaction: Update Journey Version](#)
- [Interaction: Create Event Definition](#)
- [Interaction: Fire the Event](#)
- [Interaction: Get Journey](#)
- [Interaction: Delete Journey](#)
- [Interaction: Get Journey Audit Log](#)
- [Interaction: Get Publish Status](#)
- [Interaction: Stop a Journey](#)
- [Interaction: Pause a Journey by Definition ID](#)
- [Interaction: Pause a Journey by Definition Key](#)
- [Interaction: Resume a Journey by Definition ID](#)

- [Interaction: Resume a Journey by Definition Key](#)
- [Interaction: Publish a Journey Version](#)
- [Interaction: Remove Contact from a Journey](#)
- [Interaction: Get the Status of Removing a Contact from a Journey](#)

Messaging

- [Messaging: Send Email](#)
- [Messaging: Get Email Delivery Details](#)

Objects

- [Objects: Create Nested Tags](#)
- [Objects: Get Nested Tags](#)
- [Objects: Get Nested Tag By ID](#)
- [Objects: Update Nested Tags](#)
- [Objects: Patch Nested Tags](#)
- [Objects: Delete Nested Tags](#)
- [Objects: Create Tags](#)
- [Objects: Delete Tags](#)

Platform

- [Platform: Get Endpoints](#)
- [Platform: Get Token Context](#)
- [Platform: Get Endpoint](#)
- [Platform: Get Keys](#)
- [Platform: Create or Import Key](#)
- [Platform: Get Key Types](#)
- [Platform: Get Wrapping Key](#)
- [Platform: Get Key By ID](#)
- [Platform: Delete Key By Key ID](#)

Push

- [Push: Create Push Message](#)
- [Push: Get Push Messages](#)
- [Push: Create Location](#)
- [Push: Get Locations](#)
- [Push: Get App Info](#)
- [Push: Update Push Message](#)
- [Push: Delete Push Message](#)
- [Push: Get Push Message](#)
- [Push: Get Specific Location](#)

- [Push: Update Location](#)
- [Push: Delete Location](#)
- [Push: Get Custom Keys](#)
- [Push: Update Custom Keys](#)
- [Push: Delete Custom Keys](#)
- [Push: Refresh List](#)
- [Push: Send Message to All](#)
- [Push: Send Message to Tagged Users](#)
- [Push: Send Message to List](#)
- [Push: Update Custom Key](#)
- [Push: Delete Custom Key](#)
- [Push: Send Message to Mobile Devices in Batch](#)
- [Push: Send Message to Mobile Devices](#)
- [Push: Get Refresh List Status](#)
- [Push: Get Delivery Status of MessageApp](#)
- [Push: Get Delivery Status of MessageTag](#)
- [Push: Get Delivery Status of MessageList](#)
- [Push: Get Delivery Status of MessageContact](#)

Seed-List

- [Seed-List: Seed-List Management](#)
- [Seed-List: Create a Seed-List](#)
- [Seed-List: Read an Account's Seed-Lists](#)
- [Seed-List: Read a Single Seed-List](#)
- [Seed-List: Update a Seed-List](#)
- [Seed-List: Delete a Seed-List](#)
- [Seed-List: Update an Account for a Seed-List](#)
- [Seed-List: Read Account Data for a SeedList](#)

Setup

- [Setup: File Location Management](#)
- [Setup: Create a File Location](#)
- [Setup: Get All File Locations](#)
- [Setup: Get a Single File Location](#)
- [Setup: Update a File Location](#)
- [Setup: Delete a File Location](#)

SMS

- [SMS: Create Keyword](#)
- [SMS: Queue MO Message](#)

- [SMS: Delete Keyword by ID](#)
- [SMS: Post Message to List](#)
- [SMS: Import and Send Message](#)
- [SMS: Get Subscription Status](#)
- [SMS: Post Message to Number](#)
- [SMS: Get Tracking History of Queued MO](#)
- [SMS: Queue Contact Import](#)
- [SMS: Refresh List](#)
- [SMS: Delete Keyword by Longcode](#)
- [SMS: Get Delivery Status of Queued MO](#)
- [SMS: Get Message List Status](#)
- [SMS: Get ImportSend Status](#)
- [SMS: Create ImportSend Delivery Report](#)
- [SMS: Create MessageList Delivery Report](#)
- [SMS: Get Refresh List Status](#)
- [SMS: Get Import Status](#)
- [SMS: Delete Keyword by Shortcode](#)
- [SMS: Get Message Contact Status](#)
- [SMS: Get Message Contact History](#)

Transactional Messaging Email

- [Transactional Messaging: Email - Get Send Definition by Key](#)
- [Transactional Messaging: Email - Create Send Definition](#)
- [Transactional Messaging: Email - Update Message Definition](#)
- [Transactional Messaging: Email - Delete Message Definition](#)
- [Transactional Messaging: Email - Get a List of Definitions](#)
- [Transactional Messaging: Email - Get Delete Options](#)
- [Transactional Messaging: Email - Get Queue Metrics for a Definition](#)
- [Transactional Messaging: Email - Delete Messages Queued for a Definition](#)
- [Transactional Messaging: Email - Send a Message to Multiple Recipients](#)
- [Transactional Messaging: Email - Send a Message to a Single Recipient](#)
- [Transactional Messaging: Email - Get Message Send Status for Recipient](#)
- [Transactional Messaging: Email - Get List of Messages Not Sent to Recipients](#)

Transactional Messaging Push

- [Transactional Messaging: Push - Get Send Definition by Key](#)
- [Transactional Messaging: Push - Create Send Definition](#)
- [Transactional Messaging: Push - Update Message Definition](#)
- [Transactional Messaging: Push - Delete Message Definition](#)

- [Transactional Messaging: Push - Get a List of Definitions](#)
- [Transactional Messaging: Push - Send Push Message](#)
- [Transactional Messaging: Push - Get Message Send Status for Recipients](#)

Transactional Messaging SMS

- [Transactional Messaging: SMS - Get Send Definition by Key](#)
- [Transactional Messaging: SMS - Create Send Definition](#)
- [Transactional Messaging: SMS - Update Message Definition](#)
- [Transactional Messaging: SMS - Delete Message Definition](#)
- [Transactional Messaging: SMS - Get a List of Definitions](#)
- [Transactional Messaging: SMS - Get Queue Metrics for a Definition](#)
- [Transactional Messaging: SMS - Delete Messages Queued for a Definition](#)
- [Transactional Messaging: SMS - Send a Message to Multiple Recipients](#)
- [Transactional Messaging: SMS - Send a Message to a Single Recipient](#)
- [Transactional Messaging: SMS - Get Message Send Status for Recipient](#)
- [Transactional Messaging: SMS - Get List of Messages Not Sent to Recipients](#)

REST API Permission IDs and Scopes

- [REST API Permission IDs and Scopes](#)

▼ Marketing Cloud Data Management and Analytics pdf

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/36d5243f-ee7d-4958-9ac6-98e21c69306e/mc_data_management_and_analytics.pdf

▼ Table of Contents

Marketing Cloud Einstein and Analytics	1
Analytics Builder	1
Collect Tracking Code	50
Intelligence Reports for Engagement	63
Google Analytics Integration for Marketing Cloud	172
Use Einstein to Improve Your Marketing Efforts	186

Salesforce Audience Studio and Data Studio	339
Index	341

▼ MARKETING CLOUD EINSTEIN AND ANALYTICS

Import and manage data from multiple sources to gain deep insights into the behaviors and interests of your contacts across channels.

Use these insights to set marketing goals and refine customer journeys.

Analytics Builder

Analytics Builder enables you to gain deep insights into the behaviors and interests of your contacts across channels. Use these insights to set marketing goals and refine customer journeys.

Collect Tracking Code

Collect Tracking Code is a JavaScript snippet you can use to capture data about known contacts and unknown visitor behavior. You

can use this data to enhance Contact profiles and optimize customer journeys across channels through predictive and personalized

recommendations. You can also update product and content catalog details in real time or to track email-driven revenue.

Intelligence Reports for Engagement

Intelligence Reports for Engagement enables you to generate, view, and share a detailed analysis of your Email, Push, and Journey

campaign-level data. Intelligence Reports for Engagement Advanced, a premium upgrade of Intelligence Reports, adds more features,

functionality, and analytics. The Advanced upgrade is available at an additional cost.

Google Analytics Integration for Marketing Cloud

This integration lets you track and analyze journey activity through your Google Analytics account without assistance from technical

support or services. You determine the URL tags and how data is collected and presented in the customizable Google Analytics reporting dashboard. Use Google's native authentication to securely link your Marketing Cloud and Google Analytics 360 or

Google

Analytics Free accounts.

Use Einstein to Improve Your Marketing Efforts

What is Marketing Cloud Einstein? Einstein for Marketing Cloud provides insights and data to inform the content and timing of your

marketing activities. Use powerful artificial intelligence (AI) to understand your customers' engagement and behavior so you can personalize every interaction.

Salesforce Audience Studio and Data Studio

View Salesforce Audience Studio and Data Studio documentation.

▼ Analytics Builder

Analytics Builder

Analytics Builder enables you to gain deep insights into the behaviors and interests of your contacts across channels. Use these insights

to set marketing goals and refine customer journeys.

With Analytics Builder, you can:

- Measure engagement and performance of your email campaigns.
- Configure dashboard views to visualize data in flexible tables, charts, and graphs.
- Use Collect to view contact behaviors such as clicks and purchases.

- Create targeted audience based on predictive behavior models.
- Track each interaction in a journey and measure progress towards your target in a single interface within Journey Builder.

1

Standard Reports for Marketing Cloud Data Management and Analytics

Marketing Cloud provides dozens of standard reports to track the effectiveness of your campaigns. You can run reports manually or

schedule them to run automatically. You can also choose who receives the reports.

SEE ALSO:

Standard Reports for Marketing Cloud Data Management and Analytics

▼ Standard Reports for Marketing Cloud Data Management and Analytics

Standard Reports for Marketing Cloud Data Management and Analytics

Marketing Cloud provides dozens of standard reports to track the effectiveness of your campaigns. You can run reports manually or

schedule them to run automatically. You can also choose who receives the reports.

Note: As of May 25, 2018, email reports that reference individual subscribers no longer include subscribers who were deleted using the legacy Email Studio delete functionality.

Note: After a user's account is deactivated, the standard and discover reports that they created aren't accessible. Recreate the reports.

A sending report failure is often due the attached report being larger than the maximum size.

1. To access the Reports feature, navigate to Analytics Builder and select Reports.
2. Select one of the standard reports:
 - Email Studio
 - Journey Builder
 - MobileConnect
 - MobilePush
 - GroupConnect
 - Microsites
 - CloudPages
 - Contacts
 - MC Administration
3. Ensure that your report doesn't exceed the maximum size of 6–7 MB. If so, use the Enhanced FTP option to send to recipients.

Get Started with Reports

The Reports Overview screen provides access to all reports available for Email Studio, Mobile Studio, CloudPages, SocialPages, Microsites, JourneyBuilder, and Contact Counts.

SEE ALSO:

Contacts Count Report in Analytics Builder

Microsite Tracking

Email Studio Reports in Analytics Builder

Journey Builder Reports

Mobile Studio Reports

▼ Get Started with Reports

Get Started with Reports

The Reports Overview screen provides access to all reports available for Email Studio, Mobile Studio, CloudPages, SocialPages, Microsites, JourneyBuilder, and Contact Counts.

The Reports Overview screen includes the following:

- New Reports: Lists the newest reports available in the Marketing Cloud.
- Reports Shortcut: You can create and manage shortcuts to frequently used reports. Click Add Shortcut or Select Shortcuts to create

a new shortcut or click the x next to the shortcut to delete it.

- **Reports Catalog:** Lists all available reports. To view a brief description for each report, click View Catalog. To filter by category, click

on that category in the list to view a brief description of each report in that category only. Or you can click a recently run report to view a brief description of that report.

- **Saved Reports Menu:** Click an item to populate the list to right with saved reports within that item's description.

- **Saved Reports Grid:** Lists all reports you have saved including relevant meta-data for each. You can filter and sort your view.

Click the

name of the report to view the saved version or click Run to run a new version of the report.

Note: Discover reports are indicated by a blue cube icon and can only be accessed if you have Discover enabled in your account.

The Catalog screen provides a searchable list of available reports and a brief description of each. Click Create to run a new report.

The Activity Log screen provides detailed lists all reports running today, all scheduled reports, and the history of reports you have run.

This log includes both standard reports and Discover reports. Click View Parameters report setting details.

Create a Standard Report in Analytics Builder

Learn how to create and run a standard report in Marketing Cloud's Report tool.

Marketing Cloud Analytics Builder Reports Frequently Asked Questions

Get the answers to frequently asked questions about the standard reports in Marketing Cloud Analytics Builder.

SEE ALSO:

Contacts Count Report in Analytics Builder

Microsite Tracking

Email Studio Reports in Analytics Builder

Journey Builder Reports

Mobile Studio Reports

▼ Create a Standard Report in Analytics Builder

Create a Standard Report in Analytics Builder

Learn how to create and run a standard report in Marketing Cloud's Report tool.

1. Navigate to Analytics Builder and click Reports.
2. Click View Catalog for a list of available standard report templates.
3. Click Create next to the report you want to run.
4. Set report parameters. Options vary by report.

Tip: When you set a custom date range, the system generates a report based on data as it appears at 12:00 AM on that date.

To include data from the end date, set the end date parameter to the following day. For example, if you want to include data from May 1 through the end of the day on May 17, set May 18 as the end date.

3

Standard Reports for Marketing Cloud Data Management
and Analytics

Marketing Cloud Einstein and Analytics

5. Click Submit.
6. Select an export or save option and complete the required fields.

Note: Where noted, some reports are not available as a web page and require you to select a format for file export. If you want to email reports to more than one recipient, include their email addresses separated by commas.

7. Click Save.

Contacts Count Report in Analytics Builder

The Contact Count report contains information on the total number of contacts in an account, and it runs weekly. The report also

shows the Month End Report that runs on the last day of the month.

Contact Data Portability Report

View information included in the Contact Data Portability report for Contact Builder in Reports.

Microsite Tracking

The Microsite Tracking report displays tracking data from your microsite for a specific time frame you select. You can use this report

to help evaluate the effectiveness of your email campaigns in driving traffic to your landing pages. Compare the likelihood of readers

clicking one link rather than another. This report is available in the Reports tool.

Email Studio Reports in Analytics Builder

View a list and description of available Email Studio reports available in the Reporting tool.

Journey Builder Reports

View a list and description of Journey Builder reports available in the Reports tool.

Mobile Studio Reports

View a list and description of Mobile Studio reports available in the Reports tool.

▼ Contacts Count Report in Analytics Builder

Contacts Count Report in Analytics Builder

The Contact Count report contains information on the total number of contacts in an account, and it runs weekly. The report also shows

the Month End Report that runs on the last day of the month.

Important: To view this report, enable the Access permission under Contacts. Find this permission in the Reports Catalog section

of the Reports permission node

Field Description

Total distinct contacts Number of distinct contacts used for billing purposes, it comprises:

- Count of unique Contact Keys regardless of the channel source or method in which the contact entered Salesforce Marketing Cloud. See Contact Definition and Count Determination in Contact Builder for more information.

- Contact key counts in populations to whom no sends have been attempted yet.

Number of distinct email addresses other than those addresses deleted from the email channel. Unique email addresses used by

Distinct email addresses

more than one subscriber are counted only one time in this subtotal. This subtotal doesn't apply to the formula that computes Total Distinct Contacts

Field Description

Number of mobile device addresses. This count is only for reference and doesn't apply to the formula that computes total distinct contacts.

MobilePush contacts

Number of mobile SMS numbers. This count is only for reference and doesn't apply to the formula that computes total distinct contacts.

MobileConnect contacts

Number of mobile LINE addresses. This count is only for reference and doesn't apply to the formula that computes total distinct contacts.

GroupConnect contacts

Contact key counts in populations to whom no sends have been attempted yet.

Number of contacts in populations

Total number of contacts associated with triggered sends but not associated with the All Subscribers list or another list in the

Contacts from Triggered Sends

triggered send configuration. Contacts from triggered sends that are performed at the child BU level aren't included in Total Distinct Contacts.

Number of leads synchronized from Sales and Service clouds using Synchronized Data Sources in MC Connect.

Leads from Sales and Service Clouds

Number of contacts synchronized from the Sales and Service clouds using Synchronized Data Sources in MC Connect.

Contacts from Sales and Service Clouds

Number of leads converted to contacts in the Sales and Service clouds based on the two previous subtotals.

Overlapping Leads and Contacts from Sales and Service Clouds

Number of leads synchronized from Sales and Service Clouds using version 3 or earlier of the Salesforce Connector.

V2 Leads from Sales and Service Clouds

Number of contacts synchronized from Sales and Service Clouds using version 3 or earlier of the Salesforce Connector.

V2 Contacts from Sales and Service Clouds

Number of leads, contacts, and accounts respectively synchronized from Microsoft Dynamics CRM using the Dynamics CRM connector.

Leads, Contacts, and Accounts from Microsoft Dynamics CRM

Number of leads, contacts, and accounts respectively synchronized from Microsoft Dynamics CRM using the Dynamics CRM connector

▼ Contact Data Portability Report**Contact Data Portability Report**

View information included in the Contact Data Portability report for Contact Builder in Reports.

This report contains information about a single contact record relation to data protection and privacy regulations. Use this report to

provide information to a data subject who requests a copy of their information.

5

Standard Reports for Marketing Cloud Data Management and Analytics

Marketing Cloud Einstein and Analytics

- Subscriber information

- Email statistics
- Subscriber attributes
- MobilePush information and statistics, as applicable
- MobileConnect information and statistics, as applicable
- GroupConnect information, as applicable

Note: Run this report at the top level of your account

▼ Microsite Tracking

Microsite Tracking

The Microsite Tracking report displays tracking data from your microsite for a specific time frame you select. You can use this report to

help evaluate the effectiveness of your email campaigns in driving traffic to your landing pages. Compare the likelihood of readers

clicking one link rather than another. This report is available in the Reports tool.

This five-part report provides details about views and clicks for a microsite you choose over a period you specify:

- Sheet 1: Microsite Activity
- Sheet 2: Landing Page Activity
- Sheet 3: Email Related Activity
- Sheet 4: Web Related Activity
- Sheet 5: Detailed Link Click Tracking

Microsite Activity

Learn about the information presented in this subsection of the Microsites Report in the Reports tool.

Landing Page Activity

Learn about this subsection of the Microsites Tracking report in the Reports tool.

Email Related Activity

Learn about the information contained in this subsection of the Microsites report in the Reports tool.

Web Related Activity

Learn about this subsection of the Microsites Tracking report in the Reports tool.

Detailed Link Click Tracking

Learn about this subsection of the Microsites Tracking report in the Reports tool

Microsite Activity

Microsite Activity

Learn about the information presented in this subsection of the Microsites Report in the Reports tool.

Field Name	Description
Unique Views	Number of unique views received

Field Name	Description
Total Views	Total number of views received
Unique Clicks	Number of unique clicks received
Total Clicks	Total number of clicks received

Landing Page Activity

Landing Page Activity

Learn about this subsection of the Microsites Tracking report in the Reports tool.

Field Name	Description
Page Name	Name of the landing page tracked
Unique Views	Number of unique views received
Total Views	Number of views received
Unique Clicks	Number of unique clicks received
Total Clicks	Number of clicks received

Email Related Activity

Email Related Activity

Learn about the information contained in this subsection of the Microsites report in the Reports tool.

Field Name	Description
Email Name	Name of the email containing a link to the microsite
Job ID	Job ID of the email send tracked in this row
Send Time	Send time of the email.
Unique Views	Number of unique views of a landing page that resulted from this email send. Multiple views by the same person would be counted only once.
Total Views	Number of views of a landing page that resulted from this email send. Multiple views by the same person contribute to this total.
Unique Clicks	Number of unique clicks of links in a landing page that resulted from this email send. Multiple clicks by the same person would be counted only once.
Total Clicks	Number of clicks for links in a landing page that resulted from this email send. Multiple clicks by the same person contribute to this total.

Web Related Activity

Web Related Activity

Learn about this subsection of the Microsites Tracking report in the Reports tool.

7



Marketing Cloud Einstein and Analytics

Standard Reports for Marketing Cloud Data Management and Analytics

Field Name	Description
Page Name	Name of the landing page tracked
Unique Views	Number of unique views from web traffic other than links in emails; multiple views by the same person would be counted only once.
Total Views	Number of views from web traffic other than links in emails; multiple views by the same person contribute to this total.
Unique Clicks	Number of unique clicks on the landing page from web traffic other than links from emails; multiple clicks by the same person would be counted only once.
Total Clicks	Number of clicks on the landing page from web traffic other than links from emails; multiple clicks by the same person contribute to this total.

Detailed Link Click Tracking

Detailed Link Click Tracking
Learn about this subsection of the Microsites Tracking report in the Reports tool.


Field Name	Description
Page name	Name of the landing page tracked
URL	URL for the link clicked on the landing page
Link Alias	Tracking alias for the link clicked on the landing page
Unique Clicks	Number of unique clicks of the link; multiple clicks by the same person would be counted only once.
Total Clicks	Number of clicks for the link; multiple clicks by the same person contribute to the total

Email Studio Reports in Analytics Builder

Email Studio Reports in Analytics Builder
View a list and description of available Email Studio reports available in the Reporting tool.
To create a standard Email Studio report, see [Create a Standard Report in Analytics Builder](#).

Report Name	Description
A/B Test Summary	Contains summary and comparison information about one or more A/B test campaigns for the date range you specify.
Account Send Summary	Displays all potential response counts or rates for an account organized by send. Enterprise or reseller reports also display respective member results organized by account.
Attribute by Tracking Event	Displays how your subscribers reacted to a certain email send, based on a specific tracking event, and a particular attribute.

Report Name	Description
Campaign Email Job Tracking Summary	Displays the email tracking data for each job sent that is associated with the selected campaign. Use this report to get a complete list of all sends in a campaign and a quick summary of all relevant email tracking statistics.
Campaign Email Tracking Summary	Use this report to quickly see email level tracking statistics if the emails in your campaign are sent in more than one job. This report displays the send tracking data for a specific campaign. Displays the email tracking data for each email associated with the campaign. If an email is sent through multiple jobs, Analytics Builder summarizes the tracking data for each job by email.
Carrier Deactivation Summary Report	Use this report to both verify the accuracy of the opt-out process and insight into customers to potentially retarget for SMS opt-in. The report displays a daily and historic overview of the subscriptions updated within your account. The updates are a result of deactivated phone numbers on the carrier's network or numbers ported from one carrier to another.
Conversion Tracking Statistics	Displays all conversion activity across all your email sends over a specific time period.
Email Message Frequency	This report cannot be viewed as a web page. Includes the number of customers received multiple email messages from period to period for comparison purposes. The available periods for comparison are: Yesterday vs. Day Before; Previous 2 Weeks Before Today; Previous Sunday to Sunday Weeks; Last 30 days vs. Previous 30 days; Previous 2 calendar months.
Email Performance by Attribute	Lets you evaluate email send results for different subscriber attribute and preference values.
Email Performance by Domain	Enables you to evaluate email send results for each domain sent to for a single send.
Email Performance by List	Enables you to assess response, bounce, and click-through rate for each list or group. The rate can be sent to as part of a send event that excludes email sent to a single recipient.
Email Performance for All Domains	Enables you to evaluate email send results for each domain sent to for a single send.
Email Performance Over Time	Shows data for all sends during a specific period.
Email Send	Shows how many emails have been sent from each of your accounts. This report can be run for any time period, present, or past.
Email Sends by User	Displays a list of account users who have sent emails during the specified date range and the total number of emails that each user has sent.
Email Send Performance by Audience Builder Segment	Shows send performance data by audience segment.
Forwarding Activity Details	Displays details for emails that are forwarded from subscribers to other individuals.
Impression Tracking by Job	Displays the number of times the content area was sent as part of a job. The count could be the number of impressions and the link performance for links found within the impression region.
Impression Tracking for Triggered Send by Period	Displays the performance of an impression region for triggered sends by a time period you choose.
List Demographics	Enables you to analyze your lists to see a breakdown of subscribers by status, domain, and subscription tool.
List Performance Over Time	Enables you to examine each of your lists to evaluate their effectiveness over time across multiple sends.

Report Name	Description
List Size Over Time	Displays the number of subscribers on a list over a specified time period. Includes a table showing the number of subscribers by status for each month in the specified time period. This data is also provided in a line graph.
Multi-Campaign Tracking Summary	Use this report to compare email performance across multiple campaigns. This report shows that email tracking summarized by campaign for all campaigns with a deployment date. The report shows tracking by the selected date range and the total of all selected campaigns for each tracking category.
Recent Email Sends Summary	Enables you to analyze a summary of your recent account activity.
Region Performance for Triggered Sends Over Time	Displays how a single section of content performs over a period across multiple triggered send jobs.
Region Performance Over Time	Displays how a single section of content performs over a period across multiple jobs.
Response Trend Analysis for Email Send	Displays how an email send has performed over a specified date range.
Send Classification by Email Tracking Report	Enables you to compare the send performance data of transactional and commercial sends.
Sendable Data Extension Demographics	Includes charts which show subscriber status breakdown and a top-level view of which domains are best represented in the selected sendable data extension.
Sendable Data Extension Performance Over Time	Provides an analysis of a sendable data extension over time. You can use this report to evaluate when a previously effective data extension is no longer effective.
Single Email Performance by Device	Summarizes email open and click activity on mobile vs. desktop devices for an email. This report shows the percentage of opens, unique opens, and clicks occurring on mobile devices.
Spam Complaints Over Time	Provides historical data about received spam complaints.
Subscriber Engagement	Displays which subscribers are most or least engaged with emails.
Subscriber Most Recent Activity	Displays a list of all subscribers and details their most recent open or click activity.
Subscribers Not Sent To	Displays a list of subscribers who have not received an email during a specified date range.
Triggered Sends Tracking	Displays detailed tracking data for a single triggered send over a custom time frame.  Note: The overlay option is not available under the Click Activity tab.
Unengaged Subscribers for a List	Displays which subscribers on a specified list are unresponsive after multiple emails sends.

A/B Test Summary

The A/B Test Summary contains summary and comparison information about one or more A/B test campaigns for the date range you specify.

Account Send Summary

Displays all potential response counts or rates for an account organized by send. Enterprise or Reseller reports also display respective member results organized by account.

Attribute by Tracking Event Report for Email Studio in Analytics Builder

Use the Attribute by Tracking Event Report for Email Studio in Analytics Builder to check how your subscribers reacted to a certain email send. The report lists the subscribers who interacted with an email send during the timeframe you select. Choose a subscriber attribute to include in the report. The report works only with values of the chosen attribute at the time of reporting and not the value at the time of send. For example, use this report to evaluate the effectiveness of a campaign in generating subscriber clicks for men versus women.

Campaign Email Tracking

The Campaign Email Tracking report shows send tracking data for a specific campaign.

Carrier Deactivation SMS Summary

Some US wireless carriers produce daily files for phone numbers that have deactivated from their network or been ported from one carrier to another. Marketing Cloud changes the status for all numbers included in that file from Opted-In to Opted-Out.

Conversion Tracking Statistics

Run this report to compare the effectiveness of links within campaigns or the effectiveness of campaigns in number and value of conversions. This report contains a list of conversions for links included in emails across multiple sends during a time frame.

Email Performance by Attribute Report in Analytics Builder

Run this report to view data for a specific subscriber included in a specific send. This report runs against Legacy List-Based Attributes only and doesn't account for the Sendable DEs field. To display the relevant data, give the Sendable DEs field the same name and ensure that the data is within the Legacy List-Based Attributes.

Email Performance by Domain

This report provides data for the top 10 email domains for an email send. Use this report to analyze how domains responded to a particular email send.

Email Performance by List

This report contains graphics to illustrate Emails Sent by List, Open Rate by List, Bounce Rate by List, and Click Through Rate by List.

This report shows only unique opens and clicks. Use this report to analyze which lists responded most favorably to a particular email send.

Email Performance for All Domains

This report shows performance data for each email domain included in sends during a specific period. You can use this analysis to identify domains that are bouncing your email so you can begin to resolve the delivery issue.

Email Performance Over Time

This report shows data for all sends during a specific period. You can use this report to evaluate when a previously effective email is no longer effective.

Email Send Performance by Audience Builder Segment Report

Use the Email Send Performance in Marketing Cloud Audience Builder Segment report to see send performance data by audience segment. To compare the send performance of segments in a segmented audience, run this report.

Email Send

Use the Email Send report to view account activity. This report is often used by Agency users for billing their child accounts. Enterprise users can also use this report to view the activity of their Lock and Publish accounts.

Email Sends by User

Learn about the information provided in the Email Sends by User report.
Forwarding Activity Details

Use this report to analyze forwarding activity for your email sends. You can use this report to enable providing incentives to subscribers whose forwarded emails result in new subscribers.

Track Impressions by Job Report for Email Studio in Analytics Builder

The Impression Tracking by Job report for Email Studio in Marketing Cloud Analytics Builder is two reports combined into one. The

Job Summary section shows overview tracking information. The Impression Tracking Details section lists the different areas of the

report that contained impression tracking and contains data for each area. You can only track emails that use dynamic content or

AMPscript .

Impression Tracking for Triggered Sends Report in Analytics Builder

Use the Impression Tracking for Triggered Sends report for Email Studio in Marketing Cloud Analytics Builder to monitor triggered

sends over a specific time. You can only track emails that use dynamic content or AMPscript.

List Demographics

This report displays the status, subscription method, and domains of the subscribers on a list. Use this report to see what percentage

of your subscribers are using a certain domain.

List Performance Over Time

This report provides statistics for all sends to a list during a specified time period.

List Size Over Time Report for Email Studio in Analytics Builder

To check how your list grows and subscriber statuses change over time, use the List Size Over Time report in Marketing Cloud Analytics Builder. Because the data shows first-time events for a status change, there could be overlap of data. Also, more than one

event or status can occur with a subscriber in the time period you define. The report shows you the most recent status change that

occurred during the specified time.

Multi-Campaign Email Tracking Summary

This report shows the Email Tracking by campaign, for all campaigns during the selected date range. The report also includes a grand

total of all selected campaigns for each tracking category. Use this report to compare the email performance across multiple campaigns.

Recent Email Sending Summary

The Recent Email Sending Summary provides a summary of your account activity over a recent period. You can use this report to

confirm that your jobs were completed successfully.

Region Performance for Triggered Sends Over Time

This report shows how a single section of content performs over time across multiple jobs in Triggered Emails.

Region Performance Over Time

The Region Performance Over Time report shows how a single section of content performs over a period across multiple sends.

You

can use this report to see how a specific link is performing over a certain period. Only emails that contain dynamic content or AMPscript can be tracked in this report.

Response Trend Analysis for an Email Send

Use this report to analyze when subscribers respond to your email relative to when you send it. For example, if you want to send an

announcement of an event at a specific time, you can run this report against other sends to determine how far in advance you must

send the message to ensure that most your subscribers open it.

Send Classification by Email Tracking

This report shows the send performance data of transactional and commercial sends. Use this report when you want to compare

the send performance data of transactional and commercial sends.

Single Email Performance by Device

The Single Email Performance by Device report displays email performance on mobile devices versus desktop devices.

Spam Complaints Over Time

This report provides historical data on the level of spam complaints received.

12

Standard Reports for Marketing Cloud Data Management and Analytics

Marketing Cloud Einstein and Analytics

Subscriber Engagement

Use this report to identify your most engaged subscribers. You could target these subscribers for special offers. This report includes

the number of emails that you have sent each subscriber over a specified period, and the number of emails the subscriber has opened or clicked.

Subscriber Most Recent Activity

The Subscriber Most Recent Activity report shows when each subscriber last opened or clicked a message.

Subscribers Not Sent To

This report shows a list of emails for subscribers who weren't included in a send during a time frame that you select.

Triggered Sends Tracking

The Triggered Sends Tracking report contains standard tracking information for a triggered send over a time period. You can use

this report to determine the effectiveness of a triggered send in producing opens and clicks and not causing unsubscribes.

Unengaged Subscribers for a List

This report shows a list of your most inactive subscribers. Use this information to target certain lists or subscribers or to clean up your lists.