



Customer.io – SMS & CHAT APPS INTEGRATION

How To Get Started

Customer.io is a marketing automation platform which allows you to create data-driven email, push, in-app, SMS or chat apps to engage customers where they are.

It brings customer insights, journey orchestration, and omnichannel communication.

With Mitto integration clients can send **Campaigns and build journeys using our API inside customer.io for SMS and chat apps (such as WhatsApp Business, Viber for Business, etc).** In this guide we will explain how to set it up.

PART 1

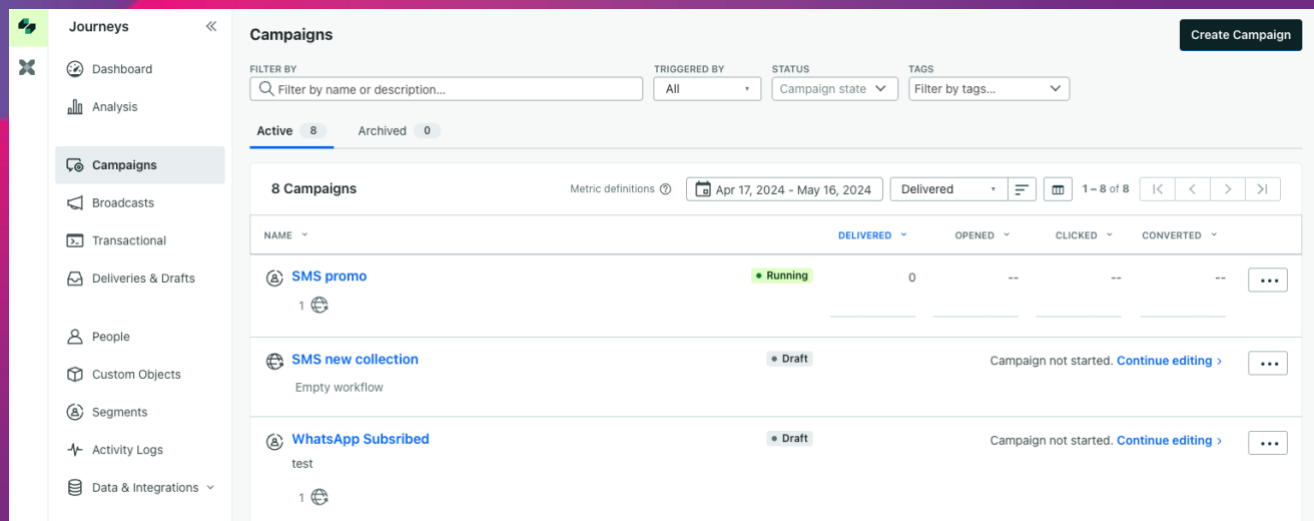
MITTO INTEGRATION

Mitto integrations enable Customer.io users to **send SMS and chat apps messages** to communicate with leads and contacts using their Mitto account.

For sending messages from Customer.io platform, **there is no need for specific installation steps.**

Navigate to Campaigns and start crafting. You will need to **use Mitto API** inside the Customer.io campaign builder **for the channel you prefer.**

- **Guide** – [How to create Mitto account](#) and find API keys
- **Guide** – [How to create WhatsApp templates](#)



The screenshot displays the Customer.io Campaigns management interface. On the left is a navigation sidebar with options like Journeys, Dashboard, Analysis, Campaigns, Broadcasts, Transactional, Deliveries & Drafts, People, Custom Objects, Segments, Activity Logs, and Data & Integrations. The main area is titled 'Campaigns' and includes a 'Create Campaign' button. It features a search bar and filters for 'TRIGGERED BY' (All), 'STATUS' (Campaign state), and 'TAGS' (Filter by tags...). Below the filters, it shows 'Active 8' and 'Archived 0' campaigns. A table lists 8 campaigns with columns for NAME, DELIVERED, OPENED, CLICKED, and CONVERTED. The first campaign, 'SMS promo', is in a 'Running' state with 0 delivered messages. The other two, 'SMS new collection' and 'WhatsApp Subscribed', are in 'Draft' states and have not started.

NAME	DELIVERED	OPENED	CLICKED	CONVERTED
SMS promo	0	--	--	--
SMS new collection				
WhatsApp Subscribed				

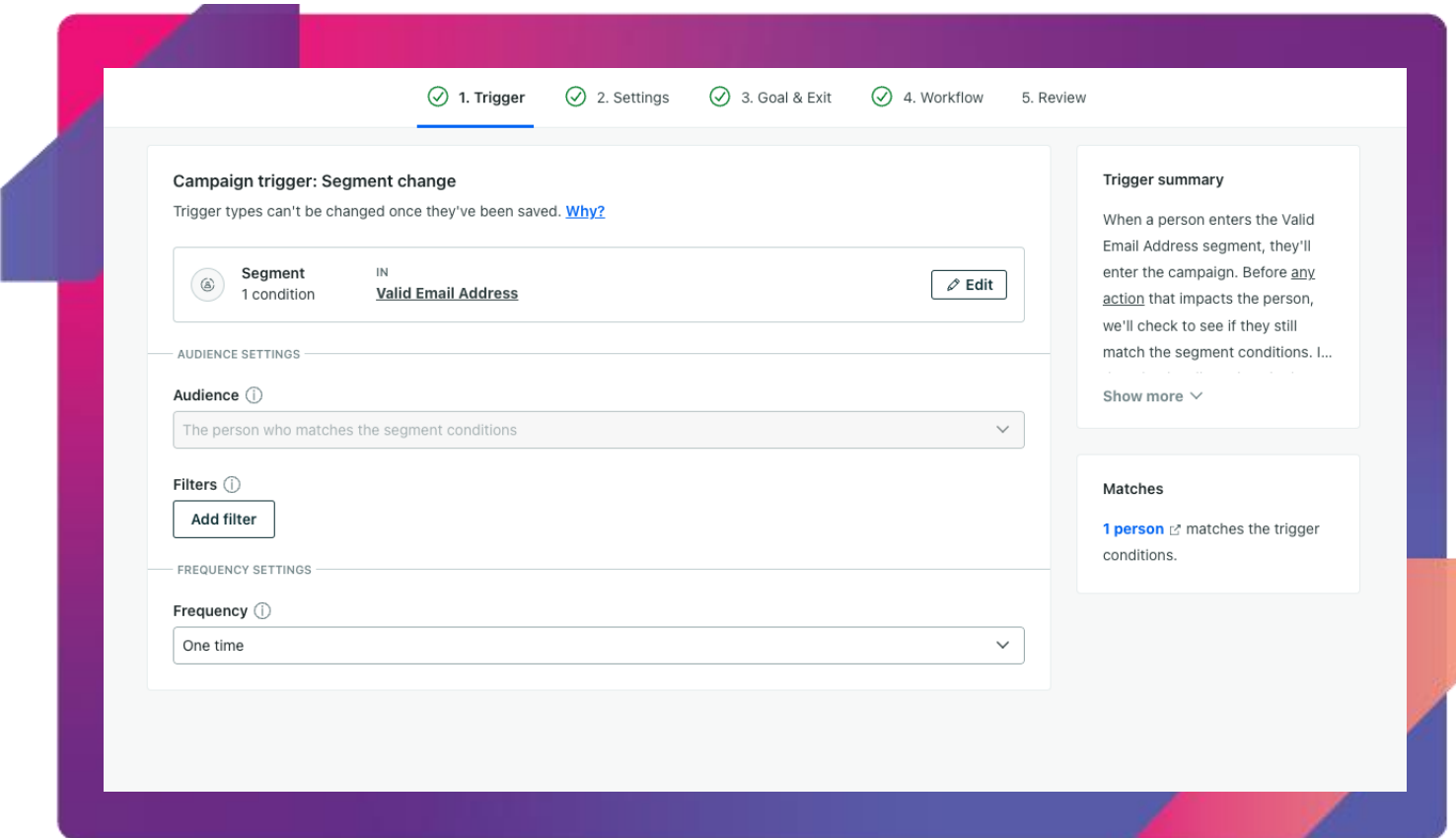
SENDING MESSAGES

Prepare your contacts and lists for campaigns.

- Go to **Segments** – create lists for targeting the right people.

Navigate to Campaigns and click New Campaign. First steps will be the same for all channels.

- First step is to set up **Trigger**. Choose who is your audience and the frequency for entering the campaign.



- Second step is to set up **Message settings** and what type of contacts will get enrolled.

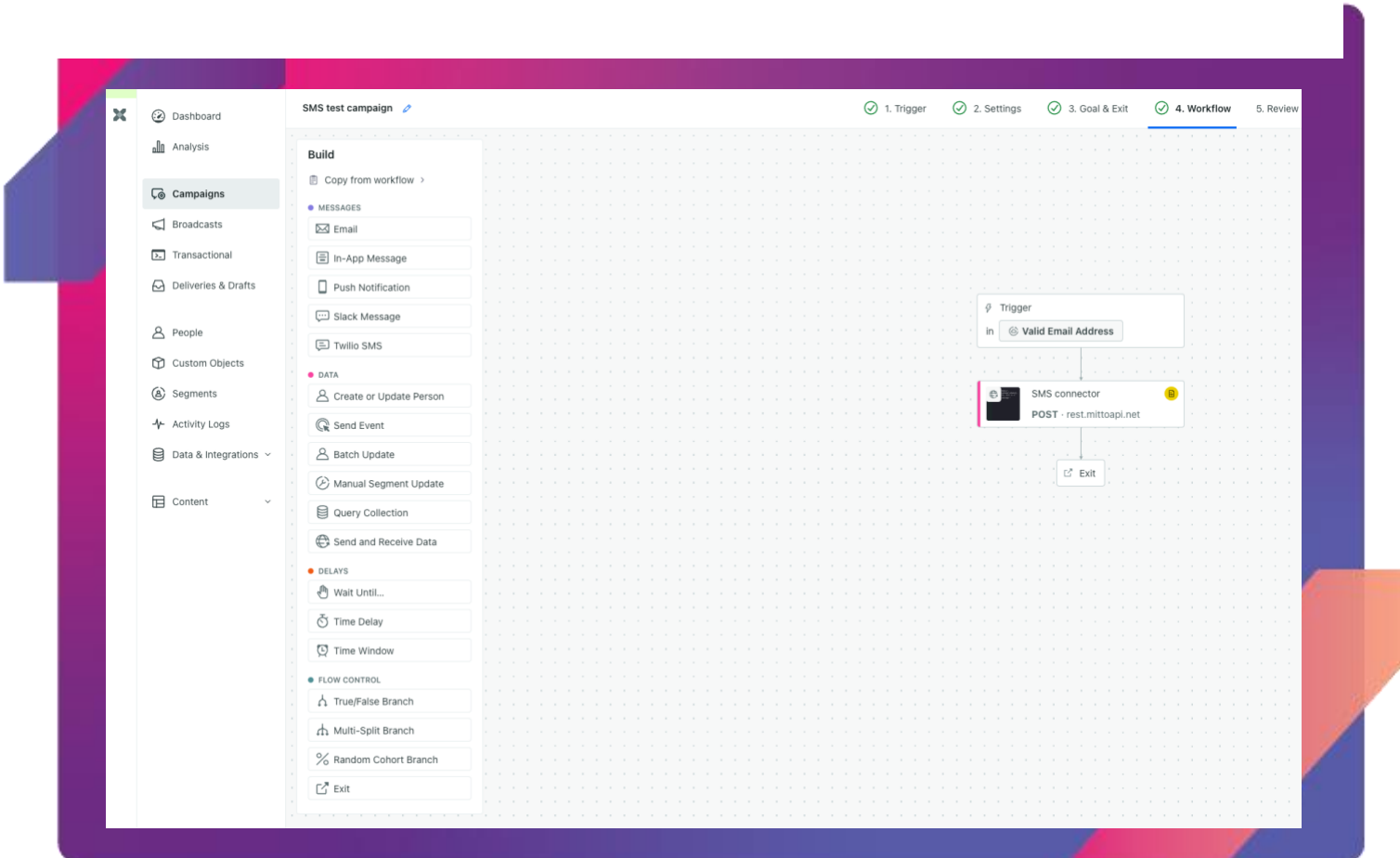
The screenshot shows the 'Message settings' step in a five-step workflow. The steps are: 1. Trigger, 2. Settings (current), 3. Goal & Exit, 4. Workflow, and 5. Review. The 'Message settings' section includes a subtitle 'Choose the default settings for messages in this workflow. You can override them later in each message'. It features a 'Subscription preference' dropdown menu set to 'All subscribed and unsubscribed', a blue informational box with a question mark icon stating 'Make sure to follow local laws about sending to unsubscribed people', and a 'Count toward message limit' toggle switch currently set to 'Off'.

- Third step is to set up **Goal and Exit conditions**.

The screenshot shows the 'Goal and Exit conditions' step in a five-step workflow. The steps are: 1. Trigger, 2. Settings, 3. Goal & Exit (current), 4. Workflow, and 5. Review. The 'Set a goal and conversion criteria' section includes a subtitle 'Goals and conversions help you measure how successful your campaign is in getting your audience to do something.' Below this are two buttons: 'Set goal' (with a target icon) and 'No goal' (with a prohibition sign icon). The 'People exit the campaign early when' section contains four radio button options: 'They achieve the goal', 'They stop matching the trigger segment or filters' (which is selected), 'They achieve the goal or they stop matching the trigger segment or filters', and 'People don't exit early, they move through the entire workflow'.

Now it's time to create the message for your campaign. First we will describe SMS scenario.

- Your workflow is created in this step. To send a message you need to choose **Send and Receive Data** from the **Data** section. Click on that step and edit message.



- Now you need to use Mitto SMS API to set it all up. Choose **POST** from the dropdown and enter link: <https://rest.mittoapi.net/sms>
- Set up headers sections, enter:
 - **Content-Type** – **application/json**
 - **X-Mitto-API-Key** – enter your personal Mitto API key
- Create a **message body** and other parameters inside the code.

The screenshot shows a REST client interface for configuring a POST request to the Mitto SMS API. The URL is `https://rest.mittoapi.net/sms`. The headers section is configured with:

NAME	VALUE
Content-Type	application/json
X-Mitto-API-Key	CZ6q0s05p6R2NCDEmlhrnqx7BApA6hNA

The body is a JSON object:

```

1 {
2   "from": "Mitto",
3   "to": "{{customer.phone}}",
4   "text": "Hi, this is a test message."
5 }

```

On the left, the 'Sample data' section shows search filters for 'email address' and 'adding filters'. The results list includes 'sand.b@mi.ch'. Below, the 'Attributes' section lists various fields such as `_created_in_customerio_at`, `cio_id`, `created_at`, `email`, `id`, and `phone`.

Mitto SMS API reference is available here: <https://docs.mitto.ch/sms-api-reference/>

Now we will show you how to create message for chat apps – example WhatsApp Business.

- Your workflow is created in this step. To send a message **you need to choose Send and Receive Data from the Data section. Click on that step and edit message.**

The screenshot displays the 'WhatsApp Test Campaign' workflow builder interface. At the top, a progress bar shows five steps: 1. Trigger, 2. Settings, 3. Goal & Exit, 4. Workflow (highlighted), and 5. Review. On the left, a 'Build' sidebar lists various actions categorized into Messages, Data, Delays, and Flow Control. The 'Data' section includes 'Send and Receive Data'. The main workspace shows a workflow diagram with three steps: a 'Trigger' step (Valid Email Address), a 'WhatsApp connector' step (POST - messaging.mittoapl.com), and an 'Exit' step. The 'WhatsApp connector' step is highlighted with a red box.

- Now you need to use Mitto chat apps API to set it all up. Choose **POST** from the dropdown and enter link: <https://messaging.mittoapi.com/api/v1.1/Messages/send>
- Set up headers sections, enter:
 - **Content-Type – application/json**
 - **Authorization – enter your personal API key**
- Keep in mind that you need to respect the rules per each chat app. For **WhatsApp**, you can use your registered templates **here**.
 - Define parameters inside the code – add trafficAccountId, set up destination (these can be numbers or simply a customer.io variable like "{{customer.phone}}") which will use numbers from the segment), insert template name and its parameters

The screenshot shows the Mitto API interface for configuring a WhatsApp connector. The breadcrumb navigation is: WhatsApp Test Campaign > Workflow > WhatsApp connector. The interface is divided into several sections:

- Request Tab:**
 - Sample data:** Includes a search bar for email addresses. Results show 'sandro.b@mitto.ch' and 'sand.b@mio.ch'.
 - Attributes:** Lists various attributes such as '_created_in_customerio_at', 'cio_id', 'created_at', 'email', 'id', 'phone', etc.
- POST Request Configuration:**
 - Method:** POST
 - URL:** https://messaging.mittoapi.com/api/v1.1/Messages/send
 - Headers:**
 - Content-Type: application/json
 - Authorization: Basic NmU1NzY2MGtMTk1MS00MWY3LWI0ZGMtOTFIND
 - JSON Body:**

```

1 {
2   "trafficAccountId": "679b453b-c876-49b0-b30e-c2b4c1c63369",
3   "destination": "{{customer.phone}}",
4   "whatsapp": {
5     "type": "template",
6     "template": {
7       "name": "mcp_account",
8       "language": {
9         "policy": "deterministic",
10        "code": "en"
11      }
12     },
13     "components": [
14       {
15         "type": "body",
16         "parameters": [
17           {
18             "type": "text",
19             "text": "John"
20           }
21         ]
22       }
23     ]
24   }
25 }

```

Mitto API reference for chat apps is [available here](#).

mitto›