

# Notify

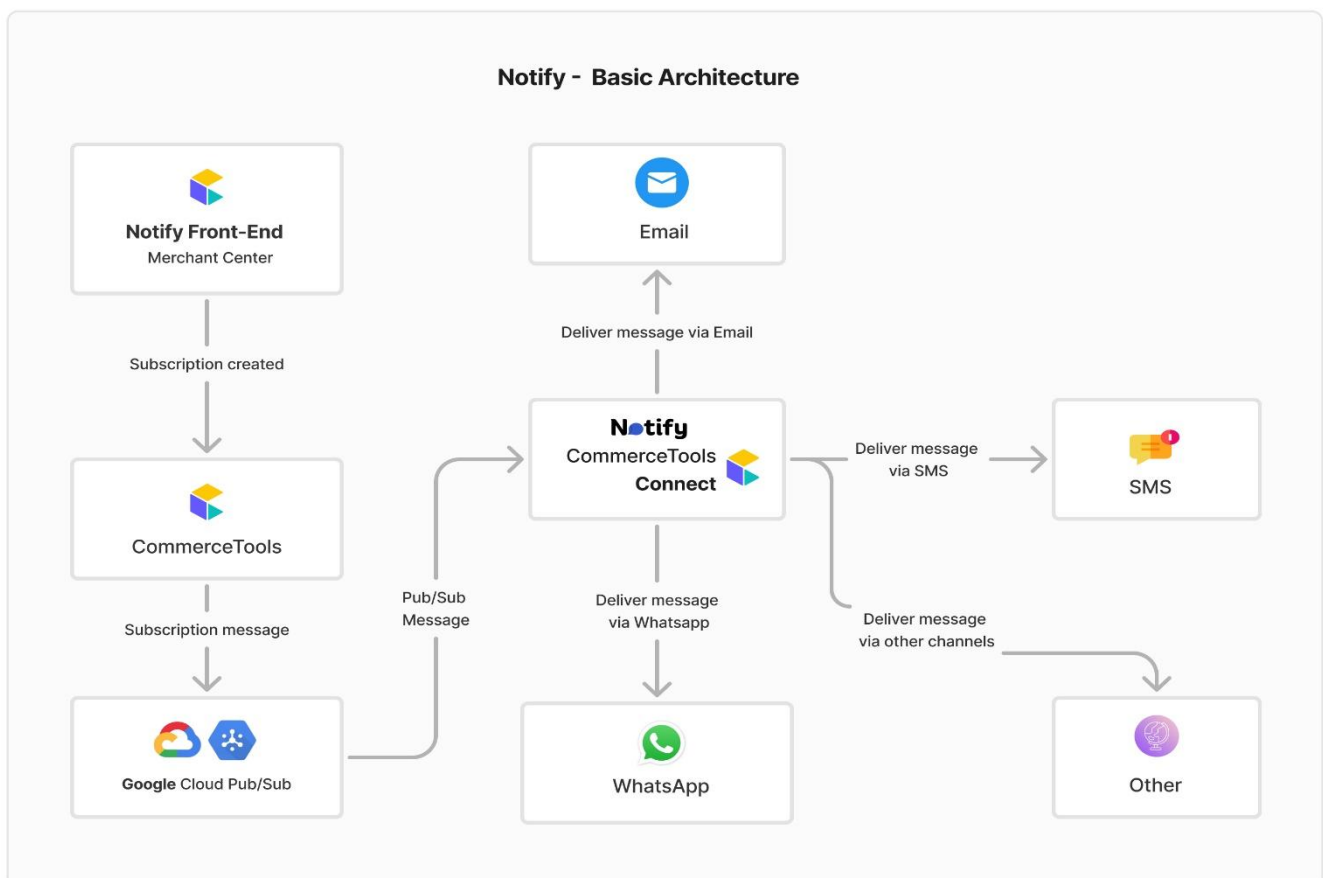
## Overview

**Notify** is an event-driven connector designed to automate real-time message delivery through multiple communication channels such as **WhatsApp**, **Email**, and **SMS**. It allows users to create subscriptions through the Notify UI and offers additional options like customizing message templates and configuring channels for message delivery. Notify listens to specific events triggered in commercetools—such as order creation or status updates—and sends notifications based on the subscriptions defined within the platform.





Notify empowers businesses to:

- 🛎 **Enhance customer engagement** with timely and relevant updates.
- 🛠 **Configure channel-specific message templates** for each subscription.
- 📊 **Track delivery status** of every message across all channels.
- ⚙ **Easily manage subscriptions**—create, edit, or remove subscriptions directly via the Notify UI

## Architecture



## Message Flow Explained

- **Subscription Creation**  
Users create subscriptions via the Notify Front-End (Merchant Center). For example, adding an OrderStateChanged subscription through the UI creates a corresponding subscription in commercetools.
- **Event Triggered**  
When a relevant event (like a change in order state) occurs, commercetools publishes a message to the configured Google Cloud Pub/Sub topic.
- **Message Received by Notify**  
The Notify Connector subscribes to the Pub/Sub topic and receives the message.
- **Message Processing**
  - Upon receiving the message, the connector:
  - Parses the event payload
  - Retrieves the user-defined configurations (e.g., selected channels, message templates, recipients)
  - Determines which channels the message should be sent to
- **Message Dispatch**  
The connector delivers the message across all configured channels:
  -  **Email**
  -  **SMS**
  -  **WhatsApp**
  -  **Other supported or custom channels**

## ✓ Prerequisites

Before setting up and running the Notify connector, ensure you have the following environment variables and accounts configured.

### 1. Commercetools Account & API Credentials

To connect to the commercetools platform, you need to generate API credentials:

1. Log in to the commercetools Merchant Center.
2. Navigate to:  
**Settings → Developer settings → Create new API client**
3. Select the required scopes for managing subscriptions and orders.
4. Capture the following credentials and update these variables:
  - CTP\_PROJECT\_KEY=your\_project\_key
  - CTP\_CLIENT\_ID=your\_client\_id

- CTP\_CLIENT\_SECRET=your\_client\_secret
  - CTP\_SCOPE=view\_orders manage\_subscriptions ...
  - CTP\_REGION=europe-west1.gcp (or applicable region)
- 

## 2. Twilio Account & Authentication Keys (for WhatsApp/SMS)

1. Sign in to your [Twilio Console](#).
2. Go to **Account Info** (available under the **For You** tab).
3. Copy the following credentials:
  - TWILIO\_ACCOUNT\_SID=your\_account\_sid
  - TWILIO\_AUTH\_TOKEN=your\_auth\_token

✅ **Note:** Make sure your Twilio phone number has WhatsApp capabilities if you're using it for WhatsApp notifications.

---

## 3. SendGrid Account & API Key (for Email Delivery)

1. Create or log in to your [SendGrid account](#).
  2. Set up **Sender Identity**:
    - Add your contact details and a **verified sender email address**.
  3. Go to:  
**Integration Guide** → **Choose Web API** → **Choose node.js** → **Create API Key**
  4. Copy the API key and update:
    - SENDGRID\_API\_KEY=your\_sendgrid\_api\_key
- 

## 4. Front-end (mc-app) credentials

1. Complete the [Custom application registration](#).
  2. Copy the **Application ID**:
  3. Copy the API key and update:
    - CUSTOM\_APPLICATION\_ID=your\_custom\_application\_id
- 

## **Configure Custom Application for the Front-End (Via Merchant Center)**

✅ **Note:** Setting up a Custom Application is a crucial step for installing the Notify connector. The front-end is the primary interface for configuring subscriptions, message templates, and channels. Without this step, you won't be able to access or configure Notify.

### **Steps to Configure Notify Front-End**

### 1. Read Official Documentation

Begin by reviewing the official commercetools [Custom Application documentation](#) to understand how the custom application deployment works.

### 2. Navigate to Merchant Center Settings

- Log into the **Commercetools Merchant Center**
- Go to **Manage organizations & teams → Your organization → Custom Applications → Configure Custom Applications**

### 3. Add a New Custom Application

Click “**Add Custom Application**”. You’ll be presented with a form requiring you to fill out key fields. Use the values exactly as specified below unless your setup requires otherwise:

Field	Value
Application name	Notify
Application URL	During the initial configuration, leave the field with the default value (e.g., <a href="https://example.com">https://example.com</a> ). After <a href="#">installing Notify</a> , you will receive the application URL. Copy & update this field with the actual application URL at that time.
Application entry point URI path	notify
<b>Permissions</b>	
OAuth Scopes for ViewMyCustomAppRoute	<a href="#">View necessary OAuth Scopes</a>
OAuth Scopes for ManageMyCustomAppRoute	<a href="#">View necessary OAuth</a> ManageMyCustomAppRoute
<b>Main menu</b>	
Default link label	Notify
Link permissions	ViewMyCustomAppRoute

#### OAuth Scopes for ViewMyCustomAppRoute

view\_orders, view\_key\_value\_documents, view\_api\_clients, view\_approval\_flows,  
view\_approval\_rules, view\_associate\_roles, view\_attribute\_groups, view\_audit\_log,  
view\_business\_units, view\_cart\_discounts, view\_categories, view\_checkout\_transactions,  
view\_customer\_groups, view\_customers, view\_discount\_codes, view\_import\_containers,  
view\_messages, view\_order\_edits, view\_payments, view\_product\_selections, view\_products,  
view\_project\_settings, view\_published\_products, view\_quote\_requests, view\_quotes,  
view\_recurring\_orders, view\_sessions, view\_shipping\_methods, view\_shopping\_lists,  
view\_staged\_quotes, view\_standalone\_prices, view\_states, view\_stores, view\_tax\_categories,  
view\_types

#### OAuth Scopes for ManageMyCustomAppRoute

manage\_orders, manage\_key\_value\_documents, manage\_subscriptions

#### 4. Register Custom Application

Click “**Register Custom Application**”. After registering the custom application an **Application ID** is created. Copy the **Application ID** that is required in [Notify installation process](#).

### Configure Custom Application for the Front-End (Via local Setup)

#### 1. Clone the repository

```
RUN : git clone https://github.com/AyataCommerce/notify-commercetools-connector.git
RUN : cd mc-notify
RUN : mc-scripts login
RUN : mc-scripts config:sync
```

#### 2. Populate the .env file with these variables

- ENABLE\_NEW\_JSX\_TRANSFORM="true"
- FAST\_REFRESH="true"
- ENTRY\_POINT\_URI\_PATH=notify
- CLOUD\_IDENTIFIER=gcp-eu
- CUSTOM\_APPLICATION\_ID=<your\_custom\_application\_id> (may not be available at first time.)
- APPLICATION\_URL=https://google.com

Read [this](#) documentation to know about how to configure locally.

### Installing Notify to Your Project

Once prerequisites and local setup are complete, follow the steps below to integrate Notify with your commercetools project.

1. To include the **Notify Front - End** to your project first you need to complete [custom application registration process](#) and get the **Application ID**
2. After completing the prerequisite:
  - Go to **Merchant center** → **your project** → **At the top right of the Merchan centre** → **Manage Organizations & teams** → **Choose your organization** → **Connect** → **Search for Notify** → **Click Install** and proceed with installation by fill all necessary fields. After installing Notify choose **mc-notify** then copy the URL string from the URL field.
3. Go to **Custom Applications** → **configure custom applications** → **Choose Notify** → then update the **Application URL** field with the **URL** that we copied from the mc-notify.
4. Click **Save**.
5. Check if the state is in **ready** state if not change the state to **ready** then click install application.

### Running the Application Locally

Follow these steps to set up and run the Notify application on your local machine:

→ git clone <https://github.com/AyataCommerce/notify-commercetools-connector.git>

### 1. Navigate to the Project Directory

Depending on the component you want to run:

**For the event-based application (backend):**

→ cd notify-commercetools-connector/notify-event

**For the custom front-end application (Merchant Center UI):**

→ cd notify-commercetools-connector/mc-notify

### 2. Install Dependencies

Run the following command in the appropriate directory:

→ yarn install

### 3. Configure Environment Variables

Make sure the .env file is created and updated with the required variables (refer to the Prerequisites section). This file should include API keys, credentials, and app-specific configurations.

### 4. Start the Application in Development Mode

**To start the event application (backend):**

→ yarn start:dev

**To start the event application (backend):**

→ yarn start

## How to Uninstall Notify

If you wish to remove the Notify connector from your commercetools project, follow these steps:

1. Go to **Connect** in the **Merchant center** → **Choose manage connectors** → **Installations** → **Select Notify** → **Click Uninstall**