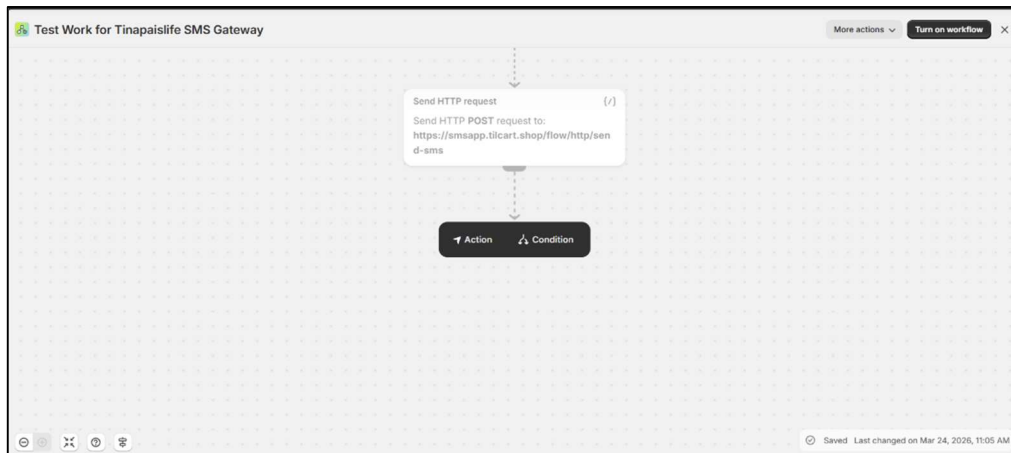


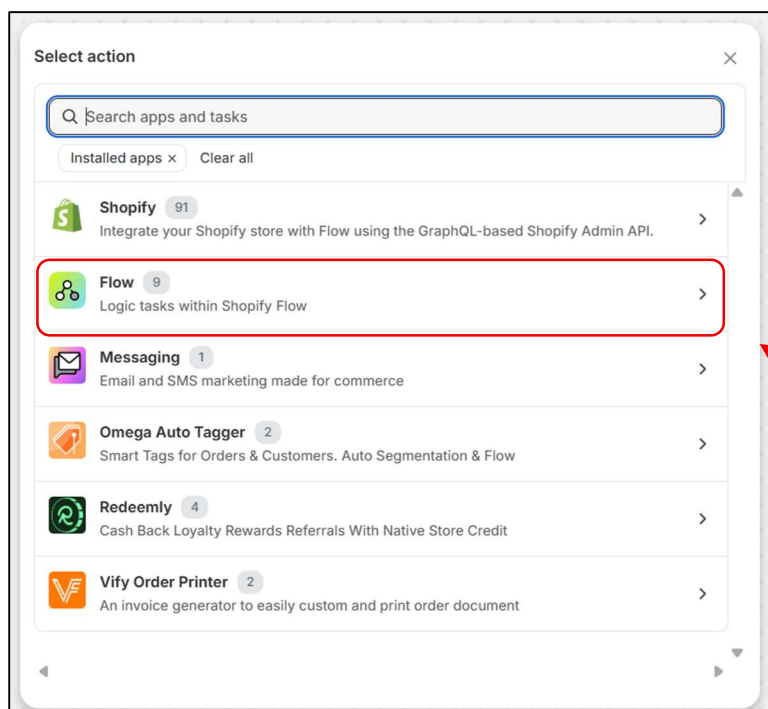


## INSTRUCTIONS ON HOW TO USE THE SMS APP IN SHOPIFY FLOW

1. First open the **Flow** in Shopify and when adding in another flow Choose **action**

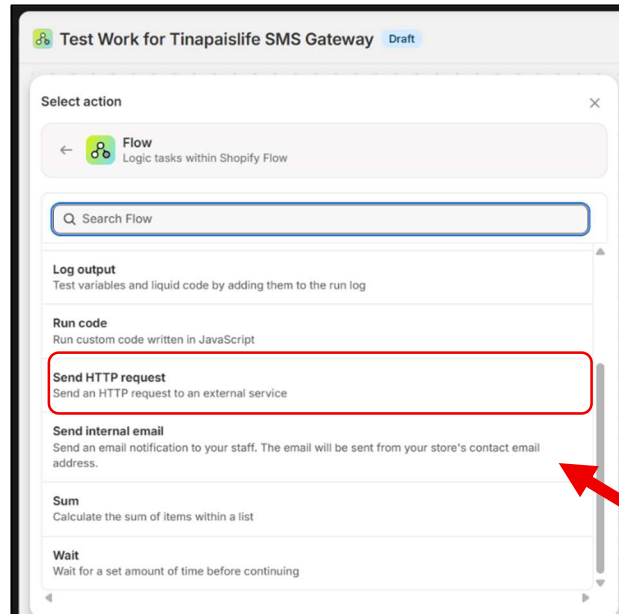


2. Then chose **Flow**

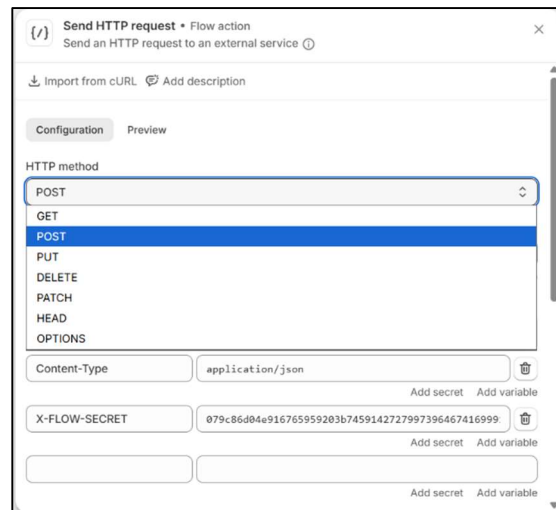




### 3. Choose **Send HTTP request**



### 4. When u added the flow action the configuration should be this, in **HTTP method** choose **POST** method.





- Then for **URL** paste this link <https://smsapp.tilcart.shop/flow/http/send-sms>

URL

Add secret Add variable

- For the Key and Value configurations put **Content-Type** in the first cell and for its value put **application/json**. Then for the second-row input **X-FLOW\_SECRET** in first cell and in second cell input the key value **079c86d04e916765959203b74591427279973964674169991436104311097314**

Headers

Key	Value	
Content-Type	application/json	<input type="button" value="Add secret"/> <input type="button" value="Add variable"/>
X-FLOW-SECRET	079c86d04e916765959203b7459142727997396467416999:	<input type="button" value="Add secret"/> <input type="button" value="Add variable"/>
		<input type="button" value="Add secret"/> <input type="button" value="Add variable"/>



7. For the body put this json data format for template handle **orders confirmed** template endpoint

```
{
  "recipient_phone": "{{order.customer.phone}}",
  "template_handle": "order_confirmed",
  "customer_first_name": "{{order.shippingAddress.phone}}",
  "customer_last_name": "{{order.customer.lastName}}",
  "customer_email": "{{order.customer.email}}",
  "order_name": "{{order.name}}",
  "order_id": "{{order.id}}",
  "reference_id": "{{order.name}}"
}
```

Body

```
{
  "recipient_phone": "09773247613",
  "template_handle": "order_confirmed",
  "customer_first_name": "{{order.shippingAddress.phone}}",
  "customer_last_name": "{{order.customer.lastName}}",
  "customer_email": "{{order.customer.email}}",
  "order_name": "{{order.name}}",
  "order_id": "{{order.id}}",
  "reference_id": "{{order.name}}"
}
```

Add secret Add variable

On client error (4XX response)

Fail

Choose how to manage errors

On server error (5XX or 429 response)

Fail

Choose how to manage errors



## FOR MAKING TEMPLATES IN THE TINAPAISLIFE SMS

1. In the dashboard there are two paths you can go to for making templates.

The screenshot shows the TinapaisLife SMS dashboard. On the left, a sidebar menu lists 'Dashboard', 'Templates', 'Logs', and 'Settings'. The 'Templates' menu item is highlighted with a red arrow. In the main content area, the 'SMS Automation' section includes a 'Create template' button, also highlighted with a red arrow. Below this, there is a table titled 'Latest Flow executions' showing message status, recipient, template, reference, and creation time.

STATUS	RECIPIENT	TEMPLATE	REFERENCE	CREATED
SUCCESS	09773247613	order_confirmed	TIL57651	2026-03-24 02:23:11
SUCCESS	09773247613	order_confirmed	TIL57650	2026-03-24 02:08:17
FAILED	09773247613	order_confirmed	TEST123	2026-03-23 08:52:23
SUCCESS	09773247613	order_confirmed	TEST123	2026-03-23 08:22:43



2. In **Template Name** you can just name it anything depending on the Flow action you want to add in **Shopify Flow** action.

- **Template Handle** – this is what u need to think about, u need to put this in the template handle in the json data `{"template_handle": "order_confirmed",}` **this will guide the flow action on what template it will use.**
- **Description** – u can leave this empty, this field is for notes.

Embedded-style admin designed for Shopify Flow SMS automation. Internal app

## Create template

Build reusable SMS content for Flow actions and test sends. [Back to templates](#)

**Template details**  
Keep names and handles clean so the team can recognize them quickly.

**Shop domain**  
tinapaislife-495.myshopify.com  
Store context this template belongs to.

**Category**  
Transactional  
Examples: Transactional, Recovery, Fulfillment.

**Template name**  
  
Human-friendly label used in the app.

**Template handle**  
  
Stable key used by Flow, for example order\_confirmed.

**Description**  
  
Optional context for the team.

**Message body**

**Recommended variables**  
Use stable placeholders that can be mapped from Flow payloads.

- {{customer\_first\_name}}
- {{order\_name}}
- {{tracking\_number}}
- {{checkout\_url}}
- {{shop\_name}}

Use variables like {{customer\_first\_name}} and {{order\_name}}



### 3. Message Body – this is where you will put the **message** that is going to be the contents of the SMS

**Description**

Optional context for the team.

**Message body**

Hi {{customer\_first\_name}}, your order {{order\_name}} has been confirmed.  
Thank you for shopping with TinapaisLife.

Use variables like {{customer\_first\_name}} and {{order\_name}}.

**Default shortcode**

**Status**  **Template is active and available for Flow actions**

- You can chose put json variables in the message body for custom data depending on the customer data

**Template metadata**

**Version:** 1

**Handle:** order\_confirmed

**Status:** Active

---

**Reference endpoint**

Customer

Orders (default)

Abandoned Checkout

Customer

Fulfillment

```
"customer_first_name": "{{ customer.firstName }}",  
"customer_last_name": "{{ customer.lastName }}",  
"customer_email": "{{ customer.email }}",  
"reference_id": "{{ customer.id }}"  
}
```

**Customer fields**

Flow Liquid uses camelCase on the customer object. Exact availability depends on your workflow trigger.

- {{ customer.id }} (id)
- {{ customer.email }} (email)
- {{ customer.createdAt }} (created\_at)
- {{ customer.updatedAt }} (updated\_at)
- {{ customer.firstName }} (first\_name)
- {{ customer.lastName }} (last\_name)
- {{ customer.ordersCount }} (orders\_count)
- {{ customer.state }} (state)
- {{ customer.totalSpent }} (total\_spent; may be a money object—use nested amount if shown in



- There are 4 dropdown selection depending on the data given by the Flow trigger, if the data is from orders you can pick the orders dropdown for orders json variables.

**Order created** • Shopify trigger  
This workflow starts when a new order is created ⓘ

**Preview**

**Trigger data**  
The trigger comes with this data ^

```
{
  "order_id": "gid://shopify/Order/6591372067044"
}
```

**Data used by this workflow**  
The workflow needs this data to work ^

```
{
  "order": {
    "id": "gid://shopify/Order/6591372067044",
    "name": "TIL57614",
    "customer": {
      "firstName": "Tessie",
      "lastName": "Fonacier",
      "email": "tvfonacier@yahoo.com"
    }
  }
},
```

The workflow changed, this preview might be outdated [Refresh](#)