# WhatsApp Templates API Management

> The template APIs allow you to create, delete, and fetch templates

## The templates API supports the following methods and are client specific

- GET
- POST
- DELETE

### Get all templates

```
curl --location 'https://api.versal.one/<client-id>/templates' \
--header 'Authorization: Bearer <client-token>'
```

### Get a template by name

```
curl --location 'https://api.versal.one/<client-id>/templates?name=order_delivery_1' \
--header 'Authorization: Bearer <client-token>'
```

### Create a template

To create a template - the following components are required

> - name - template name [cannot contain Upper case, space, special characters] can contain underscore "_" and a max of 25 characters
> - language - en [English], en_US [English US]. Click here for the complete list
> - category - Marketing/Utility
> - components - Header, Body, Footer

> for more details check the official documentation on the META website

```
curl --location 'https://api.versal.one/<client-id>/templates' \
--header 'Authorization: Bearer <client-token>' \
--data '{
    "name": "order_delivery_1",
    "language": "en_US",
    "category": "Marketing",
    "components": [
        {
            "type": "HEADER",
            "format": "Text",
            "text": "header text"
        },
        {
            "type": "BODY",
            "text": "This is a {{1}}",
            "example": {
                "body_text": [
                    [
                        "random text"
                    ]
                ]
            }
        },
        {
            "type": "FOOTER",
            "text": "lasadoasi"
        },
        {
            "type": "BUTTONS",
            "buttons": [
                {
                    "type": "PHONE_NUMBER",
                    "text": "Call",
                    "phone_number": "+918897229166"
                }
            ]
```

```
        }
    ]
}'
```

```
{
    "message": "Template created successfully",
    "response_text":
"{\"id\":\"1179134613392693\",\"status\":\"APPROVED\",\"category\":\"MARKETING\"}",
    "status": "200"
}
```

# Template Components

Templates are made up of four primary components which you define when you create a template: header, body, footer, and buttons. The components you choose for each of your templates should be based on your business needs. The only required component is the body component.

Some components support variables, whose values you can supply when using the Cloud API or On-Premises API to send the template in a template message. If your templates use variables, you must include sample variable values upon template creation.

# Headers

Headers are optional components that appear at the top of template messages. Headers support text, media (images, videos, documents), and locations.

All templates are limited to one header component

## Text Headers

You can choose to add header text to your message template. The message text in the header component accepts parameters for programmatic configuration.

Text parameters can be configured in one of two formats:

- **Positional** — Pass in an array of parameters that correspond to numeric positions in the body text with examples

    - For example: `“Hello John, your account balance is {{1}}” | [ “$1,000” ]`
- **Named** — Pass in JSON objects that contain a parameter name and examples
    - For example: `{ "param_name": "account_balance", "example": "$1,000" }`

# Component Syntax

Add this header component object into the `”components”[]` object array when calling the `POST <WHATSAPP_BUSINESS_ACCOUNT_ID>/message_templates` endpoint. Substitute the placeholder properties below using the properties table.

```
{
  "type": "HEADER",
  "format": "TEXT",
  "text": "<HEADER_TEXT>",

  "example": {
    "header_text": [
       // You must provide one of the inputs below when the <HEADER_TEXT> string contains parame
       <POSITIONAL_PARAM_EXAMPLES>
       <BODY_TEXT_NAMED_PARAMS>
    ]
  }
}
```

# Properties

| Placeholder | Description | Example Value |
|---|---|---|
| `<HEADER_TEXT>` | *`String`* \| **Required**<br>Plain text string. Can support 1 parameter.<br>If this string contains parameters, you must include the `example` property and example parameter values as shown in `<POSITIONAL_PARAM_EXAMPLES>` and `<HEADER_TEXT_NAMED_PARAMS>` below.<br>60 character maximum. | `“Our Holiday sale starts December 1st!”`<br>`“Our new sale starts {{1}}”`<br>`“Our new sale starts {{sale_start_date}}!”` |
| `<POSITIONAL_PARAM_EXAMPLES>` | *`[String]`* \| *Optional*<br>Required when using positional parameters in your header text.<br>Array of `String` in which each string is meant to illustrate the text likely to be passed in as a parameter during message send time, for example a bank account balance, or a customer name.<br>The number of strings must match the number of variables included in the string. | `["December 1st"]` |

| Placeholder | Description | Example Value |
| --- | --- | --- |
| `<HEADER_TEXT_NAMED_PARAMS>` | `[JSON Object]` \| *Optional*<br>Required when using named variables in your header text.<br>Array of JSON objects that contain `param_name` and `example` strings.<br>• `"param_name"` — Your custom parameter name. Must be lowercase letters and underscores only.<br>• `"example"` — The illustrative sample text for the parameter. | `[{ "param_name": "sale_start_date", "example": "December 1st" }]` |

## Positional Parameter Example

```
{
  "type": "HEADER",
  "format": "TEXT",
  "text": "Our new sale starts {{1}}!",
  "example": {
    "header_text": [
      "December 1st"
    ]
  }
}
```

## Named Parameter Example

```
{
  "type": "HEADER",
  "format": "TEXT",
  "text": "Our new sale starts {{sale_start_date}}!",
  "example": {
    "header_text_named_params": [
      {
        "param_name": "sale_start_date",
        "example": "December 1st"
      }
    ]
  }
}
```

# Media Headers

Media headers can be an image, video, or a document such as a PDF. All media must be uploaded with the Resumable Upload API. The syntax for defining a media header is the same for all media types.

## Syntax

```
{
  "type": "HEADER",
  "format": "<FORMAT>",
  "example": {
    "header_handle": [
      "<HEADER_HANDLE>"
    ]
  }
}
```

## Properties

| Placeholder | Description | Example Value |
| --- | --- | --- |
| `<FORMAT>` | Indicates media asset type. Set to `IMAGE`, `VIDEO`, or `DOCUMENT`. | `IMAGE` |
| `<HEADER_HANDLE>` | Uploaded media asset handle. Use the Resumable Upload API to generate an asset handle. | `4::aW...` |

## Example

```
{
  "type": "HEADER",
  "format": "IMAGE",
  "example": {
    "header_handle": [
      "4::aW..."
    ]
  }
}
```

# Body

The body component represents the core text of your message template and is a text-only template component. It is required for all templates.

The message text in the body component accepts parameters for programmatic configuration.

Text parameters can be configured in one of two formats:

- **Positional** — Pass in an array of numbered positional parameters that correspond to numeric positions in the body text with examples
  - For example: `“Hello {{1}}, your account balance is {{2}}” | [ “John”, “$1,000” ]`
- **Named** — Pass in JSON objects that contain a parameter name and examples
  - For example: `{ "param_name": "order_id", "example": "335628"}`

All templates are limited to one body component.

# Component Syntax

Add this body component object into the `”components”[]` object array when calling the `POST <WHATSAPP_BUSINESS_ACCOUNT_ID>/message_templates` endpoint. Substitute the placeholder properties below using the properties table.

```
{
  "type": "body",
  "text": "<BODY_TEXT>",
  "example": {
    "body_text": [
      [
       // You must provide one of the inputs below when the <BODY_TEXT> string contains paramete
       <POSITIONAL_PARAM_EXAMPLES>
       <BODY_TEXT_NAMED_PARAMS>
      ]
    ]
  }
}
```

# Properties

| Placeholder | Description | Example Value |
| --- | --- | --- |

| | | |
|---|---|---|
| `<HEADER_TEXT>` | *`String`* \| **Required**<br>Plain text string. Can support multiple parameters. If this string contains parameters, you must include the `example` property and example parameter values as shown in `<POSITIONAL_PARAM_EXAMPLES>` and `<BODY_TEXT_NAMED_PARAMS>` below.<br>1024 character maximum. | `“Shop our Holiday sale now!”`<br>`“Shop now through {{1}} and use code {{2}} to get {{3}} off of all merchandise.”`<br>`“Your {{order_id}}, is ready {{customer_name}}”` |
| `<POSITIONAL_PARAM_EXAMPLES>` | *`[String]`* \| *Optional*<br>Required when using positional parameters in your body text.<br>Array of string in which each string is meant to illustrate the text likely to be passed in as a parameter during message send time, for example a bank account balance, or a customer name.<br>The number of strings must match the number of variables included in the string. | `["the end of August","25OFF","25%"]` |
| `<BODY_TEXT_NAMED_PARAMS>` | *`[JSON Object]`* \| *Optional*<br>Required when using named variables in your body text.<br>Array of JSON objects that contain `param_name` and `example` strings.<br>• `“param_name”` — Your custom parameter name. Must be lowercase letters and underscores only.<br>• `“example”` — The illustrative sample text for the parameter. | `[{`<br>`"param_name": "order_id",`<br>`"example": "335628"`<br>`},`<br>`{`<br>`"param_name": "customer_name",`<br>`"example": "Shiva"`<br>`}]` |

# Positional Parameter Example

```
{
  "type": "BODY",
  "text": "Shop now through {{1}} and use code {{2}} to get {{3}} off of all merchandise.",
  "example": {
    "header_text": [
      "the end of August","25OFF","25%"
    ]
  }
}
```

# Named Parameter Example

```
{
  "type": "BODY",
  "text": "Your {{order_id}}, is ready {{customer_name}}.",
  "example": {
    "header_text_named_params": [
      {
```

```
        "param_name": "order_id",
        "example": "335628"
      },
      {
        "param_name": "customer_name",
        "example": "Shiva"
      }
    ]
  }
}
```

# Footer

Footers are optional text-only components that appear immediately after the body component. Templates are limited to one footer component.

## Syntax

```
{
  "type": "FOOTER",
  "text": "<TEXT>"
}
```

## Properties

| Placeholder | Description | Example Value |
| --- | --- | --- |
| `<TEXT>` | Text to appear in template footer when sent.<br><br>60 characters maximum. | `Use the buttons below to manage your marketing subscriptions` |

## Example

```
{
  "type": "FOOTER",
  "text": "Use the buttons below to manage your marketing subscriptions"
}
```

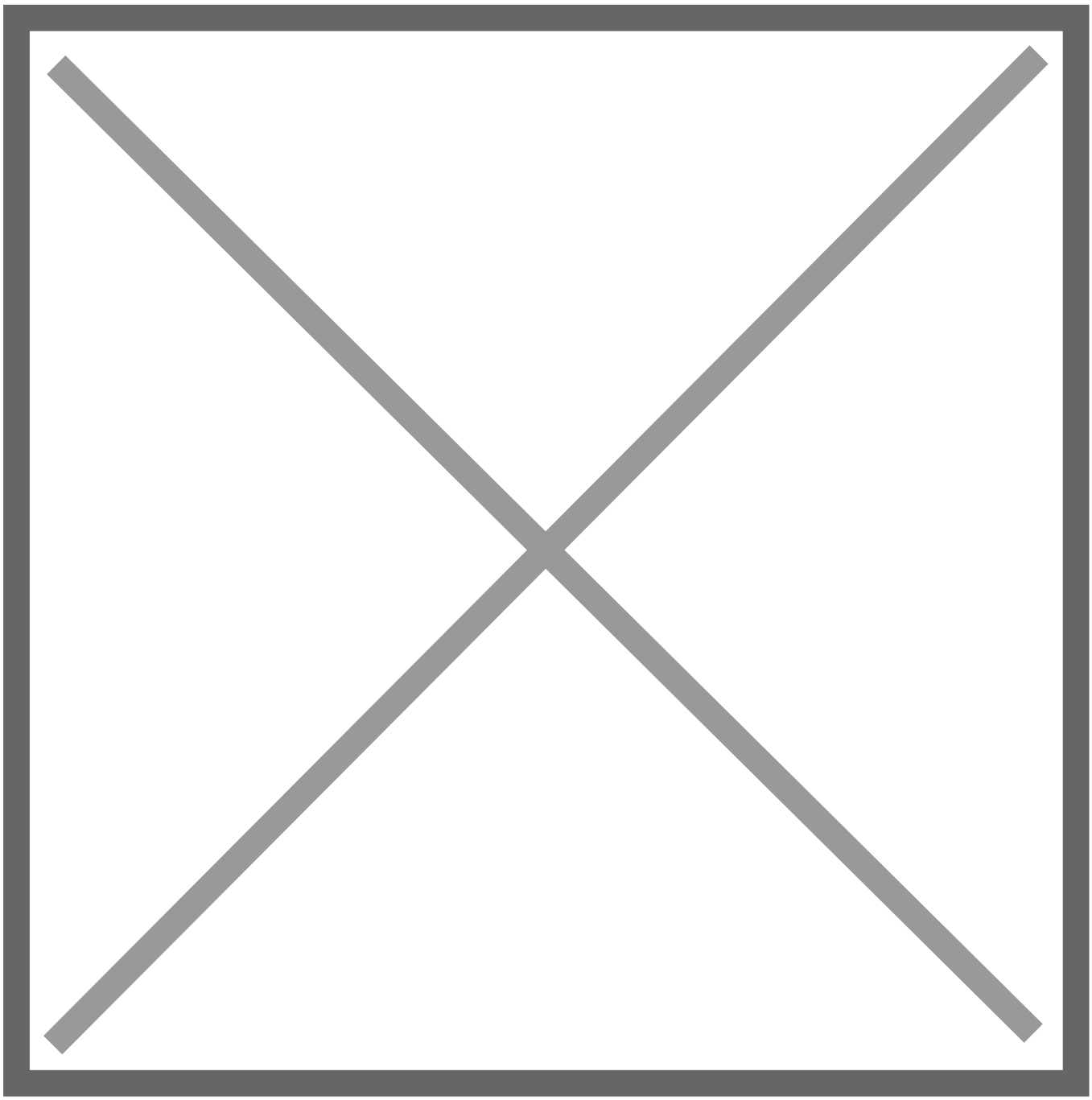
# Buttons

Buttons are optional interactive components that perform specific actions when tapped. Templates can have a mixture of up to 10 button components total, although there are limits to individual buttons of the same type as well as combination limits. These limits are described below.

Buttons are defined within a single buttons component object, packed into a single `buttons` array. For example, this template uses a phone number button and a URL button:

```
{
  "type": "BUTTONS",
  "buttons": [
    {
      "type": "PHONE_NUMBER",
      "text": "Call",
      "phone_number": "15550051310"
    },
    {
      "type": "URL",
      "text": "Shop Now",
      "url": "https://www.luckyshrub.com/shop/"
    }
  ]
}
```

If a template has more than three buttons, two buttons will appear in the delivered message and the remaining buttons will be replaced with a **See all options** button. Tapping the **See all options** button reveals the remaining buttons.



# Copy Code Buttons

Copy code buttons copy a text string (defined when the template is sent in a template message) to the device's clipboard when tapped by the app user. Templates are limited to one copy code button.

## Syntax

```
{
  "type": "COPY_CODE",
  "example": "<EXAMPLE>"
```

```
}
```

# Properties

| Placeholder | Description | Example Value |
| --- | --- | --- |
| `<EXAMPLE>` | String to be copied to device's clipboard when tapped by the app user.<br><br>Maximum 15 characters. | `250FF` |

# Example

```
{
  "type": "COPY_CODE",
  "example": "250FF"
}
```

## Delete a template

```
curl --location --request DELETE 'https://api.versal.one/<client-
id>/templates?name=harsh45021279' \
--header 'Authorization: Bearer <client-token>'
```

---

Revision #3
Created 2024-12-07 00:04:22 UTC by Admin
Updated 2024-12-07 02:39:46 UTC by Admin