Northbound Interface of DCAST Message Platform

Product Version: V2.8

Document Version: D2.8

Contents

1 Introduction

1.1 Objectives

Our DCAST service A2P Messaging is an enterprise-grade Application-to-Person (A2P) communication platform that allows DITO Business to offer the most reliable and widely-used messaging channels—including SMS, WhatsApp, and Viber and others—to leading brands. It supports marketing campaigns, utility notifications, and customer service automation to help businesses effectively engage their customers at scale.

This document provides guide to the 3rd-party business system to setup connect, send message, receive result from DCAST message platform, and provide the customer personalized experience.

Commercial Platform ServerRoot:www.a2p.ditobusiness.ph

2 Interface Design

2.1 Authentication

2.1.1 Obtaining access_token

Interface Direction: Third-party Business System→DCAST Platform

Interface Description: The **access_token** is the developer's unique credential, and the developer needs to use the access_token when invoking each interface. Under normal circumstances, the access_token is valid for 7200 **seconds**. The repeated acquisition will invalidate the access_token acquired last time. Since the number of API calls to obtain access_token is very limited, it is recommended that developers store and update access_token globally. Frequently refreshing access_token will limit API calls and affect your own business.

Developers can use AppID and AppSecret to call this API to obtain access_token.

Interface Interaction:

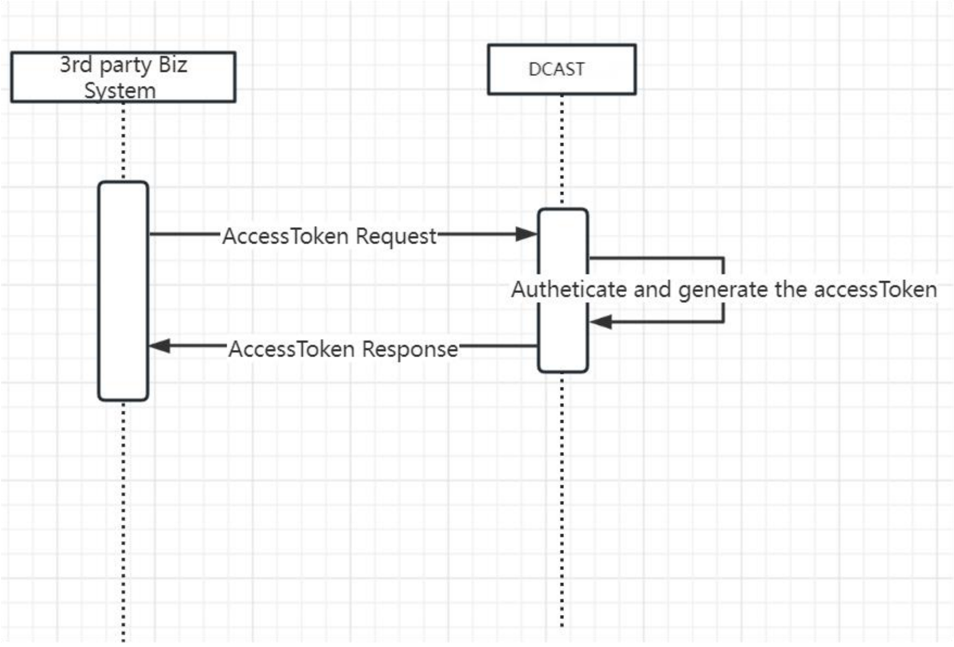

Figure 2- 1　Authentication Interaction

Interface Request:

```
https request method: POST
https://{serverRoot}/a2p/api/v1/access_token
```

Request header:

SN	Field Name	Data Type	Mandatory	Description
1	Content-Type	String	Yes	Need to set. application/json

Request body:

SN	Parameters	Type	Mandatory	Description
1	appId	String	Yes	Third party business system unique credentials, i.e. appId
2	appSecret	String	Yes	Third-party business system unique credential key, i.e. appSecret

Example request:

```
POST /a2p/api/v1/access_token HTTP/1.1
Host: serverRoot
Content-Type: application/json
Content-Length: 90


{
    "appId": "APP_ID".
    "appSecret": "APP_SECRET"
}
```

Response description:

SN	Parameters	Type	Mandatory	Description
1	resultCode	int	Yes	Return Code
2	errorMessage	String	Yes	When an error is returned, the corresponding

				error description information
3	accessToken	String	Yes	Acquired credentials
4	expires	int	Yes	Credential validity time, in seconds

Under normal circumstances, the platform will return JSON data as follows:

```
{
    "resultCode": 0.
    "accessToken": "ACCESS_TOKEN".
    "expires": 7200
}
```

The platform will return the error code and other information in the error case, the JSON packet example is as follows (the example is missing the appid parameter).

```
{
    "resultCode": 40002.
    "errorMessage": "Missing appId parameter"
}
```

2.1.2 Unified Message Header of Interface Call

All interfaces calling DCAST in the business system need to have the following message headers (Header):

SN	Parameters	Type	Mandatory	Description
1	Authorization	String	Yes	Interface credential, the data format is as follows: Bearer ACCESS_TOKEN

2.2 DCAST platform northbound interface

2.2.1 Template message Delivery

Interface direction: Third-party business system → DCAST platform

Interface Description: Third-party business systems send template messages through the DCAST platform.

Interface Interaction:

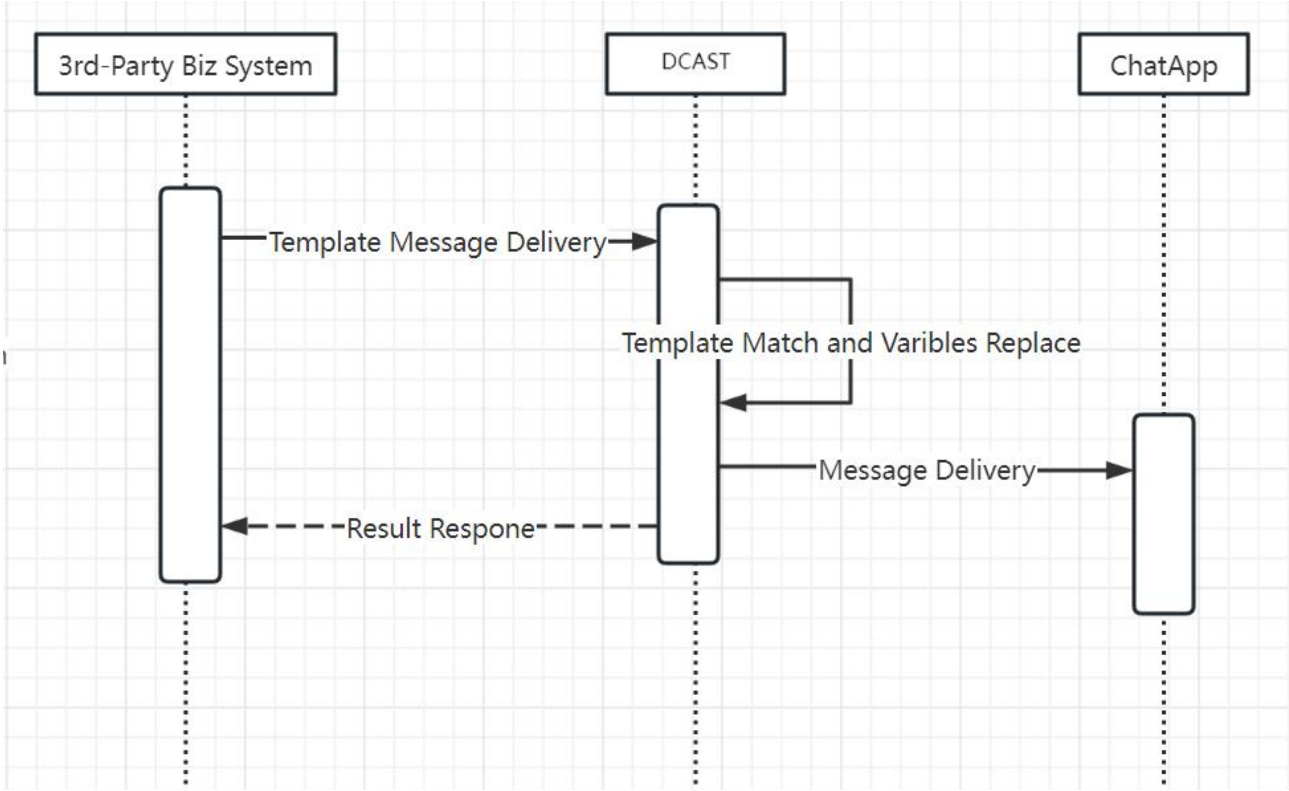

Figure 2- 2 Template Message Delivery

Interface request:

https request method: POST
https://{serverRoot}/a2p/api/v2/message/sendMessage

Request header:

SN	Parameters	Type	Mandatory	Description
1	Content-Type	String	Yes	Need to set. application/json
2	Authorization	String	Yes	Interface credential with the following data format: Bearer $ACCESS_TOKEN

Request body:

SN	Parameters		Type	Mandatory	Description
1	messageCategory		String	Yes	Message Category： WhatSAPP - WhatsApp messages VIBER - Viber messages SMS - Short messages
2	receivers		String	Yes	Receiving number, terminal number, multiple numbers separated by ","
3	templateCode		String	Yes	Template Code
4	language		String	No	Language
5	serviceNo		String	Yes	Service Number of brands/senderID
6	downgrade		String	No	Fallback method :NONE
7	signName		String	No	Signature or senderID for SMS when falling back
8	sendData			Yes	Sending data list
		data	List<Map<String,String>>	No	Business data, when passed in multiple data record, will generate Carousels based on the data

Example request:

```
POST /a2p/api/v2/message/sendMessage HTTP/1.1
```

```
Host: serverRoot
Authorization: Bearer ed6969b2-972c-4c0a-a3f5-fa493b7a9247
Content-Type: application/json
Content-Length: 468

{
    "messageCategory": "WHATSAPP",
    "receivers": "86185*******",
    "templateCode": "791531505050312704",
    "language": "en_US",
    "serviceNo": "86189******",
    "downgrade": "NONE",
    "sendData": {
        "data": [
            {
                "scene": "Login to",
                "code": "821453"
            }
        ]
    }
}
```

Response description:

SN	Parameters	Type	Mandatory	Description
1	resultCode	int	Yes	Return Code
2	errorMessage	String	No	Return an error information when the interface reports an error
3	messageId	String	No	Generated message ID if submitted successfully

The result of the returned JSON packet in the correct case is as follows:

```
{
        "resultCode": 0,
        "errorMessage": "Success",
         "messageId": "8900000100020323"
}
```

An example of a returned JSON packet in the error case is as follows:

```
{
        "resultCode": 41000,
        "errorMessage": "Missing required parameters"
}
```

2.2.2 Text message sending

Interface direction: Third-party business system → DCAST platform

Interface Description: Third-party business systems send text messages through the DCAST platform.

Interface request:

https request method: POST

https://{serverRoot}/a2p/api/v1/message/sendText

Request header:

SN	Parameters	Type	Mandatory	Description
1	Content-Type	String	Yes	Need to set. application/json
2	Authorization	String	Yes	Interface credential with the following data format: Bearer $ACCESS_TOKEN

Request body:

SN	Parameters	Type	Mandatory	Description

1	messageCategory	String	Yes	Message Category: WhatSAPP - WhatsApp messages VIBER - Viber messages SMS - Short messages
2	receivers	String	Yes	Receiving number, multiple phone numbers separated by ","
3	serviceNo	String	Yes	Service Number of Brand senderID for SMS
4	bodyText	String	Yes	Message content

Example request:

```
POST /a2p/api/v1/message/sendText HTTP/1.1

Host: serverRoot

Authorization: Bearer ed6969b2-972c-4c0a-a3f5-fa493b7a9247

Content-Type: application/json

Content-Length: 132


{

    "messageCategory": "SMS",

    "receivers": "86185*******",

    "serviceNo": "60******",

    "bodyText": "Text message content"

}
```

Response description:

SN	Parameters	Type	Mandatory	Description
1	resultCode	int	Yes	Return Code
2	errorMessage	String	No	Return an error message when the interface reports an error
3	messageId	String	No	Generated message ID if submitted successfully

The result of the returned JSON packet in the correct case is as follows:

```
{
        "resultCode": 0,
        "errorMessage": "Success",
         "messageId": "8900000100020323"
}
```

An example of a returned JSON packet in the error case is as follows:

```
{
        "resultCode": 41000,
        "errorMessage": "Missing required parameters"
}
```

2.2.3 File Messaging

Interface direction: Third-party business system → DCAST platform

Interface Description: Third-party business systems send file messages through the DCAST platform.

Interface request:

```
https request method: POST
https://{serverRoot}/a2p/api/v1/message/sendFile
```

Request header:

SN	Parameters	Type	Mandatory	Description
1	Content-Type	String	Yes	Need to set. application/json
2	Authorization	String	Yes	Interface credential with the following data format: Bearer $ACCESS_TOKEN

Request Body:

SN	Parameters	Type	Mandatory	Description
1	messageCategory	String	Yes	Message Category WhatSAPP - WhatsApp messages VIBER - Viber messages
2	receivers	String	Yes	Receiving number,　multiple phone numbers separated by ","
3	serviceNo	String	Yes	Service Number of Brand senderID for SMS
4	fileUrl	String	Yes	File Url Path
5	fileName	String	No	File name

Example request:

```
POST /a2p/api/v1/message/sendFile HTTP/1.1

Host: serverRoot

Authorization: Bearer ed6969b2-972c-4c0a-a3f5-fa493b7a9247

Content-Type: application/json

Content-Length: 172


{

      "messageCategory": "WHATSAPP",

      "receivers": "86185*******",

      "serviceNo": "86189******",

      "fileUrl": "https://******/****.png",

      "fileName": "Logo image"

}
```

Response description:

SN	Parameters	Type	Mandatory	Description
1	resultCode	int	Yes	Return Code
2	errorMessage	String	No	Return an error message when the interface

				reports an error
3	messageId	String	No	Generated message ID if submitted successfully

The result of the returned JSON packet in the correct case is as follows:

```
{
    "resultCode": 0,
    "errorMessage": "Success",
     "messageId": "8900000100020323"
}
```

An example of a returned JSON packet in the error case is as follows:

```
{
    "resultCode": 41000,
    "errorMessage": "Missing required parameters"
}
```

2.2.4 Message Status Notification Interface

Interface direction: DCAST platform → third-party business system

Interface Description: DCAST platform receives the message status notification and synchronizes the message status to the third party business system

Interface request:

Request address: WebHook address configured by the customer on the DCAST platform

Request method: POST

Request Body:

SN	Parameters	Type	Mandatory	Description
1	MessageId	String	Yes	Message ID
2	From	String	Yes	Message sender number

3	To	String	Yes	Message Recipient Number
4	Timestamp	Long	Yes	Timestamp for Unix timestamp milliseconds
5	Status	String	No	The message delivery status. Takes the following values: Sent: The message has been sent. Delivered: The message has been delivered to the user's device. Read: The message has been read by the user. Failed: Message delivery failed. Deleted: The message has been deleted by the user.
6	StatusDescription	String	No	Description of the status code
7	Error	String	No	Error Code
8	ErrorDescription	String	No	Description of the error code
9	ConversationType	String	No	The type of the session. Takes the following values: user_initiated: user initiates a session business_initiated: brand initiated session
10	ConversationId	String	No	Session ID
11	MsgFrameType	String	Yes	Message type. Takes the value: message: a message in any format, with a time limit of 24 hours after receiving the last message from the user to send such messages. template: A message template that has been approved in the console and can be sent at any time.
12	Type	String	No	The type of multimedia resource in the message content. Takes the following values:

				image: picture video：Video document: document audio: audio none: plain text

Example request:

```
Content-Type: application/json
Content-Length: 363


[
    {
        "ConversationId": "92c674c2b55d8b834122432b6dda001c",
        "ConversationType": "user_initiated",
        "From": "86185********",
        "MessageId": "202305816487032730742784",
        "MsgFrameType": "message",
        "Status": "Sent",
        "Timestamp": 1683777292000,
        "To": "86138********",
        "Type": "INTERACTIVE"
    }
]
```

Response description:

SN	Parameters	Type	Mandatory	Description
1	code	int	Yes	Return Code
2	msg	String	No	Return an error message when the interface reports an error

The result of the returned JSON packet in the correct case is as follows:

```
{
    "code": 0,
    "msg": "OK"
}
```

An example of a returned JSON packet in the error case is as follows:

```
{
    "code": 41000,
    "msg": "Handling exceptions"
}
```

2.2.5 Uplink message synchronization interface

Interface direction: DCAST platform → third-party business system

Interface Description: DCAST platform receives the uplink message from the end user and synchronizes the uplink message to the third-party business system.

Interface request:

Request address: WebHook address configured by the customer on the DCAST platform

Request method: POST

Request Body:

SN	Field Name	Data Type	Mandatory	Description
1	MessageId	String	Yes	Message ID
2	From	String	Yes	Message sender number
3	To	String	Yes	Message Recipient Number
4	Timestamp	String	Yes	Timestamp for Unix timestamp milliseconds
5	Name	String	Yes	End User Name
6	Type	String	Yes	The type of the media resource. Takes the value:

				TEXT LOCATION DOCUMENT VIDEO AUDIO REPLY IMAGE CONTACTS
7	Message	String	Yes	The content of the message.

Example request:

```
Content-Type: application/json
Content-Length: 237

[
    {
        "From": "86138********",
        "Message": "B001",
        "MessageId": "20230581648702548622 1313",
        "Name": "ya*",
        "Timestamp": 1683777285000,
        "To": "86185********",
        "Type": "TEXT"
    }
]
```

Response description:

Parameters	Type	Mandatory	Description
code	int	Yes	Return Code
msg	String	No	Return an error message when the interface reports an error

The result of the returned JSON packet in the correct case is as follows:

```
{
    "code": 0,
    "msg": "OK"
}
```

An example of a returned JSON packet in the error case is as follows:

```
{
    "code": 41000,
    "msg": "Handling exceptions"
}
```

2.3 Message Template Interface

2.3.1 Template list query interface

Interface direction: third-party business system → DCAST platform

Interface description: Third-party business systems query the list of templates in the DCAST platform.

Interface Request:

https request method: POST

https://{serverRoot}/a2p/api/waba/messageTemplates/queryTemplateList

Request header:

SN	Field Name	Data Type	Mandatory	Description
1	Content-Type	String	Yes	Need to set. application/json
2	Authorization	String	Yes	Interface credential with the following data format: Bearer $ACCESS_TOKEN

Request Body:

SN	Field Name	Data Type	Mandatory	Description
1	name	String	No	Template name
2	language	String	No	Template language
3	auditStatus	String	No	Review status. Value: Pass: Passed the review. Fail: Reject the audit. Auditing: Under review. Unaudit: Suspend audit processing.
4	templateType	String	No	Template type. Value:

					WHATSAPP VIBER
5	page			Yes	Page field
		index	Integer	Yes	Starting position, default is 1.
		size	Integer	Yes	The default number of templates for each query is 10.

Request Example:

```
POST /a2p/api/waba/messageTemplates/queryTemplateList HTTP/1.1
Host: serverRoot
api-key: 5c672f85-d196-4640-883d-b131dad5d87a
Content-Type: application/json
Content-Length: 120

{
    "auditStatus": "pass",
    "templateType": "WHATSAPP",
    "page":{
        "index": 1,
        "size": 10
    }
}
```

Response description:

SN	Paremeter	Type	Mandatory	Description
1	code	String	Yes	return code
2	message	String	No	When the interface reports an error, return an error message
3	total	Integer	Yes	Total number of records
4	data	object []	Yes	List data

5		templateType	String	Yes	Template type
6		name	String	Yes	Template name
7		auditStatus	String	Yes	Audit status
8		language	String	Yes	Template language
9		templateCode	String	Yes	Template code
10		category	String	Yes	WhatsApp template category. Value: UTILITY: Transaction related Marketing Template Authorization: Authentication Viber template category. Value: Text: plain text Image: Pure image Text_image_button: Text+Image+Button Text-button: Text+button Document: document Video: Video Text_fideo: Text+Video Text_fideoubutton: Text+Video+Button Text_image: Text+Image

The result of the returned JSON packet in the correct case is as follows:

```
{
    "data": [
        {
            "templateType": "WHATSAPP",
            "name": "cjw_location_prd_test",
            "auditStatus": "pass",
                "language": "en",
                "templateCode": "829575821869834240",
                "category": "MARKETING"
            },
            {
                "templateType": "WHATSAPP",
                "name": "cjw_auth_prd_test",
                "auditStatus": "pass",
                "language": "en",
                "templateCode": "829575583738224640",
                "category": "AUTHENTICATION"
            }
    ],
    "code": "OK",
    "total": 91,
    "requestId": "10709879-f70c-4b92-96e4-963deda82c43",
    "success": true
}
```

An example of a returned JSON packet in the error case is as follows:

```
{
    "code": "InvalidParameter.ChannelTypeNotSupport",
     "msg": "Handling exceptions"
     message": "This channel type is not supported at this time",
     "requestId": "e9447f70-4aac-4b48-a378-ca6be3214d94",
    "success": false
}
```

2.3.2 Template details query interface

Interface direction: third-party business system → DCASTplatform

Interface description: Third-party business systems query template details in the DCASTplatform.

Interface Request:

https request method: POST

https://{serverRoot}/a2p/api/waba/messageTemplates/queryTemplate

Request Header:

SN	Paremeter	Type	Mandatory	Description
1	Content-Type	String	Yes	Required settings. application/json
2	Authorization	String	Yes	Interface credentials with the following data format: Bearer $ACCESS_TOKEN

Requesting body:

SN	Paremeter	Type	Mandatory	Description
1	templateCode	String	Yes	template code
2	language	String	Yes	template language
3	templateType	String	Yes	Template type. Value as: WHATSAPP VIBER

Request Example：

```
POST /a2p/api/waba/messageTemplates/queryTemplate HTTP/1.1
```

```
Host: serverRoot
Authorization: Bearer a8bf1e65-1694-4854-8fa9-63f47ed352a9
Content-Type: application/json
Content-Length: 94

{
    "templateCode": "829575821869834240",
    "language": "en",
    "templateType": "WHATSAPP"
}
```

Response description:

SN	Parameters		Type	Mandatory	Description
1	code		String	Yes	Return code
2	message		String	No	Returns an error message when the interface reports an error
3	data		object	Yes	Template detail information
4		templateType	String	Yes	Template Type
5		name	String	Yes	Template Name
6		auditStatus	String	Yes	Audit Status
7		language	String	Yes	template language
8		templateCode	String	Yes	Template Code
9		category	String	Yes	Whatsapp template categories. Values as: UTILITY: Transaction related. MARKETING: Marketing templates. AUTHENTICATION:Authentication. Viber template classification values: text: plain text

					IMAGE: Pure Pictures text_image_button: text + image + button text_button: text + button Document video: video text_video: text + video text_video_button: text + video + butto text_image: text + image
		example	Map<String, String>	Yes	Examples of variables
		components	Object []	Yes	List of message template components.

Component Object：

SN	Parameters	Type	Mandatory	Description
1	type	String	Yes	Components type: BODY HEADER FOOTER BUTTONS Instructions For Whatsapp type templates, the character length of the BODY component does not exceed 1024

				characters.The character length of the HEADER and FOOTER components does not exceed 60 characters. FOOTER type is invalid for Viber type templates. Viber type templates where images, videos, and files are placed in the HEADER (the terminal display shows the images under the text)
2	url	String	No	Material Path
3	text	String	Yes	Text of the message
4	caption	String	No	Description of the document
5	fileName	String	No	Name of the document
6	format	String	No	formatting
7	thumbUrl	String	No	Viber Thumbnails with Video Messaging
8	duration	Integer	No	Video duration of the Viber video message, range 0 - 600
9	fileType	String	No	File types for Viber file messages
10	latitude	String	No	Location Latitude Attribute
11	longitude	String	No	Location Longitude Attributes
12	locationName	String	No	Location Name
13	locationAddress	String	No	location address
14	addSecretRecommendation	String	No	Whatsapp type template, Valid when Category is Authentication and Component Type is Body, it prompts do not share the authentication code information to other people in the

				message body
	codeExpirationMinutes	Integer	No	Whatsapp Authentication template authentication code validity (minutes), valid only for Whatsapp type messages with Category as Authentication and Component Type as Footer (this information is displayed in the Footer location)
	buttons	Object[]	No	Applies only to BUTTONS components.

Button Object:

SN	Parameters	Type	Mandatory	Description
1	type	String	Yes	Button type, Values: PHONE_NUMBER: Dialing button URL: web button QUICK_REPLY: quick reply button Description: A Quick Reply button cannot be used in a template in conjunction with a Make Call button or a Link button. A template supports the addition of

				any of the following button combinations: two link buttons, or a link button and a call button. Templates of type Viber only allow one Button, and it can only be o URL type.
2	text	String	Yes	Name of the button display
3	phoneNumber	String	No	Phone number, only valid if button type is PHONE_NUMBER
4	url	String	No	The URL that will be accessed by clicking on the link button
5	urlType	String	No	URL type, values： static：static dynamic：dynamic
6	signatureHash	String	No	Whatsapp template required when Category is Authentication and Button Type is ONE_TAP, indicating the signature hash value for involking by Whatsapp
7	packageName	String	No	Whatsapp template required when Category is Authentication and Button Type is ONE_TAP, indicating the App package for involking by Whatsapp
8	autofillText	String	No	Whatsapp template required when Category is Authentication and Button Type is ONE_TAP, indicating the button text for Whatsap Autofill operation
9	IsOptOut	Boolean	No	Whatsapp template required when Category is Marketing and Button Type is QUICK_REPLY, indicating the unsubscription of marketing information, if the customer

				clicks on this button and configures the send control action on the chatapp platform, the subsequent marketing messages will not be sent to the customer.

The result of the returned JSON packet in the correct case is as follows:

```
{
    "data": {
        "templateType": "WHATSAPP",
        "components": [
            {
                "format": "IMAGE",
                "caption": "congratulations submitted successfully",
                "type": "HEADER",
                "url":
"https://material.5gomsg.com/group1/M00/16/8C/wKjEfmKK9xqADkDzAAd6qAhOKQ0140.png"
            },
            {
                "text": "Ticket has been submitted successfully. \nMr.$(Sendername), your parcel will be sent
to$(address). We will contact you on time.",
                "type": "BODY"
            },
            {
                "text": "Regards",
                "type": "FOOTER"
            },
            {
                "buttons": [
                    {
                        "urlType": "static",
                        "text": "Official",
                        "type": "URL",
                        "url": "https://DCAST.5gomsg.com/v/684941101398605824"
                    },
                    {
                        "phone_number": "+8618570334484",
                        "text": "Call me",
                        "type": "PHONE_NUMBER"
                    }
                ],
                "type": "BUTTONS"
            }
        ],
        "name": "test_notification_2022121301",
        "qualityScore": null,
        "auditStatus": "pass",
        "language": "en",
        "templateCode": "762475642058825728",
        "category": "TRANSACTIONAL",
        "example": {
            "Sendername": "Yang",
            "address": "Lugu Avenue"
        }
    },
    "code": "OK",
    "requestId": "da934052-49c8-4e3c-ac1b-9489213d1f1d",
    "success": true
}
```

An example of a returned JSON packet in the error case is shown below:

```
{
    "code": "DetailError.TemplateNotExist",
    "message": "Template does not exist, please check the template code ",
    "requestId": "2fbeddcf-8265-4485-8715-e13b5eee1aa9",
    "success": false
}
```

2.3.3 Template Creation Interface

Interface direction: third-party business system → DCAST platform

Interface Description: The third party business system creates a message template, which can be sent after the template has been reviewed and approved.

Interface Request:

```
https Request method: POST
https://{serverRoot}/a2p/api/waba/messageTemplates/create
```

Request Header:

SN	Paremeter	Type	Mandatory	Description
1	Content-Type	String	Yes	Required settings. application/json
2	Authorization	String	Yes	Interface credentials with the following data format: Bearer $ACCESS_TOKEN

Request Body:

SN	Paremeter	Type	Mandatory	Description
1	templateType	String	Yes	The template type. Values: WHATSAPP VIBER
2	name	String	Yes	Template name
3	language	String	Yes	template language
4	category	String	Yes	Whatsapp template categories. Values as:

				UTILITY: Transaction related. MARKETING: Marketing template AUTHENTICATION:Authentication Viber template classification values: text: plain text IMAGE: Pure Pictures text_image_button: text + image + button text_button: text + button Document video: video text_video: text + video text_video_button: text + video + button text_image: text + image
5	example	Map<String,String>	Yes	Examples of variables
6	components	Object []	Yes	List of message template components.

Component Object:

The same as Components Object in chapter 2.3.2

Button Object:

The same as Components Object in chapter 2.3.2

Request Example:

```
POST /a2p/api/waba/messageTemplates/create HTTP/1.1

Host: serverRoot

Authorization: Bearer a8bf1e65-1694-4854-8fa9-63f47ed352a9

Content-Type: application/json

Content-Length: 2144
```

```
{
    "category": "MARKETING",
    "name": "document_header_043",
    "language": "en",
    "templateType": "WHATSAPP",
    "components": [
        {
            "type": "BODY",
            "text": "Reply 1 to have our local team contact you for more details for free."
        },
        {
            "type": "HEADER",
            "text": "Image Header",
            "url": "$(linkVariable)",
            "format": "DOCUMENT"
        },
        {
            "type": "FOOTER",
            "text": "Reply CANCEL to stop receiving messages."
        },
        {
            "type": "BUTTONS",
            "buttons": [
                {
                    "type": "PHONE_NUMBER",
                    "text": "call me maybe",
                    "phoneNumber": "+8613888887889"
                },
                {
                    "type": "URL",
                    "text": "visit official website",
                    "urlType": "static",
                    "url": "https://www.baidu.com"
                }
            ]
        }
    ],
    "example": {
        "linkVariable":
"https://material.5gomsg.com/group1/M00/19/99/wKjEfmNo0HOAYJaTAAKXwAcDOf8040.pdf"
    }
}
```

Response description:

SN	Parameters		Type	Mandatory	Description
1	code		String	Yes	Return code
2	message		String	No	Returns an error message when the interface reports an error
3	data		object	Yes	Template detail information
4		templateCode	String	Yes	Template code
5		templateName	String	Yes	Template name

The result of the returned JSON packet in the successful case is as follows:

```
{
    "data": {
        "templateName": "document_header_043",
        "templateCode": "836114424745811968"
    },
    "code": "OK",
    "requestId": "3876f7cb-77fd-47a1-9f34-b6739b308e8f",
    "success": true
}
```

An example of a returned JSON packet in the error case is shown below:

```
{
    "code": "InvalidParameter.TemplateTypeNotSupport",
    "message": "The template type is not supported ",
    "requestId": "ec33ce8c-e871-4764-a39f-72f5e17adeef",
    "success": false
}
```

2.3.4 Template modification interface

Interface direction: third-party business system → DCAST platform

Interface Description: When the template audit does not pass, the third-party business system modifies the message template, and the message can be sent after the template audit passes.

Interface Request:

https Request Method: POST

https://{serverRoot}/a2p/api/waba/messageTemplates/modify

Request header:

SN	Parameters	Type	Mandatory	Description
1	Content-Type	String	Yes	Required settings. application/json
2	Authorization	String	Yes	Interface credentials with the following data format: Bearer $ACCESS_TOKEN

Request Body:

SN	Parameters	Type	Mandatory	Description
1	templateCode	String	Yes	Template Code

2	templateType	String	Yes	Template type values： WHATSAPP VIBER
3	language	String	Yes	Template language
4	category	String	No	Whatsapp template categories. Values as: UTILITY: Transaction related. MARKETING: Marketing template AUTHENTICATION:Authenticatio Viber template classification values: text: plain text IMAGE: Pure Pictures text_image_button: text + image + button text_button: text + button Document video: video text_video: text + video text_video_button: text + video + button text_image: text + image
5	example	Map<String,String>	Yes	Examples of variables
6	components	Object []	Yes	List of message template components.

Component Object：

The same as Components Object in chapter 2.3.2

Button Object：

The same as Components Object in chapter 2.3.2

Request Example：

```
POST /a2p/api/waba/messageTemplates/modify HTTP/1.1
Host: serverRoot
Authorization: Bearer a8bf1e65-1694-4854-8fa9-63f47ed352a9
Content-Type: application/json
Content-Length: 1182

{
    "templateCode": "836114424745811968",
    "category": "MARKETING",
    "language": "en",
    "templateType": "WHATSAPP",
    "components": [
        {
            "type": "BODY",
            "text": "Reply 2 to have our local team contact you for more details for free."
        },
        {
            "type": "HEADER",
            "text": "Document Header",
            "url": "$(linkVariable)",
            "format": "DOCUMENT"
        },
        {
            "type": "FOOTER",
            "text": "Reply 3 to stop receiving messages."
        },
        {
            "type": "BUTTONS",
            "buttons": [
                {
                    "type": "PHONE_NUMBER",
                    "text": "call me maybe",
                    "phoneNumber": "+8613888887888"
                },
                {
                    "type": "URL",
                    "text": "visit official website",
                    "urlType": "static",
                    "url": "https://www.sina.com.cn"
                }
            ]
        }
    ],
    "example": {
        "linkVariable":
"https://material.5gomsg.com/group1/M00/19/99/wKjEfmNo0HOAYJaTAAKXwAcDOf8040.pdf"
    }
}
```

Response description：

SN	Parameters		Type	Mandatory	Description
1	code		String	Yes	Return code
2	message		String	No	Returns an error message when the interface reports an error
3	data		object	Yes	Template detail information
4		templateCode	String	Yes	Template code

The result of the returned JSON packet in the successful case is as follows：

```
{
    "data": {
        "templateCode": "836114424745811968"
    },
    "code": "OK",
    "requestId": "27728d69-5172-4dd8-a4e1-00be7ffb35fa",
    "success": true
}
```

An example of a returned JSON packet in the error case is shown below:

```
{
    "code": "InvalidParameter.TemplateTypeNotSupport",
    "message": "Template type is not supported ",
    "requestId": "cc10e54f-e756-4c33-995b-13455cc135b5",
    "success": false
}
```

2.3.5 Template deletion interface

Interface direction: third-party business system → DCAST platform

Interface Description: Third-party business systems delete templates in the DCAST platform.

Interface Request:

https Request Method: POST

https://{serverRoot}/a2p/api/waba/messageTemplates/queryTemplateList

Request header:

SN	Parameters	Type	Mandatory	Description
1	Content-Type	String	Yes	Required settings. application/json

2	Authorization	String	Yes	Interface credentials with the following data format: Bearer $ACCESS_TOKEN

Request Body：

SN	Parameters	Type	Mandatory	Description
1	templateCode	String	Yes	Template Code
2	language	String	Yes	Template language
3	templateType	String	Yes	Template type values： WHATSAPP VIBER

Request Example：

```
POST /a2p/api/waba/messageTemplates/delete HTTP/1.1
Host: serverRoot
Authorization: Bearer a8bf1e65-1694-4854-8fa9-63f47ed352a9
Content-Type: application/json
Content-Length: 98


{
    "templateCode": "836114424745811968",
    "language": "en",
    "templateType": "WHATSAPP"
}
```

Response description：

SN	Parameters	Type	Mandatory	Description
1	code	String	Yes	Return code

2	message	String	No	Returns an error message when the interface reports an error

The result of the returned JSON packet in the successful case：

```
{
      "code": "OK",
      "requestId": "ccb775bf-00dd-445d-9bb0-6e834bf1b533",
      "success": true
}
```

An example of a returned JSON packet in the error case：

```
{
      "code": "ERR-OPERATION-CONFIG-028",
      "message": "Template data is not found ",
      "requestId": "11aa2357-75a1-4ee5-8728-a55dbf2b2921",
      "success": false
}
```

2.4 Chatbot access interface

2.4.1 Interface Interaction

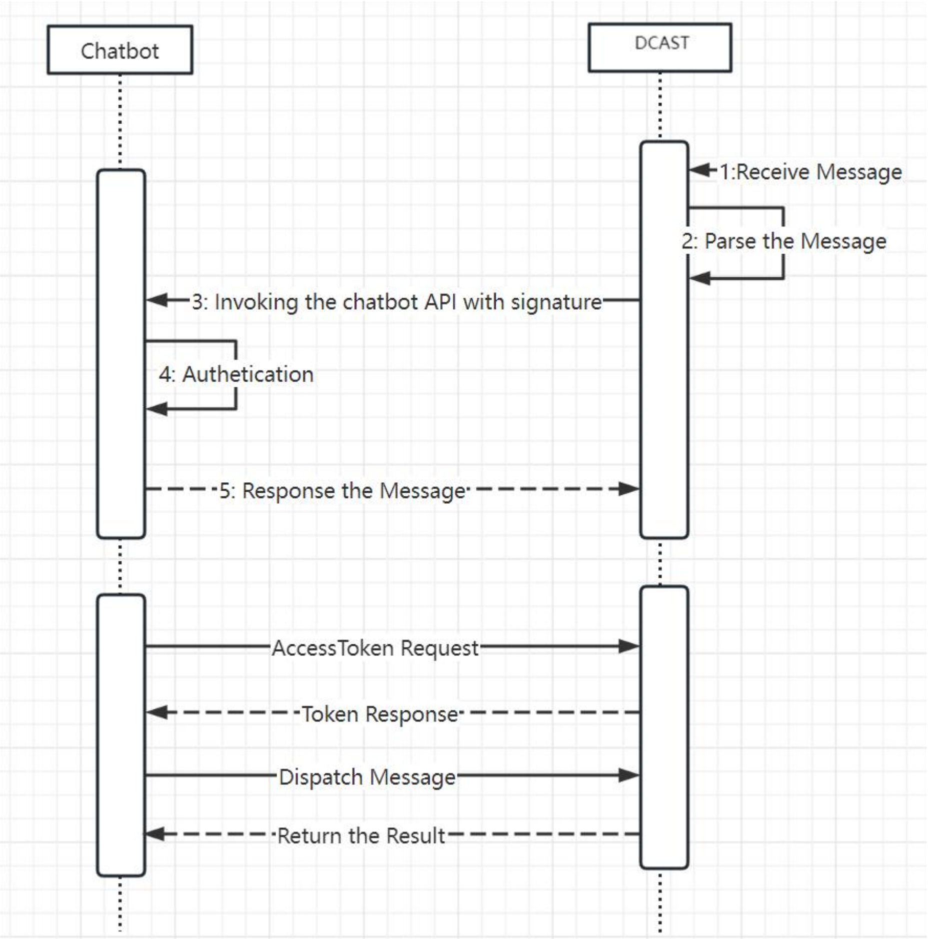

Figure 2- 3 Chatbot Interaction

2.4.2 DCAST platform sends messages to chabot

Interface direction: DCAST platform → chatbot

Interface description: Sending messages from DCAST platform to chatbots

Interface Request:

Request address: WebHook address configured by the customer on the DCAST platform

Request method: POST

Request header:

SN	Parameters	Mandatory	Description
1	AppId	Yes	Third party chatbot unique credentials, i.e. appId
2	Timestamp	Yes	The number of milliseconds in the current time
3	Signature	Yes	Signature (for specific algorithms, refer to 2.4.4 Signature Algorithm)

Request body:

SN	Parameters	Type	Mandatory	Description
1	mobileNo	String	Yes	Receiving number
2	serviceNo	String	Yes	Service Number of brand/senderID
3	msgId	String	Yes	Message ID
4	message	String	Yes	Message content

Example request:

```
{
    "mobileNo": "861380****888".
    "serviceNo": "109****333".
    "msgId": "ABC1112233343232333333DF".
    "message": "Hello World!"
}
```

Response description:

SN	Parameters	Type	Mandatory	Description
1	mobileNo	String	Yes	Receiving number
2	serviceNo	String	Yes	Service Number of Brand/senderID
3	message	String	No	Message content, if you need to return synchronously then return

				directly, if asynchronous return then set it to empty

Example response:

```
{
        "mobileNo": "86138****8888".
        "serviceNo": "109****2333".
        "message": "Hello"
}
```

2.4.3 Chatbot send messages to the DCAST platform

Please refer to.

2.2.1 Template message Delivery

2.2.2 Text message sending

2.2.3 File messaging

2.4.4 Signature Algorithm

```
import org.apache.commons.codec.digest.HmacAlgorithms;
import org.apache.commons.codec.digest.HmacUtils;

import javax.crypto.Mac;
import java.nio.charset.Charset;

public class SignatureUtils {


    /**
     * Generate a signature
     * @param appSecret
```

```
         * @param timestamp
         * @param messageBody
         * @return
         */
        public String generateSignature(String appSecret, String timestamp, String
messageBody) {
                Mac initializedMac =
HmacUtils.getInitializedMac(HmacAlgorithms.HMAC_SHA_256,
appSecret.getBytes(Charset.forName("utf8")));
                byte[] bytes = initializedMac.doFinal((timestamp +
messageBody).getBytes(Charset.forName("utf8")));
                StringBuilder hexString = new StringBuilder();
                for (byte b : bytes) {
                        String hex = Integer.toHexString(0xff & b);
                        if (hex.length() == 1) {
                                hexString.append('0');
                        }
                        hexString.append(hex);
                }
                return hexString.toString();
        }
}
```

2.5 Agent Access interface

2.5.1 Interface Interaction

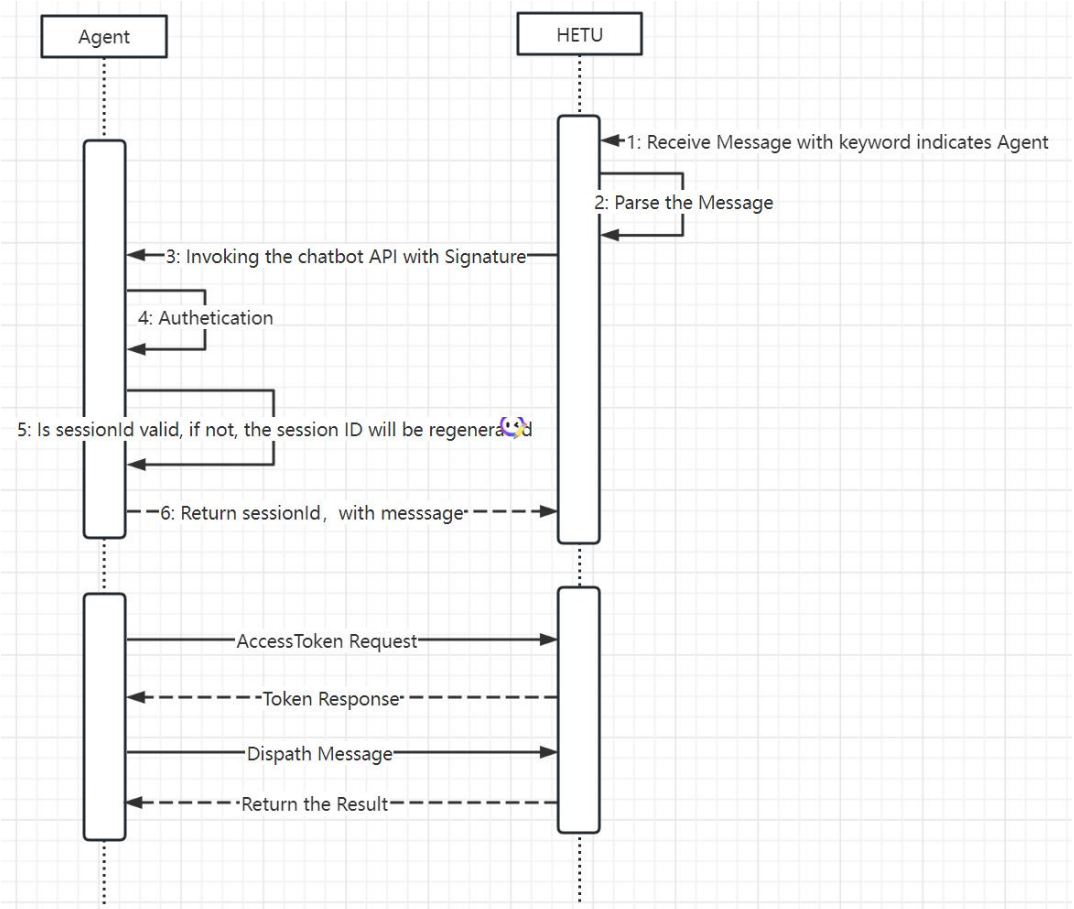
Agent
HETU
1: Receive Message with keyword indicates Agent
2: Parse the Message
3: Invoking the chatbot API with Signature
4: Authetication
5: Is sessionId valid, if not, the session ID will be regenerated
6: Return sessionId，with messsage
AccessToken Request
Token Response
Dispath Message
Return the Result

Figure 2-4 Team Inbox Agent Interaction

2.5.2 DCAST platform sends messages to Agents

Interface direction: DCAST platform →agents

Interface description: Sending messages from DCAST platform to agents

Interface Request:

Request address: WebHook address configured by the customer on the DCAST platform

Request method: POST

Request header:

SN	Parameters	Mandatory	Description
1	AppId	Yes	Third party chatbot unique credentials, i.e. appId
2	Timestamp	Yes	The number of milliseconds in the current time
3	Signature	Yes	Signature (for specific algorithms, refer to 2.4.4 Signature Algorithm)

Request body:

SN	Parameters	Type	Mandatory	Description
1	sessionId	String	No	Session Id (Session Id returned by the bot, mobileNo will be used as the primary key to create the session, DCAST platform will keep it for 30 minutes by default, expired and cleared)
2	mobileNo	String	Yes	Receiving number
3	serviceNo	String	Yes	Service Number of Brand/senderID
4	msgId	String	Yes	Message ID
5	message	String	Yes	Message content

Example request:

```
{
  "sessionId":"".
  "mobileNo": "86138****888".
  "serviceNo": "109****233".
  "msgId": "ABC1112233343232333333D".
  "message": "Hello World"
}
```

Response description:

SN	Parameters	Type	Mandatory	Description
1	sessionId	String	No	Session Id (Session Id returned by the session bot, mobileNo will be used as the primary key to create the session, DCAST platform will keep it for 30 minutes by default, expired and cleared)
2	mobileNo	String	Yes	Receiving number,
3	serviceNo	String	Yes	Service Number of Brand/senderID
4	sessionState	String	Yes	Session state (A: normal interaction state, X: closed state)
5	message	String	No	Message content, if you need to return synchronously to return directly, if the return is asynchronous then set it to empty

Response Example:

```
{
  "sessionId":"".
  "mobileNo": "86138****888".
  "serviceNo": "109****233".
  "sessionState":"".
  "message":""
}
```

2.5.3 Agents send messages to DCAST platform

Please refer to.

2.2.1 Template message Delivery

2.2.2 Text message sending

2.2.3 File messaging

3 Interface static parameters

3.1.1 Return Code Description

Return Code	Description
-1	System busy
0	Request successful
40001	AppSecret error when getting accessToken, or invalid accessToken
40002	Types of illegal credentials
40005	Illegal file types
40006	Illegal file size
40007	Illegal file id
40008	Illegal message types
40014	Illegal accessToken
40040	Number of files exceeds limit
41000	Missing required parameters
41001	Missing accessToken parameter
41002	Missing appid parameter
41004	Missing secret parameter
41005	Missing multimedia file data
41006	data max size is 100
41007	The messageType must be link or notification
42001	accessToken timeout
43001	GET request required
44001	Multimedia file is empty
44002	The POST packet is empty
45001	Interface Invoking exceeding limits
45010	Response time exceeds the limit
46001	Non-existent users
46002	Temporary user does not exist
47001	Parsing JSON content error
48001	The API function is not authorized
60001	Illegal content
60003	Invalid message types
60004	msgid Reaching the limit (up to 5 times)
60005	Missing templateCode parameter
60006	Non-existent templateCode

60007	Missing senders/receivers parameter
60008	number cannot exceed 1000
60009	Missing downgrade parameter
60010	Something went wrong, please try again later
60011	Missing bodyText parameter
60012	Incorrect parameter format
60013	Missing resourceCode parameter
60014	Invalid URL
70001	Scenario sending policy not configured

3.1.2 Language Code

Language	Code
Afrikaans	af
Albanian	sq
Arabic	ar
Azerbaijani	az
Bengali	bn
Bulgarian	bg
Catalan	ca
Simplified Chinese (Mainland China)	en_CN
Traditional Chinese (Hong Kong)	en_HK
Traditional Chinese (Taiwan)	en_TW
Croatian	hr
Czech	cs
Danish	da
Dutch	nl
English	en
English (UK)	en_GB
English (USA)	en_US
Estonian	et
Filipino	fil
Finnish	fi
French	fr
German	de
Greek	el
Gujarati	gu
Hausa	ha
Hebrew	he
Hindi	hi

Hungarian	hu
Indonesian	id
Irish language	ga
Italian	it
Japanese	ja
Kannada	kn
Kazakh	kk
Korean	ko
Laotian	lo
Latvian	lv
Lithuanian	lt
Macedonian	mk
Malay	ms
Malayalam	ml
Marathi	mr
Norwegian	nb
Persian	fa
Polish	pl
Portuguese (Brazil)	pt_BR
Portuguese (Portugal)	pt_PT
Punjabi	pa
Romanian	ro
Russian	ru
Serbian	sr
Slovakian	sk
Slovenian	sl
Spanish	es
Spanish (Argentina)	es_AR
Spanish (Spain)	es_ES
Spanish (Mexico)	es_MX
Swahili	sw
Swedish	sv
Tamil	ta
Telugu	te
Thai	th
Turkish	tr
Ukrainian	uk
Urdu	ur
Uzbek	uz
Vietnamese	vi
Zulu	zu