PINNACLE™
LET'S REACH OUT

RCS CORE API Documentation

www.pinnacle.in

Table of Contents

MANAGEMENT APIs

Base URL: https://rcsapi.pinnacle.in:447/api

Bot API

1. Get botId by botName

Method

GET

Description

This API is used to get botid from bot name.

URL

/v1/get-botId-by-botName?botName=

Request Headers

apikey	**xxxxxxxxxxxxxxxxxxxxxx**

Params

Field	**Description**	**Required/Optional**
botName	Name of bot to fetch botId	Required

Response

```
{
"status": "SUCCESS",
"code": 200,
"data": {
"message": "Request Successful!",
"botName": "botName",
"orgId": "xxxxxxxxxxxxxxx",
"botId": "xxxxxxxxxxxxxxx"
}
}
```

~~Create Bot API~~

Method

POST

Description

This API is used to create Bot.

URL

~~ /v1/create-bot~~

Request Headers

apikey	**xxxxxxxxxxxxxxxxxxxxxxxxxx**

Request Payload Description

Field Name	Description	Required / Optional
botType	Type of bot i.e. Domestic	Required
botMessageType	Bot message type (Promotional, Transactional, OTP)	Required
botName	Name of Bot Must be unique.	Required
botLogoImage.url	Url for bot logo	Required
botLogoImage.type	Type for bot logo can be one of the following image/jpeg, image/jpg, image/png	Required
botBannerImage.url	Url for banner image of bot	Required
botBannerImage.type	Type for bot banner image can be one of the following image/jpeg, image/jpg, image/png	Required
shortDescription	Description of created bot	Required
color	Specify a color for your agent with a minimum 4.5:1 contrast ratio relative to white.	Required
phoneNumbers	Array of phone numbers of organization to contact. This can be array of three phone numbers not more than that.	Required

phoneNumbers.country	Country Name	Required
phoneNumbers.countryCode	CountryCode of that country (for India: +91)	Required
phoneNumbers.number	Valid Phone Number of contact person without country code	Required
phoneNumbers.label	Label for contact person	Required
emails	Array of emails of contact persons of organization. This can be array of three emails not more than that.	Required
emails.email	Email of contact person (abc@gmail.com)	Required
emails.label	Label for the email.	Required
websites	Websites of Organization for bot.	Required
websites.url	Valid Url of website	Required
websites.label	Label for the website	Required
termsOfUseUrl	Terms of use urls of website	Required
privacyPolicyUrl	Privacy urls of website	Required
languagesSupported	Please specify the languages supported by the bot	Required
developmentPlatform	Style of APIs used “Google API”	Required
chatbotWebhook	https://rcsgateway.pinnacle.in:444 this is chatbot webhook	Required
Config. RelayWebhookUrl	Webhook url for client	Optional

Request body

Points to consider while sending the payload:

1) botLogoImage.url should be a valid image url which opens directly on browser.
2) botBannerImage.url should be a valid image url which opens directly on browser.
3) botLogoImage.type should be “image/jpeg”, ”image/jpg” or “image/png”.
4) phoneNumbers, emails, websites array cannot be more than 3.

```
{
  "botType": "Domestic",
  "botMessageType": "Promotional",
  "botName": "botName",
  "botLogoImage": {
    "url": "https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcTc9APxkj0xClmrU3PpMZglHQkx446nQPG6lA&s",
    "type": "image/jpeg"
  },
  "botBannerImage": {
    "url": "https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcTc9APxkj0xClmrU3PpMZglHQkx446nQPG6lA&s",
    "type": "image/jpeg"
  },
  "shortDescription": "Short MessageShort Message",
  "color": "#000000",
  "phoneNumbers": [
    {
      "country": "India",
      "countryCode": "+91",
      "number": "xxxxxxxxxx",
      "label": "xxx"
    },
    {
      "country": "India",
      "countryCode": "+91",
      "number": "xxxxxxxxxx",
      "label": "label"
    },
    {
      "country": "India",
      "countryCode": "+91",
      "number": "xxxxxxxxxx",
      "label": "label"
    }
  ],
  "emails": [
    {
      "email": "test@gmail.com",
      "label": "Label for Prim"
    },
    {
      "email": "test2@gmail.com",
      "label": " Other Email"
    }
  ],
  "websites": [
    {
      "url": "https://xyz.com",
      "label": "Primary Web"
    }
  ],
  "termsofUseUrl": "https://xyz.com",
  "privacyPolicyUrl": "https://xyz.com",
  "languagesSupported": "All",
  "developmentPlatform": "Google API",
  "chatbotWebhook": "https://rcsgateway.pinnacle.in:444/"
"config": { // optional field
        "relayWebhookUrl": https://rcsgateway.pinnacle.in:444/ // url used to relay message
//status to client
    }
}
```

Status

201

Response

```
{
"status": "SUCCESS",
"code": 201,
"data": {
"botId": "xxxxxxxxxx",
"botName": "botName",
"message": "Bot Created Successfully"
}
}
```

~~3. Verify Client Secret~~

Method

POST

Description

This API is used to verify client secret id.

URL

~~–/v1/verify-client-secret~~

Request Headers

apikey	xxxxxxxxxxxxxxxxxxxxxx

Request Payload Description

Fields	Description	Required/Optional
botId	Id against the bot which has to be verified. This can be fetched by get-botId-by-botName GET API.	Required
rbmClientId	Client Id for bot verification. Will be provided while create Bot process	Required
rbmClientSecret	Client secret for bot verification. Will be provided while create Bot Process	Required

Request Body

```
{
"botid": "xxxxxxxxxxxxxxxxx",
"rbmClientId": "xxxxxxxxxxxxxxxxx",
"rbmClientSecret": xxxxxxxxxxxxxxxxx"
}
```

Status

200

Response

```
Client Secret Data verification Submitted Successfully!
```

5. Get all Bots

Method

GET

Description

This API is used to get all bots

URL

/v1/get-all-bots

Request Headers

apikey	xxxxxxxxxxxxxxxxxxxxx

4. Get all Launched bots by orgId

Method

GET

Description

This API is used to get launched bots by org id.

URL

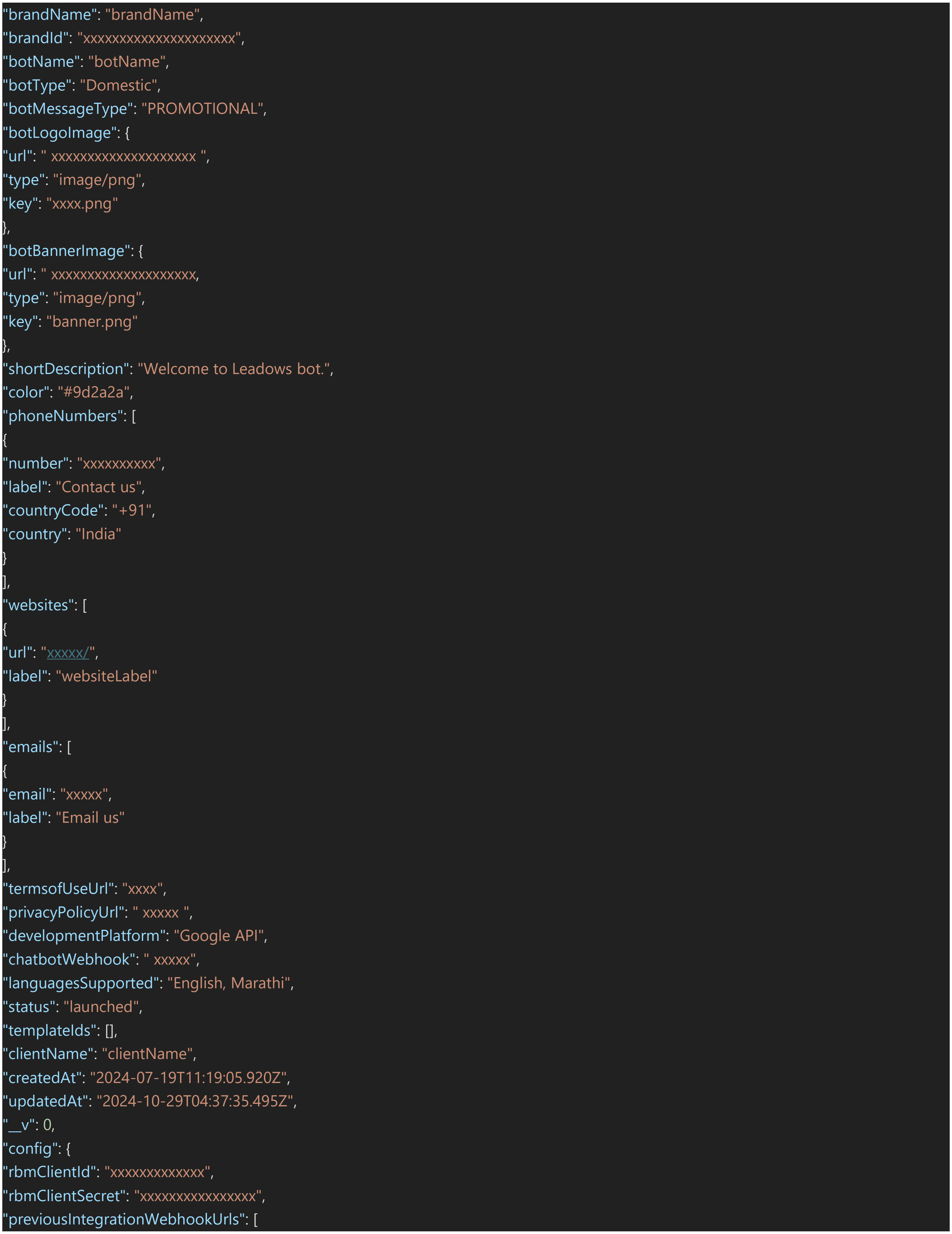
/v1/get-all-launched-bots-by-orgId

Request Headers

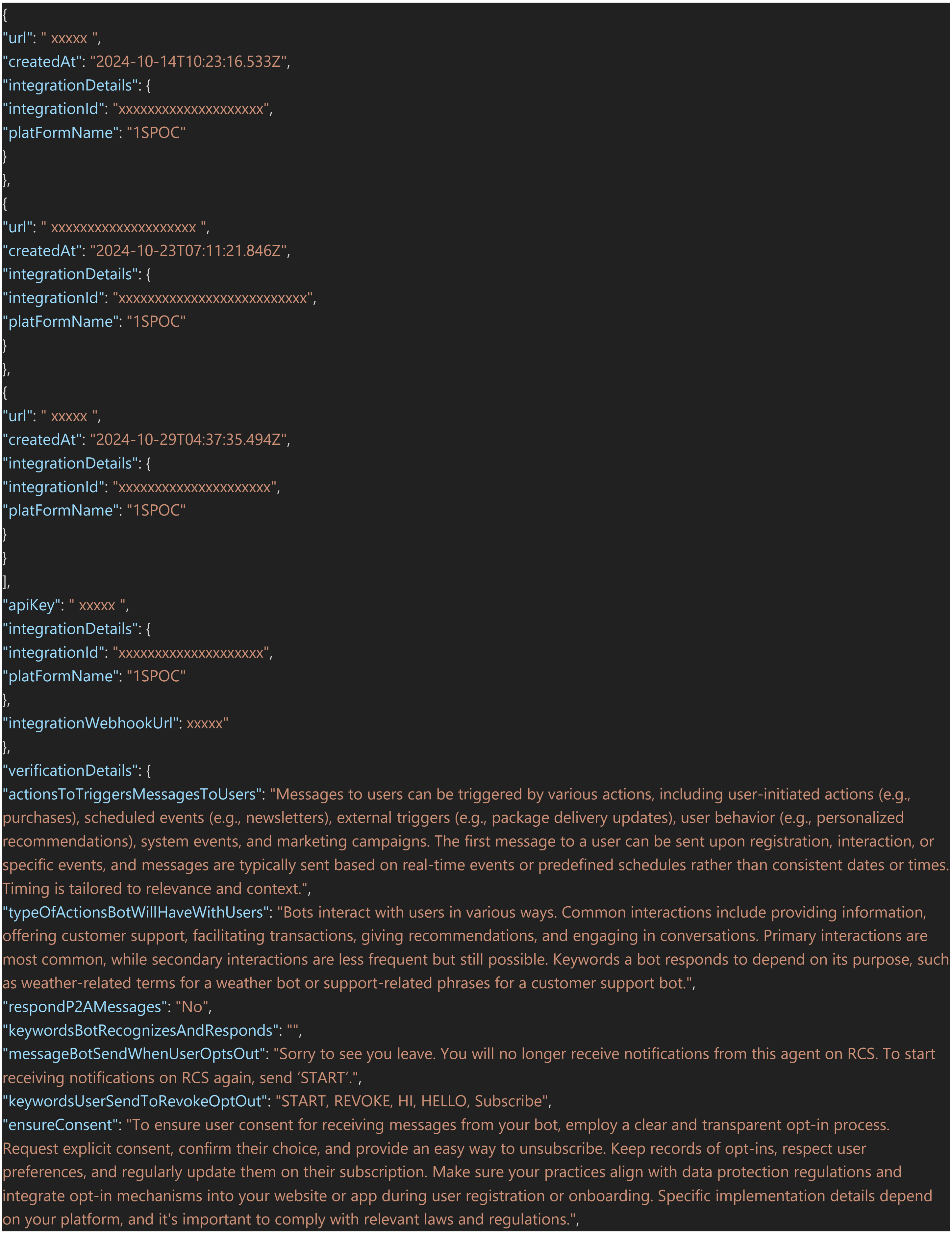
apikey	xxxxxxxxxxxxxxxxxxxxx

Response

```
{
"status": "SUCCESS",
"code": 200,
"data": {
"data": [
{
"_id": "xxxxxxxxxxxxxxxxxxxx",
"orgId": "xxxxxxxxxxxxxxxxxxxx",
```



```
"brandName": "brandName",
"brandId": "xxxxxxxxxxxxxxxxxxxxx",
"botName": "botName",
"botType": "Domestic",
"botMessageType": "PROMOTIONAL",
"botLogoImage": {
"url": " xxxxxxxxxxxxxxxxxxxx ",
"type": "image/png",
"key": "xxxx.png"
},
"botBannerImage": {
"url": " xxxxxxxxxxxxxxxxxxxx,
"type": "image/png",
"key": "banner.png"
},
"shortDescription": "Welcome to Leadows bot.",
"color": "#9d2a2a",
"phoneNumbers": [
{
"number": "xxxxxxxxxx",
"label": "Contact us",
"countryCode": "+91",
"country": "India"
}
],
"websites": [
{
"url": "xxxxx/",
"label": "websiteLabel"
}
],
"emails": [
{
"email": "xxxxx",
"label": "Email us"
}
],
"termsofUseUrl": "xxxx",
"privacyPolicyUrl": " xxxxx ",
"developmentPlatform": "Google API",
"chatbotWebhook": " xxxxx",
"languagesSupported": "English, Marathi",
"status": "launched",
"templateIds": [],
"clientName": "clientName",
"createdAt": "2024-07-19T11:19:05.920Z",
"updatedAt": "2024-10-29T04:37:35.495Z",
"__v": 0,
"config": {
"rbmClientId": "xxxxxxxxxxxxx",
"rbmClientSecret": "xxxxxxxxxxxxxxxx",
"previousIntegrationWebhookUrls": [
```



```
{
"url": " xxxxx ",
"createdAt": "2024-10-14T10:23:16.533Z",
"integrationDetails": {
"integrationId": "xxxxxxxxxxxxxxxxxxxx",
"platFormName": "1SPOC"
}
},
{
"url": " xxxxxxxxxxxxxxxxxxxx ",
"createdAt": "2024-10-23T07:11:21.846Z",
"integrationDetails": {
"integrationId": "xxxxxxxxxxxxxxxxxxxxxxxxxx",
"platFormName": "1SPOC"
}
},
{
"url": " xxxxx ",
"createdAt": "2024-10-29T04:37:35.494Z",
"integrationDetails": {
"integrationId": "xxxxxxxxxxxxxxxxxxxxx",
"platFormName": "1SPOC"
}
}
],
"apiKey": " xxxxx ",
"integrationDetails": {
"integrationId": "xxxxxxxxxxxxxxxxxxxx",
"platFormName": "1SPOC"
},
"integrationWebhookUrl": xxxxx"
},
"verificationDetails": {
"actionsToTriggersMessagesToUsers": "Messages to users can be triggered by various actions, including user-initiated actions (e.g.,
purchases), scheduled events (e.g., newsletters), external triggers (e.g., package delivery updates), user behavior (e.g., personalized
recommendations), system events, and marketing campaigns. The first message to a user can be sent upon registration, interaction, or
specific events, and messages are typically sent based on real-time events or predefined schedules rather than consistent dates or times.
Timing is tailored to relevance and context.",
"typeOfActionsBotWillHaveWithUsers": "Bots interact with users in various ways. Common interactions include providing information,
offering customer support, facilitating transactions, giving recommendations, and engaging in conversations. Primary interactions are
most common, while secondary interactions are less frequent but still possible. Keywords a bot responds to depend on its purpose, such
as weather-related terms for a weather bot or support-related phrases for a customer support bot.",
"respondP2AMessages": "No",
"keywordsBotRecognizesAndResponds": "",
"messageBotSendWhenUserOptsOut": "Sorry to see you leave. You will no longer receive notifications from this agent on RCS. To start
receiving notifications on RCS again, send ‘START’.",
"keywordsUserSendToRevokeOptOut": "START, REVOKE, HI, HELLO, Subscribe",
"ensureConsent": "To ensure user consent for receiving messages from your bot, employ a clear and transparent opt-in process.
Request explicit consent, confirm their choice, and provide an easy way to unsubscribe. Keep records of opt-ins, respect user
preferences, and regularly update them on their subscription. Make sure your practices align with data protection regulations and
integrate opt-in mechanisms into your website or app during user registration or onboarding. Specific implementation details depend
on your platform, and it's important to comply with relevant laws and regulations.",
"optOutKeywordsForBot": "UNSUBSCRIBE, QUIT, CANCEL, END, NO MORE, REMOVE, OPT-OUT, BLOCK",
"platformWhenUserRevokesOptout": "Default",
"messageBotSendWhenUserRevokesOptout": "Thanks for your request. You will start receiving RCS messages again from the Leadows
Promotional RCS Bot",
"botAccessInstructions": "HI, START, HELLO, HELP, CALL",
"screenshots": [
{
"url": " xxxxx ",
"type": "image/jpeg",
"key": " xxxxxxxxxxxxxxxxxxxx "
},
{
"url": "hxxxxx",
"type": "image/png",
"key": "image.png"
}
],
"videoUrl": " xxxxx",
"paymentStatus": "0"
},
"testDevices": [
"+91xxxxxxxxxx",
"+91xxxxxxxxxx",
"+91xxxxxxxxxx",
"+91xxxxxxxxxx",
"+91xxxxxxxxxx",
"+91xxxxxxxxxx",
]
}
],
"message": "All Bots Details Fetched"
}
}
```

5. Get bot details by botId

Method

GET

Description

This API is used to get bots details by botid.

URL

/v1/get-bot-details/{{botId}}

Request Headers

apikey	xxxxxxxxxxxxxxxxxxxxxx

Request Query

Fields	Description	Required/ Optional
botId	Id of a bot Can be fetched by get-botId-by-botName GET API	Required

Status

200

Response

```
{
"status": "SUCCESS",
"code": 200,
"data": {
"_id": "xxxxxxxxxxxxxxxxxxxxx",
"orgId": "xxxxxxxxxxxxxxxxxxxxxx",
"brandName": "brandName",
"brandId": "xxxxxxxxxxxxxxxxxx",
"botName": "botName",
"botType": "Domestic",
"botMessageType": "PROMOTIONAL",
"botLogoImage": {
"url": " xxxxx",
"type": "image/jpeg"
},
"botBannerImage": {
"url": " xxxxx",
"type": "image/jpeg"
},
"shortDescription": "Short MessageShort Message",
"color": "#000000",
"phoneNumbers": [
{
"number": "xxxxxxxxxx",
"label": " xxxxx",
"countryCode": "+91",
"country": "India"
},
{
"number": "xxxxxxxxxx",
"label": " xxxxx ",
"countryCode": "+91",
"country": "India"
},
{
"number": "xxxxxxxxxx",
"label": " xxxxx ",
"countryCode": "+91",
"country": "India"
}
],
"websites": [
{
"url": "xxxxx",
"label": " Primary Web"
}
],
"emails": [
{
"email": "test@gmail.com",
"label": "Label for Prim"
},
{
"email": "test2@gmail.com",
"label": " Other Email"
}
],
"termsofUseUrl": "xxxxx",
"privacyPolicyUrl": "xxxxx",
"developmentPlatform": "Google API",
"chatbotWebhook": "https://rcsgateway.pinnacle.in:444/",
"languagesSupported": "All",
"status": "new",
"templateIds": [],
"clientName": "clientName",
"testDevices": [],
"createdAt": "2024-10-28T04:59:07.242Z",
"updatedAt": "2024-10-28T04:59:07.242Z",
"__v": 0
}
}
```

~~6. Get Verify bot by botId~~

Method

GET

Description

This API is used to get bots details by botid.

URL

~~/v1/get-verify-bot/{{botId}}~~

Points to consider while sending the payload:

1.Bot should be verified.

Request Headers

apikey	xxxxxxxxxxxxxxxxxxxxx

Request Query

Fields	Description	Required/ Optional
botId	Id of a bot Can be fetched by get-botId-by-botName GET API	Required

Status

200

Response

```
{
"status": "SUCCESS",
"code": 200,
"data": {
"data": {
"verifyBotDetails": {
"_id": "xxxxxxxxxxxxxxxxxxxx",
"orgId": "xxxxxxxxxxxxxxxxxxx",
"brandName": "brandName",
"brandId": "xxxxxxxxxxxxxxxxxxxxx",
"botName": "botName",
"botType": "Domestic",
"botMessageType": "PROMOTIONAL",
"botLogoImage": {
"url": " xxxxx ",
"type": "image/png",
"key": "logo.png"
},
"botBannerImage": {
"url": "20banner.png",
"type": "image/png",
"key": "banner.png"
},
"shortDescription": "Welcome to xyz bot.",
"color": "#2ee5c7",
"phoneNumbers": [
{
"number": "xxxxxxxxxx",
"label": "Contact us",
"countryCode": "+91",
"country": "India"
}
],
"websites": [
{
```

```
"url": "https://xyz.com/",
"label": " websiteLabel"
}
],
"emails": [
{
"email": "xyz@gmail.com",
"label": "Email us"
}
],
"termsofUseUrl": "https://xyz.com/terms_conditions/",
"privacyPolicyUrl": "https://xyz.com/privacy_policy/",
"developmentPlatform": "Google API",
"chatbotWebhook": " xxxxx ",
"languagesSupported": "English, Marathi",
"status": "verified",
"templateIds": [],
"clientName": "clientName",
"createdAt": "2024-07-17T12:50:22.971Z",
"updatedAt": "2024-10-14T12:22:13.722Z",
"__v": 0,
"config": {
"rbmClientId": "xxxxxxxxxxxxxxxxxxxxx",
"rbmClientSecret": "xxxxxxxxxxxxxxxxxxxxxxx",
"previousIntegrationWebhookUrls": []
},
"verificationDetails": {
"actionsToTriggersMessagesToUsers": "Messages to users can be triggered by various actions, including user-initiated actions (e.g., purchases), scheduled events (e.g., newsletters), external triggers (e.g., package delivery updates), user behavior (e.g., personalized recommendations), system events, and marketing campaigns. The first message to a user can be sent upon registration, interaction, or specific events, and messages are typically sent based on real-time events or predefined schedules rather than consistent dates or times. Timing is tailored to relevance and context.",
"typeOfActionsBotWillHaveWithUsers": "Bots interact with users in various ways. Common interactions include providing information, offering customer support, facilitating transactions, giving recommendations, and engaging in conversations. Primary interactions are most common, while secondary interactions are less frequent but still possible. Keywords a bot responds to depend on its purpose, such as weather-related terms for a weather bot or support-related phrases for a customer support bot.",
"respondP2AMessages": "No",
"keywordsBotRecognizesAndResponds": "",
"messageBotSendWhenUserOptsOut": "Sorry to see you leave. You will no longer receive notifications from this agent on RCS. To start receiving notifications on RCS again, send ‘START’.",
"keywordsUserSendToRevokeOptOut": "START, REVOKE, HI, HELLO, Subscribe",
"ensureConsent": "To ensure user consent for receiving messages from your bot, employ a clear and transparent opt-in process. Request explicit consent, confirm their choice, and provide an easy way to unsubscribe. Keep records of opt-ins, respect user preferences, and regularly update them on their subscription. Make sure your practices align with data protection regulations and integrate opt-in mechanisms into your website or app during user registration or onboarding. Specific implementation details depend on your platform, and it's important to comply with relevant laws and regulations.",
"optOutKeywordsForBot": "UNSUBSCRIBE, QUIT, CANCEL, END, NO MORE, REMOVE, OPT-OUT, BLOCK",
"platformWhenUserRevokesOptout": "Default",
"messageBotSendWhenUserRevokesOptout": "Thanks for your request. You will start receiving RCS messages again from the Leadows RCS Bot",
"botAccessInstructions": "HI, START, HELLO, HELP, CALL",
"screenshots": [
{
"url": " xxxxx ",
"type": "image/jpeg",
"key": " xxxxx "
}
],
"videoUrl": " xxxxx "
}
},
"brandDetails": {
"_id": "xxxxxxxxxxxxxxxxxxxxxx",
"orgId": "xxxxxxxxxxxxxxxxxxxxxxx",
"brandName": "brandName",
"brandStatus": "VERIFIED",
"industryType": "Science, technology and engineering",
"officialBrandWebsite": "https://xyz.com/",
"brandLogo": {
"url": " xxxxx ",
"type": "image/png",
"key": “logo.png"
},
"contactPersonDetails": {
"firstName": "firstName",
"lastName": "lastName",
"emailId": "xyz@gmail.in",
"designation": "CEo",
"mobileNumber": "xxxxxxxxxx",
"country": "India",
"countryCode": "91"
},
"companyAddressDetails": {
"addressLine1": " xxxxx ",
"addressLine2": " xxxxx ",
"country": "India",
"state": " xxxxx ",
"city": " xxxxx ",
"zipCode": " xxxxx "
},
"businessVerificationDetails": {
"verifyBusinessName": [
{
"documentType": "PAN card of Company",
"selectedFile": {
"url": " xxxxx ",
"type": "application/pdf",
"key": " PAN.pdf",
"fileName": "Company PAN.pdf"
}
},
{
"documentType": "GST document",
"selectedFile": {
```

```
"url": " xxxxx ",
"type": "image/jpeg",
"key": "GST Certificate.jpeg",
"fileName": "GST Certificate.jpeg"
}
}
],
"verifyBusinessAddress": [
{
"documentType": "PAN card of Company",
"selectedFile": {
"url": xxxxx ",
"type": "application/pdf",
"key": "PAN.pdf",
"fileName": "Company PAN.pdf"
}
},
{
"documentType": "GST document",
"selectedFile": {
"url": " xxxxx ",
"type": "image/jpeg",
"key": "Certificate.jpeg",
"fileName": "GST Certificate.jpeg"
}
}
],
"termsAndConditions": false,
"agreeToPayVerificationFee": false
},
"createdAt": "2024-07-17T10:17:32.708Z",
"updatedAt": "2024-10-28T04:59:07.261Z",
"__v": 0
}
},
"message": "Bot Verified Successfully"
}
}
```

Template API :

1.Create Template API

Method

POST

Description

1. This API is used to create Templates.
2. There are three types of Templates:

- text_message
- rich_card
- carousel

URL

/v1/create-template

Request Headers

apikey	xxxxxxxxxxxxxxxxxxxxxx

Request Body

1) text_message

Request Payload Description

Field Name	Description	Required / Optional
botId	Id of bot Can be fetched by get-botId-by-botName GET API	Required
botMessageType	Bot message type can be one of the following 'Promotional', 'Transactional', 'OTP'. Bot message type should be same as botMessageType of bot.	Required
templateName	Name of the Template. Must be unique, without spaces.	Required
templateType	Type of template can be one of the following text_message	Required

	rich_card, carousel	
isSMSFallbackRequired	It is a Boolean value. If RCS message is disabled in endUser's device or if by some other reasons the RCS message doesn't reach to the endUser, by assigning isSMSFallbackRequired as true we can send the message via SMS as a fallback mechanism	Required
smsFallbackTemplateDetails	If isSMSFallbackRequired is true than we require SMS fallback details object.	Optional Required only when isSMSFallbackRequired is enabled.
smsFallbackTemplateDetails.senderId	Sender Id of SMS fallback. Will be provided.	Required if isSMSFallbackRequired is true
smsFallbackTemplateDetails.messageType	Can be 'TXT' or 'UNI'	Required if isSMSFallbackRequired is true
smsFallbackTemplateDetails.dltEntityId	Entity Id of SMS	Required if isSMSFallbackRequired is true
smsFallbackTemplateDetails.dltTempId	Template Id of SMS	Required if isSMSFallbackRequired is true
smsFallbackTemplateDetails.message	Contains message content of SMS. Message should be no longer than 2000 characters for English Messaging and for Unicode it should be 750 characters.	Required if isSMSFallbackRequired is true
textMessageDetails. messageContent	Contains content of message. Template Text message with custom variables. Variables should be in square brackets ([variable_name]).	Required

	e.g.”[Name], Time to go big or go home because it's India VS Australia FINALS!”	
textMessageDetails.suggestionList	Array of buttons. Can contains minimum = 0 and maximum= 10 buttons	Required If don’t need buttons in template send an empty array ([])
textMessageDetails.suggestionList.typeOfAction	typeOfAction can one of the following reply, url_action, dialer_action	Optional
textMessageDetails.suggestionList.suggestionText	Contains suggestion text for a button	If suggestionList contains button, then its required
textMessageDetails.suggestionList.urlAction	If ‘typeOfAction’ = ’url_action’, then this contains URL to get navigate on clicking the button	Required if ‘typeOfAction’ = ’url_action’
textMessageDetails.suggestionList.phoneNumberToDial.countryCode	If ‘typeOfAction’ = ’dialer_action’, then this contains countryCode for phoneNumbers	Required if ‘typeOfAction’=dialer_action’
textMessageDetails.suggestionList.phoneNumberToDial.number	If ‘typeOfAction’ = ’dialer_action’, then this contains number	Required if ‘typeOfAction’=dialer_action’

1) Payload (This payload is when isSMSFallbackRequired = false)

```
{
"botId": "xxxxxxxxxxxxxxxxxxxxx",
"botMessageType": "PROMOTIONAL",
"templateName": "templateName",
"templateType": "text_message",
"isSMSFallbackRequired": false,
"textMessageDetails": {
"messageContent": "Hello Morning everyone, welcome onboard! Join our community for more",
"suggestionsList": [
{
"typeOfAction": "reply",
"suggestionText": "Re-Test 1"
},
{
"typeOfAction": "url_action",
"suggestionText": "Re-Test 2",
"urlAction": "https://www.amazon.in"
},
{
"typeOfAction": "dialer_action",
"suggestionText": "button 3",
"phoneNumberToDial": {
"countryCode": "+91",
"number": "xxxxxxxxxx"
}
}
]
}
}
```

2) Payload (This payload is when isSMSFallbackRequired = true)

```
{
"botId": "xxxxxxxxxxxxxxxxxxxx",
"botMessageType": "OTP",
"templateName": "templateName",
"templateType": "text_message",
"isSMSFallbackRequired": true,
"smsFallbackTemplateDetails": {
"senderId": "PINCL",
"messageType": "TXT",
"dltEntityId": "1234",
"dltTempId": "1234",
"message": "Hello"
},
"textMessageDetails": {
"messageContent": "Hello Morning everyone, welcome onboard! Join our community for more",
"suggestionsList": [
{
"typeOfAction": "reply",
"suggestionText": "Re-Test 1"
},
{
"typeOfAction": "url_action",
"suggestionText": "Re-Test 2",
"urlAction": "https://www.amazon.in"
},
{
"typeOfAction": "dialer_action",
"suggestionText": "button 3",
"phoneNumberToDial": {
```

```
"countryCode": "+91",
"number": "xxxxxxxxxx"
}
}
]
}
}
```

Status

201

Response

```
{
"status": "SUCCESS",
"code": 201,
"data": {
"createdTemplateData": {
"botId": "xxxxxxxxxxxxxxxx",
"orgId": "xxxxxxxxxxxxxxxxxxxx",
"botMessageType": "OTP",
"templateName": "templateName",
"templateType": "text_message",
"status": "pending",
"textMessageDetails": {
"messageContent": "Hello Morning everyone, welcome onboard! Join our community for more",
"suggestionsList": [
{
"typeOfAction": "reply",
"suggestionText": "Re-Test 1",
"suggestionId": "cc07d634-5a78-4b97-b393-82525302c184"
},
{
"typeOfAction": "url_action",
"suggestionText": "Re-Test 2",
"suggestionId": "109542f9-f89e-4e63-b04b-69d389261734",
"urlAction": "https://www.amazon.in"
},
{
"typeOfAction": "dialer_action",
"suggestionText": "button 3",
"suggestionId": "93d1d547-e22c-4271-8c41-bf67d94ba3ea",
"phoneNumberToDial": {
"countryCode": "+91",
"number": "xxxxxxxxxx"
}
}
]
},
"isSMSFallbackRequired": true,
"smsFallbackTemplateDetails": {
"senderId": "PINCL",
"message": "Hello",
"dltEntityId": "xxxxxxxxxxxxx",
"dltTempId": "xxxxxxxxxxxxx",
"messageType": "TXT"
},
"variables": [],
"sampleS3File": {
"url": " xxxxx ",
"type": "text/csv",
"key": " xxxxx.csv"
},
"_id": "xxxxxxxxxxxxxxx",
"createdAt": "2024-10-29T06:19:49.392Z",
"updatedAt": "2024-10-29T06:19:49.392Z",
"__v": 0
},
"message": "Template created successfully."
}
}
```

2) rich_card

Request Payload Description

Field Name	**Description**	**Required / Optional**
botId	Id of bot Can be fetched by get-botId-by-botName GET API	Required
botMessageType	Bot message type can be one of the following ‘Promotional’, ‘Transactional’, ‘OTP’	Required
templateName	Name of the Bot. Must be unique.	Required
templateType	Type of template can be one of the following text_message rich_card, carousel	Required
isSMSFallbackRequired	It is a Boolean value. If RCS message is disabled in endUser’s device or if by some other reasons the RCS message doesn’t reach to the endUser, by assigning isSMSFallbackRequired as true we can send the message via SMS as a fallback mechanism	Required
smsFallbackTemplateDetails	If isSMSFallbackRequired is true than we require SMS fallback details object.	Optional Required only when isSMSFallbackRequired is enabled.
smsFallbackTemplateDetails.senderId	Sender Id of SMS fallback. Will be provided.	Required if isSMSFallbackRequired is true
smsFallbackTemplateDetails.messageT	Can be ‘TXT’ or ‘UNI’	Required if isSMSFallbackRequired is

ype		true
smsFallbackTemplateDetails.dltEntityId	Entity Id of SMS	Required if isSMSFallbackRequired is true
smsFallbackTemplateDetails.dltTempId	Template Id of SMS	Required if isSMSFallbackRequired is true
smsFallbackTemplateDetails.message	Contains message content of SMS	Required if isSMSFallbackRequired is true
richCardStandAloneDetails.cardOrientation	It can be ‘HORIZONTAL’ or ‘VERTICAL’	Required
richCardStandAloneDetails.cardAlignment	If cardOrientation=’HORIZONTAL’, then cardAlignment field is required. Value can be Either “LEFT” or “RIGHT”	Required if cardOrientation=’HORIZONTAL’
richCardStandAloneDetails.mediaHeight	If cardOrientation=’VERTICAL’, then mediaHeight field is required. MediaHeight can be SHORT_HEIGHT or MEDIUM_HEIGHT	Required if cardOrientation=’VERTICAL’
richCardStandAloneDetails.cardTitle	Title of the card	Required
richCardStandAloneDetails.cardDescription	Description of the card	Required
richCardStandAloneDetails.mediaType	For video mediaType will be ‘video’ and for image or gif it will be ‘image’	Required
richCardStandAloneDetails.thumbnail.url	richCardStandAloneDetails.thumbnail.url contains url for the thumbnail. The thumbnail url has to be of image only. Maximum limit of thumbnailurl is 100 KB	Required if mediaType = ’video’
richCardStandAloneDetails.media.url	Contains Url of the media. If mediaType=’video’, URL must be a valid video Url. If mediaType=’image’, URL must be a valid image Url or gif URL. Maximum limit of media url is 10 MB.	Required
richCardStandAloneDetails.suggestionList	Array of buttons. Can contains minimum = 0 and maximum= 11 buttons	Required **NOTE**: If don’t need buttons in template send an empty array ([])
richCardStandAloneDetails.suggestionList.typeOfAction	typeOfAction can one of the following reply, url_action, dialer_action	Optional
richCardStandAloneDetails.suggestionList.suggestionText	Contains suggestion text for a button	If suggestionList contains button, then its required
richCardStandAloneDetails.suggestionList.urlAction	If ‘typeOfAction’ = ’url_action’, then this conations URL to get navigate on clicking the button	Required if ‘typeOfAction’ = ’url_action’
richCardStandAloneDetails.suggestionList.phoneNumberToDial.countryCode	If ‘typeOfAction’ = ’dialer_action’, then this conations countryCode for phoneNumbers	Required if ‘typeOfAction’=dialer_action’

richCardStandAloneDetails.suggestionList.phoneNumberToDial.number	If ‘typeOfAction’ = ’dialer_action’, then this conations number	Required if ‘typeOfAction’=dialer_action’

1) **Payload (**When cardOrientation: ”HORIZONTAL”**)**

Points to consider while sending the payload:

- When cardOrientation: ”HORIZONTAL” we require a field cardAlignment.
- If mediaType is image thumbnail.url field is not required.
- If mediaType is video thumbnail.url is required field which can only be image and should have valid image extension.

```
{
"botId": "xxxxxxxxxxxxxxxxxxxxx",
"templateName": "templateName",
"templateType": "rich_card",
"botMessageType": "PROMOTIONAL",
"isSMSFallbackRequired": false,
"richCardStandAloneDetails": {
"cardOrientation": "HORIZONTAL",
"cardAlignment": "LEFT",
"cardTitle": "Happy Ganesh Chaturthi!",
"cardDescription": "Ganesh Chaturthi!",
"mediaType": "image",
"media": {
"url": " xxxxx.gif"
},
"suggestionsList": [
{
"typeOfAction": "reply",
"suggestionText": "Re-Test 1"
},
{
"typeOfAction": "url_action",
"suggestionText": "Re-Test 2",
"urlAction": "https://www.amazon.in/"
},
{
"typeOfAction": "dialer_action",
"suggestionText": "button 3",
"phoneNumberToDial": {
"countryCode": "+91",
"number": "xxxxxxxxxx"
}
}
]
}
}
```

Status

200

Response

```
{
"status": "SUCCESS",
"code": 200,
"data": {
"createdTemplateData": {
"botId": "xxxxxxxxxxxxxxxxxxxxx",
"orgId": "xxxxxxxxxxxxxxxxxxxxxx",
"botMessageType": "PROMOTIONAL",
"templateName": "templateName",
"templateType": "rich_card",
"status": "pending",
"richCardStandAloneDetails": {
"cardOrientation": "HORIZONTAL",
"cardAlignment": "LEFT",
"media": {
"url": " xxxxx.gif",
"type": "image/gif"
},
"cardTitle": "Happy Ganesh Chaturthi!",
"cardDescription": "Ganesh Chaturthi!",
"suggestionsList": [
{
"typeOfAction": "reply",
"suggestionText": "Re-Test 1",
"suggestionId": "8f07eee3-bed7-4122-be06-dc4dc1e607e3"
},
{
"typeOfAction": "url_action",
"suggestionText": "Re-Test 2",
"suggestionId": "9bb81fb7-111b-4567-9fbb-15d357229185",
"urlAction": "https://www.amazon.in/"
},
{
"typeOfAction": "dialer_action",
"suggestionText": "button 3",
"suggestionId": "f29683e0-bda2-446c-adca-5563c4f947d7",
"phoneNumberToDial": {
"countryCode": "+91",
"number": "xxxxxxxxxx"
}
}
]
},
"isSMSFallbackRequired": false,
"variables": [],
"sampleS3File": {
"url": " xxxxx ",
"type": "text/csv",
"key": “xxxxx.csv"
},
"_id": "xxxxxxxxxxxxxxxxxxxxxxxxx",
"createdAt": "2024-10-29T06:25:06.644Z",
"updatedAt": "2024-10-29T06:25:06.644Z",
"__v": 0
},
"message": "Template created successfully."
}
}
```

2) Payload (When cardOrientation: ”VERTICAL”)

Points to consider while sending the payload:

- When cardOrientation: ”VERTICAL” we require a field cardAlignment is not required.
- If mediaType is image thumbnail.url field is not required
- If mediaType is video thumbnail.url is required field which can only be image and should have valid image extension.

```
{
"botId": "xxxxxxxxxxxxxxxxxxxxx",
"templateName": "templateName",
"templateType": "rich_card",
"botMessageType": "PROMOTIONAL",
"isSMSFallbackRequired": false,
"richCardStandAloneDetails": {
"cardOrientation": "VERTICAL",
"mediaHeight": "MEDIUM_HEIGHT",
"cardTitle": "Happy Ganesh Chaturthi!",
"cardDescription": "Ganesh Chaturthi!",
"mediaType": "image",
"media": {
"url": "xxxxx"
},
"suggestionsList": [
{
"typeOfAction": "reply",
"suggestionText": "Re-Test 1"
},
{
"typeOfAction": "url_action",
"suggestionText": "Re-Test 2",
"urlAction": "https://www.amazon.in/"
},
{
```

```
"typeOfAction": "dialer_action",
"suggestionText": "button 3",
"phoneNumberToDial": {
"countryCode": "+91",
"number": "xxxxxxxxxx"
}
}
]
}
}
```

3) Payload (When mediaType: ”video”)

```
{
"botId": "xxxxxxxxxxxxxxxxxxxxx",
"templateName": "templateName",
"templateType": "rich_card",
"botMessageType": "PROMOTIONAL",
"isSMSFallbackRequired": false,
"richCardStandAloneDetails": {
"cardOrientation": "HORIZONTAL",
"cardAlignment": "LEFT",
"cardTitle": "Happy Ganesh Chaturthi!",
"cardDescription": "Ganesh Chaturthi!",
"mediaType": "video",
"media": {
"url": " xxxxx "
},
"thumbnail": {
"url": " xxxxx png"
},
"suggestionsList": [
{
"typeOfAction": "reply",
"suggestionText": "Re-Test 1"
},
{
"typeOfAction": "url_action",
"suggestionText": "Re-Test 2",
"urlAction": "https://www.amazon.in/"
},
{
"typeOfAction": "dialer_action",
"suggestionText": "button 3",
"phoneNumberToDial": {
"countryCode": "+91",
"number": "xxxxxxxxxx"
}
}
]
}
}
```

3) Carousel

Request Payload Description

Field Name	**Description**	**Require**
botId	Id of bot Can be fetched by get-botId-by-botName GET API	Required
botMessageType	Bot message type can be one of the following 'Promotional', 'Transactional', 'OTP'	Required
templateName	Name of the Bot. Must be unique.	Required
templateType	Type of template can be one of the following text_message rich_card, carousel	Required
isSMSFallbackRequired	It is a Boolean value. If RCS message is disabled in endUser's device or if by some other reasons the RCS message doesn't reach to the endUser, by assigning isSMSFallbackRequired as true we can send the message via SMS as a fallback mechanism	Required
smsFallbackTemplateDetails	If isSMSFallbackRequired is true than we require SMS fallback details object.	Optional Required isSMSFall enabled.
smsFallbackTemplateDetails.senderId	Sender Id of SMS fallback. Will be provided.	Required isSMSFall true
smsFallbackTemplateDetails.messageType	Can be 'TXT' or 'UNI'	Required isSMSFall true
smsFallbackTemplateDetails.dltEntityId	Entity Id of SMS	Required isSMSFall true
smsFallbackTemplateDetails.dltTempId	Template Id of SMS	Required isSMSFall true
smsFallbackTemplateDetails.message	Contains message content of SMS	Required isSMSFall true
richCardCarouselDetails	Contains details of carousel template	Required
richCardCarouselDetails.cardWidth	It can be 'SMALL_WIDTH' or 'MEDIUM_WIDTH'	Required
richCardCarouselDetails.mediaHeight	It can be 'SHORT_HEIGHT' or	Required

	‘MEDIUM_HEIGHT’	
richCardCarouselDetails.cardsList	Array of cards. Must contain minimum = 2 and maximum = 10 cards	Required
richCardCarouselDetails.cardsList.cardId	Id of the card	Required
richCardCarouselDetails.cardsList.cardTitle	Title of the card	Required
richCardCarouselDetails.cardsList.cardDescription	Description of the card	Required
richCardCarouselDetails.cardsList.mediaType	For video mediaType will be ‘video’ and for image or gif it will be ‘image’	Required
richCardCarouselDetails.cardsList.cardMedia.url	Contains Url of the media. If mediaType=’video’, URL must be a valid video Url. If mediaType=’image’, URL must be a valid image Url or gif URL. Maximun limit is 10 MB	Required
richCardCarouselDetails.cardsList.thumbnail.url	richCardStandAloneDetails.thumbnail.url contains url for the thumbnail. The thumbnail url has to be of image. Maximun limit is 100 KB	Required mediaTyp
richCardCarouselDetails.suggestionList	Array of buttons. Must contains minimum = 0 and maximum= 11 buttons	Required **NOTE**: If in templa array ([]
richCardCarouselDetails.suggestionList.typeOfAction	typeOfAction can one of the following reply, url_action, dialer_action	Optional
richCardCarouselDetails.suggestionList.suggestionText	Contains suggestion text for a button	If suggest button, then its r
richCardCarouselDetails.suggestionList.urlAction	If ‘typeOfAction’ = ’url_action’, then this conations URL to get navigate on clicking the button	Required ‘typeOfA
richCardCarouselDetails.suggestionList.phoneNumberToDial.countryCode	If ‘typeOfAction’ = ’dialer_action’, then this conations countryCode for phoneNumbers	Required ‘typeOfA
richCardCarouselDetails.suggestionList.phoneNumberToDial.number	If ‘typeOfAction’ = ’dialer_action’, then this conations number	Required ‘typeOfA

Payload

- If mediaType is image thumbnail.url field is not required.
- If mediaType is video thumbnail.url is required field which can only be image and should have valid image extension.

```
{
"botId": "xxxxxxxxxxxxxxx",
"botMessageType": "PROMOTIONAL",
"templateName": "templateName",
"templateType": "carousel",
"isSMSFallbackRequired": false,
"richCardCarouselDetails": {
"cardWidth": "SMALL_WIDTH",
"mediaHeight": "SHORT_HEIGHT",
"cardsList": [
{
"cardId": "123",
"cardTitle": "Greetings",
"cardDescription": "Greetings !",
"mediaType": "image",
"cardMedia": {
"url": " xxxxx.gif"
},
"suggestionsList": [
{
"typeOfAction": "dialer_action",
"phoneNumberToDial": {
"countryCode": "+91",
"number": " xxxxxxxxxxxxxxxxxxxx "
},
"suggestionText": "Call Now"
}
]
},
{
"cardId": "123",
"cardTitle": "Greetings",
"cardDescription": "Greetings !",
"mediaType": "video",
"cardMedia": {
"url": " xxxxx.mp4"
},
"cardThumbnail": {
"url": " xxxxx png"
},
"suggestionsList": [
{
"typeOfAction": "reply",
"suggestionText": "Re-Test 1"
},
{
"typeOfAction": "url_action",
"suggestionText": "Re-Test 2",
"urlAction": "https://www.amazon.in/"
},
{
"typeOfAction": "dialer_action",
"suggestionText": "button 3",
"phoneNumberToDial": {
"countryCode": "+91",
"number": "xxxxxxxxxx"
}
}
]
}
]
}
}
```

Status

200

Response

```
{
"status": "SUCCESS",
"code": 200,
"data": {
"createdTemplateData": {
"botId": "xxxxxxxxxxxxxxxxx",
"orgId": "xxxxxxxxxxxxxxxxx",
"botMessageType": "PROMOTIONAL",
"templateName": "templateName",
"templateType": "carousel",
"status": "pending",
"richCardCarouselDetails": {
"cardWidth": "SMALL_WIDTH",
"mediaHeight": "SHORT_HEIGHT",
"cardsList": [
{
"cardId": "123",
"cardMedia": {
"url": " xxxxx.gif",
"type": "image/gif"
},
"cardTitle": "Greetings",
"cardDescription": "Greetings !",
"suggestionsList": [
{
"typeOfAction": "dialer_action",
"suggestionText": "Call Now",
"suggestionPostback": "call_now",
"phoneNumberToDial": {
"countryCode": "+91",
"number": "xxxxxxxxxx"
}
}
]
},
{
"cardId": "123",
"cardMedia": {
"url": " xxxxx.mp4",
"type": "video/mp4"
},
"cardThumbnail": {
"url": " xxxxx.png",
"type": "image/png"
},
"cardTitle": "Greetings",
"cardDescription": "Greetings !",
"suggestionsList": [
{
"typeOfAction": "reply",
"suggestionText": "Re-Test 1"
},
{
"typeOfAction": "url_action",
"suggestionText": "Re-Test 2",
"urlAction": "https://www.amazon.in/"
},
{
"typeOfAction": "dialer_action",
"suggestionText": "button 3",
"phoneNumberToDial": {
"countryCode": "+91",
"number": "xxxxxxxxxx"
}
}
]
}
]
},
"isSMSFallbackRequired": false,
"variables": [],
"sampleS3File": {
"url": " xxxxx ",
"type": "text/csv",
"key": “xxxxx.csv"
},
"_id": "xxxxxxxxxxxxxxxx",
"createdAt": "2024-10-29T06:37:12.653Z",
"updatedAt": "2024-10-29T06:37:12.653Z",
"__v": 0
},
"message": "Template created successfully."
}
}
```

2. Update Template

Method

POST

Description

1. This API is used to update template, templateName and templateType, botMessageType can't be changed.
2. Only Pending and rejected template can be updated.
3. Approved template cannot be updated.

URL

/v1/update-template

Request Headers

Fields	Description	Required/Optional
apikey	xxxxxxxxxxxxxxxxxxxx	Required
templateId	Id of template to be updated. templateId can be fetched by get-templateId-by-templateName GET API	Required

NOTE: templateName and templateType cannot be updated hence not required in payload

Request Body

1) **text_message**

Request Payload Description

Field Name	Description	Required / Optional
botId	Id of bot Can be fetched by get-botId-by-botName GET API	Required
isSMSFallbackRequired	It is a Boolean value. If RCS message is disabled in endUser's device or if by some other reasons the	Required

	RCS message doesn’t reach to the endUser, by assigning isSMSFallbackRequired as true we can send the message via SMS as a fallback mechanism	
smsFallbackTemplateDetails	If isSMSFallbackRequired is true than we require SMS fallback details object.	Optional Required only when isSMSFallbackRequired is enabled.
smsFallbackTemplateDetails.senderId	Sender Id of SMS fallback. Will be provided.	Required if isSMSFallbackRequired is true
smsFallbackTemplateDetails.messageType	Can be ‘TXT’ or ‘UNI’	Required if isSMSFallbackRequired is true
smsFallbackTemplateDetails.dltEntityId	Entity Id of SMS	Required if isSMSFallbackRequired is true
smsFallbackTemplateDetails.dltTempId	Template Id of SMS	Required if isSMSFallbackRequired is true
smsFallbackTemplateDetails.message	Contains message content of SMS. Message should be no longer than 2000 characters for English Messaging and for Unicode it should be 750 characters.	Required if isSMSFallbackRequired is true
textMessageDetails. messageContent	Contains content of message.	Required
textMessageDetails.suggestionList	Array of buttons. Can contains minimum = 0 and maximum= 10 buttons	Required If don’t need buttons in template send an empty array ([])
textMessageDetails.suggestionList.typeOfAction	typeOfAction can one of the following reply, url_action, dialer_action	Optional
textMessageDetails.suggestionList.suggestionText	Contains suggestion text for a button	If suggestionList contains button, then its required

textMessageDetails.suggestionList.urlAction	If ‘typeOfAction’ = ’url_action’, then this contains URL to get navigate on clicking the button	Required if ‘typeOfAction’ = ’url_action’
textMessageDetails.suggestionList.phoneNumberToDial.countryCode	If ‘typeOfAction’ = ’dialer_action’, then this contains countryCode for phoneNumbers	Required if ‘typeOfAction’=dialer_action’

Payload

```
{
"botId": "669a4ba982ab2f0001956969",
"isSMSFallbackRequired": false,
"textMessageDetails": {
"messageContent": "Hello Morning everyone, welcome onboard! Join our community for more",
"suggestionsList": [
{
"typeOfAction": "reply",
"suggestionText": "Re-Test 1"
},
{
"typeOfAction": "url_action",
"suggestionText": "Re-Test 2",
"urlAction": "https://www.amazon.in"
},
{
"typeOfAction": "dialer_action",
"suggestionText": "button 3",
"phoneNumberToDial": {
"countryCode": "+91",
"number": " xxxxx "
}
}
]
}
}
```

Status

200

Response

```
{
"status": "SUCCESS",
"code": 200,
"data": {
"message": "Template updated successfully."
}
}
```

2) rich_card

Request Payload Description

Field Name	Description	Required / Optional
botId	Id of bot Can be fetched by get-botId-by-botName GET API	Required
isSMSFallbackRequired	It is a Boolean value. If RCS message is disabled in endUser's device or if by some other reasons the RCS message doesn't reach to the endUser, by assigning isSMSFallbackRequired as true we can send the message via SMS as a fallback mechanism	Required
smsFallbackTemplateDetails	If isSMSFallbackRequired is true than we require SMS fallback details object.	Optional Required only when isSMSFallbackRequired is enabled.
smsFallbackTemplateDetails.senderId	Sender Id of SMS fallback. Will be provided.	Required if isSMSFallbackRequired is true
smsFallbackTemplateDetails.messageType	Can be 'TXT' or 'UNI'	Required if isSMSFallbackRequired is true
smsFallbackTemplateDetails.dltEntityId	Entity Id of SMS	Required if isSMSFallbackRequired is true
smsFallbackTemplateDetails.dltTempId	Template Id of SMS	Required if isSMSFallbackRequired is true
smsFallbackTemplateDetails.message	Contains message content of SMS	Required if isSMSFallbackRequired is true
richCardStandAloneDetails.cardOrientation	It can be 'HORIZONTAL' or 'VERTICAL'	Required

richCardStandAloneDetails.cardAlignment	If cardOrientation=’HORIZONTAL’, then cardAlignment field is required. Value can be Either “LEFT” or “RIGHT”	Required if cardOrientation=’HORIZONTAL’
richCardStandAloneDetails.mediaHeight	If cardOrientation=’VERTICAL’, then mediaHeight field is required. MediaHeight can be SHORT_HEIGHT or MEDIUM_HEIGHT	Required if cardOrientation=’VERTICAL’
richCardStandAloneDetails.cardTitle	Title of the card	Required
richCardStandAloneDetails.cardDescription	Description of the card	Required
richCardStandAloneDetails.mediaType	For video mediaType will be ‘video’ and for image or gif it will be ‘image’	Required
richCardStandAloneDetails.thumbnail.url	richCardStandAloneDetails.thumbnail.url contains url for the thumbnail. The thumbnail url has to be of image only.	Required if mediaType = ’video’
richCardStandAloneDetails.media.url	Contains Url of the media. If mediaType=’video’, URL must be a valid video Url. If mediaType=’image’, URL must be a valid image Url or gif URL.	Required
richCardStandAloneDetails.suggestionList	Array of buttons. Can contains minimum = 0 and maximum= 11 buttons	Required **NOTE**: If don’t need buttons in template send an empty array ([])
richCardStandAloneDetails.suggestionList.typeOfAction	typeOfAction can one of the following reply, url_action, dialer_action	Optional
richCardStandAloneDetails.suggestionList.suggestionText	Contains suggestion text for a button	If suggestionList contains button, then its required
richCardStandAloneDetails.suggestionList.urlAction	If ‘typeOfAction’ = ’url_action’, then this conations URL to get navigate on clicking the button	Required if ‘typeOfAction’ = ’url_action’
richCardStandAloneDetails.suggestionList.phoneNumberToDial.countryCode	If ‘typeOfAction’ = ’dialer_action’, then this conations countryCode for phoneNumbers	Required if ‘typeOfAction’=dialer_action’

richCardStandAloneDetails.suggestionList.phoneNumberToDial.number	If ‘typeOfAction’ = ’dialer_action’, then this conations number	Required if ‘typeOfAction’=dialer_action’

Payload

```
{
"botId": " xxxxx ",
"isSMSFallbackRequired": false,
"richCardStandAloneDetails": {
"cardOrientation": "VERTICAL",
"mediaHeight": "MEDIUM_HEIGHT",
"cardTitle": "Happy Ganesh Chaturthi!",
"cardDescription": "Ganesh Chaturthi!",
"mediaType": "image",
"media": {
"url": "xxxxx"
},
"suggestionsList": [
{
"typeOfAction": "reply",
"suggestionText": "Re-Test 1"
},
{
"typeOfAction": "url_action",
"suggestionText": "Re-Test 2",
"urlAction": "https://www.amazon.in/"
},
{
"typeOfAction": "dialer_action",
"suggestionText": "button 3",
"phoneNumberToDial": {
"countryCode": "+91",
"number": "7507502233"
}
}
]
}
}
```

Response

```
{
"status": "SUCCESS",
"code": 200,
"data": {
"message": "Template updated successfully."
}
}
```

3) Carousel

Request Payload Description

Field Name	Description	Required / Optional
botId	Id of bot Can be fetched by get-botId-by-botName GET API	Required
isSMSFallbackRequired	It is a Boolean value. If RCS message is disabled in endUser’s device or if by some other reasons the RCS message doesn’t reach to the endUser, by assigning isSMSFallbackRequired as true we can send the message via SMS as a fallback mechanism	Required
smsFallbackTemplateDetails	If isSMSFallbackRequired is true than we require SMS fallback details object.	Optional Required only when isSMSFallbackRequired is enabled.
smsFallbackTemplateDetails.senderId	Sender Id of SMS fallback. Will be provided.	Required if isSMSFallbackRequired is true
smsFallbackTemplateDetails.messageType	Can be ‘TXT’ or ‘UNI’	Required if isSMSFallbackRequired is true
smsFallbackTemplateDetails.dltEntityId	Entity Id of SMS	Required if isSMSFallbackRequired is true
smsFallbackTemplateDetails.dltTempId	Template Id of SMS	Required if isSMSFallbackRequired is true
smsFallbackTemplateDetails.message	Contains message content of SMS	Required if isSMSFallbackRequired is true
richCardCarouselDetails	Contains details of carousel template	Required
richCardCarouselDetails.cardWidth	It can be ‘SMALL_WIDTH’ or ‘MEDIUM_WIDTH’	Required
richCardCarouselDetails.mediaHeight	It can be ‘SHORT_HEIGHT’ or ‘MEDIUM_HEIGHT’	Required

richCardCarouselDetails.cardsList	Array of cards. Must contain minimum = 2 and maximum = 10 cards	Required
richCardCarouselDetails.cardsList.cardId	Id of the card	Required
richCardCarouselDetails.cardsList.cardTitle	Title of the card	Required
richCardCarouselDetails.cardsList.cardDescription	Description of the card	Required
richCardCarouselDetails.cardsList.mediaType	For video mediaType will be ‘video’ and for image or gif it will be ‘image’	Required
richCardCarouselDetails.cardsList.cardMedia.url	Contains Url of the media. If mediaType=’video’, URL must be a valid video Url. If mediaType=’image’, URL must be a valid image Url or gif URL.	Required
richCardCarouselDetails.cardsList.thumbnail.url	richCardStandAloneDetails.thumbnail.url contains url for the thumbnail. The thumbnail url has to be of image.	Required if mediaType = ’video’
richCardCarouselDetails.suggestionList	Array of buttons. Must contains minimum = 0 and maximum= 11 buttons	Required **NOTE**: If don’t need buttons in template send an empty array ([])
richCardCarouselDetails.suggestionList.typeOfAction	typeOfAction can one of the following reply, url_action, dialer_action	Optional
richCardCarouselDetails.suggestionList.suggestionText	Contains suggestion text for a button	If suggestionList contains button, then its required
richCardCarouselDetails.suggestionList.urlAction	If ‘typeOfAction’ = ’url_action’, then this conations URL to get navigate on clicking the button	Required if ‘typeOfAction’ = ’url_action’
richCardCarouselDetails.suggestionList.phoneNumberToDial.countryCode	If ‘typeOfAction’ = ’dialer_action’, then this conations countryCode for phoneNumbers	Required if ‘typeOfAction’=dialer_action’
richCardCarouselDetails.suggestionList.phoneNumberToDial.number	If ‘typeOfAction’ = ’dialer_action’, then this conations number	Required if ‘typeOfAction’=dialer_action’

Payload

```
{
```

```
"botId": " xxxxx ",
"isSMSFallbackRequired": false,
"richCardCarouselDetails": {
"cardWidth": "SMALL_WIDTH",
"mediaHeight": "SHORT_HEIGHT",
"cardsList": [
{
"cardId": "123",
"cardTitle": "Greetings",
"cardDescription": "Greetings !",
"mediaType": "image",
"cardMedia": {
"url": xxxxx
},
"suggestionsList": [
{
"typeOfAction": "dialer_action",
"phoneNumberToDial": {
"countryCode": "+91",
"number": "9876543212"
},
"suggestionText": "Call Now",
"suggestionPostback": "call_now"
}
]
},
{
"cardId": "123",
"cardTitle": "Greetings",
"cardDescription": "Greetings !",
"mediaType": "video",
"cardMedia": {
"url": xxxxx
},
"cardThumbnail": {
"url": " xxxxx”
},
"suggestionsList": [
{
"typeOfAction": "reply",
"suggestionText": "Re-Test 1"
},
{
"typeOfAction": "url_action",
"suggestionText": "Re-Test 2",
"urlAction": "https://www.amazon.in/"
},
{
"typeOfAction": "dialer_action",
"suggestionText": "button 3",
"phoneNumberToDial": {
"countryCode": "+91",
"number": " xxxxxxxxxxxxxxxxxxxx "
}
}
]
}
]
}
}
```

Response

```
{
  "status": "SUCCESS",
  "code": 200,
  "data": {
    "message": "Template updated successfully."
  }
}
```

3. Get Template by botId

Method

GET

Description

This API is used to fetch all the templates based on BotId.

URL

/v1/get-all-templates-by-botid

Request Headers

apikey	XXXXXXXXXXXXXXXXXXXXX

Request Params

Fields	Description	Required/ Optional
botid	Id of bot whose templates need to be fetched. BotId can be fetched by get-botId-by-botName GET API	Required

Request Body

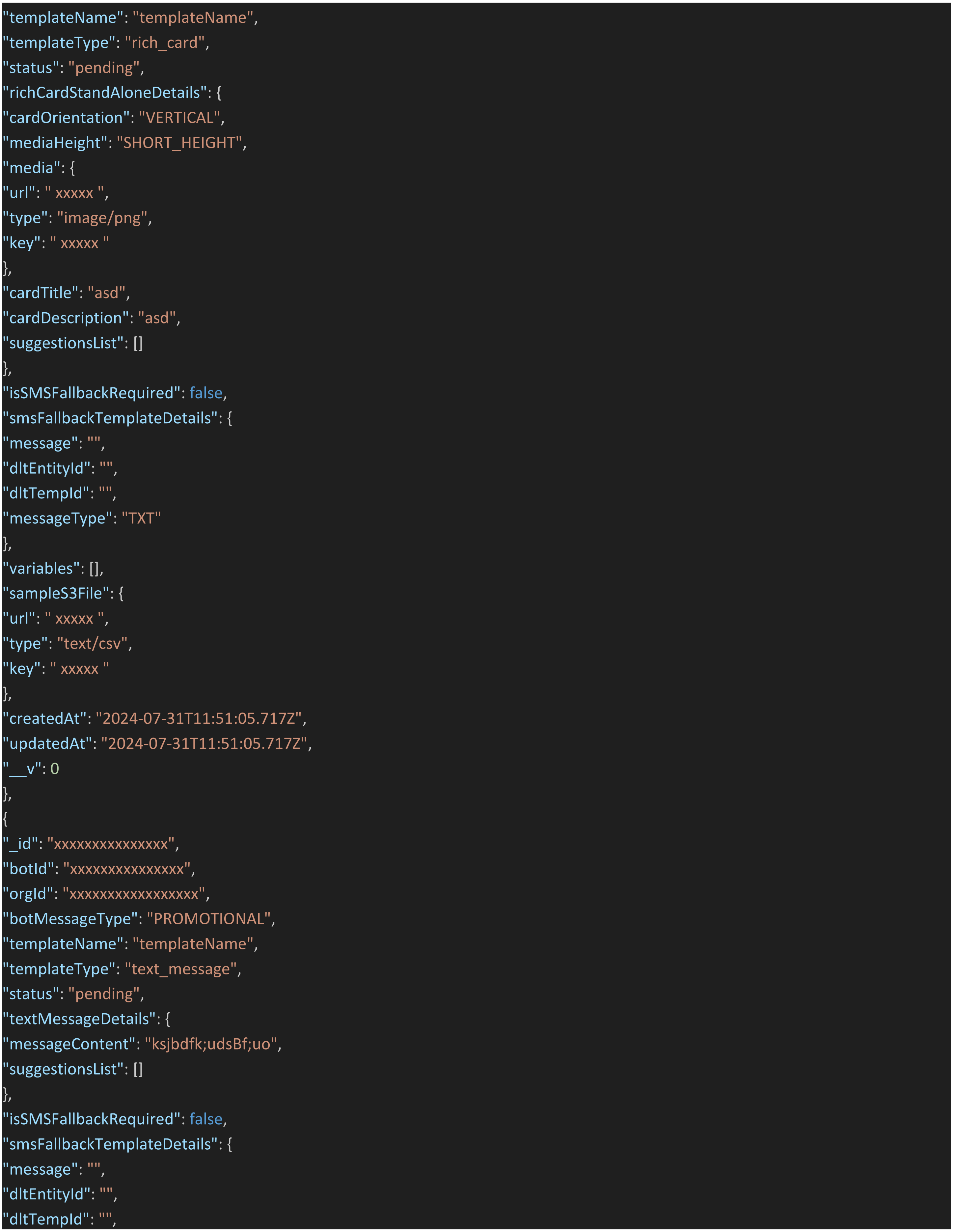
Body not required

Status

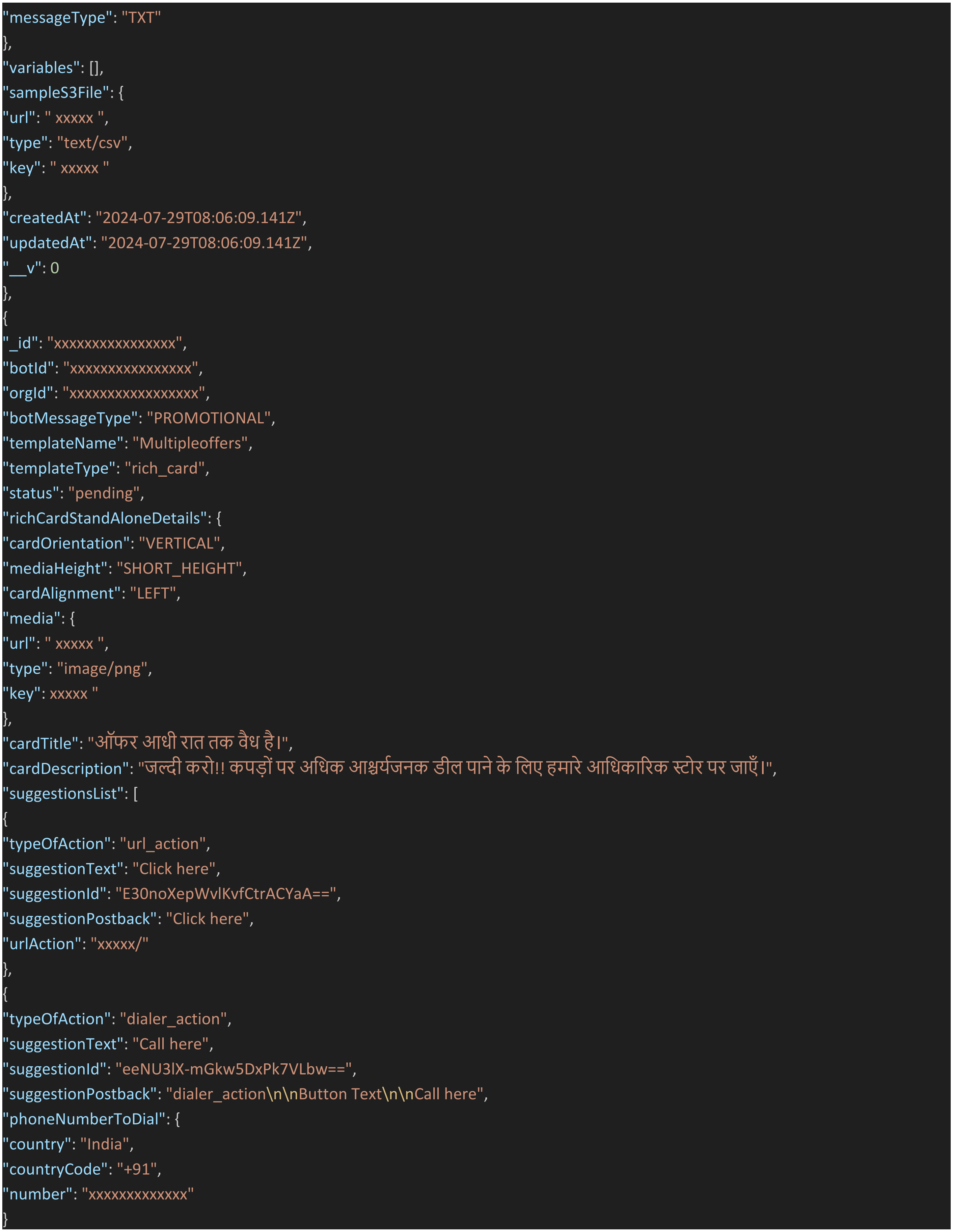
200

Response

```
[
{
"_id": "xxxxxxxxxxxxxxx",
"botId": "xxxxxxxxxxxxxx",
"orgId": "xxxxxxxxxxxxxxx",
"botMessageType": "PROMOTIONAL",
"templateName": "templateName",
"templateType": "rich_card",
"status": "pending",
"richCardStandAloneDetails": {
"cardOrientation": "VERTICAL",
"mediaHeight": "SHORT_HEIGHT",
"media": {
"url": "xxxx",
"type": "image/png",
"key": "xxxxx"
},
"cardTitle": "hi",
"cardDescription": "z",
"suggestionsList": []
},
"isSMSFallbackRequired": false,
"smsFallbackTemplateDetails": {
"message": "",
"dltEntityId": "",
"dltTempId": "",
"messageType": "TXT"
},
"variables": [],
"sampleS3File": {
"url": " xxxxx ",
"type": "text/csv",
"key": " xxxxx "
},
"createdAt": "2024-07-31T12:01:06.019Z",
"updatedAt": "2024-07-31T12:01:06.019Z",
"__v": 0
},
{
"_id": "xxxxxxxxxxxxxx",
"botId": "xxxxxxxxxxxxxx",
"orgId": "xxxxxxxxxxxxxx",
"botMessageType": "PROMOTIONAL",
```



```
"templateName": "templateName",
"templateType": "rich_card",
"status": "pending",
"richCardStandAloneDetails": {
"cardOrientation": "VERTICAL",
"mediaHeight": "SHORT_HEIGHT",
"media": {
"url": " xxxxx ",
"type": "image/png",
"key": " xxxxx "
},
"cardTitle": "asd",
"cardDescription": "asd",
"suggestionsList": []
},
"isSMSFallbackRequired": false,
"smsFallbackTemplateDetails": {
"message": "",
"dltEntityId": "",
"dltTempId": "",
"messageType": "TXT"
},
"variables": [],
"sampleS3File": {
"url": " xxxxx ",
"type": "text/csv",
"key": " xxxxx "
},
"createdAt": "2024-07-31T11:51:05.717Z",
"updatedAt": "2024-07-31T11:51:05.717Z",
"__v": 0
},
{
"_id": "xxxxxxxxxxxxxxx",
"botId": "xxxxxxxxxxxxxxx",
"orgId": "xxxxxxxxxxxxxxxxx",
"botMessageType": "PROMOTIONAL",
"templateName": "templateName",
"templateType": "text_message",
"status": "pending",
"textMessageDetails": {
"messageContent": "ksjbdfk;udsBf;uo",
"suggestionsList": []
},
"isSMSFallbackRequired": false,
"smsFallbackTemplateDetails": {
"message": "",
"dltEntityId": "",
"dltTempId": "",
```



```
"messageType": "TXT"
},
"variables": [],
"sampleS3File": {
"url": " xxxxx ",
"type": "text/csv",
"key": " xxxxx "
},
"createdAt": "2024-07-29T08:06:09.141Z",
"updatedAt": "2024-07-29T08:06:09.141Z",
"__v": 0
},
{
"_id": "xxxxxxxxxxxxxxxx",
"botId": "xxxxxxxxxxxxxxxx",
"orgId": "xxxxxxxxxxxxxxxxx",
"botMessageType": "PROMOTIONAL",
"templateName": "Multipleoffers",
"templateType": "rich_card",
"status": "pending",
"richCardStandAloneDetails": {
"cardOrientation": "VERTICAL",
"mediaHeight": "SHORT_HEIGHT",
"cardAlignment": "LEFT",
"media": {
"url": " xxxxx ",
"type": "image/png",
"key": xxxxx "
},
"cardTitle": "ऑफर आधी रात तक वैध है।",
"cardDescription": "जल्दी करो!! कपड़ों पर अधिक आश्चर्यजनक डील पाने के लिए हमारे आधिकारिक स्टोर पर जाएँ।",
"suggestionsList": [
{
"typeOfAction": "url_action",
"suggestionText": "Click here",
"suggestionId": "E30noXepWvlKvfCtrACYaA==",
"suggestionPostback": "Click here",
"urlAction": "xxxxx/"
},
{
"typeOfAction": "dialer_action",
"suggestionText": "Call here",
"suggestionId": "eeNU3lX-mGkw5DxPk7VLbw==",
"suggestionPostback": "dialer_action\n\nButton Text\n\nCall here",
"phoneNumberToDial": {
"country": "India",
"countryCode": "+91",
"number": "xxxxxxxxxxxxx"
}
}
]
},
"isSMSFallbackRequired": true,
"smsFallbackTemplateDetails": {
"message": "Dear Customer, offer on clothes valid till today midnight. Visit our website for more deals, Leadows..",
"dltEntityId": "xxxxxxxxxxxxxxxx",
"dltTempId": "xxxxxxxxxxxxxxxxx",
"messageType": "TXT"
},
"variables": [],
"sampleS3File": {
"url": " xxxxx ",
"type": "text/csv",
"key": " xxxxx "
},
"createdAt": "2024-07-23T12:57:42.967Z",
"updatedAt": "2024-07-23T12:58:25.590Z",
"__v": 0,
"lastUpdate": "Jul 23,2024"
}
]
```

4. Get Template status

Method

GET

Description

This API is used to get template status by botid and template Name.

URL

/v1/get-template-status/{{botid}}/{{templatename}}

Request Headers

apikey	xxxxxxxxxxxxxxxxxxx

Request Query

Fields	Description	Required/Optional
botid	Id of bot Can be fetched by get-botId-by-botName GET API	Required

templatename	Name of template whose status needs to be fetched	Required

Status

200

Response

```
{
"status": "SUCCESS",
"code": 200,
"data": "approved"
}
```

5. Get Templates by orgId

Method

GET

Description

This API is used to get all the templates of organisation.

Here orgId is fetched by apikey so orgId is not required in payload.

URL

/v1/get-all-template-by-orgId

Request Headers

apikey	xxxxxxxxxxxxxxxxxxxxx

Status

200

Response

```
{
 "message": "SUCCESS",
 "data": {
  "IsSmsCredentialsUsed": [
   {
    "_id": "xxxxxxxxxxxxxxxxxxx",
    "botId": "xxxxxxxxxxxxxxxxxxxx",
    "orgId": "xxxxxxxxxxxxxxxxxxxx",
    "botMessageType": "PROMOTIONAL",
    "templateName": "templateName",
    "templateType": "text_message",
    "status": "rejected",
    "textMessageDetails": {
     "messageContent": "Dear Customer, your OTP is {#var#}. it is valid for time 1 minute. Regards, Leadows.",
     "suggestionsList": [
      {
       "typeOfAction": "reply",
       "suggestionText": "Verify",
       "suggestionId": "xxxxxxxxxxxxx
       "suggestionPostback": "Verify"
      }
     ]
    },
    "isSMSFallbackRequired": true,
    "smsFallbackTemplateDetails": {
     "message": "Dear Customer, your OTP is {#var#}. it is valid for time 1 minute. Regards, Leadows.",
     "dltEntityId": "xxxxxxxxxxxxx",
     "dltTempId": "xxxxxxxxxxxxx",
     "messageType": "TXT"
    },
    "variables": [],
    "sampleS3File": {
     "url": " xxxxxxxxxxxxxxxxxxxx ",
     "type": "text/csv",
     "key": " xxxxxxxxxxxxxxxxxxxx "
    },
    "createdAt": "2024-07-23T10:07:43.213Z",
    "updatedAt": "2024-07-23T10:32:47.834Z",
    "__v": 0,
    "lastUpdate": "Jul 23,2024"
   }
  ]
 }
}
```

6. Get Templates by filter

Method

GET

Description

This API is used to get template by using filter like
pageSize
currentPage

templateName
templateType
status
botId

URL

/v1/get-template

Request Headers

apikey	xxxxxxxxxxxxxxx

Request Params

This are not required fields you can use this to apply filters accordingly.

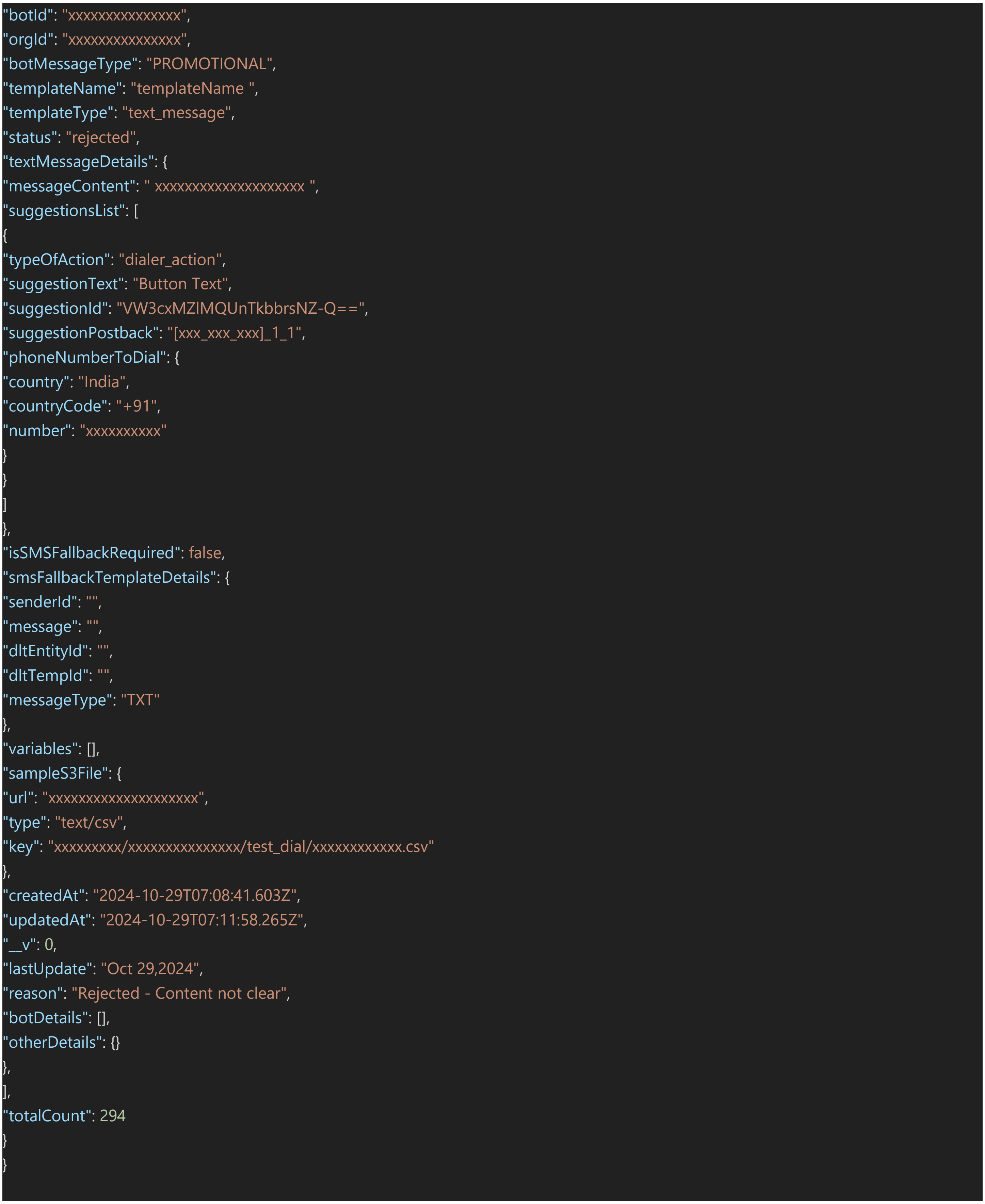
Fields	Description	Required/Optional
botId	Id of Bot Can be fetched by get-botId-by-botName GET API	Optional
templateType	Type of template “text_message”, ”rich_card”, ” carousel”	Optional
templateName	Name of template	Optional
status	Status of template “approved”, “pending” or ”rejected”	Optional
pageSize	Size number of templates to be fetched in one page	Optional
currentPage	Current page numbers	Optional

Status

200

Response

```
{
"status": "SUCCESS",
"code": 200,
"data": {
"message": "Templates fetched successfully.",
"allTemplates": [
{
"_id": "xxxxxxxxxxxxxxxx",
```



```
"botId": "xxxxxxxxxxxxxxx",
"orgId": "xxxxxxxxxxxxxxx",
"botMessageType": "PROMOTIONAL",
"templateName": "templateName ",
"templateType": "text_message",
"status": "rejected",
"textMessageDetails": {
"messageContent": " xxxxxxxxxxxxxxxxxxxx ",
"suggestionsList": [
{
"typeOfAction": "dialer_action",
"suggestionText": "Button Text",
"suggestionId": "VW3cxMZlMQUnTkbbrsNZ-Q==",
"suggestionPostback": "[xxx_xxx_xxx]_1_1",
"phoneNumberToDial": {
"country": "India",
"countryCode": "+91",
"number": "xxxxxxxxxx"
}
}
]
},
"isSMSFallbackRequired": false,
"smsFallbackTemplateDetails": {
"senderId": "",
"message": "",
"dltEntityId": "",
"dltTempId": "",
"messageType": "TXT"
},
"variables": [],
"sampleS3File": {
"url": "xxxxxxxxxxxxxxxxxxxx",
"type": "text/csv",
"key": "xxxxxxxxx/xxxxxxxxxxxxxxx/test_dial/xxxxxxxxxxxxxx.csv"
},
"createdAt": "2024-10-29T07:08:41.603Z",
"updatedAt": "2024-10-29T07:11:58.265Z",
"__v": 0,
"lastUpdate": "Oct 29,2024",
"reason": "Rejected - Content not clear",
"botDetails": [],
"otherDetails": {}
},
],
"totalCount": 294
}
}
```

7. Get TemplateId by Template name and botId

Method

GET

Description

This API is used to get templateId from template name.

URL

v1/get-templateId-by-templateName-and-botId

Request Headers

apikey	xxxxxxxxxxxxxxxxx

Params

Fields	Description	Required/Optional
templateName	Template Name	Required
botId	Id of bot to which this template belongs. BotId can be fetched by get-botId-by-botName GET API	Required

Response

```
{
"status": "SUCCESS",
"code": 200,
"data": {
"message": "Request Successful!",
"templateName": "templateName”,
"orgId": "xxxxxxxxxxxxxx",
"templateId": "xxxxxxxxxxxxxxx"
}
}
```

8. Get Template preview by Template name and botId

Method

GET

Description

This API is used to get templateId from template name.

URL

v1/template/preview

Request Headers

apikey	xxxxxxxxxxxxxxxxx

Params

Fields	Description	Required/Optional
templateName	Template Name	Required
botId	Id of bot to which this template belongs. BotId can be fetched by get-botId-by-botName GET API	Required

Response

<img src=https://rcs-bot-media-public.s3.ap-south-1.amazonaws.com/665873c0c660399dce731130/65c4eabecf32f14a18535f10/678e6616033dd76421b89b1a/templatePreview_1737435212941_templatePreview.png alt=https://rcs-bot-media-public.s3.ap-south-1.amazonaws.com/665873c0c660399dce731130/65c4eabecf32f14a18535f10/678e6616033dd76421b89b1a/templatePreview_1737435212941_templatePreview.png>

~~Campaign API :~~

1. Get all campaign by filter

Method

GET

Description

This API is used to fetch all the campaigns by
pageSize
currentPage

URL

~~/v1/get-all-campaigns-by-pagesize~~

Request Headers

apikey	xxxxxxxxxxxxxxxxx

Request Params

Fields	Description	Required/ Optional
pageSize	Page size number of data to be fetched in one page	Optional

currentPage	Current page number	Optional

Response

```
{
"allCampaigns": [
{
"_id": "xxxxxxxxxxxxxxxxx",
"botId": "xxxxxxxxxxxxxxxx",
"templateId": "xxxxxxxxxxxxxxx",
"orgId": "xxxxxxxxxxxxxxx",
"templateName": "templateName",
"botName": "botName",
"campaignName": "campaignName",
"isSMSFallbackRequired": true,
"smsVariables": [
"OTP"
],
"uploadedCSV": {
"csvFileUrl": "xxxxxxxxxxxxxxxxxxxxxx",
"csvFileType": "text/csv",
"csvFileName": "api number.csv"
},
"isDuplicateNumberAccepted": true,
"messageTypeForBilling": "Single_RBM_Message",
"ttl": "21600s",
"orgName": "orgName",
"createdAt": "2024-09-09T09:36:00.554Z",
"updatedAt": "2024-09-09T09:36:00.554Z"
}
],
"totalCount": 100
}
```

~~2. Create Campaign~~

Method

POST

Description

This API is used to create a Campaign

URL

~~/v1/create-campaign~~

Request Headers

apikey	xxxxxxxxxxxxxxxxxx
botid	xxxxxxxxxxxxxxxxxx

Request Body

```
{
"templateId": "xxxxxxxxxxxxx",
"campaignName": "campaignName",
"botName": "botName",
"templateName": "Sale",
"isSMSFallbackRequired": true,
"isDuplicateNumberAccepted": true,
"smsVariables": [
"OTP"
],
"ttl": "3600s",
"clientId": "xxxxxxxxxxxxx",
"clientSecret": "xxxxxxxxxxxxxxx",
"uploadedCSV": {
"csvFileUrl": "xxxxxxxxxxxxxx",
"csvFileType": "text/csv",
"csvFileName": "api number.csv"
},
// "status": "pending",
"orgName": "orgName"
}
```

Response

```
{
"status": "SUCCESS",
"code": 200,
"data": {
"message": "Campaign created successfully."
}
}
```

~~3. Get campaignId by campaign name~~

Method

GET

Description

This API is used to get campaign id by campaign name.

URL

~~/v1/get-campaignId-by-campaignName~~

Request Headers

apikey	xxxxxxxxxxxxxxxxxxxx

Params

Fields	Description	Required/Optional
campaignName	Name of campaign	Required

Response

```
{
"status": "SUCCESS",
"code": 200,
"data": {
"message": "Request Successful!",
"campaignName": "campaignName",
"orgId": "xxxxxxxxxxxx",
"campaignId": "xxxxxxxxxxxx"
}
}
```

~~4. Get all status campaign reports~~

Method

GET

Description

This API is used to get campaign reports for all status by botId and campaignId.

URL

~~v1/get-all-status-campaign-reports/{{botId}}/{{campaignId}}~~

Request Headers

apikey	xxxxxxxxxxxxxxx

Request Query

Fields	Description	Required/Optional
botId	Id of bot Can be fetched by get-botId-by-botName	Required
campaignId	Id of campaign Can be fetched by get-campaginId-by-campaignName GET API	Required

Status

200

Response

```
{
"status": "SUCCESS",
"code": 200,
"data": [
{
"status": "TOTAL_COUNT",
"count": 0
},
{
"status": "PENDING",
"count": 0
},
{
"status": "SENT",
"count": 0
},
{
"status": "DELIVERED",
"count": 0
},
{
"status": "READ",
"count": 0
},
{
"status": "RETRY",
"count": 0
},
{
"status": "FAILED",
"count": 0
},
{
"status": "SMS",
"count": 0
},
{
"status": "TTL_EXPIRATION_REVOKED",
"count": 0
}
]
}
```

Add Test Device :

1. Add test device against a bot

Points to consider while sending the payload:

1. Numbers which are RCS Enabled can only be added as test device.

Method

POST

Description

This API is used to add test device user.

URL

/v1/add-test-device

Request Headers

apikey	XXXXXXXXXXXXXXXXX

Request Parameters Description

Fields	Description	Required/ Optional
botId	Id of bot to which test device will be added. botId can be fetched by get-botId-by-botName API.	Required
phoneNumber	Phone Number to add as test device for the corresponding Bot. Number should be valid with valid countryCode Example: “+91xxxxxxxxx”	Required

Request Body

```
{
"botid”: “xxxxxxxxxxxxx",
"phoneNumber":"+91xxxxxxxxxx"
}
```

Response

```
{
"status": "SUCCESS",
"code": 200,
"data": "Test user invite sent successfully"
}
```

Brands API :

1. Get all Brands

Method

GET

Description

This API is used to get all brands.

URL

/v1/get-all-brands

Request Headers

apikey	xxxxxxxxxxxxxxxxxxxxx

Response

```
{
"status": "SUCCESS",
"code": 200,
"data": {
"data": [
{
"_id": "xxxxxxxxxxxxx",
"orgId": "xxxxxxxxxxxxx",
"brandName": "brandName",
"brandStatus": "VERIFIED",
"industryType": "Science, technology and engineering",
"officialBrandWebsite": "xxxxx/",
"brandLogo": {
"url": "xxxxxxxxxxxxxxxxxxxx",
"type": "image/png",
"key": " xxxxxxxxxxxxxxxxxxxx "
},
"contactPersonDetails": {
"firstName": "firstName",
"lastName": "lastName",
"emailId": "xyz@gmail.com",
"designation": "CEo",
"mobileNumber": "",
"country": "India",
"countryCode": "91"
},
"companyAddressDetails": {
"addressLine1": " xxxxxxxxxx ",
"addressLine2": " xxxxxxxxxx ",
"country": "India",
"state": " xxxxxxxxxx ",
"city": " xxxxxxxxxx ",
"zipCode": " xxxxxxxxxx "
},
"businessVerificationDetails": {
"verifyBusinessName": [
{
"documentType": "PAN card of Company",
"selectedFile": {
"url": "xxxxxxxxxxxxxxx",
"type": "application/pdf",
"key": "xxxxxxxxxxxxxxx/verifyBusinessName[0]_1721211452450_Company PAN.pdf",
"fileName": "Company PAN.pdf"
}
},
{
"documentType": "GST document",
"selectedFile": {
"url": “xxxxxxxxxxxxxxxxxxxxxx”,
"type": "image/jpeg",
"key": "xxxxxxxxxxxxxxx/verifyBusinessName[1]_1721211452466_GST Certificate.jpeg",
"fileName": "GST Certificate.jpeg"
}
}
],
"verifyBusinessAddress": [
{
"documentType": "PAN card of Company",
"selectedFile": {
"url": "xxxxxxxxxxxxxxxx",
"type": "application/pdf",
"key": "xxxxxxxxxx/verifyBusinessAddress[0]_1721211452478_Company PAN.pdf",
"fileName": "Company PAN.pdf"
}
},
{
"documentType": "GST document",
"selectedFile": {
"url": "xxxxxxxxxxxxxxxxxxxxxxxxxx",
"type": "image/jpeg",
"key": "xxxxxx/verifyBusinessAddress[1]_1721211452484_GST Certificate.jpeg",
"fileName": "GST Certificate.jpeg"
}
}
],
"termsAndConditions": false,
"agreeToPayVerificationFee": false
},
"createdAt": "2024-07-17T10:17:32.708Z",
"updatedAt": "2024-10-28T04:59:07.261Z",
"__v": 0
}
],
"message": "All Bots Details Fetched"
}
}
```

Message Sending APIs :

Promotional API Base URL: https://rcsapi.pinnacle.in/api

Note: Messages can be sent on the above url only. Messages cannot be sent on Base url

1. Promotional API

Method

POST

Description

This API is used to send Promotional messages.

URL

/v1/send-message

Request Headers

apikey	xxxxxxxxxxxxxxxxxx
botid	Xxxxxxxxxxxxxxxxxx

Request Body

```
{
"category": "promotional",
"correlationId":"434dfadf" //optional, get this field back in webhooks 10 to 30 chars limit, only alphanumeric, no special chars
"messages": [
{
"to": "+91xxxxxxxxxx",
"variables": [{"key":"OTP","value":"123"},{"key":"Name","value":"Jayant"}],
"templateId": "xxxxxxxxxxxxxxxx",
"ttl":"10",    // in hours, optional --- default 24hr,
"isSMSFallbackRequired":true,  //optional --- default false,
"smsVariables": [{"key":"OTP","value":"123"}],
}
]
}
```

Status

200

Response

```
{
"status": "SUCCESS",
"code": 200,
"correlationId": "xxxxxxxxxx",
"data": [
{
"templateId": "xxxxxxxxxxxxxx",
"message": "success",
"mobile": "+91xxxxxxxxxx",
"uniqueId": "xxxxxxxxxxxxxxxxx"
}
],
"totalMessgaes": 1
}
```

New VI APIs :

1. Get Message status by messageId

Method

GET

Description

This API is used to get message status by message Id.

URL

v1/get-message-status-by-messageId/{{botid}}/{{messageid}}

Request Headers

apikey	xxxxxxxxxxxxxxxxxxxxx

Request Query

Fields	Description	Required/Optional
botId	Id of bot. botId can be fetched by get-botId-by-botName GET API	Required
messageid	Message Id is unique Id of message send.	Required

Response

```
{
    "status": "SUCCESS",

   "code": 200,
    "data": {

      "status": {
          "msgId": "xxxxxx_xxxxx_xxxxx_xxxx",
          "status": "failed",
         "timestamp": "2024-10-14T13:18:36.441"

      }
   }
}
```

API Reports APIs :

1. Get Bot API reports

Method

POST

Description

This API is used to get Bot's API Reports.

URL

/v1/get-bot-api-reports

Request Headers

apikey	xxxxxxxxxxxxxxx

Request Body

Fields	Description	Required/Optional
botId	Id of bot.	Required

	botId can be fetched by get-botId-by-botName GET API	
startDate	Start date for the report	Required
endDate	End date for the report	Required
templateId (Optional field)	Filter to get reports by template id	Optional
userNumber (Optional field)	Filter to get reports by user number	Optional
pageSize	How many reports to fetch per api call (max 100)	Required
currentPage (Optional field – Pagination)	Current page of the report, Pagination	Required

```
{
"pageSize":10,
"currentPage":1,
"botId": "xxxxxxxxxxxxxx",
"startDate": "2024-10-10",
"endDate": "2024-10-24",
"templateId": "xxxxxxxxxxxxxx",
"userNumber": "+91xxxxxxxxxx"
}
```

Status

200

Response

```
{

  "status": "SUCCESS",

  "code": 200,

  "data": {
```

```
"reportDetails": [
  {
    "messageId": "xxxxxxxxxxxxx",
    "botId": "xxxxxxxxxxxxx",
    "correlationId": "xxxxxxxxxxxxx",
    "createdAt": "xxxxxxxxxxxxx",
    "messageTypeForBilling": "xxxxxxxxxxxxx",
    "receiverNumber": "xxxxxxxxxxxxx",
    "sendTime": "2024-11-03T06:53:48.437Z",
    "statusWithTime": {
      "PENDING": "2024-11-03T06:53:48.437Z",

      "FAILED": "2024-11-03T06:53:48.437Z",
      "SENT": "2024-11-03T06:53:48.437Z",
      "DELIVRD": "2024-11-03T06:53:48.437Z"
    },
    "templateId": "xxxxxxxxxxxxx",
    "reason": "xxxxxxxxxxxxx"
  },
  {
    "messageId": "xxxxxxxxxxxxx",
    "botId": "xxxxxxxxxxxxx",
    "correlationId": "xxxxxxxxxxxxx",
    "createdAt": "xxxxxxxxxxxxx",
        "messageTypeForBilling": "xxxxxxxxxxxxx",

        "receiverNumber": "+ xxxxxxxxxxxxxxxxxxxx ",

        "sendTime": "2024-11-03T06:53:48.437Z",

        "statusWithTime": {

          "PENDING": "2024-11-03T06:53:48.437Z",

          "SENT": "2024-11-03T06:53:47.225Z",

          "DELIVERED": "2024-11-03T06:53:46.879771Z"

        },

        "templateId": "xxxxxxxxxxxxx"

      }

    ],

    "totalCount": 14

  }

}
```

Common error codes

409 - Already exists / conflict. Example: Create template with already existing name

401 - Un authorized. Example: invalid/wrong API key, no API key provided

400 - Bad request. Example: Validation failed for required fields etc.

404 - Not found

500 - Internal server error.