Project Catalyst v2.0 - Complete API Documentation

Base URL

```
https://api.catalystvas.io/api/v2
```

Authentication

All API requests require authentication via Bearer token:

```
Authorization: Bearer YOUR_API_KEY
```

Or via API Key header:

```
X-API-Key: YOUR_API_KEY
```

Rate Limits

Channel	Max TPS	Burst	Notes
SMS	40,000	50,000	Per partner per second
WhatsApp	20,000	25,000	Respects WhatsApp rate limits
Telegram	15,000	18,000	Per bot
Viber	10,000	12,000	Per bot
Instagram	8,000	10,000	Per business account
Messenger	12,000	15,000	Per page
RCS	5,000	6,000	Per carrier
XMPP	50,000	60,000	Per user
TOTAL	**100,000**	**120,000**	Platform maximum

SMS/SMPP Channel

Send SMS

POST `/messages/sms`

Request:

```json
{
  "to": "+1234567890",
  "from": "SenderID",
  "message": {
    "body": "Your verification code is 123456"
  },
  "tracking": {
    "webhookUrl": "https://your.domain.com/dlr",
    "deliveryReport": true
  },
  "billing": {
    "rateCardId": "RC-GLOBAL-2024",
    "costCenter": "DEPT001"
  }
}
```

Response:

```json
{
  "message_id": "MSG-UUID-12345",
  "external_id": "SMPP-SEQ-999",
  "status": "submitted",
  "timestamp": "2024-12-25T18:00:00Z",
  "channel": "sms",
  "estimated_cost": 0.015,
  "currency": "USD"
}
```

Batch SMS

POST `/messages/sms/batch`

Request:

json

```
{
  "messages": [
    {
      "to": "+1234567890",
      "from": "SenderID",
      "message": {
        "body": "Message 1"
      }
    },
    {
      "to": "+0987654321",
      "from": "SenderID",
      "message": {
        "body": "Message 2"
      }
    }
  ],
  "scheduling": {
    "rateLimit": 1000,
    "maxConcurrent": 10000
  },
  "tracking": {
    "webhookUrl": "https://your.domain.com/dlr"
  }
}
```

Response:

json


```
{
  "batch_id": "BATCH-UUID-12345",
  "total_messages": 2,
  "submitted": 2,
  "failed": 0,
  "total_cost": 0.030,
  "currency": "USD",
  "messages": [
    {
      "message_id": "MSG-UUID-1",
      "to": "+1234567890",
      "status": "submitted"
    }
  ]
}
```


Check SMS Delivery Status

GET `/messages/sms/{messageId}/status`

Response:

json


```
{
  "message_id": "MSG-UUID-12345",
  "status": "delivered",
  "channel": "sms",
  "dlr": {
    "status": "DELIVRD",
    "code": "000",
    "description": "Message delivered successfully",
    "timestamp": "2024-12-25T18:05:00Z",
    "carrier": "Verizon",
    "network_id": "310410"
  },
  "billing": {
    "amount": 0.015,
    "currency": "USD",
    "rate_used": "standard"
  }
}
```

WhatsApp Channel

Send WhatsApp Message

POST /messages/whatsapp

Request:

```json
{
  "to": "11234567890",
  "from": "+1234567890",
  "message": {
    "type": "text",
    "body": "Hello from Catalyst VAS!"
  },
  "tracking": {
    "webhookUrl": "https://your.domain.com/webhook",
    "deliveryReport": true,
    "readReceipt": true
  },
  "billing": {
    "rateCardId": "RC-WHATSAPP-2024"
  }
}
```

Send WhatsApp Media

POST /messages/whatsapp/media

Request:

```json
{
  "to": "11234567890",
  "from": "+1234567890",
  "message": {
    "type": "image",
    "mediaUrl": "https://cdn.example.com/image.jpg",
    "caption": "Product Image"
  },
  "tracking": {
    "webhookUrl": "https://your.domain.com/webhook"
  }
}
```

Response:

```json
{
  "message_id": "MSG-WA-UUID-12345",
  "external_id": "wamid.12345",
  "status": "submitted",
  "channel": "whatsapp",
  "timestamp": "2024-12-25T18:00:00Z",
  "estimated_cost": 0.035,
  "currency": "USD"
}
```

Send WhatsApp Template

POST `/messages/whatsapp/template`

Request:

json

```
{
  "to": "11234567890",
  "from": "+1234567890",
  "template": {
    "name": "order_confirmation",
    "language": "en_US",
    "parameters": {
      "1": "ORDER123",
      "2": "$99.99",
      "3": "https://track.example.com/ORDER123"
    }
  },
  "tracking": {
    "webhookUrl": "https://your.domain.com/webhook"
  }
}
```

WhatsApp Interactive Message

POST `/messages/whatsapp/interactive`

Request:

json

```
{
  "to": "11234567890",
  "from": "+1234567890",
  "message": {
    "type": "interactive",
    "body": {
      "text": "Which product are you interested in?"
    },
    "footer": {
      "text": "Reply with product number"
    },
    "action": {
      "buttons": [
        {
          "type": "reply",
          "reply": {
            "id": "1",
            "title": "Product A"
          }
        },
        {
          "type": "reply",
          "reply": {
            "id": "2",
            "title": "Product B"
          }
        }
      ]
    }
  },
  "tracking": {
    "webhookUrl": "https://your.domain.com/webhook"
  }
}
```

Telegram Channel

Send Telegram Message

POST /messages/telegram

Request:

```json
{
  "to": "123456789",
  "from": "/bot_token",
  "message": {
    "type": "text",
    "body": "Hello from Catalyst VAS!",
    "parseMode": "HTML"
  },
  "tracking": {
    "webhookUrl": "https://your.domain.com/webhook"
  }
}
```

Response:

```json
{
  "message_id": "MSG-TG-UUID-12345",
  "external_id": "9876543210",
  "status": "submitted",
  "channel": "telegram",
  "timestamp": "2024-12-25T18:00:00Z"
}
```

Send Telegram Sticker

POST `/messages/telegram/sticker`

Request:

```json
{
  "to": "123456789",
  "from": "/bot_token",
  "message": {
    "type": "sticker",
    "stickerId": "AgADfCEAEqNDwWLj"
  }
}
```

Send Telegram Inline Keyboard

POST /messages/telegram/keyboard

Request:

json

```
{
  "to": "123456789",
  "from": "/bot_token",
  "message": {
    "type": "text",
    "body": "Choose an option:",
    "replyMarkup": {
      "inlineKeyboard": [
        [
          {
            "text": "Option 1",
            "callbackData": "opt_1"
          },
          {
            "text": "Option 2",
            "callbackData": "opt_2"
          }
        ]
      ]
    }
  }
}
```

Viber Channel

Send Viber Message

POST /messages/viber

Request:

```json
{
  "to": "client_id_12345",
  "from": "bot_account_id",
  "message": {
    "type": "text",
    "body": "Hello from Catalyst VAS!",
    "trackingData": "campaign_001"
  },
  "tracking": {
    "webhookUrl": "https://your.domain.com/webhook",
    "deliveryReport": true
  }
}
```

Response:

```json
{
  "message_id": "MSG-VB-UUID-12345",
  "external_id": "vb_msg_id_123",
  "status": "submitted",
  "channel": "viber",
  "timestamp": "2024-12-25T18:00:00Z"
}
```

Send Viber Rich Media

POST `/messages/viber/rich-media`

Request:

json

```
{
  "to": "client_id_12345",
  "from": "bot_account_id",
  "message": {
    "type": "rich_media",
    "altText": "Product Catalog",
    "richMedia": {
      "buttonsGroupColumns": 6,
      "buttonsGroupRows": 2,
      "buttons": [
        {
          "text": "Product A",
          "actionType": "open-url",
          "actionBody": "https://example.com/product-a"
        },
        {
          "text": "Product B",
          "actionType": "open-url",
          "actionBody": "https://example.com/product-b"
        }
      ]
    }
  }
}
```

Instagram Channel

Send Instagram DM

POST `/messages/instagram`

Request:

json

```
{
  "to": "user_id_12345",
  "from": "business_account_id",
  "message": {
    "type": "text",
    "body": "Thanks for your inquiry! We'll get back to you soon."
  },
  "tracking": {
    "webhookUrl": "https://your.domain.com/webhook"
  }
}
```

Response:

json

```
{
  "message_id": "MSG-IG-UUID-12345",
  "external_id": "ig_msg_id_999",
  "status": "submitted",
  "channel": "instagram",
  "timestamp": "2024-12-25T18:00:00Z"
}
```

Send Instagram Story Mention

POST /messages/instagram/story

Request:

json

```
{
  "to": "user_id_12345",
  "from": "business_account_id",
  "message": {
    "type": "story_mention",
    "storyId": "story_id_123",
    "caption": "Check out this amazing product!"
  }
}
```

Messenger Channel

Send Messenger Text

POST /messages/messenger

Request:

json

```
{
  "to": "user_id_12345",
  "from": "page_id",
  "message": {
    "type": "text",
    "body": "Your order has been confirmed!"
  },
  "tracking": {
    "webhookUrl": "https://your.domain.com/webhook"
  }
}
```

Response:

json

```
{
  "message_id": "MSG-FB-UUID-12345",
  "external_id": "fb_msg_id_789",
  "status": "submitted",
  "channel": "messenger",
  "timestamp": "2024-12-25T18:00:00Z"
}
```

Send Messenger Structured Message

POST /messages/messenger/structured

Request:

json

```
{
  "to": "user_id_12345",
  "from": "page_id",
  "message": {
    "type": "structured_message",
    "attachment": {
      "type": "template",
      "payload": {
        "templateType": "generic",
        "elements": [
          {
            "title": "Product A",
            "image_url": "https://example.com/img1.jpg",
            "subtitle": "$99.99",
            "buttons": [
              {
                "type": "web_url",
                "url": "https://example.com/product-a",
                "title": "View"
              }
            ]
          }
        ]
      }
    }
  }
}
```

Google RCS Channel

Send RCS Message

POST /messages/rcs

Request:

```json
{
  "to": "+11234567890",
  "from": "rcs_bot_id",
  "message": {
    "type": "text",
    "body": "Your package has been delivered!"
  },
  "tracking": {
    "webhookUrl": "https://your.domain.com/webhook",
    "deliveryReport": true
  },
  "fallback": {
    "enabled": true,
    "channel": "sms",
    "body": "Your package has been delivered!"
  }
}
```

Response:

```json
{
  "message_id": "MSG-RCS-UUID-12345",
  "external_id": "rcs_msg_id_456",
  "status": "submitted",
  "channel": "rcs",
  "timestamp": "2024-12-25T18:00:00Z",
  "fallbackChannel": null
}
```

Send RCS Rich Card

POST `/messages/rcs/rich-card`

Request:

json

```
{
  "to": "+11234567890",
  "from": "rcs_bot_id",
  "message": {
    "type": "rich_card",
    "card": {
      "title": "Order Status",
      "description": "Your order #123 is out for delivery",
      "media": {
        "height": "SHORT",
        "contentUrl": "https://example.com/tracking.jpg"
      },
      "suggestions": [
        {
          "action": {
            "urlAction": {
              "openUrl": {
                "url": "https://track.example.com/123"
              }
            }
          },
          "displayText": "Track Order"
        },
        {
          "action": {
            "urlAction": {
              "openUrl": {
                "url": "https://example.com/contact"
              }
            }
          },
          "displayText": "Contact Support"
        }
      ]
    }
  },
  "fallback": {
    "enabled": true,
    "channel": "sms"
  }
}
```

XMPP/Instant Messaging Channel

Send XMPP Message

POST /messages/xmpp

Request:

json

```
{
  "to": "user456@catalystvas.io",
  "from": "bot@catalystvas.io",
  "message": {
    "type": "text",
    "body": "Hello! How can I help you?",
    "thread": "conversation_123"
  },
  "tracking": {
    "readReceipt": true,
    "archive": true,
    "webhookUrl": "https://your.domain.com/webhook"
  }
}
```

Response:

json

```
{
  "message_id": "MSG-XMPP-UUID-12345",
  "status": "submitted",
  "channel": "xmpp",
  "timestamp": "2024-12-25T18:00:00Z",
  "archiveId": "archive_msg_999",
  "deliveryExpected": "immediate"
}
```

Send XMPP Group Chat

POST /messages/xmpp/muc

Request:

```json
{
  "to": "support_room@conference.catalystvas.io",
  "from": "bot@catalystvas.io",
  "message": {
    "type": "groupchat",
    "body": "Alert: System maintenance window 2024-12-26 02:00 UTC",
    "subject": "System Alert"
  },
  "archivePolicy": "always",
  "deliveryNotifications": true
}
```

Update XMPP Presence

POST `/messages/xmpp/presence`

Request:

```json
{
  "to": "user456@catalystvas.io",
  "from": "bot@catalystvas.io",
  "presence": {
    "show": "available|away|dnd|offline",
    "status": "In a meeting",
    "priority": 0
  }
}
```

Response:

```json
{
  "status": "updated",
  "jid": "user456@catalystvas.io",
  "presence": "available",
  "timestamp": "2024-12-25T18:00:00Z"
}
```

Retrieve XMPP Message Archive

GET /messages/xmpp/archive

Query Parameters:

```
?jid=user456@catalystvas.io
&start=2024-12-01T00:00:00Z
&end=2024-12-31T23:59:59Z
&limit=100
&page=1
```

Response:

json


```
{
  "jid": "user456@catalystvas.io",
  "total": 250,
  "limit": 100,
  "page": 1,
  "messages": [
    {
      "id": "archive_msg_001",
      "from": "bot@catalystvas.io",
      "to": "user456@catalystvas.io",
      "body": "Hello!",
      "timestamp": "2024-12-25T18:00:00Z",
      "type": "chat"
    }
  ]
}
```

Multi-Channel Orchestration

Send with Channel Preference & Failover

POST /messages/orchestrate

Request:

json

```
{
  "to": "+11234567890",
  "from": "SenderID",
  "message": {
    "body": "Time-sensitive alert!",
    "type": "text"
  },
  "channels": ["whatsapp", "sms", "telegram"],
  "channelPreference": "whatsapp>sms>telegram",
  "failover": {
    "enabled": true,
    "retryDelay": 5000,
    "maxRetries": 3,
    "sequentialFallback": true
  },
  "scheduling": {
    "sendAt": "2024-12-25T18:00:00Z",
    "timeZone": "UTC",
    "retryWindow": "24h"
  },
  "tracking": {
    "webhookUrl": "https://your.domain.com/webhook",
    "deliveryReport": true,
    "channelTracking": true
  },
  "billing": {
    "rateCardId": "RC-MULTI-CHANNEL",
    "costCenter": "ALERT-SYSTEM",
    "budgetLimit": 1.00
  }
}
```

Response:

json


```
{
  "orchestration_id": "ORCH-UUID-12345",
  "status": "initiated",
  "preferred_channel": "whatsapp",
  "channels_attempted": [],
  "channels_available": ["whatsapp", "sms", "telegram"],
  "timestamp": "2024-12-25T18:00:00Z",
  "estimatedTotalCost": 0.15,
  "messages": [
    {
      "message_id": "MSG-UUID-1",
      "channel": "pending",
      "status": "queued"
    }
  ]
}
```


Bulk Campaign

POST /messages/campaign

Request:

json


```
{
  "campaignId": "CAMP-PROMO-2024-Q4",
  "campaignName": "Holiday Promotion",
  "template": {
    "channels": {
      "whatsapp": {
        "templateName": "holiday_promo_v2",
        "parameters": ["discountPercent", "expiryDate", "coupomCode"]
      },
      "sms": {
        "body": "Get {{discountPercent}}% off! Use code {{couponCode}} until {{expiryDate}}. {{linkUrl}}"
      },
      "telegram": {
        "text": "🎉<b>Holiday Promo</b>\nGet {{discountPercent}}% off!\nCode: <code>{{couponCode}}</code>",
        "parseMode": "HTML"
      }
    }
  },
  "recipients": [
    {
      "phoneOrId": "+11234567890",
      "channel": "auto",
      "parameters": {
        "discountPercent": "25",
        "expiryDate": "2024-12-31",
        "couponCode": "PROMO25",
        "linkUrl": "https://example.com/promo"
      }
    }
  ],
  "scheduling": {
    "startTime": "2024-12-20T09:00:00Z",
    "rateLimit": 1000,
    "maxConcurrent": 10000,
    "timeZoneOptimization": true
  },
  "analytics": {
    "trackingEnabled": true,
    "conversionTracking": true,
    "attributionWindow": 30
  }
}
```

Webhook Formats

Delivery Receipt Webhook

POST {webhookUrl}

json


```
{
  "eventType": "delivery",
  "messageId": "MSG-UUID-12345",
  "orchestrationId": "ORCH-UUID-12345",
  "channel": "sms",
  "status": "delivered",
  "dlr": {
    "code": "000",
    "description": "Message delivered successfully",
    "timestamp": "2024-12-25T18:05:00Z",
    "carrier": "Verizon",
    "networkId": "310410"
  },
  "billing": {
    "amount": 0.015,
    "currency": "USD",
    "rateUsed": "standard"
  },
  "metadata": {}
}
```


Read Receipt Webhook

json


```
{
  "eventType": "read",
  "messageId": "MSG-XMPP-UUID-12345",
  "channel": "xmpp",
  "status": "read",
  "recipient": "user456@catalystvas.io",
  "timestamp": "2024-12-25T18:05:30Z"
}
```


Error Webhook

json

```
{
  "eventType": "error",
  "messageId": "MSG-UUID-12345",
  "channel": "whatsapp",
  "error": {
    "code": "RATE_LIMIT_EXCEEDED",
    "description": "Rate limit exceeded for this number",
    "timestamp": "2024-12-25T18:01:00Z",
    "retryable": true,
    "retryAfter": 3600
  }
}
```

Error Codes & Handling

Code	Description	Retryable	Action
SUCCESS	Message sent successfully	No	—
INVALID_DESTINATION	Invalid phone number or JID	No	Validate destination format
INSUFFICIENT_BALANCE	Partner account balance too low	No	Top up account
RATE_LIMIT_EXCEEDED	TPS/daily limit exceeded	Yes	Reduce sending rate
CHANNEL_UNAVAILABLE	Selected channel offline	Yes	Use failover channel
INVALID_TEMPLATE	WhatsApp template not approved	No	Use approved template
MEDIA_DOWNLOAD_FAILED	Could not download media URL	Yes	Verify media URL
INVALID_API_KEY	API key invalid/expired	No	Regenerate API key
NETWORK_ERROR	Gateway connectivity issue	Yes	Retry after delay
TIMEOUT	Gateway request timeout	Yes	Retry with exponential backoff

---

Best Practices

1. **Always use channel preferences with fallover for critical messages**
2. **Implement exponential backoff for retries** (5s, 10s, 20s, etc.)
3. **Use idempotent keys to prevent duplicate sends**
4. **Monitor webhook delivery success rate**
5. **Cache rate cards locally to reduce API calls**
6. **Implement circuit breaker pattern for failing gateways**
7. **Use batch APIs for high-volume sending**
8. **Monitor billing in real-time to prevent overspend**
9. **Encrypt sensitive data before transmission**
10. **Regular cleanup of message archives (30-90 day retention)**

Code Examples

Python

python

```python
import requests
import json

api_key = "your_api_key_here"
headers = {
    "Authorization": f"Bearer {api_key}",
    "Content-Type": "application/json"
}

# Send SMS
payload = {
    "to": "+11234567890",
    "from": "SenderID",
    "message": {
        "body": "Your OTP is 123456"
    }
}

response = requests.post(
    "https://api.catalystvas.io/api/v2/messages/sms",
    headers=headers,
    json=payload
)

print(response.json())
```

JavaScript/Node.js

javascript

```javascript
const axios = require('axios');

const apiKey = 'your_api_key_here';
const headers = {
  'Authorization': `Bearer ${apiKey}`,
  'Content-Type': 'application/json'
};

// Send WhatsApp
const payload = {
  to: '11234567890',
  from: '+1234567890',
  message: {
    type: 'text',
    body: 'Hello from Catalyst!'
  }
};

axios.post(
  'https://api.catalystvas.io/api/v2/messages/whatsapp',
  payload,
  { headers }
).then(response => {
  console.log(response.data);
}).catch(error => {
  console.error(error);
});
```

Go

```go
package main

import (
  "fmt"
  "net/http"
  "bytes"
  "encoding/json"
)

func sendMessage() {
  apiKey := "your_api_key_here"
  payload := map[string]interface{}{
    "to": "+11234567890",
    "from": "SenderID",
    "message": map[string]string{
      "body": "Your OTP is 123456",
    },
  }

  jsonPayload, _ := json.Marshal(payload)

  req, _ := http.NewRequest("POST",
    "https://api.catalystvas.io/api/v2/messages/sms",
    bytes.NewBuffer(jsonPayload))

  req.Header.Set("Authorization", fmt.Sprintf("Bearer %s", apiKey))
  req.Header.Set("Content-Type", "application/json")

  client := &http.Client{}
  resp, _ := client.Do(req)
  defer resp.Body.Close()

  fmt.Println(resp.Status)
}
```

API Version: v2.0

Last Updated: December 2024

Documentation: https://docs.catalystvas.io