Evolution API - Evolution Cloud API

API de integração WhatsApp

- Instalação e Configurações Evolution API
  - Como instalar a Evolution API 2.0
  - Instalar Evolution API v1.0
  - Instalação Evolution API oficial
- Instalação e Configurações Evolution Api Cloud
  - Instalação Evolution Api Cloud
- Instalação Evolution API Cloud update

Instalação e Configurações Evolution API

Procedimentos Evolution-API

Como instalar a Evolution API 2.0

LInk: https://leonardoamoyr.com/como-instalar-a-evolution-api-2-0/

Agosto/2024

Apontamentos de domínio da Evolution API 2.0

Acesse o local onde você adquiriu o seu domínio, entre em Zone DNS (gerenciamento de DNS) e faça os seguintes apontamentos:

Tipo A
Nome: evolutionapi
Objeto/Valor: IP da sua VPS

Tipo A
Nome: www.evolutionapi
Objeto/Valor: IP da sua VPS

Preparando o Portainer

Agora vamos para o Portainer!

Dentro do Portainer o primeiro passo é criar uma network chamada “evolutionapi”. Depois disso, acesse a aba de containers, clique no container do Traefik e role a página até o final. Selecione a network que você acabou de criar na lista e clique no botão “Join Network”.

Feito isso, agora clique em “Stacks” e clique no botão para criar uma nova stack. Copie o código abaixo e cole dentro do editor de stacks:

```
version: '3.7'
services:
```

```
  evolution2:
    image: atendai/evolution-api:v2.1.0
    networks:
      - evolutionapi
    environment:
      - SERVER_URL=https://evolutionapi.seusite.xxx
      - AUTHENTICATION_API_KEY=xxxxxxxxxxxxxxxxxx # Crie a sua API Key em https://api-
keygen.com/
      - AUTHENTICATION_EXPOSE_IN_FETCH_INSTANCES=true
      - LANGUAGE=en
      - DEL_INSTANCE=false
      - DATABASE_PROVIDER=postgresql
      -
DATABASE_CONNECTION_URI=postgresql://evolution2:evolution2@postgres:5432/evolution2?schema=pub
lic
      - DATABASE_SAVE_DATA_INSTANCE=true
      - DATABASE_SAVE_DATA_NEW_MESSAGE=true
      - DATABASE_SAVE_MESSAGE_UPDATE=true
      - DATABASE_SAVE_DATA_CONTACTS=true
      - DATABASE_SAVE_DATA_CHATS=true
      - DATABASE_SAVE_DATA_LABELS=true
      - DATABASE_SAVE_DATA_HISTORIC=true
      - DATABASE_CONNECTION_CLIENT_NAME=evolution2
      #Rabbitmq
      - RABBITMQ_ENABLED=false
      - RABBITMQ_URI=amqp://admin:admin@rabbitmq:5672/default
      - RABBITMQ_EXCHANGE_NAME=evolution
      - RABBITMQ_GLOBAL_ENABLED=false
      - RABBITMQ_EVENTS_APPLICATION_STARTUP=false
      - RABBITMQ_EVENTS_INSTANCE_CREATE=false
      - RABBITMQ_EVENTS_INSTANCE_DELETE=false
      - RABBITMQ_EVENTS_QRCODE_UPDATED=false
      - RABBITMQ_EVENTS_MESSAGES_SET=false
      - RABBITMQ_EVENTS_MESSAGES_UPSERT=true
      - RABBITMQ_EVENTS_MESSAGES_EDITED=false
      - RABBITMQ_EVENTS_MESSAGES_UPDATE=false
      - RABBITMQ_EVENTS_MESSAGES_DELETE=false
      - RABBITMQ_EVENTS_SEND_MESSAGE=false
      - RABBITMQ_EVENTS_CONTACTS_SET=false
      - RABBITMQ_EVENTS_CONTACTS_UPSERT=false
```

```
- RABBITMQ_EVENTS_CONTACTS_UPDATE=false
- RABBITMQ_EVENTS_PRESENCE_UPDATE=false
- RABBITMQ_EVENTS_CHATS_SET=false
- RABBITMQ_EVENTS_CHATS_UPSERT=false
- RABBITMQ_EVENTS_CHATS_UPDATE=false
- RABBITMQ_EVENTS_CHATS_DELETE=false
- RABBITMQ_EVENTS_GROUPS_UPSERT=false
- RABBITMQ_EVENTS_GROUP_UPDATE=false
- RABBITMQ_EVENTS_GROUP_PARTICIPANTS_UPDATE=false
- RABBITMQ_EVENTS_CONNECTION_UPDATE=true
- RABBITMQ_EVENTS_CALL=false
- RABBITMQ_EVENTS_TYPEBOT_START=false
- RABBITMQ_EVENTS_TYPEBOT_CHANGE_STATUS=false
#SqS
- SQS_ENABLED=false
- SQS_ACCESS_KEY_ID=
- SQS_SECRET_ACCESS_KEY=
- SQS_ACCOUNT_ID=
- SQS_REGION=
- WEBSOCKET_ENABLED=false
- WEBSOCKET_GLOBAL_EVENTS=false
- WA_BUSINESS_TOKEN_WEBHOOK=evolution
- WA_BUSINESS_URL=https://graph.facebook.com
- WA_BUSINESS_VERSION=v20.0
- WA_BUSINESS_LANGUAGE=pt_BR
#Webhook
- WEBHOOK_GLOBAL_URL=''
- WEBHOOK_GLOBAL_ENABLED=false
- WEBHOOK_GLOBAL_WEBHOOK_BY_EVENTS=false
- WEBHOOK_EVENTS_APPLICATION_STARTUP=false
- WEBHOOK_EVENTS_QRCODE_UPDATED=true
- WEBHOOK_EVENTS_MESSAGES_SET=true
- WEBHOOK_EVENTS_MESSAGES_UPSERT=true
- WEBHOOK_EVENTS_MESSAGES_EDITED=true
- WEBHOOK_EVENTS_MESSAGES_UPDATE=true
- WEBHOOK_EVENTS_MESSAGES_DELETE=true
- WEBHOOK_EVENTS_SEND_MESSAGE=true
- WEBHOOK_EVENTS_CONTACTS_SET=true
- WEBHOOK_EVENTS_CONTACTS_UPSERT=true
- WEBHOOK_EVENTS_CONTACTS_UPDATE=true
```

```
      - WEBHOOK_EVENTS_PRESENCE_UPDATE=true
      - WEBHOOK_EVENTS_CHATS_SET=true
      - WEBHOOK_EVENTS_CHATS_UPSERT=true
      - WEBHOOK_EVENTS_CHATS_UPDATE=true
      - WEBHOOK_EVENTS_CHATS_DELETE=true
      - WEBHOOK_EVENTS_GROUPS_UPSERT=true
      - WEBHOOK_EVENTS_GROUPS_UPDATE=true
      - WEBHOOK_EVENTS_GROUP_PARTICIPANTS_UPDATE=true
      - WEBHOOK_EVENTS_CONNECTION_UPDATE=true
      - WEBHOOK_EVENTS_LABELS_EDIT=true
      - WEBHOOK_EVENTS_LABELS_ASSOCIATION=true
      - WEBHOOK_EVENTS_CALL=true
      - WEBHOOK_EVENTS_TYPEBOT_START=false
      - WEBHOOK_EVENTS_TYPEBOT_CHANGE_STATUS=false
      - WEBHOOK_EVENTS_ERRORS=false
      - WEBHOOK_EVENTS_ERRORS_WEBHOOK=
      - CONFIG_SESSION_PHONE_CLIENT=Evolution API V2
      - CONFIG_SESSION_PHONE_NAME=Chrome
      - CONFIG_SESSION_PHONE_VERSION=2.3000.1015901307 # https://web.whatsapp.com/check-
update?version=0&platform=web
      - QRCODE_LIMIT=30
      #Openai
      - OPENAI_ENABLED=true
      #Dify
      - DIFY_ENABLED=true
      #Typebot
      - TYPEBOT_ENABLED=true
      - TYPEBOT_API_VERSION=latest
      #Chatwoot
      - CHATWOOT_ENABLED=false
      - CHATWOOT_MESSAGE_READ=true
      - CHATWOOT_MESSAGE_DELETE=true
      -
CHATWOOT_IMPORT_DATABASE_CONNECTION_URI=postgresql://evolution:evolution@postgres:5432/chatwoo
t?sslmode=disable
      - CHATWOOT_IMPORT_PLACEHOLDER_MEDIA_MESSAGE=true
      #redis
      - CACHE_REDIS_ENABLED=true
      - CACHE_REDIS_URI=redis://redis:6379/1
      - CACHE_REDIS_PREFIX_KEY=evolution
```

```
    - CACHE_REDIS_SAVE_INSTANCES=false
    - CACHE_LOCAL_ENABLED=false
    #Minio
    - S3_ENABLED=false
    - S3_ACCESS_KEY=suachave
    - S3_SECRET_KEY=suachave
    - S3_BUCKET=evolutionv3
    - S3_PORT=443
    - S3_ENDPOINT=minioserver.meubot.top
    - S3_USE_SSL=true
  labels:
    - traefik.enable=true
    - traefik.http.routers.evolution_v2.rule=Host(`evolutionapi.seusite.xxx`)
    - traefik.http.routers.evolution_v2.entrypoints=websecure
    - traefik.http.routers.evolution_v2.priority=1
    - traefik.http.routers.evolution_v2.tls.certresolver=leresolver
    - traefik.http.routers.evolution_v2.service=evolution_v2
    - traefik.http.services.evolution_v2.loadbalancer.server.port=8080
    - traefik.http.services.evolution_v2.loadbalancer.passHostHeader=true

postgres:
  image: postgres:latest
  environment:
    POSTGRES_DB: evolution2
    POSTGRES_USER: evolution2
    POSTGRES_PASSWORD: evolution2
  networks:
    - evolutionapi
  volumes:
    - evolution_postgres_data:/var/lib/postgresql/data

redis:
  image: redis:latest
  command:
    - "redis-server"
    - "--appendonly yes"
    - "--port 6379"
  volumes:
    - evolution_redis_data:/data
  networks:
```

```
      - evolutionapi

volumes:
  evolution_postgres_data:
  evolution_redis_data:

networks:
  evolutionapi:
    external: true
```

Faça as configurações conforme eu oriento no vídeo da aula e depois clique no botão “Deploy Stack” e aguarde a finalização.

Uma vez finalizado, aguarde uns 30 segundos e acesse no seu navegador o site https://evolutionapi.seusite.xxx/manager (troque pelo seu domínio).

Agora basta logar com sua API Key e conectar sua instância.

Seja feliz =)

Instalar Evolution API v1.0

Link: https://github.com/EvolutionAPI/evolution-api

git clone https://github.com/EvolutionAPI/evolution-api.git

Versões da V1 disponíveis para download anexo.

evolution-api-1.7.1.zip

evolution-api-1.7.5.zip

evolution-api-1.8.2.zip

Instalação Evolution API oficial

Link: https://doc.evolution-api.com/v2/pt/install/docker

Docker

Pré-requisitos: Antes de prosseguir com a instalação da Evolution API v2 utilizando Docker, certifique-se de que você já tenha configurado os serviços necessários, como PostgreSQL e Redis. Siga os links abaixo para mais detalhes:

Pré-requisitos: Antes de prosseguir com a instalação da Evolution API v2 utilizando Docker, certifique-se de que você já tenha configurado os serviços necessários, como PostgreSQL e Redis. Siga os links abaixo para mais detalhes:

- Configuração do Banco de Dados
- Configuração do Redis

Estas instruções de instalação assumem que você já instalou o Docker em sua máquina. Você pode encontrar informações sobre como instalar o Docker na

Documentação Oficial do Docker.

A Evolution API v2 está pronta para o Docker e pode ser facilmente implantada com Docker no modo standalone ou swarm. O repositório oficial do Evolution API contém todos os arquivos de composição necessários para instalar e executar a API.

Docker Compose

Implantar a Evolution API v2 usando o Docker Compose simplifica a configuração e o gerenciamento de seus contêineres Docker. Ele permite que você defina seu ambiente Docker em um arquivo `docker-compose.yaml` e, em seguida, use um único comando para iniciar tudo.

Arquivo Docker Compose

O exemplo a seguir ilustra como configurar o Docker Compose para ambientes standalone, ou seja, um único servidor em execução. Para a sincronização de dois servidores em paralelo ou maior escalabilidade, utilize o Docker Swarm, recomendado para usuários mais avançados.

Configuração Standalone

Atenção: Os comandos aqui descritos como `docker compose`, podem não funcionar em versões mais antigas do Docker. Caso você esteja usando uma versão mais antiga, substitua por `docker-compose`.

O Docker standalone é adequado quando a Evolution API será executada em apenas uma máquina, sem a necessidade de escalabilidade imediata. Esta é a forma mais conveniente para a maioria dos usuários.

Para começar, crie um arquivo `docker-compose.yml` com o seguinte conteúdo:

```
version: '3.9'
services:
  evolution-api:
    container_name: evolution_api
    image: atendai/evolution-api:v2.1.1
    restart: always
    ports:
      - "8080:8080"
    env_file:
      - .env
    volumes:
      - evolution_instances:/evolution/instances


volumes:
  evolution_instances:
```

Em seguida, crie um arquivo `.env` no mesmo diretório com o seguinte conteúdo mínimo:

```
AUTHENTICATION_API_KEY=mude-me
```

Para mais configurações, você pode pegar o arquivo de exemplo no repositório oficial. Confira também o guia de variáveis de ambiente aqui.

Inicializando a API

Navegue até o diretório que contém o arquivo `docker-compose.yml` e execute o seguinte comando para iniciar os serviços definidos no arquivo:

```
docker compose up -d
```

Esse comando baixará as imagens Docker necessárias, criará os serviços, redes e volumes definidos, e iniciará o serviço da Evolution API.

Verificando os Logs

Após executar o comando `docker compose up`, você pode verificar os logs para confirmar se os serviços estão em execução corretamente:

```
docker logs evolution_api
```

Parando o Serviço

Para parar o serviço, utilize o comando:

```
docker compose down
```

Acessando a API

Abra seu navegador e acesse http://localhost:8080 para verificar se a Evolution API está operacional.

Docker Swarm

Para configurar e gerenciar um cluster Docker Swarm para a Evolution API v2, siga as instruções abaixo. O Docker Swarm é ideal para ambientes que exigem escalabilidade e alta disponibilidade.

Instalação do Docker Swarm

Configurando o Servidor Manager

Se estiver utilizando um servidor da Hetzner, execute:

Copy



```
sudo apt-get update && apt-get install -y apparmor-utils
```

Etapa 1: Configuração do Hostname

1. Mude o hostname da máquina para identificá-la no cluster:

Copy



```
hostnamectl set-hostname manager1
```



2. Edite o arquivo `/etc/hosts` para adicionar o novo nome:

Copy



```
nano /etc/hosts
```

Adicione a linha:

Copy



```
127.0.0.1    manager1
```



3. Reinicie o sistema para aplicar as alterações:

Copy



```
reboot
```



4. Verifique o hostname:

Copy



```
hostnamectl
```

Etapa 2: Instalação do Docker

Instale o Docker executando:

Copy



```
curl -fsSL https://get.docker.com | bash
```

Etapa 3: Iniciando o Swarm

Inicie o Docker Swarm:

Copy



```
docker swarm init --advertise-addr IP_SERVER
```

Etapa 4: Configuração da Rede do Docker Swarm

Crie a rede overlay para o Docker Swarm:

Copy



```
docker network create --driver=overlay network_public
```

Anote o comando gerado para registrar os Workers:

Copy



```
docker swarm join --token HASH IP_SERVER:2377
```

Configurando o Servidor Worker

Se estiver utilizando um servidor da Hetzner, execute:

Copy



```
sudo apt-get update && apt-get install -y apparmor-utils
```

Etapa 1: Configuração do Hostname

1. Mude o hostname da máquina para identificá-la no cluster:

Copy

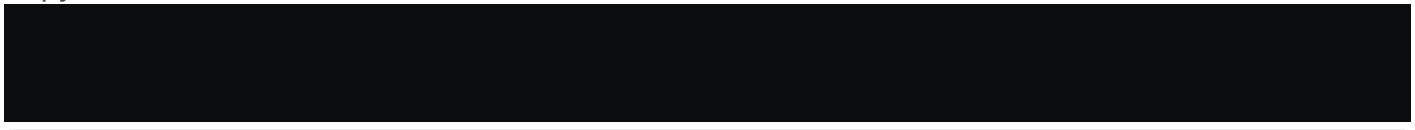


```
hostnamectl set-hostname worker1
```



2. Edite o arquivo `/etc/hosts` para adicionar o novo nome:

Copy



```
nano /etc/hosts
```

Adicione a linha:

Copy



```
127.0.0.1    worker1
```



3. Reinicie o sistema para aplicar as alterações:

Copy



```
reboot
```

Etapa 2: Instalação do Docker

Instale o Docker executando:

Copy



```
curl -fsSL https://get.docker.com | bash
```

Etapa 3: Adicionar o Worker ao Cluster

Execute o comando obtido anteriormente para adicionar o Worker ao cluster:

Copy



```
docker swarm join --token HASH IP_SERVER:2377
```

Pré-requisitos para a Evolution API via Swarm

Instalação do Traefik

Para instalar o Traefik no Docker Swarm, siga as instruções abaixo:

1. No servidor manager, crie um arquivo `traefik.yaml`:

Copy



```
nano traefik.yaml
```



2. Adicione o seguinte conteúdo ao arquivo:

Copy



```
version: "3.7"

services:
  traefik:
    image: traefik:2.11.2
    command:
      - "--api.dashboard=true"
      - "--providers.docker.swarmMode=true"
      - "--providers.docker.endpoint=unix:///var/run/docker.sock"
      - "--providers.docker.exposedbydefault=false"
      - "--providers.docker.network=network_public"
      - "--entrypoints.web.address=:80"
      - "--entrypoints.web.http.redirections.entryPoint.to=websecure"
      - "--entrypoints.web.http.redirections.entryPoint.scheme=https"
      - "--entrypoints.web.http.redirections.entrypoint.permanent=true"
      - "--entrypoints.websecure.address=:443"
      - "--certificatesresolvers.letsencryptresolver.acme.httpchallenge=true"
      - "--certificatesresolvers.letsencryptresolver.acme.httpchallenge.entrypoint=web"
      - "--certificatesresolvers.letsencryptresolver.acme.email=seu@email.com"
      - "--
certificatesresolvers.letsencryptresolver.acme.storage=/etc/traefik/letsencrypt/acme.json"
```

```
      - "--log.level=DEBUG"
      - "--log.format=common"
      - "--log.filePath=/var/log/traefik/traefik.log"
      - "--accesslog=true"
      - "--accesslog.filepath=/var/log/traefik/access-log"
    deploy:
      placement:
        constraints:
          - node.role == manager
      restart_policy:
        condition: on-failure
        delay: 5s
      labels:
        - "traefik.enable=true"
        - "traefik.http.middlewares.redirect-https.redirectscheme.scheme=https"
        - "traefik.http.middlewares.redirect-https.redirectscheme.permanent=true"
        - "traefik.http.routers.http-catchall.rule=hostregexp(`{host:.+}`)"
        - "traefik.http.routers.http-catchall.entrypoints=web"
        - "traefik.http.routers.http-catchall.middlewares=redirect-https@docker"
        - "traefik.http.routers.http-catchall.priority=1"
    volumes:
      - "/var/run

/docker.sock:/var/run/docker.sock:ro"
      - "vol_certificates:/etc/traefik/letsencrypt"
    ports:
      - target: 80
        published: 80
        mode: host
      - target: 443
        published: 443
        mode: host
    networks:
      - network_public

volumes:
  vol_certificates:
    external: true
    name: volume_swarm_certificates
```

```
networks:
  network_public:
    external: true
    name: network_public
```

3. Execute o comando abaixo para fazer o deploy da stack Traefik:

Copy

```
docker stack deploy --prune --resolve-image always -c traefik.yaml traefik
```

Deploy da Evolution API v2

Finalmente, para implantar a Evolution API v2 no Docker Swarm, use o arquivo de configuração disponível aqui com o seguinte conteúdo:

Copy

```
version: "3.7"

services:
  evolution_v2:
    image: atendai/evolution-api:v2.1.1
    volumes:
      - evolution_instances:/evolution/instances
    networks:
      - network_public
    environment:
      - SERVER_URL=https://evo2.site.com
      - DEL_INSTANCE=false
      - DATABASE_ENABLED=true
      - DATABASE_PROVIDER=postgresql
```

```
      - DATABASE_CONNECTION_URI=postgresql://postgres:SENHA@postgres:5432/evolution
      - DATABASE_SAVE_DATA_INSTANCE=true
      - DATABASE_SAVE_DATA_NEW_MESSAGE=true
      - DATABASE_SAVE_MESSAGE_UPDATE=true
      - DATABASE_SAVE_DATA_CONTACTS=true
      - DATABASE_SAVE_DATA_CHATS=true
      - DATABASE_SAVE_DATA_LABELS=true
      - DATABASE_SAVE_DATA_HISTORIC=true
      - DATABASE_CONNECTION_CLIENT_NAME=evolution_v2
      - RABBITMQ_ENABLED=false
      - RABBITMQ_URI=amqp://admin:admin@rabbitmq:5672/default
      - CACHE_REDIS_ENABLED=true
      - CACHE_REDIS_URI=redis://evo_redis:6379/1
      - CACHE_REDIS_PREFIX_KEY=evolution_v2
      - CACHE_REDIS_SAVE_INSTANCES=false
      - CACHE_LOCAL_ENABLED=false
      - S3_ENABLED=true
      - S3_ACCESS_KEY=
      - S3_SECRET_KEY=
      - S3_BUCKET=evolution
      - S3_PORT=443
      - S3_ENDPOINT=files.site.com
      - S3_USE_SSL=true
      - AUTHENTICATION_API_KEY=429683C4C977415CAAFCCE10F7D57E11
    deploy:
      mode: replicated
      replicas: 1
      placement:
        constraints:
          - node.hostname == evolution-manager
      labels:
        - traefik.enable=true
        - traefik.http.routers.evolution_v2.rule=Host(`evo2.site.com`)
        - traefik.http.routers.evolution_v2.entrypoints=websecure
        - traefik.http.routers.evolution_v2.tls.certresolver=letsencryptresolver
        - traefik.http.routers.evolution_v2.service=evolution_v2
        - traefik.http.services.evolution_v2.loadbalancer.server.port=8080
        - traefik.http.services.evolution_v2.loadbalancer.passHostHeader=true

volumes:
```

```
  evolution_instances:
    external: true
    name: evolution_v2_data

networks:
  network_public:
    external: true
    name: network_public
```

Após configurar e salvar o arquivo, faça o deploy da stack com o comando:

Copy



```
docker stack deploy --prune --resolve-image always -c evolution_api_v2.yaml evolution_v2
```

Acessando a API

Abra seu navegador e acesse https://evo2.site.com para verificar se a Evolution API está operacional.

Instalação e Configurações Evolution Api Cloud

Nova Evolution API

Instalação Evolution Api Cloud

Link: https://github.com/EvolutionAPI/evolution-api/tree/main

git clone: https://github.com/EvolutionAPI/evolution-api.git

Evolution Api

Docker Image Whatsapp Group Discord Community Postman Collection Documentation License Support Sponsors

cover.png

Evolution API

Evolution API began as a WhatsApp controller API based on CodeChat, which in turn implemented the Baileys library. While originally focused on WhatsApp, Evolution API has grown into a comprehensive platform supporting multiple messaging services and integrations. We continue to acknowledge CodeChat for laying the groundwork.

Today, Evolution API is not limited to WhatsApp. It integrates with various platforms such as Typebot, Chatwoot, Dify, and OpenAI, offering a broad array of functionalities beyond messaging. Evolution API supports both the Baileys-based WhatsApp API and the official WhatsApp Business API, with upcoming support for Instagram and Messenger.

Looking for a Lightweight Version?

For those who need a more streamlined and performance-optimized version, check out Evolution API Lite. It's designed specifically for microservices, focusing solely on connectivity without integrations or audio conversion features. Ideal for environments that prioritize simplicity and efficiency.

Types of Connections

Evolution API supports multiple types of connections to WhatsApp, enabling flexible and powerful integration options:

- *WhatsApp API - Baileys*:
  - A free API based on WhatsApp Web, leveraging the Baileys library.
  - This connection type allows control over WhatsApp Web functionalities through a RESTful API, suitable for multi-service chats, service bots, and other WhatsApp-integrated systems.
  - Note: This method relies on the web version of WhatsApp and may have limitations compared to official APIs.
- *WhatsApp Cloud API*:
  - The official API provided by Meta (formerly Facebook).
  - This connection type offers a robust and reliable solution designed for businesses needing higher volumes of messaging and better integration support.
  - The Cloud API supports features such as end-to-end encryption, advanced analytics, and more comprehensive customer service tools.
  - To use this API, you must comply with Meta's policies and potentially pay for usage based on message volume and other factors.

Integrations

Evolution API supports various integrations to enhance its functionality. Below is a list of available integrations and their uses:

- Typebot:
  - Build conversational bots using Typebot, integrated directly into Evolution with trigger management.
- Chatwoot:
  - Direct integration with Chatwoot for handling customer service for your business.
- RabbitMQ:
  - Receive events from the Evolution API via RabbitMQ.

- Amazon SQS:
  - Receive events from the Evolution API via Amazon SQS.
- Socket.io:
  - Receive events from the Evolution API via WebSocket.
- Dify:
  - Integrate your Evolution API directly with Dify AI for seamless trigger management and multiple agents.
- OpenAI:
  - Integrate your Evolution API with OpenAI for AI capabilities, including audio-to-text conversion, available across all Evolution integrations.
- Amazon S3 / Minio:
  - Store media files received in Amazon S3 or Minio.

Telemetry Notice

To continuously improve our services, we have implemented telemetry that collects data on the routes used, the most accessed routes, and the version of the API in use. We would like to assure you that no sensitive or personal data is collected during this process. The telemetry helps us identify improvements and provide a better experience for users.

Evolution Support Premium

Join our Evolution Pro community for expert support and a weekly call to answer questions. Visit the link below to learn more and subscribe:

Click here to learn more

Donate to the project.

Github Sponsors

https://github.com/sponsors/EvolutionAPI

Content Creator Partners

We are proud to collaborate with the following content creators who have contributed valuable insights and tutorials about Evolution API:

- Promovaweb
- Sandeco
- Comunidade ZDG
- Francis MNO
- Pablo Cabral
- XPop Digital
- Costar Wagner Dev
- Dante Testa
- Rubén Salazar
- OrionDesign
- IMPA 365
- Comunidade Hub Connect
- dSantana Automações
- Edison Martins
- Astra Online
- MKT Seven Automações
- Vamos automatizar

License

Evolution API is licensed under the Apache License 2.0, with the following additional conditions:

1. **LOGO and copyright information**: In the process of using Evolution API's frontend components, you may not remove or modify the LOGO or copyright information in the Evolution API console or applications. This restriction is inapplicable to uses of Evolution API that do not involve its frontend components.
2. **Usage Notification Requirement**: If Evolution API is used as part of any project, including closed-source systems (e.g., proprietary software), the user is required to display a clear notification within the system that Evolution API is being utilized. This notification should be visible to system administrators and accessible from the system's

documentation or settings page. Failure to comply with this requirement may result in the necessity for a commercial license, as determined by the producer.

Please contact contato@evolution-api.com to inquire about licensing matters.

Apart from the specific conditions mentioned above, all other rights and restrictions follow the Apache License 2.0. Detailed information about the Apache License 2.0 can be found at http://www.apache.org/licenses/LICENSE-2.0.

© 2025 Evolution API

Instalação Evolution API Cloud update

LInk: https://github.com/EvolutionAPI/evolution-api git clone: https://github.com/EvolutionAPI/evolution-api.git

Evolution API

Evolution API began as a WhatsApp controller API based on CodeChat, which in turn implemented the Baileys library. While originally focused on WhatsApp, Evolution API has grown into a comprehensive platform supporting multiple messaging services and integrations. We continue to acknowledge CodeChat for laying the groundwork.

Today, Evolution API is not limited to WhatsApp. It integrates with various platforms such as Typebot, Chatwoot, Dify, and OpenAI, offering a broad array of functionalities beyond messaging. Evolution API supports both the Baileys-based WhatsApp API and the official WhatsApp Business API, with upcoming support for Instagram and Messenger.

Looking for a Lightweight Version?

For those who need a more streamlined and performance-optimized version, check out Evolution API Lite. It's designed specifically for microservices, focusing solely on connectivity without integrations or audio conversion features. Ideal for environments that prioritize simplicity and efficiency.

Types of Connections

Evolution API supports multiple types of connections to WhatsApp, enabling flexible and powerful integration options:

- *WhatsApp API - Baileys*:
  - A free API based on WhatsApp Web, leveraging the Baileys library.
  - This connection type allows control over WhatsApp Web functionalities through a RESTful API, suitable for multi-service chats, service bots, and other WhatsApp-integrated systems.
  - Note: This method relies on the web version of WhatsApp and may have limitations compared to official APIs.
- *WhatsApp Cloud API*:
  - The official API provided by Meta (formerly Facebook).
  - This connection type offers a robust and reliable solution designed for businesses needing higher volumes of messaging and better integration support.
  - The Cloud API supports features such as end-to-end encryption, advanced analytics, and more comprehensive customer service tools.
  - To use this API, you must comply with Meta's policies and potentially pay for usage based on message volume and other factors.

Integrations

Evolution API supports various integrations to enhance its functionality. Below is a list of available integrations and their uses:

- Typebot:
  - Build conversational bots using Typebot, integrated directly into Evolution with trigger management.
- Chatwoot:
  - Direct integration with Chatwoot for handling customer service for your business.
- RabbitMQ:
  - Receive events from the Evolution API via RabbitMQ.
- Apache Kafka:
  - Receive events from the Evolution API via Apache Kafka for real-time event streaming and processing.
- Amazon SQS:
  - Receive events from the Evolution API via Amazon SQS.
- Socket.io:
  - Receive events from the Evolution API via WebSocket.
- Dify:
  - Integrate your Evolution API directly with Dify AI for seamless trigger management and multiple agents.
- OpenAI:

    - Integrate your Evolution API with OpenAI for AI capabilities, including audio-to-text conversion, available across all Evolution integrations.
- Amazon S3 / Minio:
    - Store media files received in Amazon S3 or Minio.

Community & Feedback

We value community input and feedback to continuously improve Evolution API:

□□ Feature Requests & Roadmap

- **Feature Requests**: Submit new feature ideas and vote on community proposals
- **Roadmap**: View planned features and development progress
- **Changelog**: Stay updated with the latest releases and improvements

□□ Community Support

- **WhatsApp Group**: Join our community for support and discussions
- **Discord Community**: Real-time chat with developers and users
- **GitHub Issues**: Report bugs and technical issues

□□ Security

- **Security Policy**: Guidelines for reporting security vulnerabilities
- **Security Contact**: contato@evolution-api.com

Telemetry Notice

To continuously improve our services, we have implemented telemetry that collects data on the routes used, the most accessed routes, and the version of the API in use. We would like to assure you that no sensitive or personal data is collected during this process. The telemetry helps us identify improvements and provide a better experience for users.

Evolution Support Premium

Join our Evolution Pro community for expert support and a weekly call to answer questions. Visit the link below to learn more and subscribe:

Click here to learn more

Donate to the project.

Github Sponsors

https://github.com/sponsors/EvolutionAPI

Content Creator Partners

We are proud to collaborate with the following content creators who have contributed valuable insights and tutorials about Evolution API:

- Promovaweb
- Sandeco
- Comunidade ZDG
- Francis MNO
- Pablo Cabral
- XPop Digital
- Costar Wagner Dev
- Dante Testa
- Rubén Salazar
- OrionDesign
- IMPA 365
- Comunidade Hub Connect
- dSantana Automações

- Edison Martins
- Astra Online
- MKT Seven Automações
- Vamos automatizar

License

Evolution API is licensed under the Apache License 2.0, with the following additional conditions:

1. **LOGO and copyright information**: In the process of using Evolution API's frontend components, you may not remove or modify the LOGO or copyright information in the Evolution API console or applications. This restriction is inapplicable to uses of Evolution API that do not involve its frontend components.
2. **Usage Notification Requirement**: If Evolution API is used as part of any project, including closed-source systems (e.g., proprietary software), the user is required to display a clear notification within the system that Evolution API is being utilized. This notification should be visible to system administrators and accessible from the system's documentation or settings page. Failure to comply with this requirement may result in the necessity for a commercial license, as determined by the producer.

Please contact contato@evolution-api.com to inquire about licensing matters.

Apart from the specific conditions mentioned above, all other rights and restrictions follow the Apache License 2.0. Detailed information about the Apache License 2.0 can be found at http://www.apache.org/licenses/LICENSE-2.0.

© 2025 Evolution API