
API

Update chatbot settings

The Update Chatbot Settings API allows you to edit the settings of your chatbot, such as its name, base prompt, initial messages, suggested messages, visibility status, domain restrictions, IP limits, AI model, and more.

Endpoint

POST <https://www.chatbase.co/api/v1/update-chatbot-settings>

Request Headers

The API request must include the following header:

Authorization: Bearer <Your-Secret-Key> - The secret key for authenticating the API request.

Request Body Parameters

The request payload should be sent as a JSON object in the body of the API request. The payload can include the following parameters:

chatbotId (string, required): Refers to the ID of the chatbot you want to interact with (found on the chatbot settings page).



name (string, optional): The new name for the chatbot.

instructions (string, optional): The base prompt; instructions for how the chatbot should behave.

initialMessages (array of strings, optional): An array of strings containing initial messages to be used by the chatbot. These messages will be what the chatbot greets the user with.

suggestedMessages (array of strings, optional): An array of strings containing suggested messages to be used by the chatbot. These messages will appear above the text input in the chatbot interface.

visibility (string, optional): The visibility status of the chatbot, which can be set to either 'private' or 'public'.

onlyAllowOnAddedDomains (boolean, optional): A boolean flag to enable or disable allowing the chatbot iframe and widget only on specific domains.

domains (array of strings, optional): An array of strings containing allowed domains for the chatbot.

ipLimit (integer, optional): The limit on the number of messages to be sent from one device every **ipLimitTimeframe** seconds.

ipLimitTimeframe (integer, optional): The timeframe (in seconds) in which the messages limit is applied.

ipLimitMessage (string, optional): The message to be displayed when the IP limit is exceeded.

model (string, optional): The AI model used by the chatbot, which can be set to one of the models mentioned earlier. If not set, the model set in the chatbot settings is used. The option for 'gpt-4' only works on the Standard and the Unlimited plans.

temp (number, optional): The temperature parameter for the AI model with a range of [0, 1]. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic.



collectCustomerInformation (object, optional): An object containing the information to be collected from the customer.

title (string): This field provides a title for the customer contact details section, for example, "Let us know how to contact you."

name (object): This sub-object allows you to collect the customer's name. It consists of the following parameters:

label (string): A label to describe the name field, helping customers understand what information is being requested.

active (boolean): Indicates whether the name field is activated or not. If set to true, the chatbot will prompt the customer to provide their name.

email (object): This sub-object facilitates the collection of the customer's email address. It includes the following parameters:

label (string): A label that clarifies the purpose of the email field, making it clear to customers what is being asked.

active (boolean): Determines whether the email field is active. When set to true, the chatbot will ask the customer for their email.

phone (object): This sub-object is intended for gathering the customer's phone number. It incorporates the following parameters:

label (string): A label that explains the purpose of the phone number field, helping customers understand the information being sought.

active (boolean): Specifies whether the phone number field is in use. When set to true, the chatbot will prompt the customer to provide their phone number.

styles (object, optional): An object containing the styling parameters, allowing you to customize the appearance of the chatbot's user interface.

theme (string): Choose 'dark' or 'light' for the chatbot interface theme.

userMessageColor (string): Define user message color in Hex format

buttonColor (string): Set chat bubble button color using Hex format.

displayName (string): Display a branded name within the chat interface.



Chatbase

autoOpenChatWindowAfter (integer): Automatically open chat window after a defined time (in seconds).

alignChatButton (string): Align the chatbot button in the interface, either 'left' or 'right'.

Example Request Payload

```
{
  "collectCustomerInformation": {
    "name": {
      "label": "Name",
      "active": true
    },
    "email": {
      "label": "Email",
      "active": true
    },
    "title": "Let us know how to contact you"
  },
  "styles": {
    "theme": "dark",
    "userMessageColor": "#3B81F7",
    "buttonColor": "#3B81F7",
    "displayName": "Product Hunt",
    "autoOpenChatWindowAfter": 4,
    "alignChatButton": "left"
  },
  "chatbotId": "[Your ChatbotID]",
  "name": "my Chatbot",
  "instructions": "I want you to act as a document that I am having a conversation with",
  "initialMessages": ["Hi! What can I help you with?"],
  "suggestedMessages": ["Hi! What are you?"],
  "visibility": "public",
  "onlyAllowOnAddedDomains": true,
  "domains": ["example.com"],
  "ipLimit": 20,
  "ipLimitTimeframe": 240,
```



Chatbase

```
"ipLimitMessage": "Too many messages in a row",
"model": "gpt-4",
"temp": 0
}
```

Example Requests

```
const options = {
  method: 'POST',
  headers: {accept: 'application/json', 'content-type': 'application/json'},
  body: JSON.stringify({
    collectCustomerInformation: {
      name: {label: 'Name', active: true},
      email: {label: 'Email', active: true},
      title: 'Let us know how to contact you',
    },
    styles: {
      theme: 'dark',
      userMessageColor: '#3B81F7',
      buttonColor: '#3B81F7',
      displayName: 'Product Hunt',
      autoOpenChatWindowAfter: 4,
      alignChatButton: 'left',
    },
    chatbotId: '[Your ChatbotID]',
    name: 'my Chatbot',
    instructions:
      'I want you to act as a document that I am having a conversation with. Your
initialMessages: ['Hi! What can I help you with?'],
suggestedMessages: ['Hi! What are you?'],
visibility: 'public',
onlyAllowOnAddedDomains: true,
domains: ['example.com'],
ipLimit: 20,
ipLimitTimeframe: 240,
ipLimitMessage: 'Too many messages in a row',
model: 'gpt-4',
```

```
fetch('https://www.chatbase.co/api/v1/update-chatbot-settings', options)
  .then((response) => response.json())
  .then((response) => console.log(response))
  .catch((err) => console.error(err))
```

Python

```
import requests

url = "https://www.chatbase.co/api/v1/update-chatbot-settings"

payload = {
  "collectCustomerInformation": { "name": {
    "label": "Name",
    "active": True
  } },
  "styles": {
    "theme": "dark",
    "userMessageColor": "#3B81F7",
    "buttonColor": "#3B81F7",
    "displayName": "Product Hunt",
    "autoOpenChatWindowAfter": 4,
    "alignChatButton": "left"
  },
  "chatbotId": "[Your ChatbotID]",
  "name": "my Chatbot",
  "instructions": "I want you to act as a document that I am having a conversat.",
  "initialMessages": ["Hi! What can I help you with?"],
  "suggestedMessages": ["Hi! What are you?"],
  "visibility": "private",
  "onlyAllowOnAddedDomains": True,
  "domains": ["example.com"],
  "ipLimit": 20,
  "ipLimitTimeframe": 240,
```



Chatbase

```
"ipLimitMessage": "Too many messages in a row",
"model": "gpt-4",
"temp": 0.2
```

```
}
headers = {
    "accept": "application/json",
    "content-type": "application/json"
}
```

```
response = requests.post(url, json=payload, headers=headers)
```

```
print(response.text)
```

Shell

```
curl --request POST \
      --url https://www.chatbase.co/api/v1/update-chatbot-settings \
      --header 'accept: application/json' \
      --header 'content-type: application/json' \
      --data '
{
  "collectCustomerInformation": {
    "name": {
      "label": "Name",
      "active": true
    }
  },
  "styles": {
    "theme": "dark",
    "userMessageColor": "#3B81F7",
    "buttonColor": "#3B81F7",
    "displayName": "Product Hunt",
    "autoOpenChatWindowAfter": 4,
    "alignChatButton": "left"
  },
  "chatbotId": "[Your ChatbotID]",
  "name": "my Chatbot",
  "instructions": "I want you to act as a document that I am having a conversation"
```



Chatbase

```
"initialMessages": [
  "Hi! What can I help you with?"
],
"suggestedMessages": [
  "Hi! What are you?"
],
"visibility": "private",
"onlyAllowOnAddedDomains": true,
"domains": [
  "example.com"
],
"ipLimit": 20,
"ipLimitTimeframe": 240,
"ipLimitMessage": "Too many messages in a row",
"model": "gpt-4",
"temp": 0.2
},
{
```

HTTP

POST /api/v1/update-chatbot-settings HTTP/1.1

Accept: application/json

Content-Type: application/json

Host: www.chatbase.co

Content-Length: 898

```
{"collectCustomerInformation":{"name":{"label":"Name","active":true}},"styles":{"
```

Responses

202: Returns a success message indicating that the chatbot settings have been updated successfully.

400: If the request is missing the `chatbotId` parameter.



401: If the API request is unauthorized.

Chatbase

404: If the `chatbotId` provided is invalid and does not correspond to any existing chatbot.

500: If there is an internal server error while processing the request.

Error Handling

If there are any errors during the API request, appropriate HTTP status codes will be returned along with error messages in the response body. Make sure to handle these errors gracefully in your application.

That's it! You should now be able to edit your chatbot settings using the Update Chatbot Settings API.

[◀ Delete chatbot profile picture](#)

[Message a chatbot ▶](#)

Powered by Mintlify