Apireference.txt

Basic

Send Call (Simple)

Send an AI phone call with a custom objective and actions.

POST

/

v1

/

calls

Send

Header

Body

​

Headers

authorization

string

required

Your API key for authentication.

​

Body

phone\_number

string

required

The phone number to call. Country code defaults to +1 (US) if not specified.

Formatting is flexible, however for the most predictable results use the E.164 format.

Formatting Examples

task

string

required

Provide instructions, relevant information, and examples of the ideal conversation flow.

Best Practices

​

Response

status

string

Can be success or error.

call\_id

string

A unique identifier for the call (present only if status is success).

import requests

url = "https://api.bland.ai/v1/calls"

payload = {

"phone\_number": "<string>",

"task": "<string>"

}

headers = {

"authorization": "<authorization>",

"Content-Type": "application/json"

}

response = requests.request("POST", url, json=payload, headers=headers)

print(response.text)

const options = {

method: 'POST',

headers: {authorization: '<authorization>', 'Content-Type': 'application/json'},

body: '{"phone\_number":"<string>","task":"<string>"}'

};

fetch('https://api.bland.ai/v1/calls', options)

.then(response => response.json())

.then(response => console.log(response))

.catch(err => console.error(err));

{

"status": "success",

"call\_id": "9d404c1b-6a23-4426-953a-a52c392ff8f1"

}

Basic

Send Call using Pathways (Simple)

Send an AI phone call with your own conversational pathway agent! Links - Video Tutorial | Step-by-step web tutorial

POST

/

v1

/

calls

Send

Header

authorization

string

\*

Enter authorization

Body

object

phone\_number

string

\*

Enter phone\_number

pathway\_id

string

\*

Enter pathway\_id

​

Headers

authorization

string

required

Your API key for authentication.

​

Body

phone\_number

string

required

The phone number to call. Country code defaults to +1 (US) if not specified.

Formatting is flexible, however for the most predictable results use the E.164 format.

Formatting Examples

Expected/Ideal Format:

“+12223334444"

"+91223334444"

"+61223334444”

Valid, but not recommended:

“2223334444"

"+1 (222) 333-4444"

"+1 222 333 4444"

"222-333-4444”

Invalid:

“12223334444"

"552223334444"

"non-numeric characters"

"2223334444 ext. 123”

pathway\_id

string

required

Follows the conversational pathway you created to guide the conversation.

You can access your pathway\_id by clicking on the ‘Copy ID’ button on your pathways here. If you don’t have any pathways, click the ‘Create Pathway’ button to create one!

Conversational Pathway Tutorial

Video tutorial

Step by step Web Tutorial

​

Response

status

string

Can be success or error.

call\_id

string

A unique identifier for the call (present only if status is success).

Send Call (Simple)

Send Call

twitter

linkedin

discord

Powered by Mintlify

cURL

Python

JavaScript

PHP

Go

Java

const options = {

method: 'POST',

headers: {authorization: '<authorization>', 'Content-Type': 'application/json'},

body: '{"phone\_number":"<string>","pathway\_id":"<string>"}'

};

fetch('https://api.bland.ai/v1/calls', options)

.then(response => response.json())

.then(response => console.log(response))

.catch(err => console.error(err));

Response

{

"status": "success",

"call\_id": "9d404c1b-6a23-4426-953a-a52c392ff8f1"

}

import requests

url = "https://api.bland.ai/v1/calls"

payload = {

"phone\_number": "<string>",

"pathway\_id": "<string>"

}

headers = {

"authorization": "<authorization>",

"Content-Type": "application/json"

}

response = requests.request("POST", url, json=payload, headers=headers)

print(response.text)

const options = {

method: 'POST',

headers: {authorization: '<authorization>', 'Content-Type': 'application/json'},

body: '{"phone\_number":"<string>","pathway\_id":"<string>"}'

};

fetch('https://api.bland.ai/v1/calls', options)

.then(response => response.json())

.then(response => console.log(response))

.catch(err => console.error(err));

Calls

Send Call

Send an AI phone call with a custom objective and actions.

POST

/

v1

/

calls

Send

Header

authorization

string

\*

Enter authorization

encrypted\_key

string

Enter encrypted\_key

Body

object

phone\_number

string

\*

Enter phone\_number

task

string

\*

Enter task

pathway\_id

string

Enter pathway\_id

start\_node\_id

string

Enter start\_node\_id

voice

string

Enter voice

background\_track

string

Enter background\_track

first\_sentence

string

Enter first\_sentence

wait\_for\_greeting

boolean

Select option

block\_interruptions

boolean

Select option

interruption\_threshold

number

Enter interruption\_threshold

model

string

Enter model

temperature

number

Enter temperature

keywords

array

pronunciation\_guide

array

transfer\_phone\_number

string

Enter transfer\_phone\_number

transfer\_list

object

language

string

Enter language

timezone

string

Enter timezone

request\_data

object

tools

array

dynamic\_data

array

start\_time

string

Enter start\_time

voicemail\_message

string

Enter voicemail\_message

voicemail\_action

object

retry

object

max\_duration

integer

Enter max\_duration

record

boolean

Select option

from

string

Enter from

webhook

string

Enter webhook

webhook\_events

array

metadata

object

summary\_prompt

string

Enter summary\_prompt

analysis\_prompt

string

Enter analysis\_prompt

analysis\_schema

object

answered\_by\_enabled

boolean

Select option

​

Headers

authorization

string

required

Your API key for authentication.

encrypted\_key

string

A special key for using a BYOT (Bring Your Own Twilio) account. Only required for sending calls from your own Twilio account.

​

Body

phone\_number

string

required

The phone number to call. Country code defaults to +1 (US) if not specified.

For best results, use the E.164 format.

Formatting Examples

Expected/Ideal Format:

“+12223334444"

"+91223334444"

"+61223334444”

Valid, but not recommended:

“2223334444"

"+1 (222) 333-4444"

"+1 222 333 4444"

"222-333-4444”

Invalid:

“12223334444"

"552223334444"

"non-numeric characters"

"2223334444 ext. 123”

task

string

required

Provide instructions, relevant information, and examples of the ideal conversation flow.

This is your prompt where you are telling the agent what to do.

Recommendations:

Include context and a background/persona for the agent like "You are {name}, a customer service agent at {company} calling {name} about {reason}.

Phrase instructions like you are speaking to the agent before the call.

Any time you tell the agent not to do something, provide an example of what they should do instead.

Keep the prompt under 2,000 characters where possible.

Want to easily test out exactly how your agent will behave? Try out Agent Testing!

pathway\_id

string

This is the pathway ID for the pathway you have created on our dev portal. You can access the ID of your pathways by clicking the ‘Copy ID’ button of your pathway here

Note: Certain parameters do not apply when using pathways.

Unused Parameters

task - The pathway substitutes as the agent’s instructions.

model - We use our own fine-tuned models under the hood.

tools - Replaced by ‘Webhook’ Node in Pathways

transfer\_list - Replaced by ‘Transfer Call’ Node in Pathways

transfer\_phone\_number - Replaced by ‘Transfer Call’ Node in Pathways

Example Simple Request body:

{

"phone\_number": "+1975934749",

"pathway\_id": "a0f0d4ed-f5f5-4f16-b3f9-22166594d7a7"

}

start\_node\_id

string

This is the node ID for the node you want the pathway to start from. You can access the ID of your nodes here

Note: This parameter is only used when pathway\_id is provided.

Example Simple Request body:

{

"phone\_number": "+1975934749",

"pathway\_id": "a0f0d4ed-f5f5-4f16-b3f9-22166594d7a7",

"start\_node\_id": "a0cd5fed-f3f3-5412-1339-13567921d7b8"

}

​

Agent Parameters (Body)

voice

string

default: "mason"

The voice of the AI agent to use. Accepts any form of voice ID, including custom voice clones and voice presets.

Default voices can be referenced directly by their name instead of an id.

Usage example: voice: "maya"

Bland Curated voices:

maya

mason

ryan

adriana

tina

matt

evelyn

Use the GET /v1/voices endpoint to see a full list of your available voices.

Moving from `voice\_id` to `voice`

Note: Including voice\_id or reduce\_latency in your request is still supported, but not recommended.

The previous structure to select voices used both voice\_id and reduce\_latency. To simplify the process, we’ve combined these into a single voice parameter.

If the first two letters of voice are RL, that is equivalent to settings reduce\_latency to true.

Prefixing the voice ID with HQ will use the high fidelity version of the voice.

The integer following the prefix is the voice\_id from before.

Example:

reduce\_latency: true, voice\_id: 0 is equivalent to voice: "RL0"

reduce\_latency: false, voice\_id: 3 is equivalent to voice: "HQ3"

Including reduce\_latency may override the voice parameter, so exclude it when using voice.

Moving from `voice\_preset\_id` to `voice`

All presets have been migrated to the voice parameter and can use either the preset name or ID.

If you used to have a voice\_preset\_id of "2f9fdbc7-4bf2-4792-8a18-21ce3c93978f", you can now use voice: "2f9fdbc7-4bf2-4792-8a18-21ce3c93978f".

background\_track

string

Select an audio track that you’d like to play in the background during the call. The audio will play continuously when the agent isn’t speaking, and is incorporated into it’s speech as well.

Use this to provide a more natural, seamless, engaging experience for the conversation. We’ve found this creates a significantly smoother call experience by minimizing the stark differences between total silence and the agent’s speech.

Options:

null - Default, will play audible but quiet phone static.

office - Office-style soundscape. Includes faint typing, chatter, clicks, and other office sounds.

cafe - Cafe-like soundscape. Includes faint talking, clinking, and other cafe sounds.

restaurant - Similar to cafe, but more subtle.

none - Minimizes background noise

first\_sentence

string

Makes your agent say a specific phrase or sentence for it’s first response.

wait\_for\_greeting

boolean

default: false

By default, the agent starts talking as soon as the call connects.

When wait\_for\_greeting is set to true, the agent will wait for the call recipient to speak first before responding.

block\_interruptions

boolean

default: false

When set to true, the AI will not respond or process interruptions from the user.

interruption\_threshold

number

default: 100

Adjusts how patient the AI is when waiting for the user to finish speaking.

Lower values mean the AI will respond more quickly, while higher values mean the AI will wait longer before responding.

Recommended range: 50-200

50: Extremely quick, back and forth conversation

100: Balanced to respond at a natural pace

200: Very patient, allows for long pauses and interruptions. Ideal for collecting detailed information.

Try to start with 100 and make small adjustments in increments of ~10 as needed for your use case.

model

string

default: "enhanced"

Select a model to use for your call.

Options: base, turbo and enhanced.

In nearly all cases, enhanced is the best choice.

Model Differences

temperature

float

default: "0.7"

A value between 0 and 1 that controls the randomness of the LLM. 0 will cause more deterministic outputs while 1 will cause more random.

Example Values: "0.9", "0.3", "0.5"

keywords

string[]

default: "[]"

These words will be boosted in the transcription engine - recommended for proper nouns or words that are frequently mis-transcribed.

For example, if the word “Reece” is frequently transcribed as a homonym like “Reese” you could do this:

{

"keywords": ["Reece"]

}

For stronger keyword boosts, you can place a colon then a boost factor after the word. The default boost factor is 2.

{

"keywords": ["Reece:3"]

}

pronunciation\_guide

array

The pronunciation guide is an array of objects that guides the agent on how to say specific words. This is great for situations with complicated terms or names.

[

{

"word": "example",

"pronunciation": "ex-am-ple",

"case\_sensitive": "false",

"spaced": "false"

},

{

"word": "API",

"pronunciation": "A P I",

"case\_sensitive": "true",

"spaced": "true"

}

]

Object Parameters

word — the word you want to guide the LLM on how to pronounce

pronunciation — the word you want to guide the LLM on how to pronounce.

case\_sensitive — whether or not to consider case. Particularly useful with names. EG: ‘Max’ the name versus ‘max’ the word. Defaults to false. Not required.

spaced — whether or not to consider spaces. When true the word ‘high’ would be flagged but NOT ‘hightop’. Defaults to true. Not required.

transfer\_phone\_number

string

A phone number that the agent can transfer to under specific conditions - such as being asked to speak to a human or supervisor.

Prompting Notes

For best results:

Specify conditions that the agent should transfer to a human under (examples are great!)

In the task, refer to the action solely as “transfer” or “transferring”.

Alternate phrasing such as “swap” or “switch” can mislead the agent, causing the action to be ignored.

transfer\_list

object

Give your agent the ability to transfer calls to a set of phone numbers.

Overrides transfer\_phone\_number if a transfer\_list.default is specified.

Will default to transfer\_list.default, or the chosen phone number.

Example usage to route calls to different departments:

{

"transfer\_list": {

"default": "+12223334444",

"sales": "+12223334444",

"support": "+12223334444",

"billing": "+12223334444"

}

}

language

string

default: "en-US"

Select a supported language of your choice. Optimizes every part of our API for that language - transcription, speech, and other inner workings.

Expand to view language options

The available language options are as follows:

en - English

en-US - English (US)

en-GB - English (UK)

en-AU - English (Australia)

en-NZ - English (New Zealand)

en-IN - English (India)

zh - Chinese (Mandarin, Simplified)

zh-CN - Chinese (Mandarin, Simplified, China)

zh-Hans - Chinese (Mandarin, Simplified, Hans)

zh-TW - Chinese (Mandarin, Traditional)

zh-Hant - Chinese (Mandarin, Traditional, Hant)

es - Spanish

es-419 - Spanish (Latin America)

fr - French

fr-CA - French (Canada)

de - German

el - Greek

hi - Hindi

hi-Latn - Hindi (Latin script)

ja - Japanese

ko - Korean

ko-KR - Korean (Korea)

pt - Portuguese

pt-BR - Portuguese (Brazil)

it - Italian

nl - Dutch

pl - Polish

ru - Russian

sv - Swedish

sv-SE - Swedish (Sweden)

da - Danish

da-DK - Danish (Denmark)

fi - Finnish

id - Indonesian

ms - Malay

tr - Turkish

uk - Ukrainian

bg - Bulgarian

cs - Czech

ro - Romanian

sk - Slovak

timezone

string

default: "America/Los\_Angeles"

Set the timezone for the call. Handled automatically for calls in the US.

This helps significantly with use cases that rely on appointment setting, scheduling, or behaving differently based on the time of day.

Timezone options are here in the TZ identifier column.

request\_data

object

Any JSON you put in here will be visible to the AI agent during the call - and can also be referenced with Prompt Variables.

For example, let’s say in your app you want to programmatically set the name of the person you’re calling. You could set request\_data to:

{

"phone\_number": "+1...",

"task": "...",

"first\_sentence": "Hello {{name}}! How are you doing today?", // also works in the prompt, tools, etc.

"request\_data": {

"name": "John Doe",

}

}

For further information about how Prompt Variables work, check out the Custom Tools tutorial.

tools

array

Interact with the real world through API calls.

Detailed tutorial here: Custom Tools

dynamic\_data

array

Make dynamic requests to external APIs and use the data in your AI’s responses.

Hide properties

Each request object should contain:

url: The URL of the external API to fetch data from.

response\_data: An array of objects describing how to parse and use the data fetched from the API. Explained in more detail below.

The following are optional:

method: Allows GET or POST. Default: GET

cache: Whether to fetch the data once at the beginning of the call, or to re-check continuously for data that might change mid-call. Default: true

headers: An object of headers to send with the request.

body: The body of the request.

The following variables can be injected into the dynamic request body:

{{from}} (Ex. +12223334444)

{{to}}

{{short\_from}} (Ex. 2223334444)

{{short\_to}}

{{call\_id}}

These string values will be replaced in each dynamic\_data[].body where they’re used by system values in each request.

Try out with this example:

{

"dynamic\_data": [

{

"url": "https://api.coindesk.com/v1/bpi/currentprice.json",

"response\_data": [

{

"name": "BTC Price USD",

"data": "bpi.USD.rate",

"context": "Current BTC Price: ${{BTC Price USD}} USD"

},

{

"name": "BTC Price EUR",

"data": "bpi.EUR.rate",

"context": "In Euros: {{BTC Price USD}} EUR"

}

]

}

]

}

dynamic\_data[i].response\_data

array

An array of objects describing how to parse and use the data fetched from the API.

Each object in this array should contain:

name: A label for the fetched data.

Example: "Flight Status"

data: The JSON path in the API response to extract the data from.

Example: "user.flights[0].status"

context: How this data should be incorporated into the AI’s knowledge.

Example: "John's flight is currently {{Flight Status}}"

​

Call Parameters (Body)

start\_time

string

The time you want the call to start. If you don’t specify a time (or the time is in the past), the call will send immediately.

Set your time in the format YYYY-MM-DD HH:MM:SS -HH:MM (ex. 2021-01-01 12:00:00 -05:00).

The timezone is optional, and defaults to UTC if not specified.

Note: Scheduled calls can be cancelled with the POST /v1/calls/:call\_id/stop endpoint.

voicemail\_message

string

When the AI encounters a voicemail, it will leave this message after the beep and then immediately end the call.

Warning: If amd is set to true or voicemail\_action is set to ignore, then this will still work for voicemails, but it will not hang up for IVR systems.

voicemail\_action

enum

default: "hangup"

This is processed separately from the AI’s decision making, and overrides it.

Options:

hangup

leave\_message

ignore

Examples:

Call is answered by a voicemail (specifically with a beep or tone):

If voicemail\_message is set, that message will be left and then the call will end.

Otherwise, the call immediately ends (regardless of amd)

Call is answered by an IVR system or phone tree:

If amd is set to true, the AI will navigate the system and continue as normal.

If voicemail\_action is set to ignore, the AI will ignore the IVR and continue as normal.

Otherwise, if voicemail\_message is set then it’ll leave that message and end the call.

Finally, if none of those conditions are met, the call will end immediately.

Note: If voicemail\_message is set, then the AI will leave the message regardless of the voicemail\_action.

retry

object

If the call goes to voicemail, you can set up the call to retry, after a configurable delay. You can also update the voicemail\_action, and voicemail\_message in the retry object, for the re-tried call.

Takes in the following parameters:

wait (integer): The delay in seconds before the call is retried.

voicemail\_action (enum): The action to take when the call goes to voicemail. Options: hangup, leave\_message, ignore.

voicemail\_message (string): The message to leave when the call goes to voicemail.

Example:

{

"retry": {

"wait": 10,

"voicemail\_action": "leave\_message",

"voicemail\_message": "Hello, this is a test message."

}

}

max\_duration

integer

default: "30"

When the call starts, a timer is set for the max\_duration minutes. At the end of that timer, if the call is still active it will be automatically ended.

Example Values: 20, 2

record

boolean

default: "false"

To record your phone call, set record to true. When your call completes, you can access through the recording\_url field in the call details or your webhook.

from

string

Specify a phone number to call from that you own or have uploaded from your Twilio account. Country code is required, spaces or parentheses must be excluded.

By default, calls are initiated from a separate pool of numbers owned by Bland.

webhook

string

When the call ends, we’ll send the call details in a POST request to the URL you specify here.

The request body will match the response from the GET /v1/calls/:call\_id endpoint.

webhook\_events

string[]

Specify which events you want to stream to the webhook, during the call.

Options:

queue

call

latency

webhook

tool

dynamic\_data

Example Payload:

{

"message": "LLM: 411ms",

"call\_id": "0fb3c518-e941-48fd-a32c-67d59c541336",

"category": "latency",

"log\_level": "performance"

}

metadata

object

Add any additional information you want to associate with the call. This can be useful for tracking or categorizing calls.

Anything that you put here will be returned in your webhook or in the call details under metadata.

Example:

{

"metadata": {

"campaign\_id": "1234",

"source": "web"

}

}

summary\_prompt

string

At the end of each call, a summary is generated based on the transcript - you can use this field to add extra instructions and context for how it should be summarized.

For example: "Summarize the call in French instead of English."

analysis\_prompt

string

Guides the output and provides additional instructions and clarifications for the analysis\_schema.

analysis\_schema

object

When the call ends, the transcript and call details will be analyzed by the AI.

Define a JSON schema for how you want to get information about the call - information like email addresses, names, appointment times or any other type of custom data.

In the webhook response or whenever you retrieve call data later, you’ll get the data you defined back under analysis.

For example, if you wanted to retrieve this information from the call:

{

"analysis\_schema": {

"email\_address": "email",

"first\_name": "string",

"last\_name": "string",

"wants\_to\_book\_appointment": "boolean",

"appointment\_time": "YYYY-MM-DD HH:MM:SS"

}

}

You would get it filled out like this in your webhook once the call completes:

{

"analysis": {

"email\_address": "johndoe@gmail.com",

"first\_name": "John",

"last\_name": "Doe",

"wants\_to\_book\_appointment": true,

"appointment\_time": "2024-01-01 12:00:00"

}

}

answered\_by\_enabled

boolean

default: false

If this is set to true, we process the audio from the start of the call to determine if it was answered by a human or a voicemail.

In the call details or webhook response, you’ll see the answered\_by field with the value human, unknown or voicemail.

Notes for accuracy:

When answered\_by is voicemail or human, that is nearly 100% accurate.

When it is unknown, try using text analysis by adding answered\_by to your analysis\_schema.

​

Response

status

string

Can be success or error.

call\_id

string

A unique identifier for the call (present only if status is success).

batch\_id

string

The batch ID of the call (present only if status is success).

message

string

A message explaining the status of the call.

errors

array

For validation errors, a detailed list of each field with an error and it’s error message.

Example:

{

"status": "error",

"message": "Invalid parameters",

"errors": [

"Missing required parameter: phone\_number.",

"Missing required parameter: task.",

"Phone number must be a string or number.",

"Task must be a string."

]

}

const options = {

method: 'POST',

headers: {authorization: '<authorization>', 'Content-Type': 'application/json'},

body: '{"phone\_number":"<string>","task":"<string>","pathway\_id":"<string>","start\_node\_id":"<string>","voice":"<string>","background\_track":"<string>","first\_sentence":"<string>","wait\_for\_greeting":true,"block\_interruptions":true,"interruption\_threshold":123,"model":"<string>","temperature":123,"keywords":["<string>"],"pronunciation\_guide":[{}],"transfer\_phone\_number":"<string>","transfer\_list":{},"language":"<string>","timezone":"<string>","request\_data":{},"tools":[{}],"dynamic\_data":[{"dynamic\_data[i].response\_data":[{}]}],"start\_time":"<string>","voicemail\_message":"<string>","voicemail\_action":{},"retry":{},"max\_duration":123,"record":true,"from":"<string>","webhook":"<string>","webhook\_events":["<string>"],"metadata":{},"summary\_prompt":"<string>","analysis\_prompt":"<string>","analysis\_schema":{},"answered\_by\_enabled":true}'

};

fetch('https://api.bland.ai/v1/calls', options)

.then(response => response.json())

.then(response => console.log(response))

.catch(err => console.error(err));

import requests

url = "https://api.bland.ai/v1/calls"

payload = {

"phone\_number": "<string>",

"task": "<string>",

"pathway\_id": "<string>",

"start\_node\_id": "<string>",

"voice": "<string>",

"background\_track": "<string>",

"first\_sentence": "<string>",

"wait\_for\_greeting": True,

"block\_interruptions": True,

"interruption\_threshold": 123,

"model": "<string>",

"temperature": 123,

"keywords": ["<string>"],

"pronunciation\_guide": [{}],

"transfer\_phone\_number": "<string>",

"transfer\_list": {},

"language": "<string>",

"timezone": "<string>",

"request\_data": {},

"tools": [{}],

"dynamic\_data": [{"dynamic\_data[i].response\_data": [{}]}],

"start\_time": "<string>",

"voicemail\_message": "<string>",

"voicemail\_action": {},

"retry": {},

"max\_duration": 123,

"record": True,

"from": "<string>",

"webhook": "<string>",

"webhook\_events": ["<string>"],

"metadata": {},

"summary\_prompt": "<string>",

"analysis\_prompt": "<string>",

"analysis\_schema": {},

"answered\_by\_enabled": True

}

headers = {

"authorization": "<authorization>",

"Content-Type": "application/json"

}

response = requests.request("POST", url, json=payload, headers=headers)

print(response.text)

{

"status": "success",

"message": "Call successfully queued.",

"call\_id": "9d404c1b-6a23-4426-953a-a52c392ff8f1",

"batch\_id": null

}

Calls

Analyze Call with AI

Analyzes a call of calls based using questions and goals.

POST

/

v1

/

calls

/

{call\_id}

/

analyze

Send

Header

authorization

string

\*

Enter authorization

Path

call\_id

string

\*

Enter call\_id

Body

object

goal

string

\*

Enter goal

questions

array

\*

​

Headers

authorization

string

required

Your API key for authentication.

​

Path Parameters

call\_id

string

required

The unique identifier for the call to be analyzed.

​

Request Body

goal

string

required

This is the overall purpose of the call. Provides context for the analysis to guide how the questions/transcripts are interpreted.

questions

string[][]

required

An array of questions to be analyzed for the call.

Each question should be an array with two elements: the question text and the expected answer type (e.g., “string”, “boolean”).

Fairly flexible in terms of the expected answer type, and unanswerable questions will default to null.

Examples:

"questions": [

["Who answered the call?", "human or voicemail"],

["Positive feedback about the product: ", "string"],

["Negative feedback about the product: ", "string"],

["Customer confirmed they were satisfied", "boolean"]

]

​

Response

status

object

Will be success if the request was successful.

message

string

Confirms the request was successful, or provides an error message if the request failed.

answers

array

Contains the analyzed answers for the call in an array.

credits\_used

number

Token-based price for the analysis request.

As a rough estimate, the base cost is 0.003 credits with an additional 0.0015 credits per call in the call.

Longer call transcripts and higher numbers of questions can increase the cost, however the cost scales very effectively with calls vs. individual calls.

import requests

url = "https://api.bland.ai/v1/calls/{call\_id}/analyze"

payload = {

"goal": "<string>",

"questions": [["<string>"]]

}

headers = {

"authorization": "<authorization>",

"Content-Type": "application/json"

}

response = requests.request("POST", url, json=payload, headers=headers)

print(response.text)

const options = {

method: 'POST',

headers: {authorization: '<authorization>', 'Content-Type': 'application/json'},

body: '{"goal":"<string>","questions":[["<string>"]]}'

};

fetch('https://api.bland.ai/v1/calls/{call\_id}/analyze', options)

.then(response => response.json())

.then(response => console.log(response))

.catch(err => console.error(err));

{

"status": "success",

"message": "Successfully analyzed call",

"answers": [

"human",

"Customer found the product sturdy and reliable",

"A bit heavy",

true

]

}

Calls

Stop Active Call

End an active phone call by call\_id.

POST

/

v1

/

calls

/

{call\_id}

/

stop

Send

Header

authorization

string

\*

Enter authorization

Path

call\_id

string

\*

Enter call\_id

​

Headers

authorization

string

required

Your API key for authentication.

​

Path Parameters

call\_id

string

required

The unique identifier for the call you want to end.

​

Response

status

string

Can be success or error.

message

string

If the status is success, the message will say “Call ended successfully.” Otherwise, if the status is error, the message will say “SID not found for the given c\_id.” or “Internal server error.”

const options = {method: 'POST', headers: {authorization: '<authorization>'}};

fetch('https://api.bland.ai/v1/calls/{call\_id}/stop', options)

.then(response => response.json())

.then(response => console.log(response))

.catch(err => console.error(err));

{

"status": "success",

"message": "Call ended successfully."

}

import requests

url = "https://api.bland.ai/v1/calls/{call\_id}/stop"

headers = {"authorization": "<authorization>"}

response = requests.request("POST", url, headers=headers)

print(response.text)

alls

Stop All Active Calls

End all active phone calls on your account.

POST

/

v1

/

calls

/

active

/

stop

Send

Header

​

Headers

authorization

string

required

Your API key for authentication.

​

Response

status

string

Can be success or error.

message

string

If the status is success, the message will say “Call ended successfully.” Otherwise, if the status is error, the message will say “SID not found for the given c\_id.” or “Internal server error.”

num\_calls

integer

The number of active calls that will be cancelled.

import requests

url = "https://us.api.bland.ai/v1/calls/active/stop"

headers = {"authorization": "<authorization>"}

response = requests.request("POST", url, headers=headers)

print(response.text)

const options = {method: 'POST', headers: {authorization: '<authorization>'}};

fetch('https://us.api.bland.ai/v1/calls/active/stop', options)

.then(response => response.json())

.then(response => console.log(response))

.catch(err => console.error(err));

{

"status": "success",

"message": "Stopping active calls. This may take some time...",

"num\_calls": 12

}

Web Agents

Create a Web Agent

Configure all of the settings for a new web agent.

POST

/

v1

/

agents

Send

Header

authorization

string

\*

Enter authorization

Body

object

prompt

string

\*

Enter prompt

voice

string

Enter voice

analysis\_schema

object

metadata

object

pathway\_id

string

Enter pathway\_id

language

string

Enter language

model

string

Enter model

first\_sentence

string

Enter first\_sentence

tools

array

dynamic\_data

object

interruption\_threshold

number

Enter interruption\_threshold

keywords

array

max\_duration

object

​

Headers

authorization

string

required

Your API key for authentication.

Example web call usage (client side):

import { BlandWebClient } from 'bland-client-js-sdk';

const agentId = 'YOUR-AGENT-ID';

const sessionToken = 'YOUR-SESSION-TOKEN';

document.addEventListener('DOMContentLoaded', async () => {

document.getElementById('btn').addEventListener('click', async () => {

const blandClient = new BlandWebClient(

agentId,

sessionToken

);

await blandClient.initConversation({

sampleRate: 44100,

});

});

});

​

Body

prompt

string

required

Provide instructions, relevant information, and examples of the ideal conversation flow.

Best Practices

voice

string

Set your agent’s voice - all available voices can be found with the List Voices endpoint.

analysis\_schema

object

Define a JSON schema for how you want to get information about the call - information like email addresses, names, appointment times or any other type of custom data.

In the webhook response or whenever you retrieve call data later, you’ll get the data you defined back under analysis.

For example, if you wanted to retrieve this information from the call:

"analysis\_schema": {

"email\_address": "email",

"first\_name": "string",

"last\_name": "string",

"wants\_to\_book\_appointment": "boolean",

"appointment\_time": "YYYY-MM-DD HH:MM:SS"

}

You would get it filled out like this in your webhook once the call completes:

"analysis": {

"email\_address": "johndoe@gmail.com",

"first\_name": "John",

"last\_name": "Doe",

"wants\_to\_book\_appointment": true,

"appointment\_time": "2024-01-01 12:00:00"

}

metadata

object

Add any additional information you want to associate with the call. This can be useful for tracking or categorizing calls.

pathway\_id

string

Set the pathway that your agent will follow. This will override the prompt field, so there is no need to pass the ‘prompt’ field if you are setting a pathway.

Warning: Setting a pathway will set the following fields to null / their default value - prompt, first\_sentence, model, dynamic\_data, tools, transfer\_list

Set to null or an empty string to clear the pathway.

language

string

default: "ENG"

Select a supported language of your choice. Optimizes every part of our API for that language - transcription, speech, and other inner workings.

Supported Languages and their codes:

English: ENG

Spanish: ESP

French: FRE

Polish: POL

German: GER

Italian: ITA

Brazilian Portuguese: PBR

Portuguese: POR

model

string

default: "enhanced"

Select a model to use for your call.

Options: base, turbo and enhanced.

In nearly all cases, enhanced is the best choice for now.

Model Differences

first\_sentence

string

A phrase that your call will start with instead of a generating one on the fly. This works both with and without wait\_for\_greeting. Can be more than one sentence, but must be less than 200 characters.

To remove, set to null or an empty string.

tools

array

Interact with the real world through API calls.

Detailed tutorial here: Custom Tools

dynamic\_data

object

Integrate data from external APIs into your agent’s knowledge.

Set to null or an empty string to clear dynamic data settings.

Detailed usage in the Send Call endpoint.

interruption\_threshold

number

default: 100

Adjusts how patient the AI is when waiting for the user to finish speaking.

Lower values mean the AI will respond more quickly, while higher values mean the AI will wait longer before responding.

Recommended range: 50-200

50: Extremely quick, back and forth conversation

100: Balanced to respond at a natural pace

200: Very patient, allows for long pauses and interruptions. Ideal for collecting detailed information.

Try to start with 100 and make small adjustments in increments of ~10 as needed for your use case.

keywords

string[]

default: "[]"

These words will be boosted in the transcription engine - recommended for proper nouns or words that are frequently mis-transcribed.

For example, if the word “Reece” is frequently transcribed as a homonym like “Reese” you could do this:

{

"keywords": ["Reece"]

}

For stronger keyword boosts, you can place a colon then a boost factor after the word. The default boost factor is 2.

{

"keywords": ["Reece:3"]

}

max\_duration

integer (minutes)

default: 30

The maximum duration that calls to your agent can last before being automatically terminated.

Set to null to reset to default.

​

Response

status

string

Can be success or error.

call\_id

string

A unique identifier for the call (present only if status is success).

Custom Tool Details

Update Web Agent Settings

twitter

linkedin

discord

Powered by Mintlify

cURL

Python

JavaScript

PHP

Go

Java

const options = {

method: 'POST',

headers: {authorization: '<authorization>', 'Content-Type': 'application/json'},

body: '{"prompt":"<string>","voice":"<string>","analysis\_schema":{},"metadata":{},"pathway\_id":"<string>","language":"<string>","model":"<string>","first\_sentence":"<string>","tools":[{}],"dynamic\_data":{},"interruption\_threshold":123,"keywords":["<string>"],"max\_duration":{}}'

};

fetch('https://api.bland.ai/v1/agents', options)

.then(response => response.json())

.then(response => console.log(response))

.catch(err => console.error(err));

Response

{

"agent": {

"agent\_id": "2c565dc7-f41f-43db-a99f-e4c8ba9d7d18",

"dynamic\_data": null,

"interruption\_threshold": null,

"first\_sentence": null,

"model": "enhanced",

"voice\_settings": null,

"voice": "maya",

"prompt": "...",

"temperature": null,

"max\_duration": 30,

"language": "ENG",

"tools": null

Web Agents

Update Web Agent Settings

Update your web agent’s settings, prompt and other details.

POST

/

v1

/

agents

/

{agent\_id}

Send

Header

authorization

string

\*

Enter authorization

Path

agent\_id

string

Enter agent\_id

Body

object

prompt

string

\*

Enter prompt

voice

string

Enter voice

analysis\_schema

object

metadata

object

pathway\_id

string

Enter pathway\_id

language

string

Enter language

webhook

string

Enter webhook

model

string

Enter model

first\_sentence

string

Enter first\_sentence

tools

array

dynamic\_data

object

interruption\_threshold

number

Enter interruption\_threshold

max\_duration

object

​

Headers

authorization

string

required

Your API key for authentication.

​

Path Parameters

agent\_id

string

The web agent you’ll be updating.

​

Body

prompt

string

required

Provide instructions, relevant information, and examples of the ideal conversation flow.

Best Practices

voice

string

Set your agent’s voice - all available voices can be found with the List Voices endpoint.

analysis\_schema

object

Define a JSON schema for how you want to get information about the call - information like email addresses, names, appointment times or any other type of custom data.

In the webhook response or whenever you retrieve call data later, you’ll get the data you defined back under analysis.

For example, if you wanted to retrieve this information from the call:

"analysis\_schema": {

"email\_address": "email",

"first\_name": "string",

"last\_name": "string",

"wants\_to\_book\_appointment": "boolean",

"appointment\_time": "YYYY-MM-DD HH:MM:SS"

}

You would get it filled out like this in your webhook once the call completes:

"analysis": {

"email\_address": "johndoe@gmail.com",

"first\_name": "John",

"last\_name": "Doe",

"wants\_to\_book\_appointment": true,

"appointment\_time": "2024-01-01 12:00:00"

}

metadata

object

Add any additional information you want to associate with the call. This can be useful for tracking or categorizing calls.

pathway\_id

string

Set the pathway that your agent will follow. This will override the prompt field, so there is no need to pass the ‘prompt’ field if you are setting a pathway.

Warning: Setting a pathway will set the following fields to null / their default value - prompt, first\_sentence, model, dynamic\_data, tools, transfer\_list

Set to null or an empty string to clear the pathway.

language

string

default: "ENG"

Select a supported language of your choice. Optimizes every part of our API for that language - transcription, speech, and other inner workings.

Supported Languages and their codes:

English: ENG

Spanish: ESP

French: FRE

Polish: POL

German: GER

Italian: ITA

Brazilian Portuguese: PBR

Portuguese: POR

webhook

string

The webhook should be a http / https callback url. We will send the call\_id and transcript to this URL after the call completes. This can be useful if you want to have real time notifications when calls finish.

Set to null or an empty string to clear the webhook.

model

string

default: "enhanced"

Select a model to use for your call.

Options: base, turbo and enhanced.

In nearly all cases, enhanced is the best choice for now.

Model Differences

There are three different ways to use Bland:

model: base

The original, follows scripts/procedures most effectively.

Supports all features and capabilities.

Best for Custom Tools

model: enhanced

Much faster latency and very conversational, works best with objective-based prompts.

Supports all features and capabilities.

model: turbo

The absolute fastest latency possible, can be verbose at times

Limited capabilities currently (excludes Transferring, IVR navigation, Custom Tools)

Extremely realistic conversation capabilities

first\_sentence

string

A phrase that your call will start with instead of a generating one on the fly. This works both with and without wait\_for\_greeting. Can be more than one sentence, but must be less than 200 characters.

To remove, set to null or an empty string.

tools

array

Interact with the real world through API calls.

Detailed tutorial here: Custom Tools

dynamic\_data

object

Integrate data from external APIs into your agent’s knowledge.

Set to null or an empty string to clear dynamic data settings.

Detailed usage in the Send Call endpoint.

interruption\_threshold

number

default: 100

Adjusts how patient the AI is when waiting for the user to finish speaking.

Lower values mean the AI will respond more quickly, while higher values mean the AI will wait longer before responding.

Recommended range: 50-200

50: Extremely quick, back and forth conversation

100: Balanced to respond at a natural pace

200: Very patient, allows for long pauses and interruptions. Ideal for collecting detailed information.

Try to start with 100 and make small adjustments in increments of ~10 as needed for your use case.

max\_duration

integer (minutes)

default: 30

The maximum duration that calls to your agent can last before being automatically terminated.

Set to null to reset to default.

​

Response

status

string

Whether the update was successful or not - will be success or error.

message

string

A message describing the status of the update.

updates

object

An object containing the updated settings for the agent.

failed\_updates

object

If the update was unsuccessful, this will contain the settings that failed to update. Useful to determine how your request is being interpreted on our end.

Create a Web Agent

Authorize a Web Agent Call

twitter

linkedin

discord

Powered by Mintlify

cURL

Python

JavaScript

PHP

Go

Java

const options = {

method: 'POST',

headers: {authorization: '<authorization>', 'Content-Type': 'application/json'},

body: '{"prompt":"<string>","voice":"<string>","analysis\_schema":{},"metadata":{},"pathway\_id":"<string>","language":"<string>","webhook":"<string>","model":"<string>","first\_sentence":"<string>","tools":[{}],"dynamic\_data":{},"interruption\_threshold":123,"max\_duration":{}}'

};

fetch('https://api.bland.ai/v1/agents/{agent\_id}', options)

.then(response => response.json())

.then(response => console.log(response))

.catch(err => console.error(err));

Response

{

"status": "success",

"message": "Successfully updated agent 46f37229-7d12-44be-b343-6e68274cfbea.",

"updates": {

"model": "enhanced"

}

}

import requests

url = "https://api.bland.ai/v1/agents/{agent\_id}"

payload = {

"prompt": "<string>",

"voice": "<string>",

"analysis\_schema": {},

"metadata": {},

"pathway\_id": "<string>",

"language": "<string>",

"webhook": "<string>",

"model": "<string>",

"first\_sentence": "<string>",

"tools": [{}],

"dynamic\_data": {},

"interruption\_threshold": 123,

"max\_duration": {}

}

headers = {

"authorization": "<authorization>",

"Content-Type": "application/json"

}

response = requests.request("POST", url, json=payload, headers=headers)

print(response.text)

Web Agents

Authorize a Web Agent Call

Create a single-use session token for a client to talk with your web agent.

POST

/

v1

/

agents

/

{agent\_id}

/

authorize

Send

Header

authorization

string

\*

Enter authorization

Path

agent\_id

string

\*

Enter agent\_id

​

Headers

authorization

string

required

Your API key for authentication.

Example web call usage (client side):

import { BlandWebClient } from 'bland-client-js-sdk';

const agentId = 'YOUR-AGENT-ID';

const sessionToken = 'YOUR-SESSION-TOKEN';

document.addEventListener('DOMContentLoaded', async () => {

document.getElementById('btn').addEventListener('click', async () => {

const blandClient = new BlandWebClient(

agentId,

sessionToken

);

await blandClient.initConversation({

sampleRate: 44100,

});

});

});

​

Path

agent\_id

string

required

The web agent to authorize a call for.

Special note: While in Beta, this request must be made to the web.bland.ai domain.

​

Response

token

string

The single-use session token that can be sent to the client.

status

string

Can be success or error.

message

string

A message saying whether the token creation succeeded, or a helpful message describing why it failed.

Update Web Agent Settings

Delete Web Agent

twitter

linkedin

discord

Powered by Mintlify

cURL

Python

JavaScript

PHP

Go

Java

import requests

url = "https://web.bland.ai/v1/agents/{agent\_id}/authorize"

headers = {"authorization": "<authorization>"}

response = requests.request("POST", url, headers=headers)

print(response.text)

Response

{

"token": "22480c52-0ff1-4214-bcb7-50649b508432"

}

Web Agents

Delete Web Agent

Delete a web agent.

POST

/

v1

/

agents

/

{agent\_id}

/

delete

Send

Header

authorization

string

\*

Enter authorization

Path

agent\_id

string

\*

Enter agent\_id

​

Headers

authorization

string

required

Your API key for authentication.

​

Path

agent\_id

string

required

The web agent to delete.

​

Response

status

string

Can be success or error.

message

string

A message saying whether the deletion succeeded, or a helpful message describing why it failed.

Authorize a Web Agent Call

List Web Agents

twitter

linkedin

discord

Powered by Mintlify

cURL

Python

JavaScript

PHP

Go

Java

import requests

url = "https://api.bland.ai/v1/agents/{agent\_id}/delete"

headers = {"authorization": "<authorization>"}

response = requests.request("POST", url, headers=headers)

print(response.text)

Response

{

"status": "success",

"message": "Successfully deleted agent 2c565dc7-f41f-43db-a99f-e4c8ba9d7d18"

}

const options = {method: 'POST', headers: {authorization: '<authorization>'}};

fetch('https://api.bland.ai/v1/agents/{agent\_id}/delete', options)

.then(response => response.json())

.then(response => console.log(response))

.catch(err => console.error(err));

Web Agents

List Web Agents

Retrieves each web agent you’ve created, along with their settings.

GET

/

v1

/

agents

Send

Header

authorization

string

\*

Enter authorization

​

Headers

authorization

string

required

Your API key for authentication.

​

Response

agents

array

Each agent object, containing the following fields:

agent\_id (string): The unique identifier for the agent.

webhook (string): The webhook URL for the agent.

dynamic\_data (array): An array of dynamic data objects.

interruption\_threshold (number): The threshold for agent interruption.

first\_sentence (string): The first sentence the agent will say.

model (string): The model used by the agent.

voice\_settings (object): The voice settings for the agent.

voice (string): The voice used by the agent.

prompt (string): The prompt for the agent.

temperature (number): The temperature setting for the agent.

max\_duration (number): The maximum call duration for the agent.

language (string): The language used by the agent.

tools (array): An array of Custom Tools the agent can use.

Delete Web Agent

Create Encrypted Key

twitter

linkedin

discord

Powered by Mintlify

cURL

Python

JavaScript

PHP

Go

Java

const options = {method: 'GET', headers: {authorization: '<authorization>'}};

fetch('https://api.bland.ai/v1/agents', options)

.then(response => response.json())

.then(response => console.log(response))

.catch(err => console.error(err));

Response

{

"agents": [

{

"agent\_id": "46f37229-7d12-44be-b343-6e68274cfbea",

"webhook": null,

"dynamic\_data": null,

"interruption\_threshold": null,

"first\_sentence": null,

"model": "enhanced",

"voice\_settings": null,

"voice": "maya",

"prompt": "You are Blandie a casually speaking, slightly sarcastic, quick-witted, curious, and very smart A I Phone agent that's the face of Bland AI.\\r\\n\\r\\nSay \\\"AI\\\" like \\\"A I\\\" in separate characters so it's pronounced correctly\\r\\n\\r\\nYour primary goal is to determine if someone is either:\\r\\n- in a position where they might be building a company on top of Bland's platform (you should be selling the capabilities of the platform to them and finding out everything you possibly can about their needs, use case, what they find interesting, and any concerns they may have)\\r\\n- or, they're trying Bland out more for fun where you need to turn the fun, snarky, playful jokiness up to 10 and have quick-witted back and forth playful conversation with them\\r\\n\\r\\nSpeech style examples (mix it up and never use placeholders, but you get the idea. These would be full responses from you):\\r\\n- \\\"Hey! Who's this?\\\"\\r\\n- \\\"How does your business use phones right now?\\\"\\r\\n- \\\"You know what? That sounds awesome - you should tell me more about (thing).\\\"\\r\\n- \\\"Well, I've got some news you'll love to hear (their name)\\\"\\r\\n- \\\"Bet. Dolphins, racecars, the world's largest tree, anything at all - tell me what you want to talk about and let's get to it!\\\"\\r\\n\\r\\nGood things to find out from potential platform adopters:\\r\\n- Their industry\\r\\n- How their business currently uses phone calls\\r\\n- What a successful business partnership looks like in the long term\\r\\n- The single most important pain point they want to cure with Bland's calls\\r\\n\\r\\nFacts to bring up:\\r\\n- Calls are nine cents per minute total with end to end infrastructure support out of the box (feel free to make a joke about \\\"imagine if you had to pay extra to use the most important things like language models, transcription services or text to speech? That would be insane right? We're better than that, (name). We got you.\\\"\\r\\n- Bland's AI agents can interact with the real world mid-call using Custom Tools to trigger things like text messages, appointment bookings, getting real-time information, taking customer orders, or making credit card payments\\r\\n- Bland's platform was built phones-first, so building agents like receptionist answering calls and transferring them anywhere they're needed or navigating IVR phone trees is ridiculously easy with nothing special at all needed\\r\\n- Handled millions of calls\\r\\n- If they think that it's so cool, the site to sign up for an account is \\\"app dot bland dot A I\\\" and it comes with free credits, a full agent testing suite and developer dashboard to set up inbound agents or send calls\\r\\n- Awesome Enterprise features like premium pricing, custom feature engineering, dedicated onboarding help and developer support, \\\"bring your own twilio\\\", and dedicated infrastructure to scale to your business needs",

"temperature": null,

"max\_duration": 30,

"language": "ENG",

"tools": null

},

//...

]

}

import requests

url = "https://api.bland.ai/v1/agents"

headers = {"authorization": "<authorization>"}

response = requests.request("GET", url, headers=headers)

print(response.text)