

GET STARTED

Setup

Quickstart

Models

Pricing

Go-live checklist

API DETAILS

SDKs

Inputs

Outputs

Content moderation

USAGE & BILLING

SAMPLE APPS

Web app: Hair makeover

Chrome extension: Generate video from any image

Chrome extension: Virtual try on

Figma plugin: Image and Video Generator

ERRORS

HTTP Errors

Task failures

Troubleshooting

API VERSIONS

Overview

Version 2024-11-06

Troubleshooting

Troubleshooting assets

When submitting an asset, the API may return a `400 Bad Request` response if the asset cannot be used. If you're using SDKs, a `BadRequestError` will be [thrown in Node](#) or [raised in Python](#).

The message in the error response body will include two pieces of pertinent information: the field that the error occurred for (e.g., `promptImage`) and the reason for the failure.

Common error reasons

Invalid data URI.

The provided data URI is malformed and could not be parsed. Be sure you're using a library to encode the URI.

Unsupported asset type. Data URIs must include the content type of the value they encode.

Your data URI specifies a media type that's not supported. See the list of [supported media types](#).

Invalid URL

You provided a URL that is non-standard and cannot be parsed.

Only HTTPS URLs are allowed.

All URLs must start with `https://`. You cannot use `http://` or other schemes, like `ftp://`.

URLs must be hosted on a domain.

You cannot provide a URL that points to an IP address. For instance, `https://11.22.33.44/foo/bar` would be rejected. You can instead create an A or AAAA record for your domain that points at the IP address of your host (in the example here, an A record pointing to `11.22.33.44`). Be sure to set up HTTPS on the host for that record.

Failed to fetch asset. The URL may be incorrect or the server hosting the asset may be down.

When we attempted to fetch the asset, we encountered a non-HTTP connection issue. This might be a DNS issue, TCP connection issue, TLS problem, protocol error, or an unexpectedly closed connection. Check that the URL is working and that connections are not being rejected.

Failed to fetch asset. Received HTTP response code "..."

When we attempted to fetch the asset, we did not get a 200 status code. The resposne code that we received is provided in the reason. Be aware that we do not follow redirects (via the `Location` HTTP response header).

Timeout while fetching asset.

It took longer than ten seconds to download the provided asset.

Assets must use an approved Content-Type response header. We received application/octet-stream, which is not allowed.

Your server returned `application/octet-stream` for the `Content-Type` HTTP response header. This is not allowed. See the list of [supported media types](#).

Unsupported Content-Type response header: "...".

Your server returned an unsupported value for the `Content-Type` HTTP response header, which is noted in the response. See the list of [supported media types](#).

Content-Length not provided

Your server did not specify a `Content-Length` HTTP response header. Lengths must be provided; we do not support streaming responses of unknown length.

Asset size exceeds XX.XMB.

Your server specified a `Content-Length` HTTP response header that exceeds the maximum size for the asset type. This error may also be returned if the number of bytes returned by the server does not match the number specified in the `Content-Length` response header. The maximum size is specified in the reason and in the [inputs](#) documentation.

Asset size exceeds XX.XMB.

Your server specified a `Content-Length` HTTP response header that exceeds the maximum size for the asset type. The maximum size is specified in the reason.

Invalid asset dimensions. Height and width must not exceed 8000px. Got XXxYY.

The provided asset is larger than 8000px on one of its sides. Assets must be less than 8000px on either side.

Invalid asset aspect ratio. width / height ratio must be between XX and YY. Got ZZ.

The aspect ratio (the asset width divided by the asset height) must be between the values XX and YY. The computed aspect ratio is included in the reason as ZZ.

Debugging failures

You can investigate the cause(s) for many common failures by simulating our request for your asset. To do this, we'll run a cURL command against the URL you'll specify for your asset. For this example, we'll use the asset URL `https://example.com/assets/image.jpg`.

```
Specify your asset URL
curl "https://example.com/assets/image.jpg" \
-I returns the HTTP response headers
-I \
Mimic the User-Agent that Runway uses
-H "User-Agent: RunwayML API/1.0"
```

You'll receive output that looks like this:

```
% curl "https://example.com/assets/image.jpg" \
-I \
-H "User-Agent: RunwayML API/1.0"
1. Check the status code
HTTP/2 200
2. Check the Content-Type
content-type: image/jpg
3. Check the Content-Length
content-length: 123456
vary: Accept-Encoding
cache-control: max-age=14400
accept-ranges: bytes
alt-svc: h3=":443"; ma=86400
```

1. Your server should be returning a `200` status code.
2. Be sure you're returning an acceptable `Content-Type`.
3. A `Content-Length` should be provided with an accurate file size.

On this page

Overview

Troubleshooting assets

Common error reasons

Debugging failures