GET STARTED
Setup
Quickstart
Models
Pricing
Go-live checklist
API DETAILS
USAGE & BILLING
SAMPLE APPS
ERRORS
API VERSIONS

Using the API

On this page
Overview
Talking to the API

Before starting, make sure you have followed the instructions in the setup page.

Talking to the API

Generating Video | Generating Images

In this example, we'll use the `gen4_turbo` model to generate a video from an image using the text prompt "Generate a video". You'll want to replace the `promptImage` with a URL of an image and a `promptText` with your own text prompt.

Node | Python | Just testing

First, you'll want to install the Runway SDK. You can do this with npm:

```
npm install --save @runwayml/sdk
```

In your code, you can now import the SDK and start making requests:

```
import RunwayML, { TaskFailedError } from '@runwayml/sdk';

const client = new RunwayML();

// Create a new image-to-video task using the "gen4_turbo" model
try {
  const task = await client.imageToVideo
    .create({
      model: 'gen4_turbo',
      // Point this at your own image file
      promptImage: 'https://example.com/image.jpg',
      promptText: 'Generate a video',
      ratio: '1280:720',
      duration: 5,
    })
    .waitForTaskOutput();

  console.log('Task complete:', task);
} catch (error) {
  if (error instanceof TaskFailedError) {
    console.error('The video failed to generate.');
    console.error(error.taskDetails);
  } else {
    console.error(error);
  }
}
```

Uploading base64 encoded images as data URIs

You can also upload base64 encoded images (as a data URI) instead of pointing to an external URL. This can be useful if you're working with a local image file and want to avoid an extra network round trip to upload the image.

To do this, simply pass the base64 encoded image string as a data URI in the `promptImage` field instead of a URL. For more information about file types and size limits, see the Inputs page.

Node | Python

```
import fs from 'node:fs';
import RunwayML, { TaskFailedError } from '@runwayml/sdk';

const client = new RunwayML();

// Read the image file into a Buffer. Replace `example.png` with your own image path.
const imageBuffer = fs.readFileSync('example.png');

// Convert to a data URI. We're using `image/png` here because the input is a PNG.
const dataUri = `${imageBuffer.toString('base64')}`;

// Create a new image-to-video task using the "gen4_turbo" model
try {
  const imageToVideo = await client.imageToVideo
    .create({
      model: 'gen4_turbo',
      // Point this at your own image file
      promptImage: dataUri,
      promptText: 'Generate a video',
      ratio: '1280:720',
      duration: 5,
    })
    .waitForTaskOutput();

  console.log('Task complete:', task);
} catch (error) {
  if (error instanceof TaskFailedError) {
    console.error('The video failed to generate.');
    console.error(error.taskDetails);
  } else {
    console.error(error);
  }
}
```