

Built-in AI Early Preview Program

The Summarization API

Authors

Kenji Baheux

ContactSee [this section](#)**Last-updated**

Aug 29, 2024

See [changelog](#).

Intro

Thanks for participating in our Early Preview Program for built-in AI capabilities ([article](#), [talk at Google I/O 2024](#)). As always we are [eager to hear your feedback](#) about this program and our APIs.



Know of other folks who would love to join this program? Or perhaps you got access to this document from a friend?

Sign up to get the latest updates directly in your inbox.

In this update, we are thrilled to introduce the summarization API to our Early Preview Program participants. This initial iteration represents our first step towards giving web developers access to summarization capabilities that run on-device. We believe that your early feedback will be invaluable in shaping the future of the API and ensuring it meets the needs of both developers and users.

Summarization API

Purpose

The summarization API is provided for local experimentation. It lets you use [Gemini Nano](#) to condense long-form content or large amounts of content, thereby making information more accessible and useful for your users.

Early Preview Goals

The goals for this early preview are to hear your feedback on the following aspects:

1. The [quality of the summaries](#), via [this feedback channel](#).
2. **Issues with Chrome's current implementation**, via [this feedback channel](#).
3. The [eventual shape of the API](#), via [this feedback channel](#).

Availability


The summarization API is available, behind an experimental flag, from Chrome 129+ on desktop platforms.

- You'll need Version 129.0.6639.0 or above
- We recommend using [Chrome Canary](#).

Requirements

Our Built-in AI program is **currently focused on desktop platforms**. In addition, the following conditions are required for Chrome to download and run Gemini Nano.

Aspect	Windows	MacOS	Linux
OS version	10, 11	≥ 13 (Ventura)	Not specified
Storage	At least 22 GB on the volume that contains your Chrome profile. <i>Note that the model requires a lot less storage, it's just for the sake of having an ample storage margin. If after the download the available storage space falls below 10 GB, the model will be deleted again.</i>		
GPU	Integrated GPU, or discrete GPU (e.g. video card).		
Video RAM	4 GB (minimum)		
Network connection	A non-metered connection		

	These are not necessarily the final requirements for Gemini Nano in Chrome.
	Not yet supported: <ul style="list-style-type: none">• Chrome for Android• Chrome for iOS• Chrome for ChromeOS

Setup

Prerequisites

1. Acknowledge [Google's Generative AI Prohibited Uses Policy](#).
2. Download Chrome [Canary channel](#), and [confirm that your version](#) is equal or newer than 129.0.6639.0.
3. Check that your device meets the [requirements](#).
 - Don't skip this step, in particular make sure that you have **at least 22 GB of free storage space**.
 - If after the download the available storage space falls below 10 GB, the model will be **deleted again**.
 - Note that this is the same Gemini Nano model used by [the Prompt API](#), so if you already have the Prompt API set up, you don't need to worry about storage space.

Enable Gemini Nano

 **Ignore this section if you already have [set up the Prompt API!](#)**

Follow these steps to enable Gemini Nano:

1. Open a new tab in Chrome, go to
`chrome://flags/#optimization-guide-on-device-model`
2. Select **Enabled BypassPerfRequirement**
 - This bypass performance checks which might get in the way of having Gemini Nano downloaded on your device.
3. Relaunch Chrome.

Enable the Summarization API

Follow these steps to enable the summarization API [flag](#) for local experimentation:

1. Open a new tab in Chrome, go to
`chrome://flags/#summarization-api-for-gemini-nano`
2. Select **Enabled**
3. Relaunch Chrome.

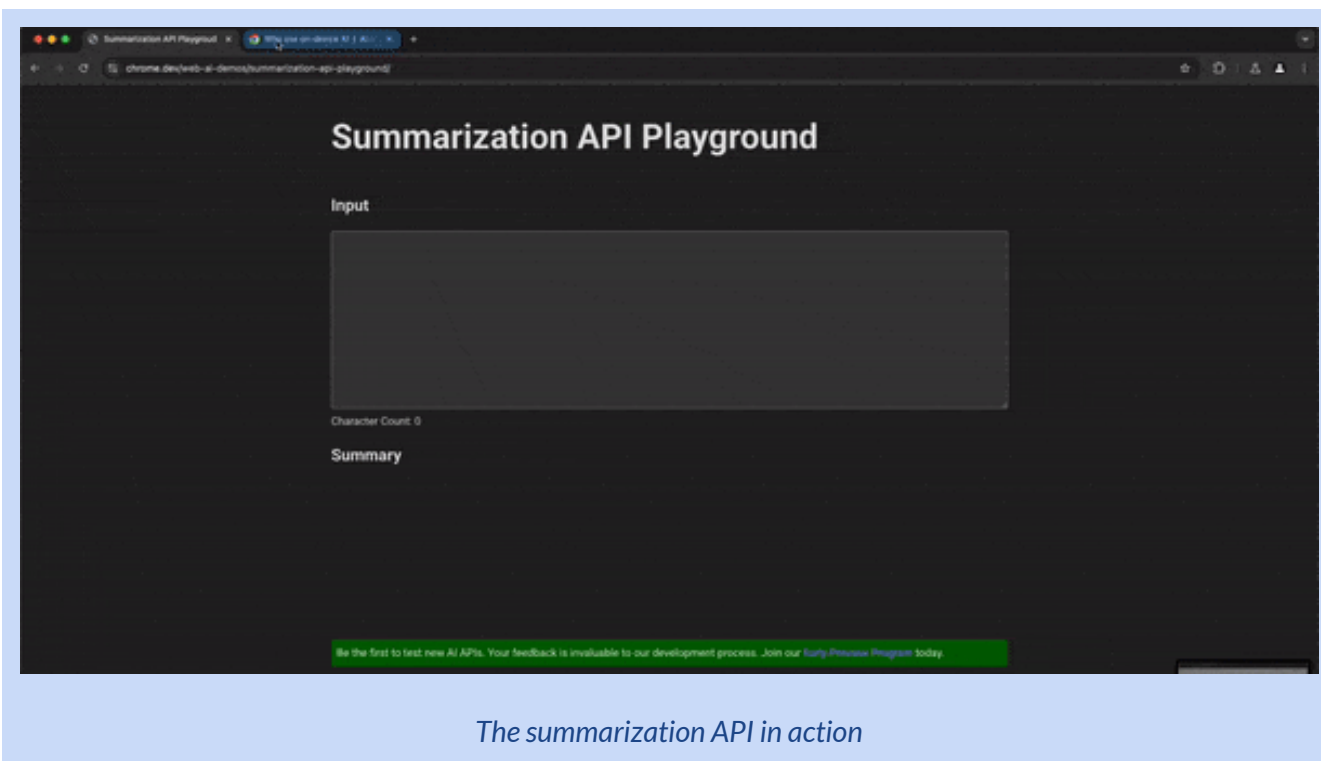
Finalize the overall set up

1. Open DevTools and send `await ai.summarizer.create();` in the console.
 - Don't worry if the call fails, as the point is to force Chrome to schedule a model download.
2. Now try `await ai.summarizer.capabilities();` in the console until the response changes to **“readily”**.
 - This may take about 3 to 5 minutes depending on your network connection, so let your Chrome instance run for a while.
 - If this still fails after waiting for a while:
 - If you see “The model was available but there was not an execution config available for the feature.” in the console, then you might need to wait for another day. This is because the config for this API is fetched once a day.
 - Last resort: please see the [troubleshooting section](#).

Demo

With the summarization API enabled, head over to this Chrome Dev Playground to try it out:

- <https://chrome.dev/web-ai-demos/summarization-api-playground/> ([code](#))




The summarization API in action

API overview

Sample code

Note: the current implementation isn't final and will evolve to support additional options and get closer to the design pattern described in the most recent [Prompt API explainer](#).

	Explainer, explained.
	An explainer is a document that describes a proposed web platform feature or collection of features. As work progresses, explainers facilitate discussion and, hopefully, consensus around the approach and feature design. Explainers are updated as design progresses.

Checking if the summarizer is available

```
JavaScript
const canSummarize = await ai.summarizer.capabilities();
let summarizer;
if (canSummarize && canSummarize.available !== 'no') {
```

```

if (canSummarize.available === 'readily') {
  // The summarizer can immediately be used.
  summarizer = await ai.summarizer.create();
} else {
  // The summarizer can be used after the model download.
  summarizer = await ai.summarizer.create();
  summarizer.addEventListener('downloadprogress', (e) => {
    console.log(e.loaded, e.total);
  });
  await summarizer.ready;
}
} else {
  // The summarizer can't be used at all.
}

```

Summarizing a piece of text

JavaScript

const someUserText = 'Hiroshi chuckled as he took a sip of his green tea. It was a typical Monday morning in the life of a Chrome engineer, but the project on his desk was far from ordinary. He was tasked with developing the "isTeapot?" API, a quirky new feature for web developers that would return a HTTP 418 "I\'m a teapot" status code if the requested resource was, in fact, a teapot. The day began with a flurry of code reviews and discussions with his team. They debated the finer points of the API\'s design, including whether to support different teapot types like "kyusu" or "tetsubin". Hiroshi argued for a more inclusive approach, allowing developers to specify any teapot-like object in the request headers. After a lively debate, they settled on a flexible design that allowed for custom teapot definitions. Hiroshi dove into the implementation, his fingers dancing across the keyboard as he crafted the code that would bring this peculiar API to life. He added a few Easter eggs for developers who might stumble upon the feature, including a hidden reference to the Hyper Text Coffee Pot Control Protocol. By lunchtime, Hiroshi had a working prototype. He tested it with a few sample requests, grinning as the "418 I\'m a teapot" response popped up on his screen. He imagined the amusement it would bring to web developers who discovered this hidden gem. As the afternoon progressed, Hiroshi fine-tuned the API, adding documentation and examples to help developers get started. He envisioned a future where websites would display playful teapot animations when the "isTeapot?" API was triggered, adding a touch of whimsy to the internet. As the day wound down, Hiroshi pushed his code to the repository, a sense of satisfaction washing over him. The "isTeapot?" API was a small, quirky feature, but it brought a smile to his

face. He knew that somewhere out there, a web developer was going to have a lot of fun with it.';

```
const result = await summarizer.summarize(someUserText);  
console.log(result);
```

```
// [TEMPORARY CAVEAT] No longer necessary in the latest Canary (as of  
08/29/2024)
```

```
// Destroy the summarizer to release resources
```

```
// summarizer.destroy();
```

```
// Create a new one before attempting to summarize other text
```

```
// summarizer = await ai.summarizer.create();
```

```
const someUserText2 = 'The gentle thud reverberated through the lander,  
followed by a soft silence. Emily unbuckled her harness, her heart pounding in  
her chest. A million miles away from home, she was the first human to set foot  
on Mars. She peered through the small window, the rust-colored landscape  
stretching out as far as the eye could see. A barren, rocky desert bathed in  
the pale sunlight filtering through the dusty atmosphere. It was breathtakingly  
beautiful. Taking a deep breath, she grabbed her helmet and put it on. A hiss  
signaled the pressurization, and her voice echoed in her ears as she spoke,  
"Emily to Guiana, Eos has landed." She stepped out of the lander, the fine  
Martian dust crunching under her boots. The low gravity made her feel lighter,  
each step a small leap. She planted the flag, the blue, white, and red tricolor  
fluttering in the gentle Martian breeze. Her visor displayed a stream of data:  
temperature -62°C, atmospheric pressure 6.5 mbar, radiation levels within  
acceptable limits. She marveled at the technology that allowed her to survive  
in this harsh environment. She began to explore, her boots leaving the first  
human footprints on the red planet. Every rock, every dune was a potential  
scientific treasure. She collected samples, documenting her findings with the  
helmet camera. As the sun began to set, painting the Martian sky in hues of  
orange and red, Emily made her way back to the lander. The first day on Mars  
was a success. She was filled with a sense of awe and wonder, realizing that  
she was part of something truly historic. The dream of Mars had finally become  
a reality.';
```

```
const result2 = await summarizer.summarize(someUserText2);  
console.log(result2);
```

```
// Destroy the summarizer to release resources
```

```
summarizer.destroy();
```

Caveats

Here are known temporary limitations:

- To summarize different texts, you will first need to destroy the current summarizer to release resources, and create a new one.
 - We'll address this inconvenience so that the same summarizer can be used to summarize different pieces of text without interference.
- Only English input and output are supported.
 - We intend to go beyond English content over time. If you haven't done so already, consider responding to [our second survey](#) to help us prioritize the next languages we focus on.
- No support of any options (e.g. length guidance, style, etc) for the time being.
- The context window is currently limited to 1024 tokens but we use about 26 of those under the hood.
 - Thanks to your feedback via [our second survey](#), we are exploring how to expand this feature to 4096 tokens, which should meet most developers' needs while maintaining performance and resource usage.
 - While we work on adding token counting and tracking to our APIs, you can estimate the number of tokens in English content by assuming that 1 token is roughly equal to 4 characters.

Appendix

General feedback

Quality or technical issues

If you experience quality or technical issues, consider [sharing details](#). Your reports will help us refine and improve our models, APIs, and components in the AI runtime layer, to ensure safety and responsible use.

- Handy shortlink: goo.gle/chrome-ai-dev-preview-feedback-quality

Feedback about Chrome's behavior / implementation of the API

If you want to report bugs or other issues related to Chrome's behavior / implementation of the API, provide as many details as possible (e.g. repro steps) in a [public chromium bug report](#).

Feedback about the API

If you want to report ergonomic issues or other problems related to the API itself, see if there is any related issue first and if not then file a public spec issue:

- [summarization API spec issues](#)

Other feedback

For other questions or issues, reach out directly by sending an email to [the mailing list owners](mailto:chrome-ai-dev-preview+owners@chromium.org) (chrome-ai-dev-preview+owners@chromium.org). We'll do our best to be as responsive as possible or update existing documents when more appropriate.

FAQ

Participation in the Early Preview Program

Opt-out and unsubscribe

To opt-out from the Early Preview Program, simply send an email to:

- chrome-ai-dev-preview+unsubscribe@chromium.org.

Opt-in

If you know someone who would like to join the program, ask them to fill out [this form](#) and that they communicate their eagerness to provide feedback when answering the last question of the survey!

Other updates

Links to all previous updates and surveys we've sent can be found in [The Context Index](#) also available via goo.gle/chrome-ai-dev-preview-index

Changelog

Date	Changes
Aug 8, 2024	<ul style="list-style-type: none">• First version.
Aug 20, 2024	<ul style="list-style-type: none">• Added note about model deletion if less than 10 GB of storage remain some time later after an initial successful model download.
Aug 29, 2024	<ul style="list-style-type: none">• Commented out the temporary caveat around the need to avoid re-using an existing summarizer to summarize different texts. This is no longer necessary in the latest Canary.• Added troubleshooting step in the setup section for "The model was available but there was not an execution config available for the feature."