

Built-in AI APIs for the Web

Thomas Steiner
tomac@google.com

Adding AI to your web app

Imagine you build an app to let users record and organize creative ideas:

ideas.example.com  .

Features:

- Record ideas by voice (e.g., musical ideas like a tune or a melody, so OS-native dictation isn't an option).
- Enter ideas by keyboard (e.g., a catchy headline for a blog post).
- Sketch ideas with a drawing (e.g., the dimensions of a piece of furniture).
- Grab ideas by photo (e.g., to recall the design of a fabric).

Adding AI to your web app (cont'd)

Possible AI use:

- *"List all my ideas where I hum."*
- *"Based on this headline, write an opening paragraph."*
- *"How much wood do I need given this sketch?"*
- *"Where can I buy a t-shirt with this design?"*
- *"Rewrite this rough idea more formally."*
- *"Share this idea translated to Japanese with my friend 英治."*
- *"Summarize all my ideas from yesterday afternoon."*

Adding AI to your web app (cont'd)

Option 1: Use an AI provider in the cloud 🌤️

- + Some of the most powerful models run (exclusively) in the cloud.
 - + The app can remain small and doesn't consume much disk space.
 - + No device hardware requirements, as all AI inference happens on the server.
-
- Can get expensive and make cost prediction unclear.
 - Requires user data to be sent to the cloud.
 - Doesn't work offline.

Adding AI to your web app (cont'd)

Option 2: Download AI models locally and run on the client 

- + Privacy-preserving, as all processing happens locally.
 - + Works offline once the initial model download has happened.
 - + Is free to the developer as all inference happens on the client.
-
- Model downloads can be prohibitively large, and require lots of disk space.
 - May not work on all hardware or be slow.
 - Limits what models can be used, no way to protect a model.

Adding AI to your web app (cont'd)

Option 2: Download AI models locally and run on the client

- + Privacy-preserving, as all processing happens locally
- + Works offline once the initial model download has happened
- + Is free to the developer as all inference happens on the client

shameless plug



Tuesday 3rd			
Time CEST	HTML Room (rooms 9 + 10)	JS Room (rooms 4 + 5)	CSS Room (rooms 1 + 2)
10:00-11:30	Web Platform: What are we working on?	TC39 proposals and integration with the web platform	Hackathon: Browser integration of Fontations Rust font stack
11:30-12:00	Break	Break	Break
12:00-13:30	Browser Funding After Search Deals	Cross-Origin Storage (COS)	Multimedia in WebKit
13:30-15:00	Lunch	Lunch	Lunch
15:00-16:30	Browser fingerprinting defense: review of the strategies used by different browsers	JavaScript subsets for custom embedding use-cases	tvOS as a new platform for Chromium
16:30-18:00	HTML-in-Canvas: Proposal and Progress	Gosub & Yavashark: A New Browser & JS/TS Engine	Chromium Embedding: Experiences and challenges

- Model downloads can be prohibitively large, and require lots of disk space.
- May not work on all hardware or be slow.
- Limits what models can be used, no way to protect a model.

Adding AI to your web app (cont'd)

Proposed option 3: Make a shared AI model available to Web APIs (built-in AI) 

- + Privacy-preserving, as all processing happens locally.
 - + Works offline once the initial model download has happened.
 - + Is free to the developer as all inference happens on the client.
 - ± Model downloads can be prohibitively large, but it's a one-time operation.
-
- May not work on all hardware or be slow.
 - Limits what models can be used, as you're bound to what the browser ships.

Built-in AI APIs



Web Machine Learning Community Group Charter

- This Charter: <https://webmachinelearning.github.io/charter/>
- Previous Charter: [a4da99c](#)
- Start Date: 2025-05-19
- Last Modified: [commits/master](#)

webmachinelearning.github.io/charter/#scope-of-work

Scope of Work

Task-specific APIs

A set of higher-level APIs that:


- Detect the language of the given input text;
- Translate input text to other languages;
- Produce textual summaries of input text;
- Produce new textual material given a writing task;
- Rephrase input text;
- Proofread input text.

Prompt API

A general-purpose API to prompt a language model directly using common prompting techniques to accomplish a variety of tasks, including:

- Classification, tagging, and keyword extraction of arbitrary text;
- Helping users compose text, such as blog posts, reviews, or biographies;
- Summarizing, e.g. of articles, user reviews, or chat logs;
- Generating titles or headlines from article contents;
- Answering questions based on the unstructured contents of a web page;
- Translation between languages.


Translator and Language Detector APIs



W3C Community Group
Draft Report

Translator and Language Detector APIs

Draft Community Group Report, 9 May 2025



▼ More details about this document

This version:
<https://webmachinelearning.github.io/translation-api>

Issue Tracking:
[GitHub](#)

Editor:
[Domenic Denicola](#) (Google) d@domenic.me

Copyright © 2025 the Contributors to the Translator and Language Detector APIs Specification, published by the [Web Machine Learning Community Group](#) under the [W3C Community Contributor License Agreement \(CLA\)](#). A human-readable [summary](#) is available.

TABLE OF CONTENTS

- 1 Introduction
- 2 Dependencies
- 3 The translator API
 - 3.1 Creation
 - 3.2 Availability
 - 3.3 The Translator class
 - 3.4 Translation
 - 3.4.1 The algorithm
 - 3.4.2 Usage
 - 3.4.3 Errors
 - 3.5 Permissions policy integration
- 4 The language detector API
 - 4.1 Creation
 - 4.2 Availability
 - 4.3 The LanguageDetector class
 - 4.4 Language detection
 - 4.4.1 The algorithm
 - 4.4.2 Usage

Abstract


The translator and language detector APIs gives web pages the ability to translate text between languages, and detect the language of such text.

Status of this document

This specification was published by the [Web Machine Learning Community Group](#). It is not a W3C Standard nor is it on the W3C Standards Track. Please note that under the [W3C Community Contributor License Agreement \(CLA\)](#) there is a limited opt-out and other conditions apply. Learn more about [W3C Community and Business Groups](#).

webmachinelearning.github.io/translation-api/


Writing Assistance APIs



W3C Community Group
Draft Report

Writing Assistance APIs

Draft Community Group Report, 22 May 2025



▼ More details about this document

This version:
<https://webmachinelearning.github.io/writing-assistance-apis>

Issue Tracking:
[GitHub](#)

Editor:
[Domenic Denicola](#) (Google) d@domenic.me

Copyright © 2025 the Contributors to the Writing Assistance APIs Specification, published by the [Web Machine Learning Community Group](#) under the [W3C Community Contributor License Agreement \(CLA\)](#). A human-readable [summary](#) is available.

Abstract

The summarizer, writer, and rewriter APIs provide high-level interfaces to call on a browser or operating system's built-in language model to help with writing tasks.

Status of this document

This specification was published by the [Web Machine Learning Community Group](#). It is not a W3C Standard nor is it on the W3C Standards Track. Please note that under the [W3C Community Contributor License Agreement \(CLA\)](#) there is a limited opt-out and other conditions apply. Learn more about [W3C Community and Business Groups](#).

TABLE OF CONTENTS

- 1 Introduction
- 2 The summarizer API
 - 2.1 Creation
 - 2.2 Availability
 - 2.3 The Summarizer class
 - 2.4 Summarization
 - 2.4.1 The algorithm
 - 2.4.2 Usage
 - 2.4.3 Options
 - 2.4.4 Errors
- 2.5 Permissions policy integration
- 3 The writer API
 - 3.1 Creation
 - 3.2 Availability
 - 3.3 The Writer class
 - 3.4 Writing
 - 3.4.1 The algorithm
 - 3.4.2 Usage
 - 3.4.3 Options

webmachinelearning.github.io/writing-assistance-apis/

Prompt API

The screenshot displays the GitHub interface for the repository `webmachinelearning / prompt-api`. The top navigation bar includes links for Code, Issues (26), Pull requests, Actions, Projects, Security, and Insights. A search bar is located on the right. The left sidebar shows the file structure with `main` as the selected branch. The main content area displays the `prompt-api / README.md` file. The README content includes a title "Explainer for the Prompt API", a disclaimer, a description of the project's purpose, and a list of tasks that can be accomplished using the API.

webmachinelearning / prompt-api

Type `/` to search

<> Code Issues 26 Pull requests Actions Projects Security Insights

Files

main

Go to file

CONTRIBUTING.md

LICENSE.md

README.md

security-privacy-questionnaire....

w3c.json

prompt-api / README.md

domenic Add a permissions policy and remove worker support 618d4e5 · last week History

Preview Code Blame 859 lines (619 loc) · 45.9 KB

Raw

Explainer for the Prompt API

This proposal is an early design sketch by the Chrome built-in AI team to describe the problem below and solicit feedback on the proposed solution. It has not been approved to ship in Chrome.

Browsers and operating systems are increasingly expected to gain access to a language model. ([Example](#), [example](#), [example](#).) Language models are known for their versatility. With enough creative [prompting](#), they can help accomplish tasks as diverse as:

- Classification, tagging, and keyword extraction of arbitrary text;
- Helping users compose text, such as blog posts, reviews, or biographies;
- Summarizing, e.g. of articles, user reviews, or chat logs;
- Generating titles or headlines from article contents
- Answering questions based on the unstructured contents of a web page
- Translation between languages
- Proofreading

github.com/webmachinelearning/prompt-api/

Adding built-in AI to your web app

Possible AI use:

- *"List all my ideas where I hum."* → **Prompt API with audio input**
- *"Based on this headline, write an opening paragraph."* → **Writer API**
- *"How much wood do I need given this sketch?"* → **Prompt API w/ image input**
- *"Where can I buy a t-shirt with this design?"* → **Prompt API w/ image input**
- *"Rewrite this rough idea more formally."* → **Rewriter API**
- *"Share this idea translated to Japanese with my friend 英治."* → **Translator API**
- *"Summarize all my ideas from yesterday afternoon."* → **Summarizer API**

Live demo of the APIs

Built-in AI APIs

Web Engines Hackfest rocks!



Thomas Steiner

Developer Relations Engineer on the **Google Chrome** team (tomac@google.com)

Biography

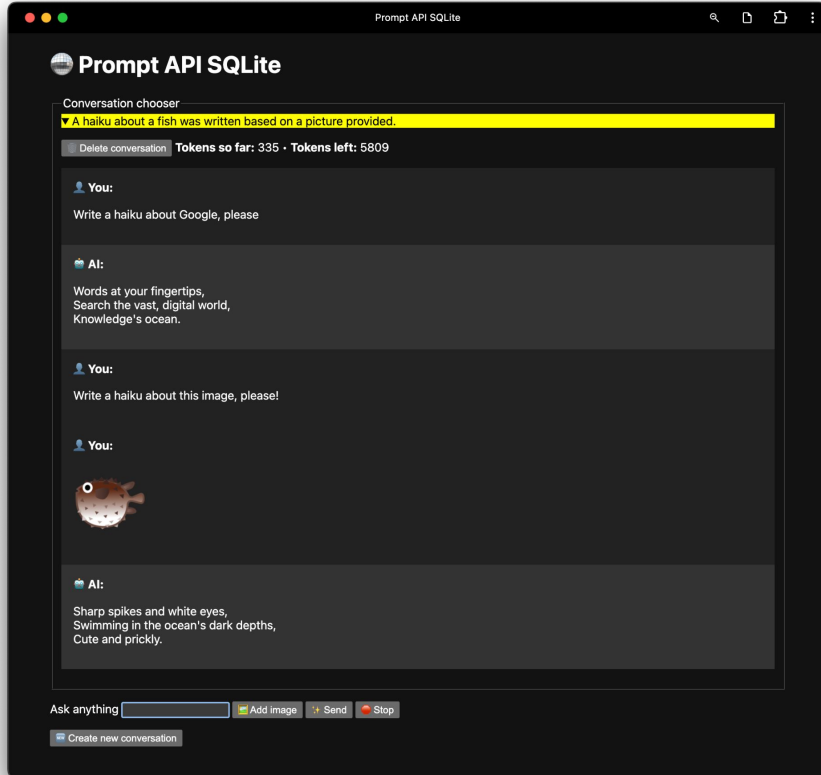
Thomas Steiner is a Developer Relations Engineer at Google, focused on Web AI, WebAssembly, and Project Fugu. He's an alumnus of University of Lyon (Postdoc), Polytechnic University of Barcelona (Ph. D.), and University of Karlsruhe (MA).

He blogs at blog.tomayac.com and posts as [@tomayac@toot.cafe](https://t.me/tomayac). On all other places on the Internet, chances are that he's there under his online alias *tomayac*.

Prompt:

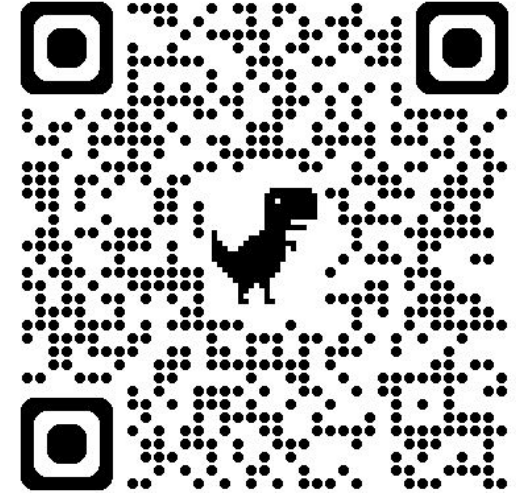
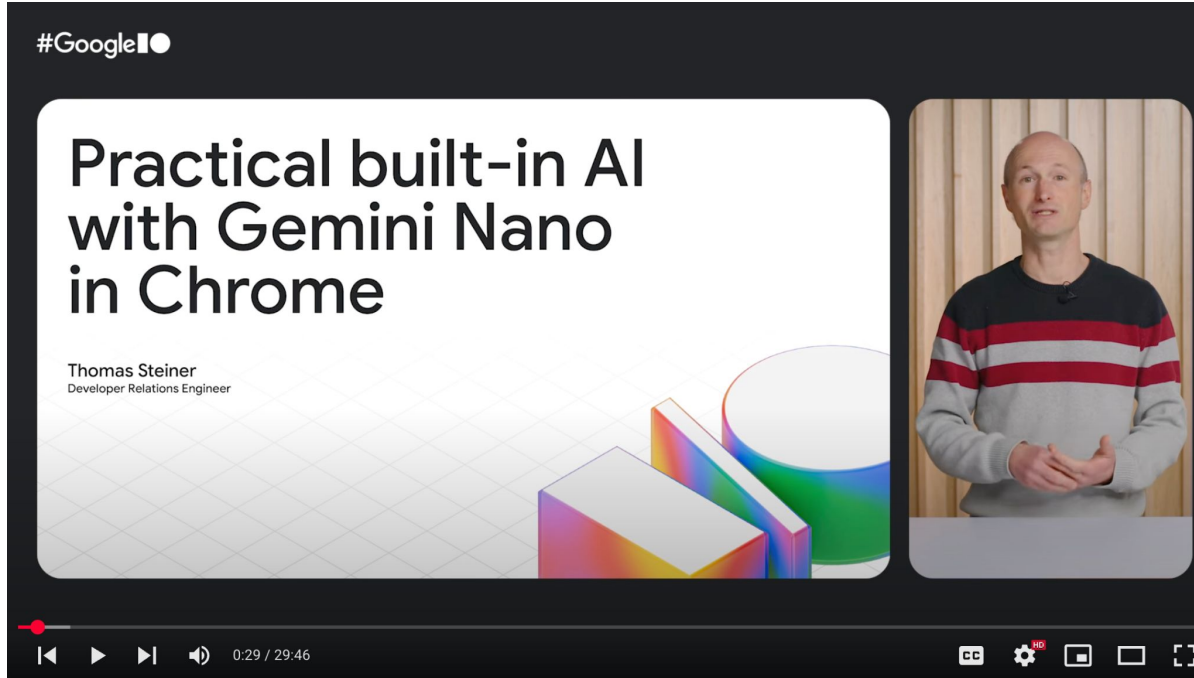
web-engines-hackfest-rocks.glitch.me/

Live demo of the APIs (cont'd)



tomayac.github.io/prompt-api-sqlite

Example partner implementations

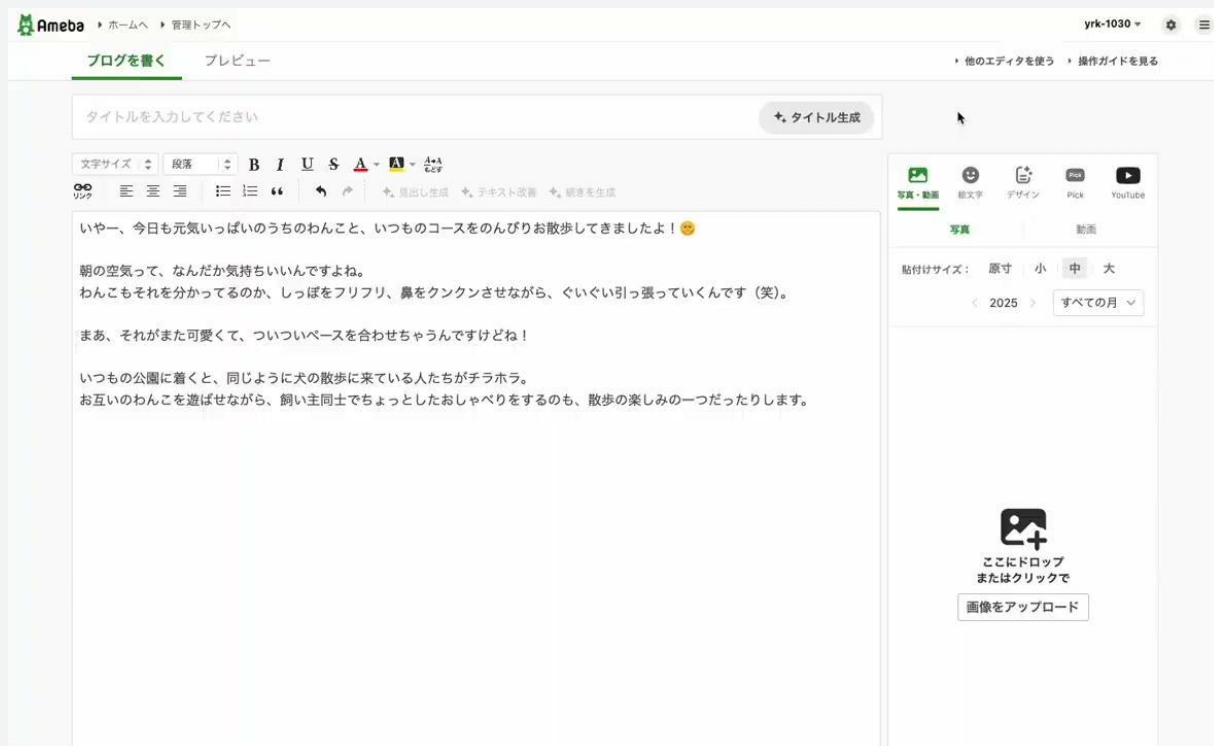


youtu.be/CjpZCWYrSxM



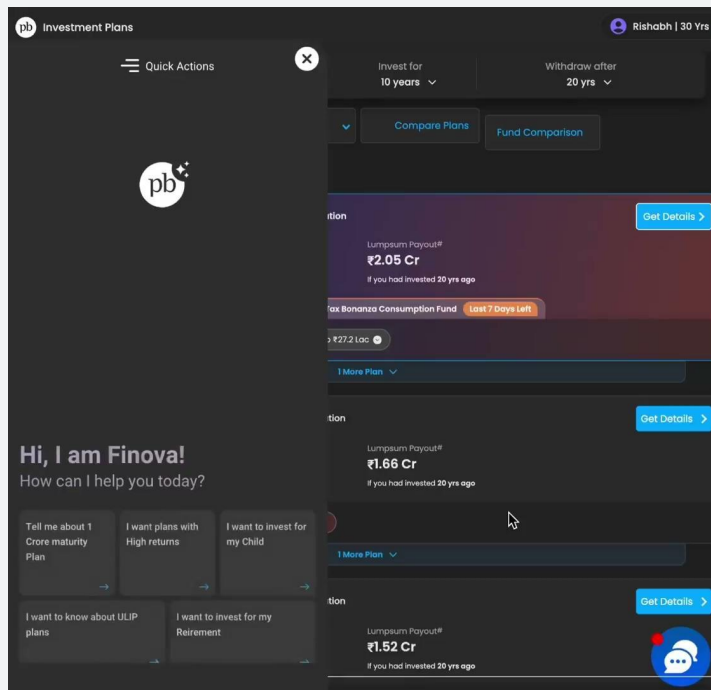
Blogger helper function
extension for blog writers
powered by the **Prompt API**
for **Chrome Extensions**.

- Generating blog titles and headings.
- Improving and editing sentences.
- Drafting subsequent paragraphs.



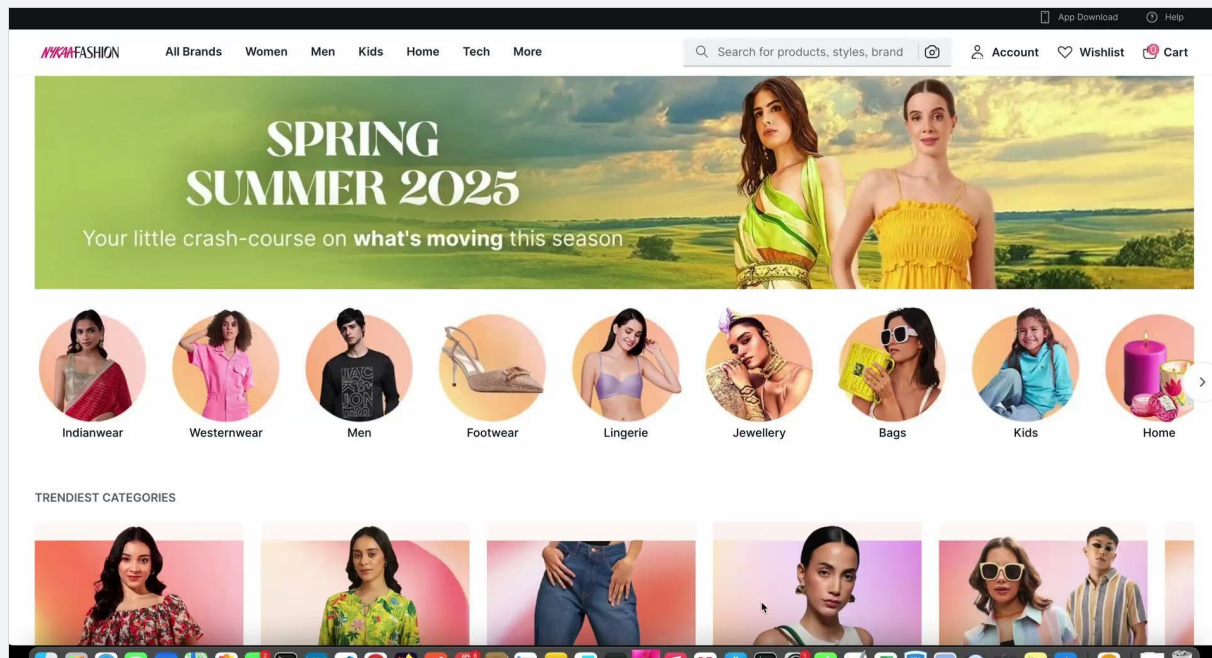


Policybazaar used the **Language Detector** API to detect the user's input language and the **Translator API** to translate messages to and from English. This enables efficient and accessible insurance assistance to users in their native language.



NYKAA FASHION

Nykaa, India's leading beauty and lifestyle destination, tested the **Prompt API with image input** to let users run visual search sessions to find fitting outfits.



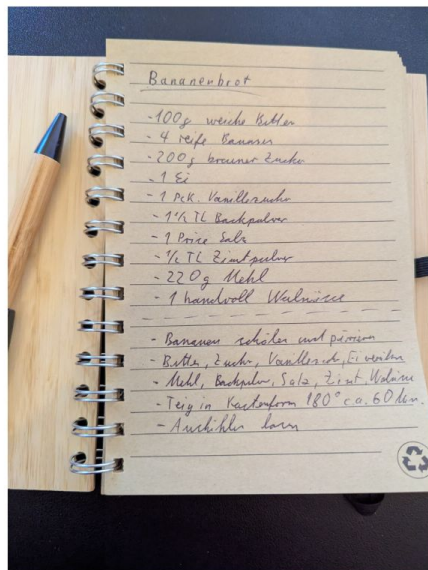


Chefkoch streamlined the recipe publishing process for their users by integrating the **Prompt API with image input**. The API functions as client-side OCR, converting handwritten recipes into structured JSON output that can then be edited on the Chefkoch site.



Menu

Digitize



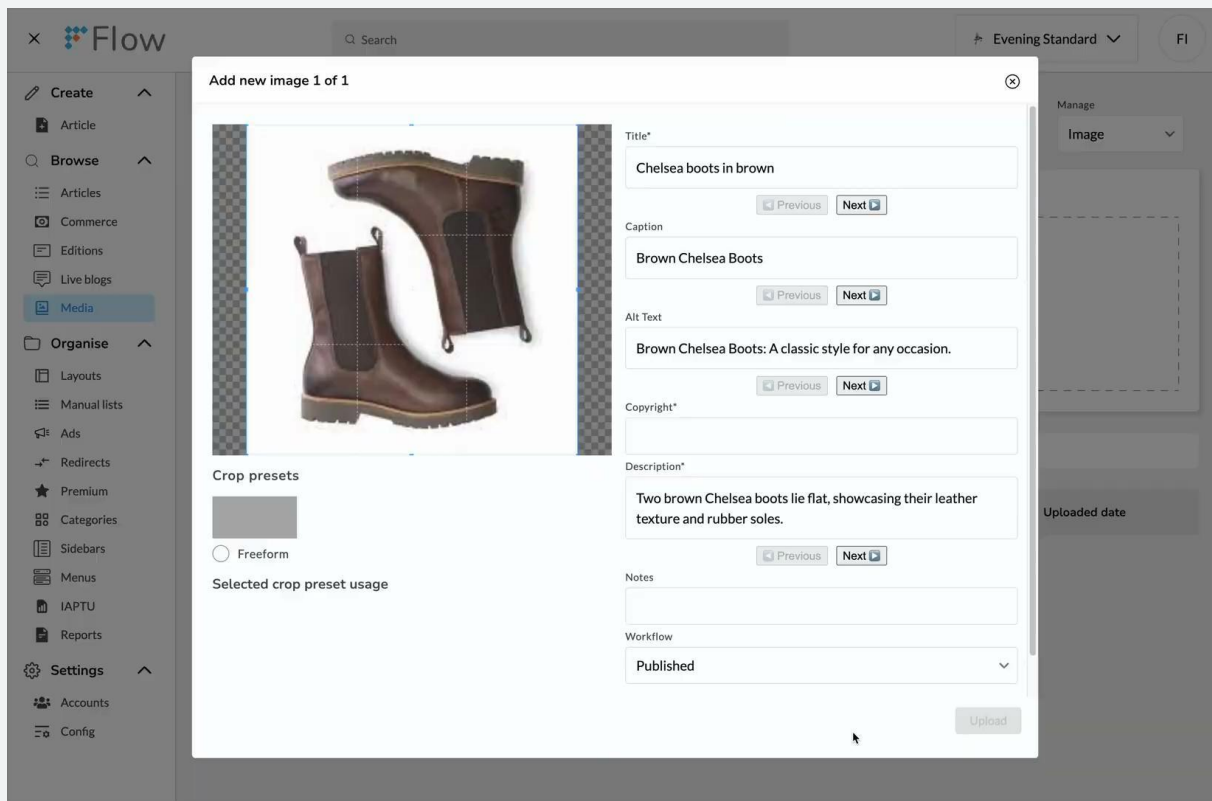
Digitize

Bananenbrot

- 100g weiches Butter
- 4 reife Bananen
- 200g brauner Zucker
- 1 Ei
- 1 Pck. Vanillezucker
- 1 1/2 TL Backpulver
- 1 Prise Salz
- 1/2 TL Zimt
- 220g Mehl
- 1 handvoll Walnüsse
- Bananen reiben und pürieren
- Butter, Zucker, Vanillezucker, Ei verquirlen
- Mehl, Backpulver, Salz, Zimt, Walnüsse unterrühren
- Teig in eine Kastenform 180° ca. 60 min.
- Auskühlen lassen



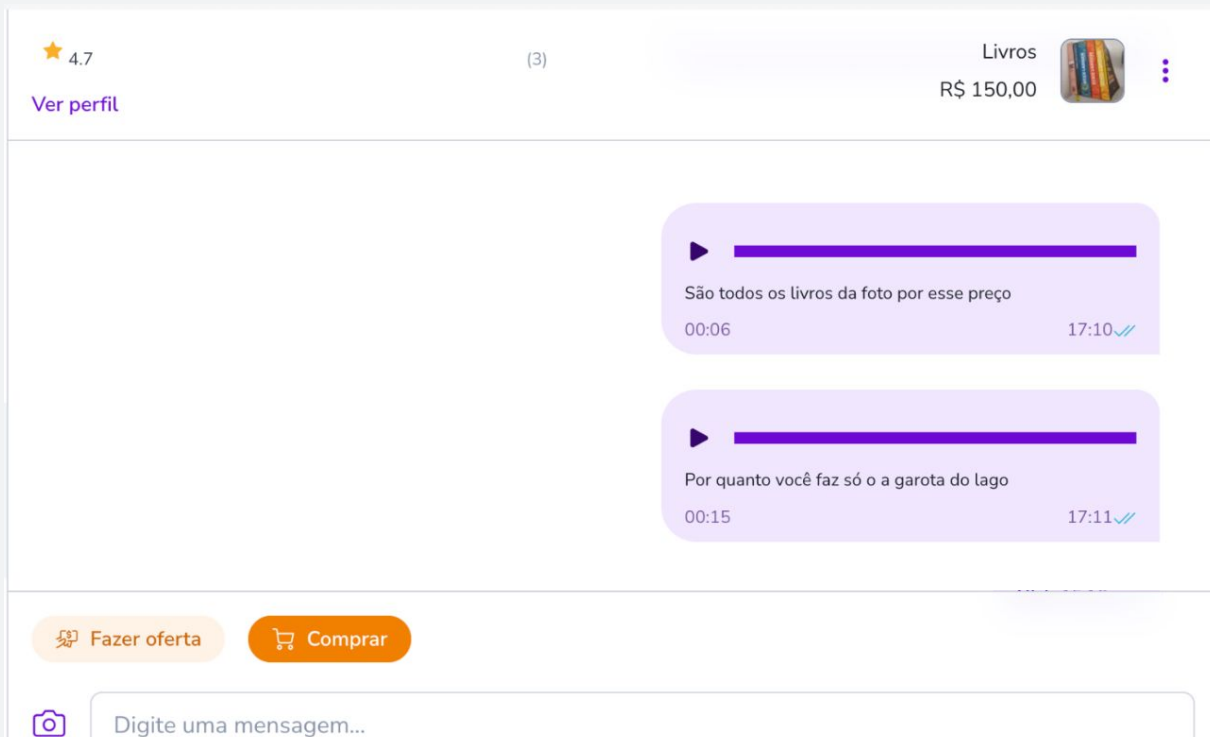
BrightSites' integration of the **Prompt API with image input** empowers journalists to automatically generate titles, captions, alt text, and descriptions directly from images within their CMS.





olx zap VivaReal

OLX used the **Prompt API with audio input** to transcribe audio messages, enhancing accessibility and enabling them to be read in more places and less time than the original audio.



Firestore AI Logic

[Build](#)[Run](#)[Solutions](#)[Pricing](#)[Docs](#)[Community](#)[Support](#)[/](#)[English](#)[Documentation](#) > [Firestore AI Logic](#)[Overview](#)[Fundamentals](#)[AI](#)[Build](#)[Run](#)[Reference](#)[Samples](#)[Overview](#)[DEVELOP USING AI](#)[Firestore Studio](#)[Gemini in Firestore](#)[Firestore MCP Server](#)[BUILD AI-POWERED APPS](#)[Firestore AI Logic](#)[Introduction](#)[Get started](#)

Here's everything we announced at I/O, from new Firestore Studio features to more ways to integrate AI. [Read blog.](#)

[Firestore](#) > [Documentation](#) > [Firestore AI Logic](#) > [AI](#)

Was this helpful?

[Send feedback](#)

Build hybrid experiences with on-device and cloud-hosted models

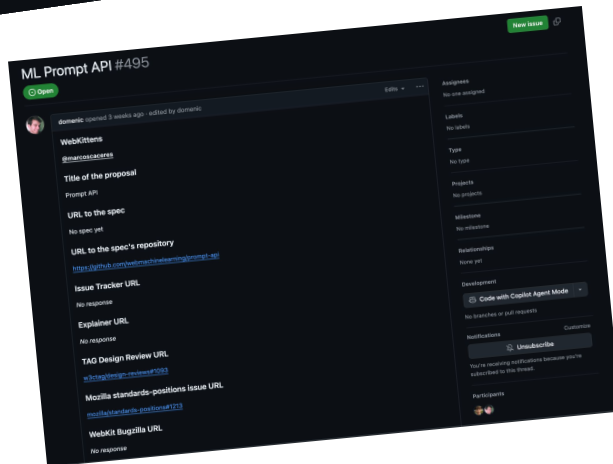
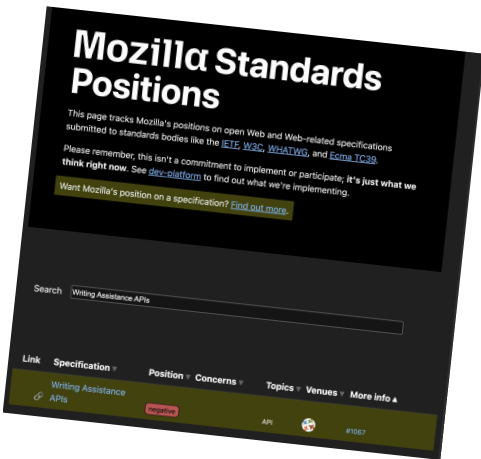
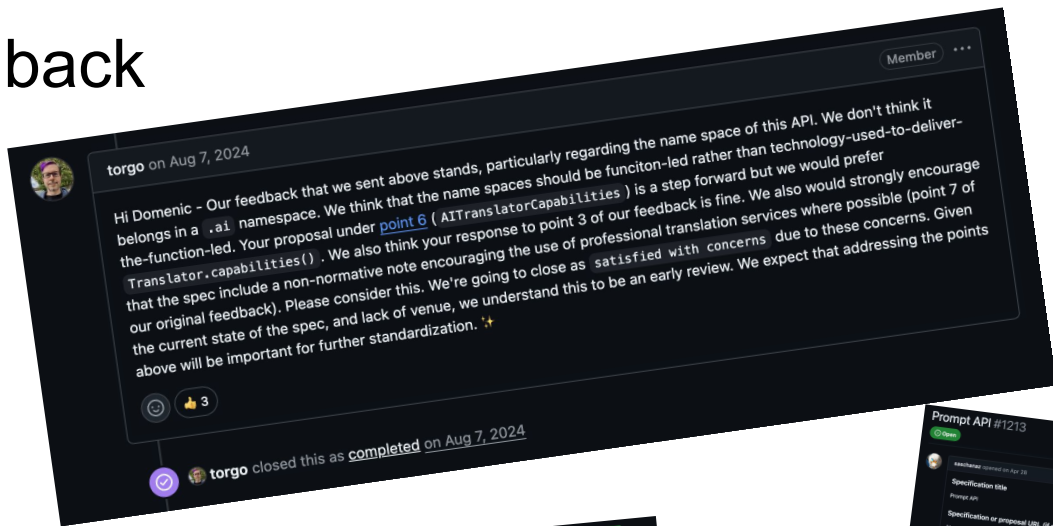


Experimental: Using the Firestore AI Logic SDKs to build hybrid experiences is an Experimental feature, which means that it isn't subject to any SLA or deprecation policy and could change in backwards-incompatible ways.

This initial release only **supports on-device inference for web apps running on Chrome on Desktop.**

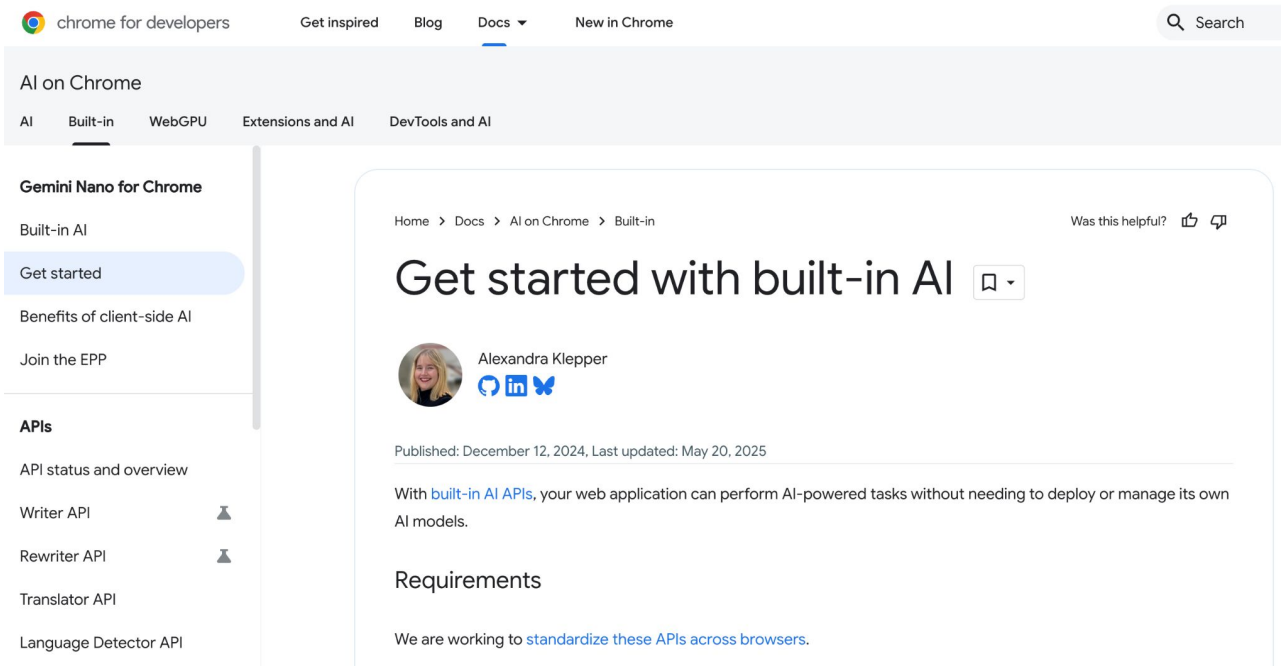
[Info](#)[Chat](#)

Seeking feedback



Mozilla

Learn more about Chrome's implementation



The screenshot shows the 'chrome for developers' website. The top navigation bar includes links for 'Get inspired', 'Blog', 'Docs', and 'New in Chrome', along with a search bar. The 'Docs' section is expanded, showing 'AI on Chrome' as the selected category. Under 'AI on Chrome', there are sub-sections: 'AI', 'Built-in' (which is highlighted), 'WebGPU', 'Extensions and AI', and 'DevTools and AI'. On the left sidebar, under 'Gemini Nano for Chrome', the 'Built-in AI' section is expanded, showing links for 'Get started', 'Benefits of client-side AI', and 'Join the EPP'. Below this, under 'APIs', there are links for 'API status and overview', 'Writer API', 'Rewriter API', 'Translator API', and 'Language Detector API'. The main content area is titled 'Get started with built-in AI' and is authored by Alexandra Klepper. It includes a breadcrumb trail: 'Home > Docs > AI on Chrome > Built-in'. The article text states: 'With built-in AI APIs, your web application can perform AI-powered tasks without needing to deploy or manage its own AI models.' It also has a 'Requirements' section that says: 'We are working to standardize these APIs across browsers.'

chrome for developers

Get inspired Blog Docs New in Chrome

Search

AI on Chrome

AI Built-in WebGPU Extensions and AI DevTools and AI

Gemini Nano for Chrome

Built-in AI

Get started

Benefits of client-side AI

Join the EPP

APIs

API status and overview

Writer API

Rewriter API

Translator API

Language Detector API

Home > Docs > AI on Chrome > Built-in

Was this helpful?

Get started with built-in AI

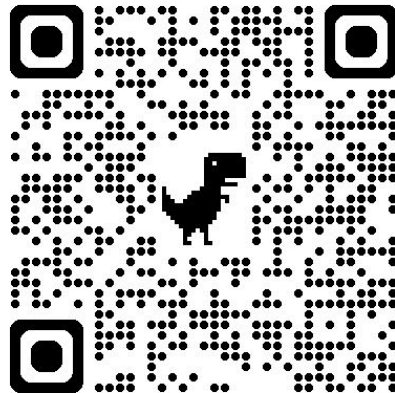
Alexandra Klepper

Published: December 12, 2024, Last updated: May 20, 2025

With [built-in AI APIs](#), your web application can perform AI-powered tasks without needing to deploy or manage its own AI models.

Requirements

We are working to [standardize these APIs across browsers](#).



Thank you

Thomas Steiner
tomac@google.com