CHROME Update #6

Built-in AI Early Preview Program

The Translation API

Authors: Thomas Steiner, Kenji Baheux

Contact: See this section

Last-updated: Oct 15, 2024
See changelog.

Latest news 📯

Oct 22, 2024

- Updated Setup steps to reflect a change in how the components are downloaded.
- Added a note about the option to remove limits on language packs.
- Added a section about language packs management.

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.

You can now explore the translation API, which works alongside the language detection API from Update #4. This update includes a preview of the API and instructions for local testing. plus implementation details to get you started.

Translation API

Purpose

The translation API is provided for local experimentation. Its purpose is to let you translate from one language to another. Currently, the API supports bi-directional translations between:

- English and Spanish.
- English and Japanese.

We intend to add support for more languages over time based on interest and viability. You can inform our prioritization through this form.

Early Preview Goals

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

1. The **quality of the translation,** 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.
   - Note: the current implementation is based on an old version of the API design.
4. The pair of languages we should prioritize next via this form.

Availability

The translation API is available, behind an experimental flag, from Chrome 131+ on desktop platforms.

- **You'll need Version 131.0.6778.2 or above.**
- We recommend using Chrome Canary or Chrome dev channel.

Requirements

There are no particular hardware requirements for the translation API, and the API is supported on the following desktop platforms: Windows, Mac, and Linux.

Setup

Prerequisites

1. Make sure that you are using Chrome on one of these platforms: Windows, Mac, or Linux.
2. Download Chrome Canary, and confirm that your version is equal to or newer than 131.0.6778.2 .

Enable the translation API

1. Navigate to `chrome://flags/#translation-api`
2. Select `Enabled`
   - If you want to try many language pairs, select `Enabled without language pack limit`
3. Relaunch Chrome.

Trigger downloads for the translation components

1. Navigate to https://translation-demo.glitch.me/
2. Click "from en to es" and "from en to ja" under translate().
3. Wait (language models and translation components are downloaded in the background).
   - Navigate to `chrome://components` and look for `TranslateKit` components to monitor progress if you want.
4. Now the translation API should be ready to translate for English ⇔ Spanish and English ⇔ Japanese language pairs.

Confirm availability of the Translation API

- Open DevTools and send `await translation.canTranslate({sourceLanguage: "en", targetLanguage: "es"});` in the console. If this returns `"readily"`, then you are all set.
  - Otherwise, double check that you have followed the prerequisites and the enabling steps.
  - If this still fails, please reach out with details about your environment and any error messages you may have.

Language packs management

Navigate to `chrome://on-device-translation-internals/` to manually install or uninstall language packs.

API overview

Sample code

Note: the current implementation doesn't correspond to the API as documented in the explainer. The following sample code represents the current implementation, and will eventually break as we adjust the code to be in sync with the 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 translator is available for the required language pair

JavaScript

```javascript
const languagePair = {
  sourceLanguage: 'en', // Or detect the source language with the Language Detection API
  targetLanguage: 'es',
};

const canTranslate = await translation.canTranslate(languagePair);
let translator;
if (canTranslate !== 'no') {
  if (canTranslate === 'readily') {
    // The translator can immediately be used.
    translator = await translation.createTranslator(languagePair);
  }/* else {
    // The translator can be used after the model download.
    translator = await translation.createTranslator(languagePair);
    translator.addEventListener('downloadprogress', (e) => {
      console.log(e.loaded, e.total);
    });
    await translator.ready;
  } */
} else {
    // The translator can't be used at all.
}
```

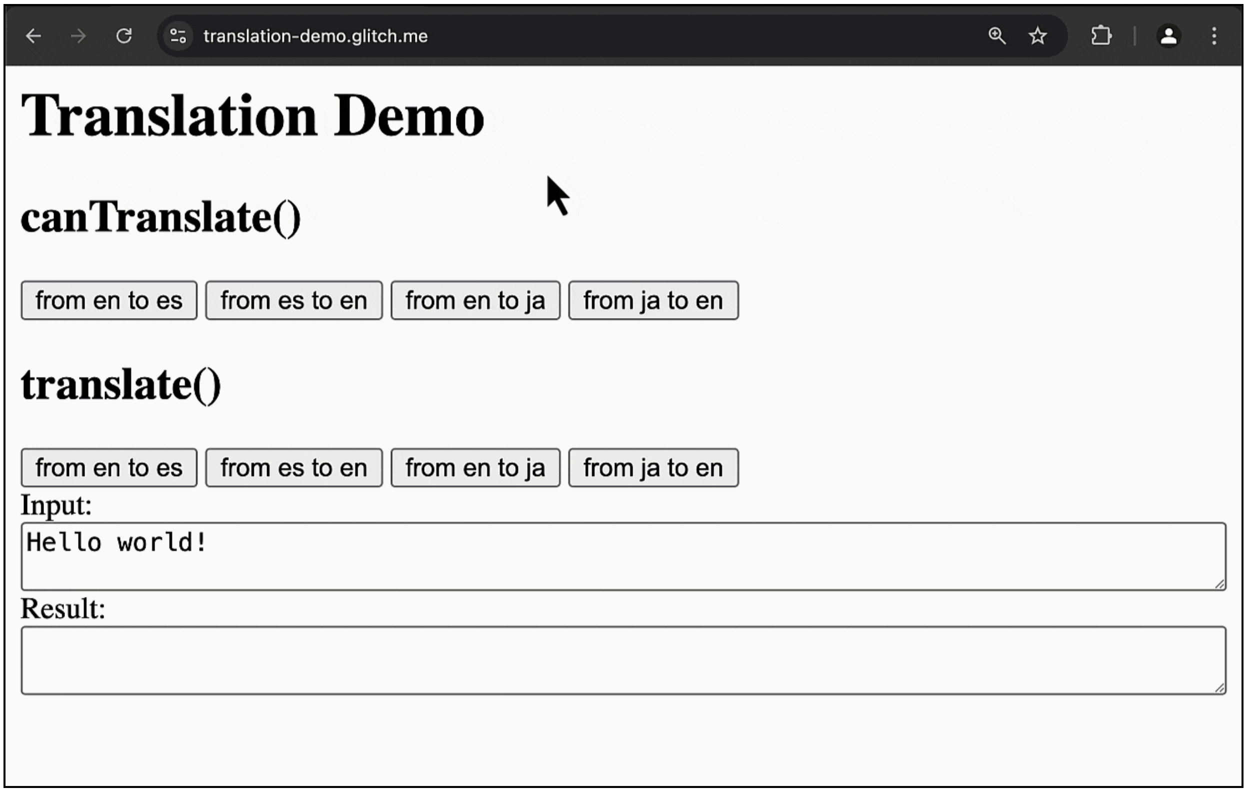
Translating from one language to another

JavaScript

```
const someUserText = 'Hello and a warm welcome to the Early Preview Program!';
const translation = await translator.translate(someUserText);
console.log(translation);
// Logs "¡Hola y una cálida bienvenida al programa de previsualización
anticipada!"
```

Demo

For a live demo, see https://translation-demo.glitch.me/.



Translation Demo

canTranslate()

translate()

Input:

Hello world!

Result:

Caveats

Here are known temporary limitations:

- **Performance with long text:** long test translation request blocks the succeeding translation requests.
- **Restrictions on language pairs:** in preparation for the upcoming origin trial, only specific languages will be allowed to mitigate fingerprinting concerns. We expect to improve this over time. For now, here are the heuristics we are using:
    - Either the source or the destination language must be in the user's preferred languages.
    - The other language must be among the top 12 most popular languages (reference).
    - For local prototyping, you can bypass these checks by adding this command line option: `--disable-features=TranslationAPIAcceptLanguagesCheck`

Appendix

Full API surface

The full API surface *for the current implementation* is described below. See Web IDL for details on the language.

Unset

```
[Exposed=(Window,Worker)]
interface Translation {
  Promise<TranslationAvailability> canTranslate(TranslationLanguageOptions
options);
  Promise<LanguageTranslator> createTranslator(TranslationLanguageOptions
options);
};

[Exposed=(Window,Worker)]
interface LanguageTranslator {

  Promise<(DOMString)> translate(DOMString input);
};

partial interface WindowOrWorkerGlobalScope {
  readonly attribute Translation translation;
};

enum TranslationAvailability { "readily", "after-download", "no" };

dictionary TranslationLanguageOptions {
  required DOMString targetLanguage;
  required DOMString sourceLanguage;
};
```

Supported language pairs

en ⇔ ar	en ⇔ bn	en ⇔ de	en ⇔ es
en ⇔ fr	en ⇔ hi	en ⇔ it	en ⇔ ja
en ⇔ ko	en ⇔ nl	en ⇔ pl	en ⇔ pt
en ⇔ ru	en ⇔ th	en ⇔ tr	en ⇔ vi
en ⇔ zh	en ⇔ zh-Hant		

See BCP 47 codes for details.

Head over to `chrome://on-device-translation-internals/` to manage language packages.

General feedback

Feedback form for 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 APIs

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:

- Translation API spec issues

Feedback about language pairs to support

Want to see other language pairs supported? Tell us your preferences in this form..

Other feedback

For other questions or issues, reach out directly by sending an email to the mailing list owners (`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
Sep 30, 2024	- First version.
Oct 15, 2024	- Updated the Setup steps with the chrome://flags instead of the command line options approach. - Added info about new languages, bringing the total to 18. - Added info about an internals page allowing developers to manage language packages. - Added info about some limitations on valid language pairs for the upcoming Origin Trial to mitigate fingerprinting concerns.
Oct 22, 2024	- Simplified setup steps.