




















transloadit / uppy Public

<> Code Issues 169 Pull requests 51 Actions Projects Security and quality

47 Branches 4442 Tags Go to file Go to file <> Code

qxprakash @uppy/core: fix stale comments (#6272)

588e1c0 · 7 hours ago

 .changeset	@uppy/dashboard: My Device ...	8 hours ago
 .cursor/rules	Headless components (#5727)	11 months ago
 .github	remove stale heroku job (#6254)	2 weeks ago
 .vscode	Migrate from Eslint/Prettier/Sty...	10 months ago
 .yarn/patches	Improve yarn changesets patch	9 months ago
 assets	Delete old, unused files (#5788)	10 months ago
 examples	@uppy/examples: update tran...	last month
 output	refactor: make companion out...	8 years ago
 packages	@uppy/core: fix stale comment...	7 hours ago
 private	Introduce @uppy/image-gener...	4 months ago
 scripts	Update versions in readme's a...	8 months ago
 .browserslistrc	Remove IE polyfills and specia...	5 years ago
 .dockerignore	run type checks also for comp...	3 years ago
 .editorconfig	Add tests and eslint	11 years ago
 .env.example	Google Picker (#5443)	2 years ago
 .gitignore	Migrate from Cypress to Vitest ...	9 months ago
 .yarnrc.yml	fixup! Add .npmrc	9 months ago
 BUNDLE-README.md	[ci] release (#6170)	last month
 CHANGELOG.md	Release: uppy@4.18.0 (#5796)	10 months ago

CLAUDE.md	Add CLAUDE.md	9 months ago
Dockerfile	Fix Companion build output an...	8 months ago
LICENSE	Update LICENSE	7 years ago
README.md	[ci] release (#6170)	last month
SECURITY.md	meta: enable prettier for markd...	2 years ago
biome.json	Run biome check on main (#5...	8 months ago
package.json	Upgrade yarn (#6133)	3 months ago
tsconfig.shared.json	From Babel to TS (#5792)	10 months ago
turbo.json	Introduce @uppy/image-gener...	4 months ago
yarn.lock	@UPPY: upgrade esbuild from...	last month

[README](#) [Contributing](#) [MIT license](#) [Security](#)

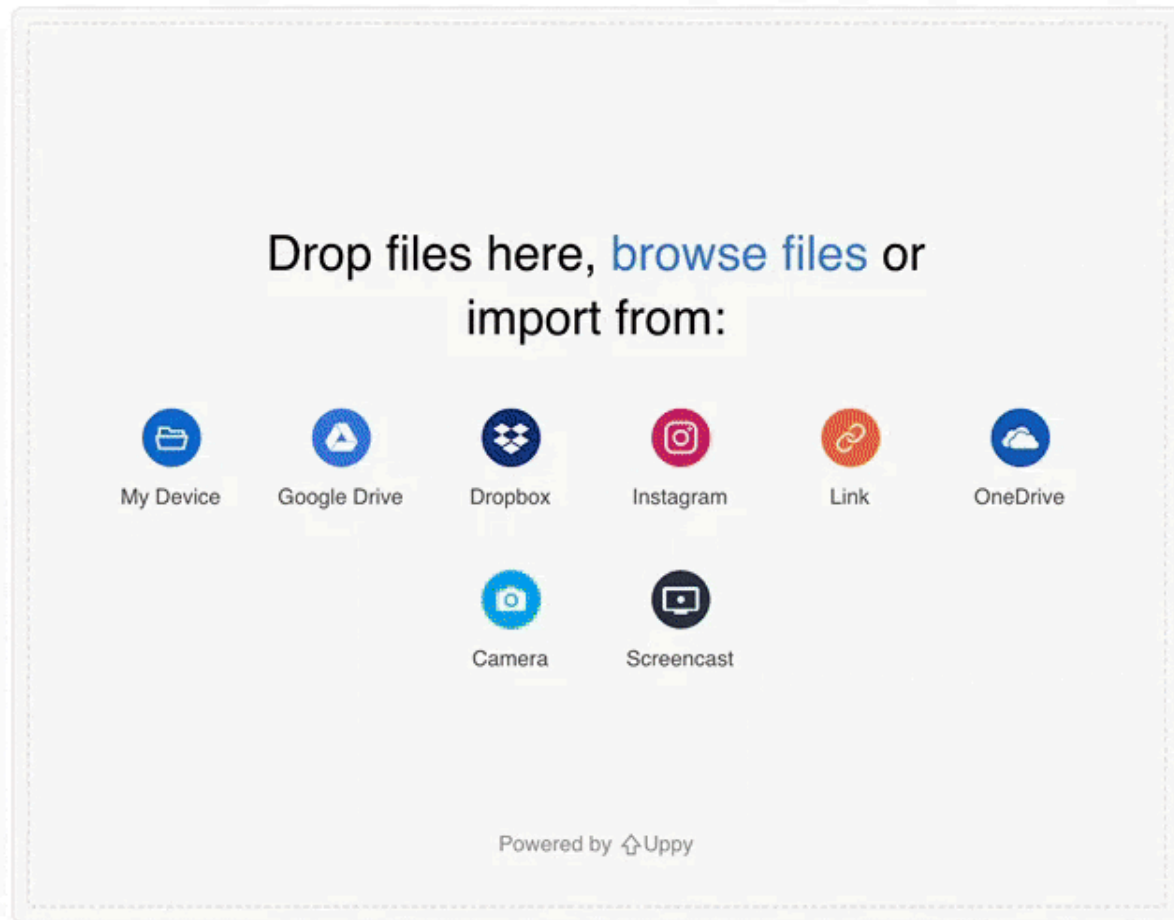


**Uppy** npm v5.2.4

Uppy is a sleek, modular JavaScript file uploader that integrates seamlessly with any application. It's fast, has a comprehensible API and lets you worry about more important problems than building a file uploader.

- **Fetch** files from local disk, remote URLs, Google Drive, Dropbox, Box, Instagram or snap and record selfies with a camera
- **Preview** and edit metadata with a nice interface
- **Upload** to the final destination, optionally process/encode





[Read the docs](#) | [Try Uppy](#)



Uppy is being developed by the folks at [Transloadit](#), a versatile API to handle any file in your app.

Tests	CI <span>passing</span>	Companion <span>passing</span>	
Deploys	Release <span>passing</span>	Companion Edge Deploy <span>passing</span>	Deploy to GitHub Pages <span>passing</span>

## Example

Code used in the above example:

```
import Uppy from '@uppy/core'
import Dashboard from '@uppy/dashboard'
import RemoteSources from '@uppy/remote-sources'
import ImageEditor from '@uppy/image-editor'
import Webcam from '@uppy/webcam'
import Tus from '@uppy/tus'

const uppy = new Uppy()
```



```
.use(Dashboard, { trigger: '#select-files' })
.use(RemoteSources, { companionUrl: 'https://companion.uppy.io' })
.use(Webcam)
.use(ImageEditor)
.use(Tus, { endpoint: 'https://tusd.tusdemo.net/files/' })
.on('complete', (result) => {
  console.log('Upload result:', result)
})
```

[Try it online](#) or [read the docs](#) for more details on how to use Uppy and its plugins.

## Integrations

Uppy has first-class support for plain JS/HTML, [React](#), [Svelte](#), [Vue](#), and [Angular](#).

For the supported frameworks (except Angular) Uppy offers three ways to build user interfaces:

1. **Pre-composed, plug-and-play components.** Mainly `<Dashboard />`. The downside is that you can't customize the UI.
2. **Headless components.** Smaller components, easier to override the styles or compose them together with your own components.
3. **Hooks.** Attach our logic to your own components, no restrictions, create a tailor-made UI.

## Features

- Lightweight, modular plugin-based architecture, light on dependencies ⚡
- Resumable file uploads via the open [tus](#) standard, so large uploads survive network hiccups
- Supports picking files from: Webcam, Dropbox, Box, Google Drive, Instagram, bypassing the user's device where possible, syncing between servers directly via [@uppy/companion](#)
- Works great with file encoding and processing backends, such as [Transloadit](#), works great without (all you need is to roll your own Apache/Nginx/Node/FFmpeg/etc backend)
- Sleek user interface ✨
- Optional file recovery (after a browser crash) with [Golden Retriever](#)
- Speaks several languages (i18n) 🌐
- Built with accessibility in mind
- Free for the world, forever (as in beer 🍺, pizza 🍕, and liberty 🗽)
- Cute as a puppy, also accepts cat pictures 🐶

## Installation

```
npm install @uppy/core @uppy/dashboard @uppy/tus
```



Add CSS [uppy.min.css](https://releases.transloadit.com/uppy/v5.2.4/uppy.min.css), either to your HTML page's `<head>` or include in JS, if your bundler of choice supports it.

Alternatively, you can also use a pre-built bundle from Transloadit's CDN: Smart CDN. In that case `uppy` will attach itself to the global `window.Uppy` object.

**!** The bundle consists of most Uppy plugins, so this method is not recommended for production, as your users will have to download all plugins when you are likely using only a few.

```
<!-- 1. Add CSS to `<head>` -->
<link
  href="https://releases.transloadit.com/uppy/v5.2.4/uppy.min.css"
  rel="stylesheet"
/>

<!-- 2. Initialize -->
<div id="files-drag-drop"></div>
<script type="module">
  import {
    Uppy,
    Dashboard,
    Tus,
  } from 'https://releases.transloadit.com/uppy/v5.2.4/uppy.min.mjs'

  const uppy = new Uppy()
  uppy.use(Dashboard, { target: '#files-drag-drop' })
  uppy.use(Tus, { endpoint: 'https://tusd.tusdemo.net/files/' })
</script>
```

## Documentation

- [Uppy](#) — full list of options, methods and events
- [Companion](#) — setting up and running a Companion instance, which adds support for Instagram, Dropbox, Box, Google Drive and remote URLs
- [React](#) — components to integrate Uppy UI plugins with React apps
- [Architecture & Writing a Plugin](#) — how to write a plugin for Uppy

## Plugins

### UI Elements

- [Dashboard](#) — universal UI with previews, progress bars, metadata editor and all the cool stuff. Required for most UI plugins like Webcam and Instagram
- Headless components ([react](#), [svelte](#), [vue](#))

## Sources

- [Drag & Drop](#) — plain drag and drop area
- [File Input](#) — even plainer “select files” button
- [Webcam](#) — snap and record those selfies 📷
- © [Google Drive](#) — import files from Google Drive
- © [Dropbox](#) — import files from Dropbox
- © [Box](#) — import files from Box
- © [Instagram](#) — import images and videos from Instagram
- © [Facebook](#) — import images and videos from Facebook
- © [OneDrive](#) — import files from Microsoft OneDrive
- © [Import From URL](#) — import direct URLs from anywhere on the web

The © mark means that [@uppy/companion](#), a server-side component, is needed for a plugin to work.

## Destinations

- [Tus](#) — resumable uploads via the open [tus](#) standard
- [XHR Upload](#) — regular uploads for any backend out there (like Apache, Nginx)
- [AWS S3](#) — plain upload to AWS S3 or compatible services

## File Processing

- [Transloadit](#) — support for [Transloadit](#)'s robust file uploading and encoding backend

## Miscellaneous

- [Golden Retriever](#) — restores files after a browser crash, like it's nothing
- [Thumbnail Generator](#) — generates image previews (included by default with `Dashboard`)
- [Form](#) — collects metadata from `<form>` right before an Uppy upload, then optionally appends results back to the form

## Browser Support

---

We aim to support recent versions of Chrome, Firefox, and Safari.

## FAQ

---

Why not use `<input type="file">`?

Having no JavaScript beats having a lot of it, so that's a fair question! Running an uploading & encoding business for ten years though we found that in cases, the file input leaves some to be desired:

- We received complaints about broken uploads and found that resumable uploads are important, especially for big files and to be inclusive towards people on poorer connections (we also launched [tus.io](https://tus.io) to attack that problem). Uppy uploads can survive network outages and browser crashes or accidental navigate-aways.
- Uppy supports editing meta information before uploading.
- Uppy allows cropping images before uploading.
- There's the situation where people are using their mobile devices and want to upload on the go, but they have their picture on Instagram, files in Dropbox or a plain file URL from anywhere on the open web. Uppy allows to pick files from those and push it to the destination without downloading it to your mobile device first.
- Accurate upload progress reporting is an issue on many platforms.
- Some file validation — size, type, number of files — can be done on the client with Uppy.
- Uppy integrates webcam support, in case your users want to upload a picture/video/audio that does not exist yet :)
- A larger drag and drop surface can be pleasant to work with. Some people also like that you can control the styling, language, etc.
- Uppy is aware of encoding backends. Often after an upload, the server needs to rotate, detect faces, optimize for iPad, or what have you. Uppy can track progress of this and report back to the user in different ways.
- Sometimes you might want your uploads to happen while you continue to interact on the same single page.

Not all apps need all these features. An `<input type="file">` is fine in many situations. But these were a few things that our customers hit / asked about enough to spark us to develop Uppy.

## Why is all this goodness free?

Transloadit's team is small and we have a shared ambition to make a living from open source. By giving away projects like [tus.io](https://tus.io) and [Uppy](https://uppy.io), we're hoping to advance the state of the art, make life a tiny little bit better for everyone and in doing so have rewarding jobs and get some eyes on our commercial service: [a content ingestion & processing platform](https://transloadit.com).

Our thinking is that if only a fraction of our open source userbase can see the appeal of hosted versions straight from the source, that could already be enough to sustain our work. So far this is working out! We're able to dedicate 80% of our time to open source and haven't gone bankrupt yet. :D

## Does Uppy support S3 uploads?

Yes, please check out the [docs](#) for more information.

## Can I use Uppy with Rails/Node.js/Go/PHP?

Yes, whatever you want on the backend will work with `@uppy/xhr-upload` plugin, since it only does a `POST` or `PUT` request. Here's a [PHP backend example](#).

If you want resumability with the Tus plugin, use [one of the tus server implementations](#) 🔥

And you'll need `@uppy/companion` if you'd like your users to be able to pick files from Instagram, Google Drive, Dropbox or via direct URLs (with more services coming).

## Contributions are welcome

---

- Contributor's guide in [.github/CONTRIBUTING.md](#)
- Changelog to track our release progress (we aim to roll out a release every month): [CHANGELOG.md](#)

## Used by

---

Uppy is used by: [Photobox](#), [Issuu](#), [Law Insider](#), [Cool Tabs](#), [Soundoff](#), [Scrumi](#), [Crive](#) and others.

Use Uppy in your project? [Let us know!](#)

## Contributors

---



## Releases 397

 [@uppy/companion@6.2.2](#) Latest

on Mar 25

[+ 396 releases](#)

## Packages

No packages published

## Contributors 450



[+ 436 contributors](#)

## Languages

