How to receive shared files

- Using the Web Share Target API
- Further reading
- Demo

Using the Web Share Target API

First, declare a `share_target` in your web app manifest that lists an `action` (your URL that handles shared files), a `method` (`"POST"` for files) and an `enctype` (`"multipart/form-data"` for files) and a `params` object that contains a `files` property with an array of objects with a `name` and `accept` property that list the sharable file types and the name to obtain them.

```
{
  "share_target": {
    "action": "/receive-files/",
    "method": "POST",
    "enctype": "multipart/form-data",
    "params": {
      "files": [
        {
          "name": "image",
          "accept": ["image/jpeg", "image/png", "image/webp", "image/gif"]
        }
      ]
    }
  }
}
```

You then need to handle incoming `POST` requests in the service worker. The file is temporarily stored in a media cache, so it can be consumed in the client. This can be done by redirecting the app to a URL with a special marker query parameter like `share-target`.

```
self.addEventListener('fetch', (fetchEvent) => {
  if (fetchEvent.request.url.endsWith('/receive-files/') && fetchEvent.request.method ===
'POST') {
    return fetchEvent.respondWith(
      (async () => {
        const formData = await fetchEvent.request.formData();
        const image = formData.get('image');
        const keys = await caches.keys();
        const mediaCache = await caches.open(keys.filter((key) =>
key.startsWith('media'))[0]);
        await mediaCache.put('shared-image', new Response(image));
        return Response.redirect('./?share-target', 303);
      })(),
    );
  }
});
```

Finally, you need to consume the file in the client.

```
window.addEventListener('load', async () => {
  if (location.search.includes('share-target')) {
    const keys = await caches.keys();
    const mediaCache = await caches.open(
      keys.filter((key) => key.startsWith('media'))[0],
    );
    const image = await mediaCache.match('shared-image');
    if (image) {
      const blob = await image.blob();
      await mediaCache.delete('shared-image');
      // Handle the shared file somehow.
    }
  }
});
```

Further reading

- Receiving shared data with the Web Share Target API

- W3C Web Share Target API

Demo

□□□ #3
□ □□□ □□□ 2026-10-31 23:02:43 UTC
□ cocobook-sync □□□ 2026-11-04 17:39:42 UTC