Window: showOpenFilePicker() method

- Syntax
  - Parameters
  - Return value
  - Exceptions
- Security
- Examples
- Specifications
- Browser compatibility
- See also

The **`showOpenFilePicker()`** method of the Window interface shows a file picker that allows a user to select a file or multiple files and returns a handle for the file(s).

Syntax

```
showOpenFilePicker()
showOpenFilePicker(options)
```

Parameters

- `options`
  - : An object containing options, which are as follows:
    - `excludeAcceptAllOption`

    - : A boolean value that defaults to `false`. By default the picker should include an option to not apply any file type filters (instigated with the type option below). Setting this option to `true` means that option is *not* available.
  - `id`
    - : By specifying an ID, the browser can remember different directories for different IDs. If the same ID is used for another picker, the picker opens in the same directory.
  - `multiple`
    - : A boolean value that defaults to `false`. When set to `true` multiple files may be selected.
  - `startIn`
    - : A FileSystemHandle or a well-known directory (`"desktop"`, `"documents"`, `"downloads"`, `"music"`, `"pictures"`, or `"videos"`) to open the dialog in.
  - `types`
    - : An Array of allowed file types to pick. Each item is an object with the following options:
      - `description`
        - : An optional description of the category of files types allowed. Defaults to an empty string.
      - `accept`
        - : An Object with the keys set to the MIME type and the values an Array of file extensions (see below for an example).

Return value

A Promise whose fulfillment handler receives an Array of FileSystemFileHandle objects.

Exceptions

- `AbortError` DOMException
  - : Thrown if the user dismisses the prompt without making a selection, or if the user agent deems any selected files too sensitive or dangerous.
- `SecurityError` DOMException
  - : Thrown if the call was blocked by the same-origin policy or it was not called via a user interaction such as a button press.
- TypeError
  - : Thrown if accept types can't be processed, which may happen if:
    - Any key string of the `accept` options of any item in `types` options can't parse a valid MIME type.

- Any value string(s) of the `accept` options of any item in `types` options is invalid, for example, if it does not start with `.` and if end with `.`, or if it contains any invalid code points and its length is more than 16.
- The `types` options is empty and the `excludeAcceptAllOption` options is `true`.

Security

Transient user activation is required. The user has to interact with the page or a UI element in order for this feature to work.

Examples

Here we set the options object for passing into the method. We'll allow a selection of image file types, with no option to allow for all files types, or multiple file selection.

```
const pickerOpts = {
  types: [
    {
      description: "Images",
      accept: {
        "image/*": [".png", ".gif", ".jpeg", ".jpg"],
      },
    },
  ],
  excludeAcceptAllOption: true,
  multiple: false,
};
```

Next we can create an asynchronous function which show the file picker and return the selected file.

```
// create a reference for our file handle
let fileHandle;

async function getFile() {
  // open file picker, destructure the one element returned array
  [fileHandle] = await window.showOpenFilePicker(pickerOpts);
```

```
  // run code with our fileHandle
}
```

Specifications

Browser compatibility

See also

- File System API
- The File System Access API: simplifying access to local files

---

□□□ #3
□ □□□ □□□ 2026-10-31 03:27:53 UTC
□ cocobook-sync □□□ 2026-11-04 17:35:22 UTC