parse-icon-set

- parseIconSet()
  - Usage
    - Callback
  - parseIconSetAsync
  - Example
    - Validation
    - JSON Modules


```
title: parseIconSet() in Iconify Utils
types:
  IconifyJSON: '/docs/types/iconify-json.md'
  IconifyInfo: '/docs/types/iconify-info.md'
  IconifyIcon: '/docs/types/iconify-icon.md'
functions:
  validateIconSet: './validate-icon-set.md'
  quicklyValidateIconSet: './quickly-validate-icon-set.md'
```

parseIconSet()

This function is part of Iconify Utils package.

Functions `[func]parseIconSet()` and `[func]parseIconSetAsync()` parse icon set, calling custom function for every icon.

Usage

Function has the following parameters:

- `[prop]data`, `[type]IconifyJSON`. Icon set data.
- `[prop]callback`, `[type]function`. Callback, called for each icon found in icon set.

Function returns names of parsed icons as `[type]string[]`.

Callback

Callback function has 2 parameters:

- `[prop]name`, `[type]string`. Icon name.
- `[prop]data`, `[type]IconifyIcon | null`. Icon data, `null` if icon is invalid.

Callback is called for each entry in icon set: all icons, aliases and all entries in `[prop]not_found` property.

If `[prop]data` parameter in callback is `null`, icon is invalid or missing.

parseIconSetAsync

Function `[func]parseIconSetAsync()` is identical to `[func]parseIconSet()`, except that it is asynchronous and callback is also an asynchronous function:

```
await parseIconSetAsync(iconSet, async (name, data) => {
	// do async stuff
});
```

Example

Exporting all icons from icon set:

```
src: libraries/utils/parse.ts
title: 'usage.ts'
```

Counting icons in icon set:

```
src: libraries/utils/count.ts
title: 'count.ts'
```

Code used in example above should be used to count icons in icon set when calculating number of icons for `[type]IconifyInfo` data. It counts all icons, excluding hidden icons and basic aliases.

Validation

Function does not check icon set for errors. Before using it, validate icon set using `[func]validateIconSet()` or `[func]quicklyValidateIconSet()`.

JSON Modules

When using ES modules, examples above might require running node with `[str]--experimental-json-modules` flag.

Until JSON modules support in Node.js becomes stable, you can avoid using flag by replacing `[func]import` with `[func]require()` because `[func]require()` supports JSON files:

```
const { icons } = require('@iconify-json/codicon');
```

□□□ #2
□ □□□ □□□ 2026-10-31 14:29:17 UTC
□ □□□ □□□ 2026-10-31 14:29:18 UTC