

---

If this tool saves you time, please consider making a donation towards the continued maintenance and development: <https://donate.biesbjerg.com>



## ngx-translate-extract

Extract translatable (ngx-translate) strings and save as a JSON or Gettext pot file. Merges with existing strings if the output file already exists.

### Install

Install the package in your project:

```
npm install @biesbjerg/ngx-translate-extract --save-dev
```

Add a script to your project's **package.json**:

```
1  ...
2  "scripts": {
3    "i18n:init": "ngx-translate-extract --input ./src --output ./src/
      assets/i18n/template.json --key-as-default-value --replace --
      format json",
4    "i18n:extract": "ngx-translate-extract --input ./src --output ./src/
      assets/i18n/{en,da,de,fi,nb,nl,sv}.json --clean --format json"
5  }
6  ...
```

You can now run `npm run i18n:extract` and it will extract strings from your project.

### Usage

#### Extract from dir and save to file

```
ngx-translate-extract --input ./src --output ./src/assets/i18n/
strings.json
```

#### Extract from multiple dirs

```
ngx-translate-extract --input ./src-a ./src-b --output ./src/assets/
i18n/strings.json
```

#### Extract and save to multiple files using path expansion

---

```
ngx-translate-extract --input ./src --output ./src/i18n/{da,en}.json
```

## JSON indentation

Tabs are used by default for indentation when saving extracted strings in json formats:

If you want to use spaces instead, you can do the following:

```
ngx-translate-extract --input ./src --output ./src/i18n/en.json --  
format-indentation ''
```

## Marker function

If you want to extract strings that are not passed directly to `TranslateService's get() /instant() /stream()` methods, you can wrap them in a marker function to let `ngx-translate-extract` know you want to extract them.

Install marker function: `npm install @biesbjerg/ngx-translate-extract-marker`

```
1 import { marker } from '@biesbjerg/ngx-translate-extract-marker';  
2  
3 marker('Extract me');
```

You can alias the marker function if needed:

```
1 import { marker as _ } from '@biesbjerg/ngx-translate-extract-marker';  
2  
3 _('Extract me');
```

*Note: `ngx-translate-extract` will automatically detect the import name*

## Commandline arguments

```
1 Usage:  
2 ngx-translate-extract [options]  
3  
4 Output  
5 --format, -f          Format [string] [choices: "json", "  
    namespaces-json", "pot"] [default: "json"]  
6 --format-indentation, --fi Format indentation (JSON/Namedspaced JSON  
    ) [string] [default: ""]  
7 --sort, -s           Sort strings in alphabetical order  
                        [boolean]
```

---

```

8  --clean, -c          Remove obsolete strings after merge
                        [boolean]
9  --replace, -r        Replace the contents of output file if it
                        exists (Merges by default) [boolean]
10
11  Extracted key value (defaults to empty string)
12  --key-as-default-value, -k    Use key as default value
                        [boolean]
13  --null-as-default-value, -n    Use null as default value
                        [boolean]
14  --string-as-default-value, -d  Use string as default value
                        [string]
15
16  Options:
17  --version, -v        Show version number
                        [
18  boolean]
19  --help, -h          Show help
                        [boolean]
20  --input, -i          Paths you would like to extract strings from. You can
                        use path expansion, glob patterns and
                        multiple paths [array] [required] [default: ["/
                        Users/kim/apps/ngx-translate-extract"]]
21  --output, -o         Paths where you would like to save extracted strings.
                        You can use path expansion, glob
                        patterns and multiple paths
                        [
22  array] [required]
23
24  Examples:
25  ngx-translate-extract -i ./src-a/ -i ./src-b/ -o strings.json
                        Extract (ts, html) from multiple paths
26  ngx-translate-extract -i './{src-a,src-b}/' -o strings.json
                        Extract (ts, html) from multiple paths using brace
                        expansion
27  ngx-translate-extract -i ./src/ -o ./i18n/da.json -o ./i18n/en.json
                        Extract (ts, html) and save to da.json and en.json
28  ngx-translate-extract -i ./src/ -o './i18n/{en,da}.json'
                        Extract (ts, html) and save to da.json and en.json
                        using brace
                        expansion
29  ngx-translate-extract -i './src/**/*.{ts,tsx,html}' -o strings.json
                        Extract from ts, tsx and html
30  ngx-translate-extract -i './src/**/*.!(*.spec).{ts,html}' -o
                        Extract from ts, html, excluding files with ".spec"
31  strings.json

```

---

---

### **Note for GetText users**

Please pay attention of which version of `gettext-parser` you actually use in your project. For instance, `gettext-parser:1.2.2` does not support HTML tags in translation keys.