

bcp-47-match

 main no status

Match BCP 47 language tags with language ranges per RFC 4647.

Contents

[What is this?](#)

[When should I use this?](#)

[Install](#)

[Use](#)

[API](#)

[basicFilter\(tags\[, ranges='*'\]\)](#)

[extendedFilter\(tags\[, ranges='*'\]\)](#)

[lookup\(tags, ranges\)](#)

[Types](#)

[Compatibility](#)

[Security](#)

[Related](#)

[Contribute](#)

[License](#)

What is this?

This package can match [BCP 47](#) language tags with “language ranges” per [RFC 4647](#). This is done by the `:lang()` pseudo class in CSS, the Accept-Language HTTP header, and a few other places.

When should I use this?

You can use this package if you want to choose a certain document based on language tags.

Install

This package is [ESM only](#). In Node.js (version 14.14+, 16.0+), install with [npm](#):

```
npm install bcp-47-match
```

In Deno with [esm.sh](#):

```
import * as bcp47Match from 'https://esm.sh/bcp-47-match@2'
```

In browsers with [esm.sh](#):

Contribute

Yes please! See [How to Contribute to Open Source](#).

License

[MIT](#) © [Titus Wormer](#)

Compatibility

This package is at least compatible with all maintained versions of Node.js. As of now, that is Node.js 14.14+ and 16.0+. It also works in Deno and modern browsers.

Security

This package is safe.

Related

- [wooorm/bcp-47](https://github.com/wooorm/bcp-47) — parse and serialize BCP 47 language tags
- [wooorm/bcp-47-normalize](https://github.com/wooorm/bcp-47-normalize) — normalize, canonicalize, and format BCP 47 tags
- [wooorm/iso-3166](https://github.com/wooorm/iso-3166) — ISO 3166 codes
- [wooorm/iso-639-2](https://github.com/wooorm/iso-639-2) — ISO 639-2 codes
- [wooorm/iso-639-3](https://github.com/wooorm/iso-639-3) — ISO 639-3 codes
- [wooorm/iso-15924](https://github.com/wooorm/iso-15924) — ISO 15924 codes
- [wooorm/un-m49](https://github.com/wooorm/un-m49) — UN M49 codes

```
<script type="module">
  import * as bcp47Match from 'https://
    esm.sh/bcp-47-match@2?bundle'
</script>
```

Use

```
import {basicFilter, extendedFilter,
  lookup} from 'bcp-47-match'

const tags = ['en-GB', 'de-CH', 'en',
  'de']

console.log(basicFilter(tags, '*')) //
  => [ 'en-GB', 'de-CH', 'en',
    'de' ]
console.log(basicFilter(tags, 'en')) //
  => [ 'en-GB', 'en' ]
console.log(basicFilter(tags, 'en-
  GB')) // => [ 'en-GB' ]
console.log(basicFilter(tags, ['en-GB',
  'en'])) // => [ 'en-GB', 'en' ]
console.log(basicFilter(tags, 'jp')) //
  => []

console.log(extendedFilter(tags,
  '*')) // => [ 'en-GB', 'de-CH',
  'en', 'de' ]
console.log(extendedFilter(tags,
  'en')) // => [ 'en-GB', 'en' ]
```

This package exports the identifier `basicFilter`, `extendedFilter`, and `lookup`. There is no default export.

API

```
console.log(extendedFilter(tags, 'en-GB' ])
console.log(extendedFilter(tags, 'en-GB' ]) // => [ 'en-GB' ]
console.log(extendedFilter(tags, 'en-GB' ]) // => [ 'en-GB' ]
console.log(lookup(tags, 'en')) // =>
'en'
console.log(lookup(tags, 'en')) // =>
console.log(lookup(tags, ['en-GB', 'en-GB' ]) // => [ 'en-GB' ]
console.log(lookup(tags, ['en-GB', 'en-GB' ]) // => [ 'en-GB' ]
console.log(lookup(tags, 'jp')) // =>
undefined
```

Lookup		zh	zh-Hans	zh-Hant
* de	de-CH	<div>?</div>	<div>?</div>	<div>?</div>
de-	de-DE	<div>?</div>	<div>?</div>	<div>?</div>
de-*	de-DE	<div>?</div>	<div>?</div>	<div>?</div>
*-	CH	<div>?</div>	<div>?</div>	<div>?</div>
Parameters				
• tags (string or Array<string>) — list of BCP 47 tags				
• ranges (string or Array<string>) — list of RFC 4647 basic ranges (but * is ignored)				
Returns				
First matching tag in tags, undefined otherwise (string?).				

Types

This package is fully typed with [TypeScript](#). It exports no additional types.

Returns

Possibly empty list of matching tags in the order they matched
(Array<string>).

lookup(tags, ranges)

👉 *Note:* See [Lookup spec](#)

Find the best language tag that matches a list of ranges.
Searches for a match between the first range and all tags, and
continues with further ranges. Returns the first match, if any.
View matching table

Lookup	* de	de - CH	de - DE	de - * - DE	* - CH
de	?	?	?	?	?
de-CH		?			?
de-CH-1996					?
de-DE			?		?
de-DE-1996					?
de-DE-x-goethe					?
de-Deva					?
de-Deva-DE					?
de-Latf-DE					?
de-Latn-DE					?
de-Latn- DE-1996					?
de-x-DE					?
en					?
en-GB					?

basicFilter(tags[, ranges='*'])

👉 *Note:* See [Basic Filtering spec](#)

Match language tags to a list of simple ranges. Searches for
matches between the first range and all tags, and continues with
further ranges. Returns a list of matching tags in the order they
matched.

View matching table

Basic Filter	* de	de - CH	de - DE	de - * - DE	* - CH
de	?	?			
de-CH	?	?	?		
de-CH-1996	?	?	?		
de-DE	?	?		?	
de-DE-1996	?	?		?	
de-DE-x-goethe	?	?		?	
de-Deva	?	?			
de-Deva-DE	?	?			
de-Latf-DE	?	?			
de-Latn-DE	?	?			
de-Latn- DE-1996	?	?			
de-x-DE	?	?			
en	?				
en-GB	?				
zh	?				
zh-Hans	?				
zh-Hant	?				

Parameters

Extended * Filter

de de-CH de-DE de-* - *-CH

- tags (string or Array<string>) — list of BCP 47 tags
- ranges (string or Array<string>, default: '*') — list of RFC 4647 [basic ranges](#) (aka, matching /[^](*|[a-z]{1,8})(-[a-z0-9]{1,8}))*\$/i)

Returns

Possibly empty list of matching tags in the order they matched (Array<string>).

extendedFilter(tags[, ranges='*'])

👉 **Note:** See [Extended Filtering spec](#)

Match language tags to a list of extended ranges. Searches for matches between the first range and all tags, and continues with further ranges.

View matching table

Extended * Filter

de de-CH de-DE de-* - *-CH

de	?	?					
de-CH	?	?	?				
de-CH-1996	?	?	?	?			
de-DE	?	?	?				
de-DE-1996	?	?	?	?			
de-DE-1996	?	?	?				

- tags (string or Array<string>) — list of BCP 47 tags
- ranges (string or Array<string>, default: '*') — list of RFC 4647 [extended ranges](#) (aka, matching /[^](*|[a-z]{1,8})(-[a-z0-9]{1,8}))*\$/i)

Parameters

de	?	?					
de-DE	?	?	?				
de-DE-1996	?	?	?	?			
de-X-DE	?	?	?				
de	?	?	?				
de-GB	?	?	?				
zh	?	?	?				
zh-Hans	?	?	?				
zh-Hant	?	?	?				