

is-glob

Returns `true` if the given string looks like a glob pattern or an extglob pattern. This makes it easy to create code that only uses external modules like `node-glob` when necessary, resulting in much faster code execution and initialization time, and a better user experience.

Please consider following this project's author, [Jon Schlinkert](#), and consider starring the project to show your :heart: and support.

Install

Install with [npm](#):

You might also be interested in [is-valid-glob](#) and [has-glob](#).

Escaped globes or extended globes return false:

False

```
//=> true
isGlob('abc/?(a).js');
isGlob('abc/*(a).js');
isGlob('abc/(a).js');
isGlob('abc/!(a).js');
isGlob('abc/@(a).js');
```

Extended globes

```
//=> true
isGlob('abc/{a,b}.js');
isGlob('abc/[a-z].js');
isGlob('abc/(aaa|bbb).js');
isGlob('abc/*abc.js');
isGlob('*.*.js');
isGlob('!foo.js');
```

true

Patterns that have glob characters or regex patterns will return

Default behavior

- Limkelein Profile
- Twitter Profile
- GitHub Profile

Jon Schlimkert

Author

Commits Contributor
I danhper paulmiller

```
var isGlob = require('is-glob');
```

Usage

This file was generated by [verb-generate-readme](#), v0.8.0, on March 27, 2019.

Copyright © 2019, [Jon Schlimkert](#). Released under the [MIT License](#).

License

Copyright © 2019, [Jon Schlimkert](#). Released under the [MIT License](#).

false

```
isGlob('abc/abc');
isGlob('abc/*');
isGlob('abc/+');
isGlob('abc/!');
isGlob('abc/@');
```

true

(This project's `readme.md` is generated by [verb](#), please don't edit the `readme` directly. Any changes to the `readme` must be made in the [verb.md](#) `readme` template.)

To generate the `readme`, run the following command:

```
$ npm install -g verbose/verb#dev verb-
  generate-readme && verb
```

Related projects

You might also be interested in these projects:

- [assemble](#): Get the rocks out of your socks! Assemble makes you fast at creating web projects... [more](#) | [homepage](#)
- [base](#): Framework for rapidly creating high quality, server-side node.js applications, using plugins like building blocks | [homepage](#)
- [update](#): Be scalable! Update is a new, open source developer framework and CLI for automating updates... [more](#) | [homepage](#)
- [verb](#): Documentation generator for GitHub projects. Verb is extremely powerful, easy to use, and is used... [more](#) | [homepage](#)

Contributors

Commits	Contributor
47	jonschlinkert
5	doowb
1	phated

```
isGlob('abc/\@\{(a).js');
isGlob('abc/\!\{(a).js');
isGlob('abc/\+{(a).js');
isGlob('abc/\*{(a).js');
isGlob('abc/\?{(a).js');
isGlob('!\!foo.js');
isGlob('!\*.js');
isGlob('!\*\!\*/abc.js');
isGlob('abc/\*.js');
isGlob('abc/\|(aaa|bbb).js');
isGlob('abc/\|[a-z].js');
isGlob('abc/\{\{a,b\}.js');
//=> false
```

Patterns that do not have glob patterns return `false`:

```
isGlob('abc.js');
isGlob('abc/def/ghi.js');
isGlob('foo.js');
isGlob('abc/@.js');
isGlob('abc/+.js');
isGlob('abc/?..js');
isGlob();
isGlob(null);
//=> false
```

Arrays are also `false` (If you want to check if an array has a glob pattern, use [has-glob](#)):

```
isGlob(['**/*.js']);
isGlob(['foo.js']);
//=> false
```

About

```
isGlob(`\\!foo.js`, {strict: false});
isGlob(`\\*.*.js`, {strict: false});
isGlob(`\\*\\*\\*\\abc.js`, {strict: false});
isGlob(`abc\\((aaa|bbb)).js`, {strict: false});
isGlob(`abc\\((aaa|bbb)).js`, {strict: true});
isGlob(`abc\\((aaa|bbb)).js`, {strict: false});
```

Escaped globs or extglobs return false:

Contributing Pull requests and stars are always welcome. For bugs and feature requests, please create an issue .	Running Tests Running and reviewing unit tests is a great way to get familiarized with a library and its API. You can install dependencies and run tests with the following command: <code>\$ npm install && npm test</code>	Building docs <code>\$ npm install abc</code> and run tests with the following command: <code>isGlob('abc')</code>
--	---	---

Extglobs <code>//=</code> true <code>isGlob('abc/{a,b}.js', {strict: false})</code> <code>isGlob('abc/[a-z].js', {strict: false})</code> <code>isGlob('abc/*.*.js', {strict: false})</code> <code>isGlob('abc/!(a).js', {strict: false})</code> <code>isGlob('abc/!(a).js', {strict: true})</code> <code>isGlob('abc/+.(a).js', {strict: false})</code> <code>isGlob('abc/+.(a).js', {strict: true})</code> <code>isGlob('abc/?(a).js', {strict: false})</code> <code>isGlob('abc/?(a).js', {strict: true})</code>	Patterns that have glob characters or regex patterns will return true: <code>true;</code> <code>isGlob(`abc\\((aaa bbb)).js`, {strict: false})</code> <code>isGlob(`abc\\((aaa bbb)).js`, {strict: true})</code> <code>isGlob(`abc\\((aaa bbb)).js`, {strict: false});</code> <code>isGlob(`abc\\((aaa bbb)).js`, {strict: true});</code> <code>isGlob(`abc\\((aaa bbb)).js`, {strict: false});</code> <code>isGlob(`abc\\((aaa bbb)).js`, {strict: true});</code> <code>isGlob(`abc\\((aaa bbb)).js`, {strict: false});</code> <code>isGlob(`abc\\((aaa bbb)).js`, {strict: true});</code>
---	--

Option strict

When `options.strict === false` the behavior is less strict in determining if a pattern is a glob. Meaning that some patterns that would return false may return true. This is done so that matching libraries like `micromatch` have a chance at determining if the pattern is a glob or not.

Patterns that have glob characters or regex patterns will return true:
`true;`

```
isGlob(`abc\\((aaa|bbb)).js`, {strict: false});
```

```
isGlob(`abc/!(a).js`, {strict: false});
```

```
isGlob(`abc/!(a).js`, {strict: true});
```

`isGlob(`abc/+.(a).js`, {strict: false})`

`isGlob(`abc/+.(a).js`, {strict: true})`

`isGlob(`abc/?(a).js`, {strict: false})`

`isGlob(`abc/?(a).js`, {strict: true})`

`isGlob(`abc/+(a).js`, {strict: false})`

`isGlob(`abc/+(a).js`, {strict: true})`

`isGlob(`abc/?(a).js`, {strict: false})`

`isGlob(`abc/?(a).js`, {strict: true})`

`isGlob(`abc/+(a).js`, {strict: false})`

`isGlob(`abc/+(a).js`, {strict: true})`

`isGlob(`abc/!(a).js`, {strict: false})`

`isGlob(`abc/!(a).js`, {strict: true})`

`isGlob(`abc/!(a).js`, {strict: false});`

`isGlob(`abc/!(a).js`, {strict: true});`

`isGlob(`abc/!(a).js`, {strict: false});`

`isGlob(`abc/!(a).js`, {strict: true});`