



npm v3.0.0 CI passing gzip size 218 B

Mitt

Tiny 200b functional event emitter / pubsub.

- **Microscopic:** weighs less than 200 bytes gzipped
- **Useful:** a wildcard `"*"` event type listens to all events
- **Familiar:** same names & ideas as [Node's EventEmitter](#)
- **Functional:** methods don't rely on `this`
- **Great Name:** somehow [mitt](#) wasn't taken

Mitt was made for the browser, but works in any JavaScript runtime. It has no dependencies and supports IE9+.

Table of Contents

- [Install](#)
- [Usage](#)
- [Examples & Demos](#)
- [API](#)
- [Contribute](#)
- [License](#)

Install

This project uses [node](#) and [npm](#). Go check them out if you don't have them locally installed.

```
$ npm install --save mitt
```

Then with a module bundler like [rollup](#) or [webpack](#), use as you would anything else:

```
// using ES6 modules
import mitt from 'mitt'

// using CommonJS modules
var mitt = require('mitt')
```

The [UMD](#) build is also available on [unpkg](#):

```
<script src="https://unpkg.com/mitt/dist/mitt.umd.js"></script>
```

You can find the library on `window.mitt` .

Usage

```
import mitt from 'mitt'

const emitter = mitt()

// listen to an event
emitter.on('foo', e => console.log('foo', e) )

// listen to all events
emitter.on('*', (type, e) => console.log(type, e) )

// fire an event
emitter.emit('foo', { a: 'b' })

// clearing all events
emitter.all.clear()

// working with handler references:
function onFoo() {}
emitter.on('foo', onFoo) // listen
emitter.off('foo', onFoo) // unlisten
```

Typescript

```
import mitt from 'mitt';
const emitter: mitt.Emitter = mitt();
```

Examples & Demos

[Preact + Mitt Codepen Demo](#)

```
<PubSub>
  <div>
    <Observe onFoo={ e => alert(e.a) } />
    <Emit type="foo" a="b" />
  </div>
</PubSub>
```

API

Table of Contents

- [mitt](#)
- [all](#)
- [on](#)
 - [Parameters](#)
- [off](#)
 - [Parameters](#)
- [emit](#)
 - [Parameters](#)

mitt

Mitt: Tiny (~200b) functional event emitter / pubsub.

Returns **Mitt**

all

A Map of event names to registered handler functions.

on

Register an event handler for the given type.

Parameters

- `type` ([string](#) | [symbol](#)) Type of event to listen for, or `"*"` for all events
- `handler` [Function](#) Function to call in response to given event

off

Remove an event handler for the given type.

Parameters

- `type` ([string](#) | [symbol](#)) Type of event to unregister `handler` from, or `"*"`
- `handler` [Function](#) Handler function to remove

emit

Invoke all handlers for the given type. If present, `"*"` handlers are invoked after type-matched handlers.

Note: Manually firing `"*"` handlers is not supported.

Parameters

- `type` ([string](#) | [symbol](#)) The event type to invoke
- `evt` **Any?** Any value (object is recommended and powerful), passed to each handler

Contribute

First off, thanks for taking the time to contribute! Now, take a moment to be sure your contributions make sense to everyone else.

Reporting Issues

Found a problem? Want a new feature? First of all see if your issue or idea has [already been reported](#). If don't, just open a [new clear and descriptive issue](#).

Submitting pull requests

Pull requests are the greatest contributions, so be sure they are focused in scope, and do avoid unrelated commits.

- Fork it!
- Clone your fork: `git clone https://github.com/<your-username>/mitt`
- Navigate to the newly cloned directory: `cd mitt`
- Create a new branch for the new feature: `git checkout -b my-new-feature`
- Install the tools necessary for development: `npm install`
- Make your changes.
- Commit your changes: `git commit -am 'Add some feature'`
- Push to the branch: `git push origin my-new-feature`
- Submit a pull request with full remarks documenting your changes.

License

[MIT License](#) © [Jason Miller](#)