# Contributing guidelines

> Note: this document is intended as a draft, it will be updated soon

## Requirements:

\* Nodejs ^8.x

\* Visual Studio Code

\* GPG Tool

- [Contributing guidelines](#contributing-guidelines)

- [Requirements:](#requirements)

- [Installing and compiling source](#installing-and-compiling-source)

- [Testing the theme](#testing-the-theme)

- [Adding new Material Theme commands](#adding-new-material-theme-commands)

- [Financial contributions](#financial-contributions)

- [Credits](#credits)

- [Contributors](#contributors)

- [Backers](#backers)

- [Sponsors](#sponsors)

## Installing and compiling source

First you will have to install node\_modules through npm or yarn

```shell

npm install

# or

yarn install

```

To compile to the source code, you have to execute the build task through visual studio code.

First you need to invoke to quick command (on MacOS `?P`, while on Linux/windows is `ctrl+p`)

then type `task build` and wait until VSC will have finished the task.

### Testing the theme

To test the theme, you will have to go to the debug section, select the \*Launch Extension\* from debug and execute it.

### Adding new Material Theme commands

Material Theme commands are located inside the `extensions/commands` directory. They are split by folders so you can find a single specific `index.ts` file inside a `command` folder. This is a patter choice: so you can add related command helpers inside that folder.

#### index.ts

The main file must return a `function`. The `function` can be `async` or `sync`.

We want to make this clear, so the `async` declaration must be present even if you want to use `then()` for chaining inside the `function`.

##### Example of `async` exported `function`

```js

export default async (): Promise<boolean> => {

...

```

##### Example of `sync` exported `function`

```js

export default (): boolean => {

...

```

#### Using external "mini" modules

We encourage you to split up your command in other small modules so they can be \*\*tested\*\* more easily and outside of the context.

#### Register the command

You must register the new command to execute it when it's called by the user.

Firstly add the command to our commands `extensions/commands/index.ts`

```js

export {default as awesomeCommand} from './awesome-command';

```

Commands are registered within the `extensions/material.theme.config.ts` file.

Just add a new registration after the others that will run your command:

```js

Commands.registerCommand('materialTheme.awesomeCommand', () => ThemeCommands.awesomeCommand());

```

#### Add to package.json

When you are ready to test your new command you must add its declaration within the `package.json`. The command \*\*must be prefixed with\*\* `materialTheme.` prefix.

As the others your command will look like this:

Level: `contribute.commands -> Array`

```json

{

"command": "materialTheme.awesomeCommand",

"title": "Awesome Title For Command",

"category": "? Material Theme"

}

```

#### Build and run

Now you can re-build the theme and just run the debug, you will see your command in the command list of VSCode just searching for its title.

#### Test your command

A test suite is currently in developing. Update soon.

### Adding new custom setting

Soon(ish)®

## Financial contributions

We also welcome financial contributions in full transparency on our [open collective](https://opencollective.com/vsc-material-theme).

Anyone can file an expense. If the expense makes sense for the development of the community, it will be "merged" in the ledger of our open collective by the core contributors and the person who filed the expense will be reimbursed.

## Credits

### Contributors

Thank you to all the people who have already contributed to vsc-material-theme!

<a href="graphs/contributors"><img src="https://opencollective.com/vsc-material-theme/contributors.svg?width=890" /></a>

### Backers

Thank you to all our backers! [[Become a backer](https://opencollective.com/vsc-material-theme#backer)]

<a href="https://opencollective.com/vsc-material-theme#backers" target="\_blank"><img src="https://opencollective.com/vsc-material-theme/backers.svg?width=890"></a>

### Sponsors

Thank you to all our sponsors! (please ask your company to also support this open source project by [becoming a sponsor](https://opencollective.com/vsc-material-theme#sponsor))

<a href="https://opencollective.com/vsc-material-theme/sponsor/0/website" target="\_blank"><img src="https://opencollective.com/vsc-material-theme/sponsor/0/avatar.svg"></a>

<a href="https://opencollective.com/vsc-material-theme/sponsor/1/website" target="\_blank"><img src="https://opencollective.com/vsc-material-theme/sponsor/1/avatar.svg"></a>

<a href="https://opencollective.com/vsc-material-theme/sponsor/2/website" target="\_blank"><img src="https://opencollective.com/vsc-material-theme/sponsor/2/avatar.svg"></a>

<a href="https://opencollective.com/vsc-material-theme/sponsor/3/website" target="\_blank"><img src="https://opencollective.com/vsc-material-theme/sponsor/3/avatar.svg"></a>

<a href="https://opencollective.com/vsc-material-theme/sponsor/4/website" target="\_blank"><img src="https://opencollective.com/vsc-material-theme/sponsor/4/avatar.svg"></a>

<a href="https://opencollective.com/vsc-material-theme/sponsor/5/website" target="\_blank"><img src="https://opencollective.com/vsc-material-theme/sponsor/5/avatar.svg"></a>

<a href="https://opencollective.com/vsc-material-theme/sponsor/6/website" target="\_blank"><img src="https://opencollective.com/vsc-material-theme/sponsor/6/avatar.svg"></a>

<a href="https://opencollective.com/vsc-material-theme/sponsor/7/website" target="\_blank"><img src="https://opencollective.com/vsc-material-theme/sponsor/7/avatar.svg"></a>

<a href="https://opencollective.com/vsc-material-theme/sponsor/8/website" target="\_blank"><img src="https://opencollective.com/vsc-material-theme/sponsor/8/avatar.svg"></a>

<a href="https://opencollective.com/vsc-material-theme/sponsor/9/website" target="\_blank"><img src="https://opencollective.com/vsc-material-theme/sponsor/9/avatar.svg"></a>