# Contribution guide

:v: Thanks for your contributions :v:

## When you contribute...

- ???Issue????????????????????????????????????????????/????????????????????Issue?????????????????????????????????????????????????????????????????????????????

- ??????????????????????????????

- ????????????????????PR?????????????????????????

- ???????????????????????

## Issues

Feature suggestions and bug reports are filed in https://github.com/syuilo/misskey/issues .

\* Please search existing issues to avoid duplication. If your issue is already filed, please add your reaction or comment to the existing one.

\* If you have multiple independent issues, please submit them separately.

## Branches

\* \*\*master\*\* branch is tracking the latest release and used for production purposes.

\* \*\*develop\*\* branch is where we work for the next release.

\* \*\*l10n\_develop\*\* branch is reserved for localization management.

## Localization (l10n)

Misskey uses [Crowdin](https://crowdin.com/project/misskey) for localization management.

You can improve our translations with your Crowdin account.

Your changes in Crowdin are automatically submitted as a PR (with the title "New Crowdin translations") to the repository.

The owner [@syuilo](https://github.com/syuilo) merges the PR into the develop branch before the next release.

If your language is not listed in Crowdin, please open an issue.

![Crowdin](https://d322cqt584bo4o.cloudfront.net/misskey/localized.svg)

## Internationalization (i18n)

Misskey uses the Vue.js plugin [Vue I18n](https://github.com/kazupon/vue-i18n).

Documentation of Vue I18n is available at http://kazupon.github.io/vue-i18n/introduction.html .

## Documentation

\* Documents for contributors are located in [`/docs`](/docs).

\* Documents for instance admins are located in [`/docs`](/docs).

\* Documents for end users are located in [`/src/docs`](/src/docs).

## Test

\* Test codes are located in [`/test`](/test).

## Continuous integration

Misskey uses CircleCI for executing automated tests.

Configuration files are located in [`/.circleci`](/.circleci).

## Adding MisskeyRoom items

\* Use English for material, object and texture names.

\* Use meter for unit of length.

\* Your PR should include all source files (e.g. `.png`, `.blend`) of your models (for later editing).

\* Your PR must include the glTF binary files (`.glb`) of your models.

\* Add a locale key `room.furnitures.YOUR\_ITEM` at [`/locales/ja-JP.yml`](/locales/ja-JP.yml).

\* Add a furniture definition at [`/src/client/app/common/scripts/room/furnitures.json5`](/src/client/app/common/scripts/room/furnitures.json5).

If you have no experience on 3D modeling, we suggest to use the free 3DCG software [Blender](https://www.blender.org/).

You can find information on glTF 2.0 at [glTF 2.0 — Blender Manual]( https://docs.blender.org/manual/en/dev/addons/io\_scene\_gltf2.html).

## FAQ

### How to resolve conflictions occurred at yarn.lock?

Just execute `yarn` to fix it.

## Glossary

### AP

Stands for \_\*\*A\*\*ctivity\*\*P\*\*ub\_.

### MFM

Stands for \_\*\*M\*\*isskey \*\*F\*\*lavored \*\*M\*\*arkdown\_.

### Mk

Stands for \_\*\*M\*\*iss\*\*k\*\*ey\_.

### SW

Stands for \_\*\*S\*\*ervice\*\*W\*\*orker\_.

### Nyaize

Convert ?(na) to ??(nya)

#### Denyaize

Revert Nyaize

## TypeScript Coding Style

### Do not omit semicolons

This is to avoid Automatic Semicolon Insertion (ASI) hazard.

Ref:

\* https://www.ecma-international.org/ecma-262/#sec-automatic-semicolon-insertion

\* https://github.com/tc39/ecma262/pull/1062

### Do not omit curly brackets

Bad:

``` ts

if (foo)

bar;

else

baz;

```

Good:

``` ts

if (foo) {

bar;

} else {

baz;

}

```

As a special case, you can omit the curly brackets if

\* the body of the `if`-statement have only one statement and,

\* the `if`-statement does not have `else`-clause.

Good:

``` ts

if (foo) bar;

```

Make sure that the condition and the body statement are on the same line.

### Do not use `==` when it can simply be replaced with `===`.

?

### Use only boolean (or null related) values in the condition of an `if`-statement.

Bad:

``` ts

if (foo.length)

```

Good:

``` ts

if (foo.length > 0)

```

### Do not use `export default`

This is because the current language support does not work well with `export default`.

Ref:

\* https://basarat.gitbooks.io/typescript/docs/tips/defaultIsBad.html

\* https://gfx.hatenablog.com/entry/2017/11/24/135343

Bad:

``` ts

export default function(foo: string): string {

```

Good:

``` ts

export function something(foo: string): string {

```

## Directory structure

```

src ... Source code

@types ... Type definitions

prelude ... Independence utils for coding JavaScript without side effects

misc ... Independence utils for Misskey without side effects

service ... Common functions with side effects

queue ... Job queues and Jobs

server ... Web Server

client ... Client

mfm ... MFM

test ... Test code

```

## Notes

### placeholder

SQL????????????????????????????????????

???

``` ts

query.andWhere(new Brackets(qb => {

for (const type of ps.fileType) {

qb.orWhere(`:type = ANY(note.attachedFileTypes)`, { type: type });

}

}));

```

??????????`type`???????????????????????????

???????????????

```ts

query.andWhere(new Brackets(qb => {

for (const type of ps.fileType) {

const i = ps.fileType.indexOf(type);

qb.orWhere(`:type${i} = ANY(note.attachedFileTypes)`, { [`type${i}`]: type });

}

}));

```

### Not `null` in TypeORM

```ts

const foo = await Foos.findOne({

bar: Not(null)

});

```

???????(`bar`?`null`????)????????????

????????:

```ts

const foo = await Foos.findOne({

bar: Not(IsNull())

});

```

### `null` in SQL

SQL?????????????`null`????????????SQL??????????????

???

``` ts

query.where('file.folderId = :folderId', { folderId: ps.folderId });

```

???????`ps.folderId`?`null`??????`file.folderId = null`???????????????????????SQL??????????????????

???????????????

``` ts

if (ps.folderId) {

query.where('file.folderId = :folderId', { folderId: ps.folderId });

} else {

query.where('file.folderId IS NULL');

}

```

### `[]` in SQL

SQL???????`IN`???????`[]`(????)????????????SQL??????????????

???

``` ts

const users = await Users.find({

id: In(userIds)

});

```

???????`userIds`?`[]`??????`user.id IN ()`???????????????????????SQL??????????????????

???????????????

``` ts

const users = userIds.length > 0 ? await Users.find({

id: In(userIds)

}) : [];

```

### ????????? in SQL

SQL????????????\*\*1???\*\*?

`[a, b, c]`? `a`??????????`[0]`????`[1]`???

### `undefined`????

MongoDB???????findOne?????????????????????????? \*\*`undefined`\*\* ???????????

MongoDB?`null`???????????????`if (x === null)`?????????????`if (x == null)`????????

### ???`undefined`????

???????????????????????????????(??)??`undefined`???????????`undefined`????????TypeScript???????

????????????????????????`undefined`???????????????`ensure`?????????????????????

????

``` ts

const user = await Users.findOne(userId);

// ????? user ??? User | undefined

if (user == null) {

throw 'missing user';

}

// ????? user ??? User

```

??????`ensure`????

``` ts

const user = await Users.findOne(userId).then(ensure);

// ????? user ??? User

```

??????????

????`ensure`???????????????????????????`undefined`??????Promise?Reject????????????????

``` ts

const user = await Users.findOne(userId).then(ensure);

// ??? Users.findOne ???? undefined ?????ensure ???????????

// ?????????????

// ????.then(ensure) ?

// if (user == null) {

// throw 'missing user';

// }

// ?????????????

```

### Migration????

```

npx ts-node ./node\_modules/typeorm/cli.js migration:generate -n ?????

```

???????????????????????????????

## ???

### HTML?????? follow ??????????

??????????????????