# Contributing to obs-websocket

## Translating obs-websocket to your language

Localization happens on [Crowdin](https://crowdin.com/project/obs-websocket)

## Branches

\*\*Development happens on `4.x-current`\*\*

## Writing code for obs-websocket

### Code Formatting Guidelines

\* Function and variable names: snake\_case for C names, camelCase for C++ method names

\* Request and Event names should use MixedCaps names

\* Request and Event json properties should use camelCase. For more detailed info on property naming, see [Google's JSON Style Guide](https://google.github.io/styleguide/jsoncstyleguide.xml)

\* Code is indented with Tabs. Assume they are 8 columns wide

\* 80 columns max code width. (Docs can be larger)

\* New and updated requests/events must always come with accompanying documentation comments (see existing protocol elements for examples).

These are required to automatically generate the [protocol specification document](docs/generated/protocol.md).

### Code Best-Practices

\* Favor return-early code and avoid wrapping huge portions of code in conditionals. As an example, this:

```cpp

if (success) {

return req->SendOKResponse();

} else {

return req->SendErrorResponse("something went wrong");

}

```

is better like this:

```cpp

if (!success) {

return req->SendErrorResponse("something went wrong");

}

return req->SendOKResponse();

```

\* Some example common response/request property names are:

\* `sceneName` - The name of a scene

\* `sourceName` - The name of a source

\* `fromScene` - From a scene - scene name

### Commit Guidelines

\* Commits follow the 50/72 standard:

\* 50 characters max for the commit title (excluding scope name)

\* One empty line after the title

\* Description wrapped to 72 columns max width per line.

\* Commit titles:

\* Use present tense

\* Prefix the title with a "scope" name

\* e.g: "CI: fix wrong behaviour when packaging for OS X"

\* Typical scopes: CI, General, Requests, Events, Server

\*\*Example commit:\*\*

```

Requests: Add GetTransitionPosition

Adds a new request called `GetTransitionPosition` which gets the current

transition's state from 0.0f to 1.0f. Works with both auto and manual

transitions.

```

### Pull Requests

\* Pull Requests must never be based off your fork's main branch (in this case, `4.x-current`).

\* Start your work in a newly named branch based on the upstream main one (e.g.: `feature/cool-new-feature`, `bugfix/fix-palakis-mistakes`, ...)

\* Only open a pull request if you are ready to show off your work.

\* If your work is not done yet, but for any reason you need to PR it (like collecting discussions, testing with CI, getting testers),

create it as a Draft Pull Request (open the little arrow menu next to the "Create pull request" button, then select "Create draft pull request").