

Contribution Guideline

Code Style

Use a linter along with an auto formatter to have consistent code style in the project.

Typescript

Use <https://eslint.org/> and <https://prettier.io/> to autoformat the code based on `.eslintrc.js` configuration file.

▼ setup ESLint

JetBrains IDE's

1. Go to Preferences > Languages & Frameworks > JavaScript > Code Quality Tools > ESLint
2. Select "Manual ESLint configuration"
 - a. Select `/frontend/node_modules/eslint` as ESLint package (should be suggested in dropdown)
 - b. Select `/frontend/.eslintrc.js` as configuration file (should also be suggested)
3. Check "Run eslint --fix on save" checkbox to autoformat when saving the file

Visual Studio Code

1. Install [ESLint Plugin](#)
2. use the command palette to open **Preferences: Open Workspace Settings (JSON)** and add following config to it to format on save.

```
{
  "editor.codeActionsOnSave": {
    "source.fixAll": true
  },
  "eslint.validate": ["javascript"]
}
```

Code Documentation

Comments

Document functions using JSDoc Format (across different languages)

Only document single lines of code if others can't understand what the function does with the given description.

```
/**
 * @author Simon Morgenstern
 * @summary short summary of function (if needed)
 * @description what does my function do
 * @param {Type} Name - Description
 * @returns {Type} Description
 */
function doSomething(x: number, y: String) {...}
```

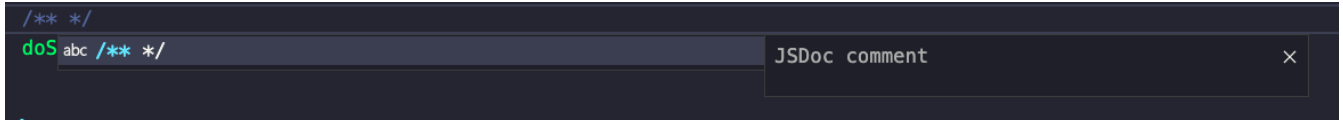
▼ JSDoc shortcut

Jetbrains IDE's

```
/**doSomething(x: number, y: string) {  
  
}
```

Add `/**` in front of your function definition (as shown above) then hit enter.

Visual Studio Code



Add `/**` above your function (as shown above) then hit enter.

Diagrams & external documentation

Does your code add a new module, new layer of functionality? Does your change affect the architecture of the code? Create / Edit a suitable UML Diagram.

Use the [Visual Studio Code - UML extension](#) to create diagrams in VS Code.

We try to document everything else directly in the `README.md`.

Git

Branch Naming: SKIOS-<TicketNumber>-title-of-ticket

-> Branches can be linked to Jira Tickets.

Pull requests

1. Pull current master branch, rebase your branch onto the master branch (resolve conflicts if necessary)
2. Create Pull Request
 - a. service lead must be optional reviewer
 - b. add used resources (if 3rd party code was used)
 - c. add and fill out the checklist

```
- [x] added test(s), or explain why one is not needed  
/possible...  
- [ ] added comments to all the functions  
- [ ] added documentation to latex project  
- [ ] code can't be split up into smaller logical units  
- [ ] code is formatted and linted
```

3. Move ticket to "Needs Review"

squash commits when merging branch. Keep commit messages if they make sense 🤔.

Code review

- Check if tests are running fine
- Check for code structure and variable names (can you understand what the code does by looking at the code and comments?)
- Write comments directly in the pull request
- Merge code after approving PR (**service lead is responsible for merging!**)