

Documentation style guide

- [Sentences and grammar](#)
- [Word choice](#)
- [Headings and capitalization](#)
- [Technical terms and abbreviations](#)
- [Code formatting](#)
- [Confluence formatting](#)
- [Font conventions](#)

This guide introduces documentation conventions applicable to the Kaa project. The intent of this guide is to unify the look and feel of the documentation, make navigation predictable and usage simple.

Sentences and grammar

- Use consistent terminology, so that every term means the same thing and is spelled in the same way throughout entire documentation. If not properly introduced, synonyms can confuse the reader.
- Give preference to present tense over past and future tenses; give preference to simple verb forms over perfect and continuous forms.
- Use numbered lists for step-by-step procedures and bulleted list for an unordered set of items. Capitalize the first word of each list entry.
- Use parallel sentence structures in bulleted and numbered lists.
- End each entry of the list with a period if all the entries are complete sentences or if they are a mixture of fragments and complete sentences. If all entries are fragments, do not end them with periods or any other punctuation mark.
- Use imperative mood in procedures, that is, formulate your instructions as giving commands to the user.
- Avoid inventing new words or assigning words an unusual meaning.

Word choice

- Avoid using *please* in instructions unless some step in the procedure causes inconvenience for the user or represents a workaround for some system limitation.

Headings and capitalization

- Use **Heading 2** as the first level heading.
- Use **Heading 4** as the last level heading.
- Capitalize only the first word in titles and headings.
- Do not use capitalization for no apparent reason. Use lowercase unless uppercase is justified.

Technical terms and abbreviations

- Spell out a technical abbreviation on its first mention on the page.
- Ensure that an important technical term or abbreviation is included in [Glossary](#).
- Use *italic* formatting to emphasize a new technical term on its first mention.

Code formatting

- Format all JSON files with [this online tool](#), "4 space tab" profile.
- Enable line numbers for all code examples which are more than 3 lines long.
- When a code example is available in several programming languages (as with SDK usage examples), represent the alternatives using [a tabbed container](#) with tab names presenting the language name ("Java", "C++", etc.). Leave the code example within the container without a heading.
- Enable the syntax highlight for the language of the code example, whenever available.
- Start the schema namespace, which we use for examples and documentation purposes, with the `org.kaaproject.kaa.schema.sample` prefix.
- Use monospace font style for the inline code in documentation (e.g. `kaa.start()`).

Confluence formatting

- Confluence does not automatically create anchors for the headings in the text. It is up to you to create such anchors for the sections that you anticipate anchor links to. It is advised that the anchor names are identical to the section names.
- Add the table of contents to the top of every page having two or more headings.
- For every documentation page, include an introduction section that immediately follows the table of contents (or top of the page for pages with no TOC). In the introduction, explain the purpose of the page and what the reader can learn from it.
- For every documentation page, create a copyright section at the bottom by clicking **Page layout** and then **Add section** on the menu. The copyright section must have a horizontal delimiting line and the following text: "Copyright © 2014[-<current year>], [CyberVision, Inc.](#)".
- Links are good! Think of the documentation modularity similarly to how you think about the code modularity. Rather than explaining the same topic multiple times in multiple locations, modularize and use links to fragment identifiers when applicable.

Font conventions

- Use **bold** for the titles of windows and dialog boxes; for the names of commands, attributes, constants, methods, fields, predefined classes, databases, events, UI elements; for the user input.
- Use *italic* for terms on first mention and emphasis; for placeholders and parameter names; for section names.
- Use regular font for other purposes, like the names of user-defined classes, folders, files and file extensions; for key names and combinations; for strings (enclose in quotation marks) and values; for environment variables and error message names; for the names of programs and utilities.
- Use links for page names.
- Optionally, use `monospace` for parameter names, folder paths, and values, in case this font will make the text easier to read.