SUB RDD Technical Reference

Date of last change:
5. Juni 2018

1 Purpose

This guideline should help you getting started a new software development project (or improving an existing one!) in the Research and Development Department of the Göttingen State and University Library.

Our goal is to establish better software quality by following standards the developer team has mutually agreed upon. Roughly basing on the DARIAH Technical Reference, these standards are discussed, worked out, and decided in the Software Quality Working Group, which meets biweekly on Tuesdays at 12:30-13:30. However, they aren't cast in stone, so in case you have a good idea for a better standard, feel free to contribute!

2 Status

This document is a living document and will be extended as soon as the Software Quality Working Group has agreed upon a new standard for software projects in RDD.

3 Guidelines

3.1 Do you stick to our code style guides?

3.1.1 General

The basic definitions are given by our EditorConfig, i.e. unix line breaks and 2 space indentation.

Unfortunately, not all editors support EditorConfig. In case you use **eXide**, the IDE that comes with exist-db, you can set 2 space indentation as default by editing `/db/apps/eXide/src/preferences.js`.

3.1.2 Specific for programming languagues

For the more prominent programming languages we have formatting and general style guides we ask you to follow:

- **Java**: The Java style guide can be found here. It's based on the Google style guide for Java with some minor RDD specific setting. You can configure Eclipse to use it automatically at *Eclipse > Preferences > Java > Code Style > Formatter*. Just load the RDD Eclipse Java Google Style in the formatter preferences and use it in your RDD projects.
- **JavaScript**: For JS we use the Airbnb JavaScript Style Guide. @TODO: How to use in editor?
- **HTML/CSS**: For HTML/CSS we agreed upon the Google HTML/CSS Style Guide. @TODO: How to use in editor?
- **XQuery**: We use the xqdoc style guide with the following addenda:
  - use double quotes instead of single quotes
- **XSLT**: Since there is no official style guide for XSLT, we decided to write our own, resulting from common best practices and own experiences within the department.
- **SPARQL**: For SPARQL there is not really any official style guide and there is no possibility to simply include any code style automatically using a code style file. We just collect some advices how to format and use SPARQL code.
  - Declaration of variables should start with a **?** (and not with a **$**).
  - **{** paranthesis should be at the end of the line. @TODO: examples
  - Group concatenations in SELECT command should be in seperate lines.
  - @TODO: more

3.2 Is your software fully documented?

3.2.1 General issues

- don't document computer language's interna
- best use languagee structure to document
- write the best documentation you can

- where to document the code? where is it documented in RDD?
- documentation and variable language is American English
- should be as code-near as possible
- exit strategies!
- TODO fugu: see `https://wiki.de.dariah.eu/pages/viewpage.action?pageId=64957922`
- code quality level for RDD

3.2.2 Developer Documentation

- API documentation
  - used parameters, author and since annotations
  - links to callers? who is calling this method, and when?
  - see Dennis' LABSUBBLOG entry

3.2.3 Admin Documentation

- how to install the software, how to run and/or restart it, how to test the installation, ...
- server documentation

3.2.4 User Documentation

- how to use the software and APIs, FAQs, walkthroughs, ...

3.3 Which version control do you use? You do use version control, do you?

3.4 What is your test coverage?

3.5 Code building and continuous integration

4 Helpful links and references

- DARIAH Technical Reference: `https://dariah-eric.github.io/technical-reference`
- DHTech – An international grass-roots community of Digital Humanities software engineers: `https://dh-tech.github.io`
- The Software Sustainability Institute, Guidelines and publicartions: `https://www.software.ac.uk`

- The Joel Test: 12 Steps to Better Code: https://www.joelonsoftware.com/2000/08/09/the-joel-test-12-steps-to-better-code
- Software Quality Guidelines: https://github.com/CLARIAH/software-quality-guideli
- Software Testing Levels: http://softwaretestingfundamentals.com/software-testing-levels