Coding Standards

- Indentation: 3-spaces, no tabs
- maximum 120-column line length
- jclouds uses Checkstyle to enforce a consistent coding style. Configure your IDE or run mvn checkstyle:checkstyle --quiet -Dcheckstyle.output.file=/dev/stdout -Dcheckstyle.output.format=plain before submitting pull requests.
- You can also see the Checkstyle results of your pull requests by clicking on the link to the DEV@cloud build in the comments (look for the cloudbees-pull-request-builder)
- Both Eclipse and Idea support Checkstyle integration.
- jclouds Checkstyle does not enforce all coding styles that should be followed in pull requests. However, developers can use this jclouds profile (compatible for import in both Eclipse and Idea): eclipse-code-formatter.xml; Also when using Idea ensure that .* imports are not used (under code style -> imports).
- Java language level: jclouds targets Java language level 6 and thus cannot use newer language features or API additions in Java 7. Note that jclouds Maven configuration prevents incorrect use of newer language features but not newer APIs (yet). The only exception to this rule is the filesystem api, which is configured by Maven to use Java language level 7.
- Guava: jclouds extensively uses Guava to provide additional functionality and consistent abstractions as opposed to other libraries, e.g., Apache Commons. That said, Guava usage should be limited in public apis, and not used in domain/value classes. This is to avoid runtime incompatibilities and confusion between types defined in Guava and Java 8.