How to add a new validator to *linteddata*

Contents

1 General architecture

- image of the class diagram
- description of the three main groups
  - check → extend
  - report → no changes
  - execution → only add the new check
- describe different severity levels

2 Check types

- describe the different types of test
- SPARQL checks might be implemented more efficient as a different type, but should be implemented as them → can be used in other frameworks
- file level
  - errors that get lost when the file is parsed
  - manual processing of the file
- multi graph level
  - file parsed as dataset
  - information extracted from all models
- graph level
  - executed on default model and all named models
  - check always performs on only one model at the same time → no information about other models at this point

- SPARQL level
  - executed on a single model
  - query = string → attribute
  - `execute` → only processing of the query result into failures

3 Implementation of a new validator

1. choose a corresponding level
2. implementation of the constructor
   - description of the attributes
   - choose severity
3. implementation of the `execute`
   - how to customise `failureDescription`
   -
4. test check
   - JUnit Test
   - test the 'lowest' `execute` function
5. add test to the list of all checks
   - add check in `Runner.createAllChecks()`

4 Example

- documentation of creating a check