

Software Architecture Description, Deliverables

Week 1

Diagram of runtime Components:

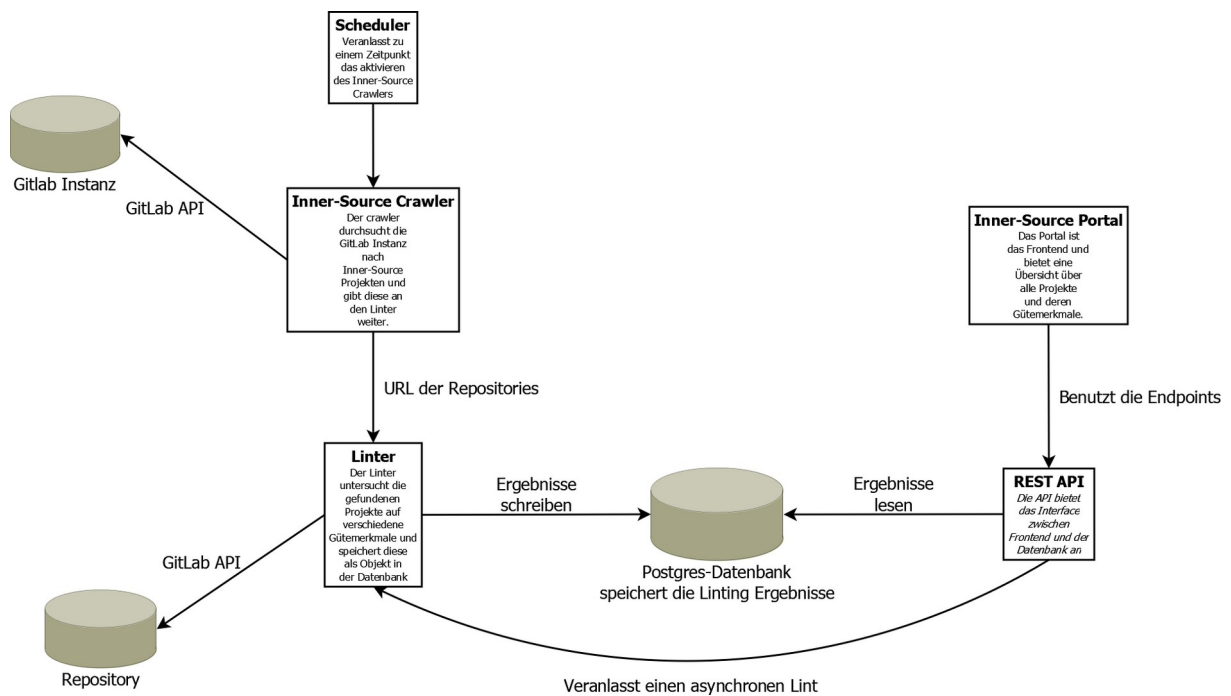


Figure 1 Diagram of runtime Components

Diagrams of code components

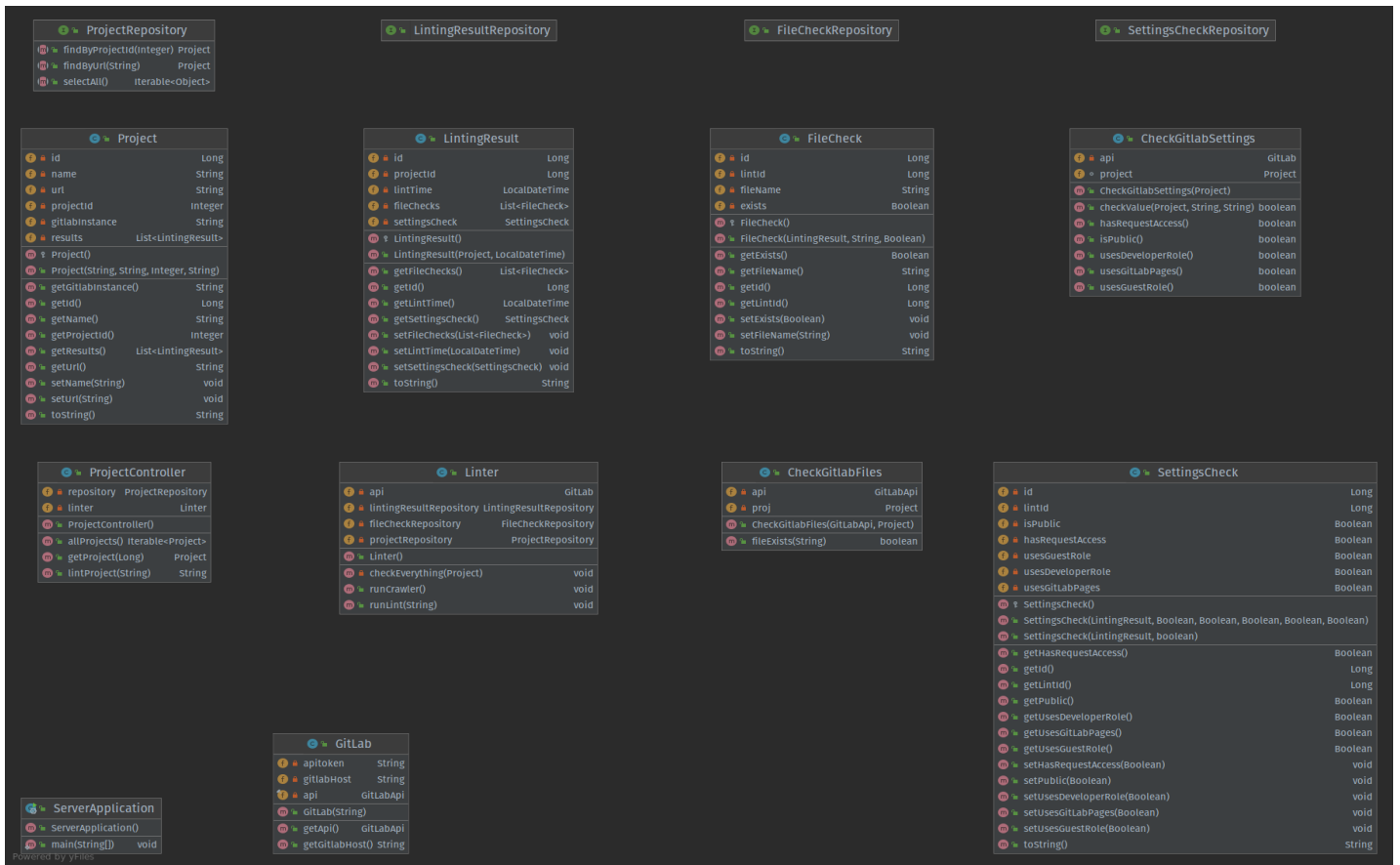


Figure 2 UML diagram

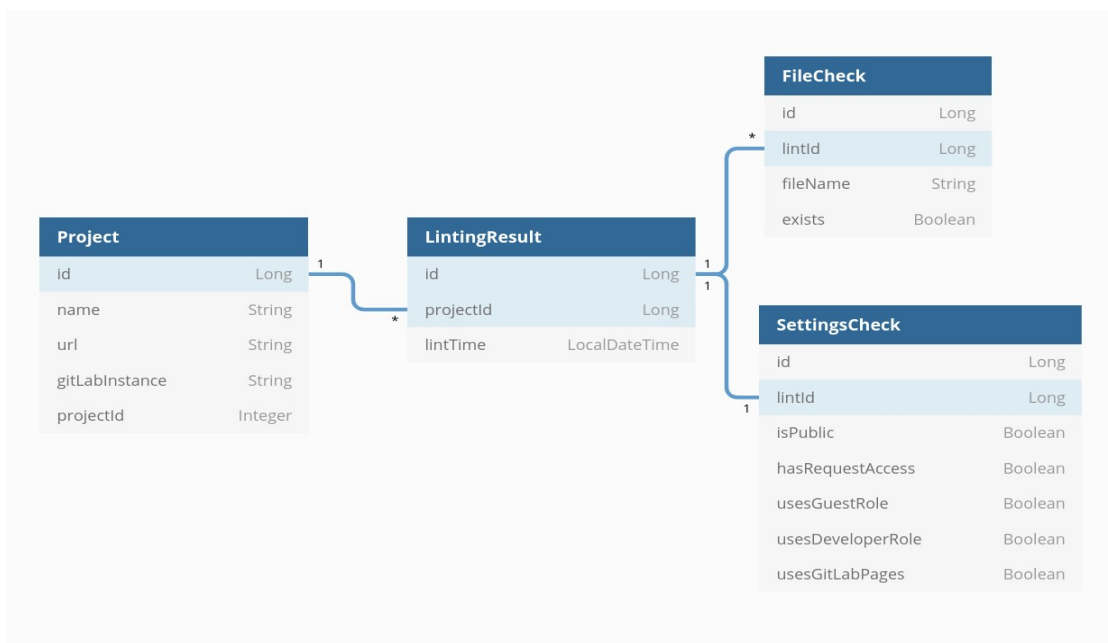


Figure 3 Database ER diagram

Summary of underlying technology stack:

| \ | Context | Name | Version | License | Comment |
|----|----------------------|-----------------------|---------|--|------------------------|
| 1 | Frontend Framework | Angular | 11.2.11 | MIT | |
| 2 | Backend Framework | Spring (Boot) | 2.4.5 | Apache License 2.0 | |
| 3 | Backend Database | Postgres SQL | 12 | PostgreSQL Licence | Similar to MIT |
| 4 | Backend runtime | Docker | 20.10.6 | Apache License 2.0 | |
| 5 | Backend orchestrator | Docker-Compose | 1.29.1 | Apache License 2.0 | |
| 6 | Backend Framework | GitLab4J-API | 4.8.0 | MIT | |
| 7 | Backend testing | JUnit | 5 | Eclipse Public License | |
| 8 | Backend scheduler | awaitility | 3.1.2 | Apache License 2.0 | |
| 9 | Testing Database | H2 in-memory database | 1.0.200 | Eclipse Public License or Mozilla Public License 2.0 | |
| 10 | Http Testing | HttpClient | 4.5.13 | Apache License | |
| 11 | Java runtime | OpenJDK | 11.0.11 | GNU GPL v2 | with linking exception |

Figure 4 Technology Stack

Description:

Angular is used to create the front-end for application.

Spring (Boot) is used to create a stand-alone web application using the embedded Tomcat server.

PostgreSQL is used as relational database management system with SQL compliance and transactions with ACID properties.

Docker and Docker-Compose are used for porting the application to different platforms.

The GitLab4J-API supports the usage of the GitLab-API inside Java.

Awaitility is used as a scheduler to run a job every day.

JUnit, H2 and HttpClient are used to test the back-end, the database and the front-end.

Java is used as the programming language and Java 11 is the runtime environment.

Descriptions:

Figure 1: Diagram of runtime components

The inner-source portal displays a website built via angular, which takes an URL as an input and sends it to the REST-API. Either there is already an entry in the database then the results are returned or the URL gets sent to the linter where it is checked if a repository exists or not. If so then the linter extracts required information on quality criteria and saves it the database. The database consists of a list of inner-source repositories, where each entry has a number of linting results with their corresponding timestamps. This creates the possibility to display the development of a repository over time.

There is also a total check of all inner-source repositories under a certain GitLab instance. For this task the scheduler runs each day at a certain timestamp and starts the crawler, which extracts a list of all inner-source repositories and hands them over to the linter. The linter then creates a linting result which is saved to the database with a corresponding timestamp.

Figure 2: UML diagram

The ServerApplication starts the SpringBoot App.

The Project class contains all information about a repository, which is stored in the database. The information itself is saved via linting results (see figure 2).

The LintingResult class is used as container for all needed information, e.g. time stamps, file checks and setting checks (see figure 2).

The FileCheck class currently only stores the existence of certain required files. A linting result can have multiple FileCheck objects (see figure 2).

The SettingsCheck class stores the current settings, like visibility, allowance of forking, etc., of a repository at a timestamp specified in the LintingResult (see figure 2).

The Linter class is used for extracting all required information via the GitLab API for a given repository. The method checkEverything() then creates a LintingResult object where all information is saved. The LintingResult object is then returned.

The LintingResult class stores the references on all information of the executed checks.

The classes CheckGitlabSettings and CheckGitlabFiles are being called in checkEverything() and are used to perform needed checks.

The GitLab class is the entry point of the gitlab4j library.

The ProjectController governs the access for the ProjectRepository and is used by the API to get all projects, a project by id or add a new repository with a string.