

## Developing an Open Source Performance Benchmarking Git Bot

Proposal author / Supervisor

Philipp Leitner (Software Engineering)

### Background

Software developers increasingly use bots, small tools to automate tedious manual tasks, as part of their daily work. Bots are usually integrated with standard continuous integration pipelines, such as GitHub or GitLab, and are often configured to automatically run on every commit to the project repository. A very well-known software development bot is [Dependabot](#), which automatically keeps project dependencies updated with new releases.

Another routine task that many developers want to automate is (continuous) performance testing, that is, repeatedly running a project's benchmarks to assess if recent code changes have led to unexpected slowdowns. However, so far no standard tool for this purpose exists – instead, projects that want to use a benchmarking bot usually have to [build it themselves](#).

### Project Description

The goal of this project is to develop an open source performance benchmarking bot to address this practical gap. The bot shall integrate with GitHub, and be able to execute (at least) Java benchmarks written using the [JMH framework](#) (more generic support for other frameworks or programming languages would be desirable). The bot shall be able to (a) execute the benchmarks of the project using a well-defined existing execution environment, (b) collect and store benchmark results, (c) automatically analyze the results using robust statistical methods of performance analysis (e.g., comparison of bootstrapped confidence intervals), and (d) report back discovered performance degradations as GitHub issue or code review comment.

### Suggested Reading Material

Interested students will need to familiarize themselves with benchmarking libraries such as [JMH](#) as well as the [GitHub API](#). Further, students should read up on [bots in software development](#) (the proposed bot would be an automation-focused bot according to the classification in the linked paper, no AI or chat interface will be required).

### Target Group

This project is suitable for all students in the D, DV and IT program.

### Special Prerequisites

Interested students should have a strong background in object-oriented programming with Java, including experience with Java build tools such as Maven and Gradle. Further, some familiarity with Git and GitHub, and how these tools are used in real-life software projects, will be required. Finally, students should be able to interface with existing REST or GraphQL based Web APIs.

Prior experience in performance measurement and software performance testing is not required (but interested students will have to read up on these topics as part of the project).