

# Assignment 3 – The Article

**Title:** Selenium-Jupiter: Enhancing Automated Testing in Java

## **Abstract**

Selenium WebDriver is a library that allows controlling web browsers (e.g., Chrome, Firefox, etc.) programmatically. It provides a cross-browser programming interface in several languages used primarily to implement end-to-end tests for web applications. JUnit is a popular unit testing framework for Java. Its latest version (i.e., JUnit5) provides a programming and extension model called Jupiter. This paper presents Selenium-Jupiter, an open-source JUnit 5 extension for Selenium WebDriver. Selenium-Jupiter aims to ease the development of Selenium WebDriver tests thanks to an automated driver management process implemented in conjunction with the Jupiter parameter resolution mechanism. Moreover, Selenium-Jupiter provides seamless integration with Docker, allowing the use of different web browsers in Docker containers out of the box. This feature enables cross-browser testing, load testing, and troubleshooting (e.g., configurable session recordings). This paper presents an example case in which Selenium-Jupiter is used to evaluate the performance of video conferencing systems based on WebRTC. This example case shows that Selenium-Jupiter can build and maintain the required infrastructure for complex tests effortlessly.

©2022 Elsevier Inc. All rights reserved.

---

**Introduction:** This paper delves into "Selenium-Jupiter," an extension of JUnit 5 for Selenium WebDriver, addressing the integration challenges in automated testing in Java.

**Background:** Selenium WebDriver is pivotal in web application testing, and JUnit 5 is a popular framework in Java. This paper explores their integration.

**Problem Statement:** The authors aim to simplify the configuration and management of WebDriver instances in Java testing environments.

**Methodology:** The approach involves the development of a JUnit 5 extension, enabling easier instantiation and management of WebDriver.

**Results:** The extension demonstrates efficiency in managing browser instances, contributing significantly to streamlined testing processes.

**Validity:** The paper validates the results through practical implementation scenarios, though it acknowledges certain limitations in scope.

**Future Work:** It suggests further exploration in enhancing integration with other testing frameworks and broader application contexts.

**Personal Evaluation:** The paper effectively addresses a common challenge in Java testing. Its practical applicability is commendable, though exploration of its integration with other technologies would be beneficial.

- Why? What problem are the author(s) trying to solve?

They attempt to evaluate the performance of WebRTC-based video conferencing systems using Selenium-Jupiter, verifying its capability to construct and maintain infrastructure for complex tests.

- How? What method has been used to answer the question?
- What is the result from the method?
- What is the validity of the result according to the author?
- Does the research solve the problem/answer the question?
- What are to be done next? future work? and why?
- And lastly something from you. What do you think and why? If you make claims support those with evidence or logic.