

## 5.3 Automate Aspects of Testing

As we are working with Java and Maven, test automation is naturally integrated into the build process. When Maven builds the project, all automated tests are executed, with the build failing if a test fails or errors.

This behaviour was leveraged in the CI pipeline by using GitHub Actions to build the project on every push to the master branch and on all pull requests targeting the master branch. This pipeline runs in a clean Ubuntu environment using Java 21, ensuring that test results are consistent and not influenced by local configurations. I have included a screenshot of my yaml file used to configure the GitHub Actions.

```
name: Java CI with Maven

on:
  push:
    branches: [ "master" ]
  pull_request:
    branches: [ "master" ]

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v4
    - name: Set up JDK 21
      uses: actions/setup-java@v4
      with:
        java-version: '21'
        distribution: 'temurin'
        cache: maven
    - name: Build with Maven
```

Automating testing in this way ensures that unit, integration and system level tests are always run before changes are merged. This helps prevent regressions, enforces a consistent testing discipline, and increases confidence that the software continues to meet its requirements as it evolves.