CI / CD Pipeline on GitHub

Describing the CI / CD pipeline with GitHub Actions for the frontend and services



**Group Name:** Marinda Boshoff, Michael Osuntuyi, Milan Koster van Groos, Victoria Fong

**Documented by:** Victoria Fong

**Year:** 2022

**Class:** S-A-RB-CMK 4

# Introduction

The Stress Wearables project is uploaded and stored on GitHub. GitHub Actions are designed to help simplify workflows with flexible and secure automation and offer easy-to-use CI/CD capabilities. GitHub actions have been added to each repository (frontend and backend) of the project.

This document will explain what type of GitHub action have been added and how to configure Sonar Cloud.

# Table of Contents

[Introduction 2](#_Toc125050371)

[Table of Contents 3](#_Toc125050372)

[Front-End 4](#_Toc125050373)

[Pipeline 4](#_Toc125050374)

[GitHub Actions 5](#_Toc125050375)

[Dependabot 5](#_Toc125050376)

[Back-End 6](#_Toc125050377)

[Pipeline 6](#_Toc125050378)

[GitHub Actions 8](#_Toc125050379)

[Dependabot 8](#_Toc125050380)

[Configure Sonar Cloud 9](#_Toc125050381)

# Front-End

GitHub Links to:

* [Caregiver Dashboard](https://github.com/Fontys-Stress-Wearables/Caregiver-Dashboard)
* [Organisation Dashboard](https://github.com/Fontys-Stress-Wearables/Organisation-Dashboard)

The front-end of the project consists of currently two dashboards, the Organisation Dashboard and Caregiver Dashboard.

More detail of these dashboards can be found in the reading guide under the System Architecture & Design section named “Software Architecture Document (Design folder)”

The repositories of these Dashboards have the same Ci/Cd pipeline built for consistency. Below is the pipeline explained with the added GitHub actions.

## Pipeline

The pipeline for the frontend has two workflows, one is for pull requests where we build and format the project with also running CodeQL & Sonar Cloud. The next workflow is for merge requests where the same applies but with building the Docker file.

Graphical user interface

Description automatically generated with low confidence

**Image 1: “Pull request workflow for the frontend dashboards”**

Graphical user interface, text, application

Description automatically generated

**Image 2: “Merge request workflow for the frontend dashboards”**

In the background of running these workflows is Sonar Cloud as well running as seen in image 3. To be able to configure this to another user account, this is discussed the [Configure Sonar Cloud](#_Configure_Sonar_Cloud) section.

Graphical user interface, application, Teams

Description automatically generated

**Image 3: “Sonar Cloud running on Dashboards”**

## GitHub Actions

The GitHub actions used are:

|  |  |  |
| --- | --- | --- |
| **GitHub Action** | **Description** | **Workflow (request type)** |
| Build | Build and formats the code with NPM. | Push & Merge |
| CodeQL | Finds security vulnerabilities and uploads vulnerabilities in the security tab on GitHub. | Push & Merge |
| Sonar Cloud | Detects bugs, vulnerabilities, and code smells. | Push & Merge |
| Docker | Builds Docker image. | Merge |

## Dependabot

Graphical user interface, text, application, email

Description automatically generated

**Image 4: “Dependabot running on Dashboards”**

Dependabot has been added to both repositories of the dashboards to make it easier to fix vulnerable dependencies in the repositories.

Dependabot is ran when a merge request has been made and is set to run weekly for dependencies checks.

# Back-End

GitHub Links to:

* [Organisation Service](https://github.com/Fontys-Stress-Wearables/Organisation-Service)
* [Patient Service](https://github.com/Fontys-Stress-Wearables/Patient-Service)
* [Patient Group Service](https://github.com/Fontys-Stress-Wearables/Patient-Group-Service)
* [Stress Data Service](https://github.com/Fontys-Stress-Wearables/Stress-Data-Service)
* [Feedback Service](https://github.com/Fontys-Stress-Wearables/Feedback-Service)
* [User Service](https://github.com/Fontys-Stress-Wearables/User-Service)

The Back end of the project consists of in total 6 microservices.

More detail of these dashboards can be found in the reading guide under the System Architecture & Design section named “Software Architecture Document (Design folder)”

The repositories of these microservices have the same Ci/Cd pipeline built for consistency. Below is the pipeline explained with the added GitHub actions.

## Pipeline

The pipeline for the backend has two workflows, one is for pull requests where we build & test the project with also running CodeQL & Sonar Cloud. The next workflow is for merge requests where the same applies but with building the Docker file.

Graphical user interface

Description automatically generated

**Image 5: “Pull request workflow for the frontend dashboards”**

Diagram

Description automatically generated

**Image 6: “Merge request workflow for the frontend dashboards”**

In the background of running these workflows is Sonar Cloud as well running as seen in image 7. To be able to configure this to another user account, this is discussed the [Configure Sonar Cloud](#_Configure_Sonar_Cloud) section.

Graphical user interface, application

Description automatically generated

**Image 7: “Sonar Cloud running on Dashboards”**

## GitHub Actions

The GitHub actions used are:

|  |  |  |
| --- | --- | --- |
| **GitHub Action** | **Description** | **Workflow (request type)** |
| Build | Build the services with DotNet. | Push & Merge |
| Unit Tests | Runs the Unit Tests made for services. | Push & Merge |
| CodeQL | Finds security vulnerabilities and uploads vulnerabilities in the security tab on GitHub. | Push & Merge |
| Sonar Cloud | Detects bugs, vulnerabilities, and code smells. | Push & Merge |
| Docker | Builds Docker image. | Merge |

## Dependabot

Graphical user interface, text, application, chat or text message

Description automatically generated

**Image 4: “Dependabot pull requests on Feedback-Service”**

Dependabot has been added to all the repositories of the microservices to make it easier to fix vulnerable dependencies in the repositories.

Dependabot is ran when a merge request has been made and is set to run weekly for dependencies checks.

# Configure Sonar Cloud

Currently the project is under a Fontys student account from one of the group members. In the case sonar cloud fails for not being able to run sonar cloud because of misfunctions of the account not being available a new account will have to be made.

You can use the [Getting Started Documentation](https://docs.sonarcloud.io/getting-started/github/#:~:text=If%20your%20code%20is%20on,in%20using%20your%20GitHub%20credentials.) from Sonar Cloud to change accounts.