Teknisk dokumentation team05, *Release 3*

February 28, 2022

1 Introduktion

Följande dokument är en beskrivning av den bakomliggande tekniska strukturen för ett tidtagarprogram för enduro-tävlingar. Programmet består i stort sett av tre delar: en registeringsdel (i form av ett användargränssnitt) som samlar in information, en serverapplikationer som behandlar data från användargränssnittet och ett resultprogram som sköter behandlingen av den information som samlats in och genereringen av resultatfiler.

2 Övergripande arkitektur

Programmet består av två delar. En Node serverapplikation skriven i ramverket Express.js som ansvarar för att ta emot och skicka datan från webb gränssnittet. Webb gränssnittet är skrivet i React.js som är ett populärt ramverk för att utveckla webb appar. Samt ett Java program som ansvarar för själva funktionaliteten av tidtagar programmet.

2.1 Filstruktur

2.1.1 README.md

Fil som beskriver hur man kör de olika programmen.

2.1.2 /Acceptanstester

Mapp med acceptanstester och automatiserade acceptanstester som kan köras med Python och körs automatiskt när man kör gradlew test.

2.1.3 /Web

Mapp som innehåller kod för serverapplikationen och webb gränssitet.

2.1.4 /docs

Mapp som innehåller all dokumentation.

2.1.5 /release_files

Mapp som innehåller skript och konfigurationsfiler som ska med i releasen.

2.1.6 /result

Mapp som innehåller koden för resultatprogrammet.

2.1.7 /register

Mapp som innehåller koden för den deprikerade Java swing applikationen.

2.2 Köra program i utvecklar miljö

Se till att Java, Node.js och Python är installerat.

2.2.1 Java programmet

För att köra java programmet är det bara att använda gradle eller köra ./gradlew :result:run. ./gradlew test kör alla tester samt acceptanstesterna.

2.2.2 Server och react gui

Se till att vara i mappen /Web och kör npm install. När det är färdigt kör man servern och gui:t med npm run dev, som startar både serven och gui:t och startar om de båda programmen när det görs ändringar i koden.

2.3 Bygga och starta program

För att bygga projektet följ dem relevanta kommandona för ditt operativsystem nedan.

2.3.1 Linux / Mac:

1. Stega in i rotmappen för projektet.
2. Kör kommandot ”./build.sh”.
3. En zip-fil och motsvarande unzippade katalog finns nu under mappen ”build”.
4. Stega in i mappen med det byggda projektet via ”cd build/distributions”.
5. Starta nu programmet via kommandot ”./start.sh”.
6. Du kan nu besöka hemsidan genom att klicka på länken som skrivits ut i terminalen eller genom att besöka URL:en ”localhost:PORT”, där ”PORT” är portnumret som är specifierat i config-filen, i din webbläsare. Standard portnumret är 4000.

2.3.2 Windows (Powershell & CMD):

1. Stega in i rotkatalogen för projektet.
2. Kör kommandot ".\build.bat" (notera användningen av backslash).
3. Stega in i katalogen "build" via kommandot "cd build", där finns nu en zip-fil med ett komplett bygge av projektet.
4. Om du har Windows 10 build 17063 eller nyare så kan du köra kommandot "tar -xf team05-race-clock-vX.X.zip" för att unzippa filen. Om du har en äldre Windows version eller detta inte fungerar så öppna utforskaren och unzippa filen via det grafiska gränssnittet.
5. När du har unzippat filen så stega in i den nya katalogen via kommandot "cd team05-race-clock-vX.X".
6. Starta nu programmet via kommandot ".\start.bat".
7. Du kan nu besöka hemsidan via URL:en "localhost:PORT", där "PORT" är portnumret som är specifierat i config-filen, i din webbläsare. Standard portnumret är 4000.

2.4 Applikationsbeskrivningar

2.4.1 Resultatprogram

Resultatprogrammet använder sig utav data från servern i form av filer som skapats av användarnas inmatningar i användargränssnittet. Baserat på det datat samt en konfigurationsfil skapar resultatprogrammet en resultatfil. Programmet kan formattera resultatet på ett antal olika sätt, exempelvis sorterat eller icke sorterat. För att ändra resultatprogrammets beteende använder man sig av en konfigureringsfil då man kör programmet.

Resultatprogrammet använder sig av Java, och enligt kundens önskemål ska programmet kunna köras med Java 11.

2.4.2 Server/API

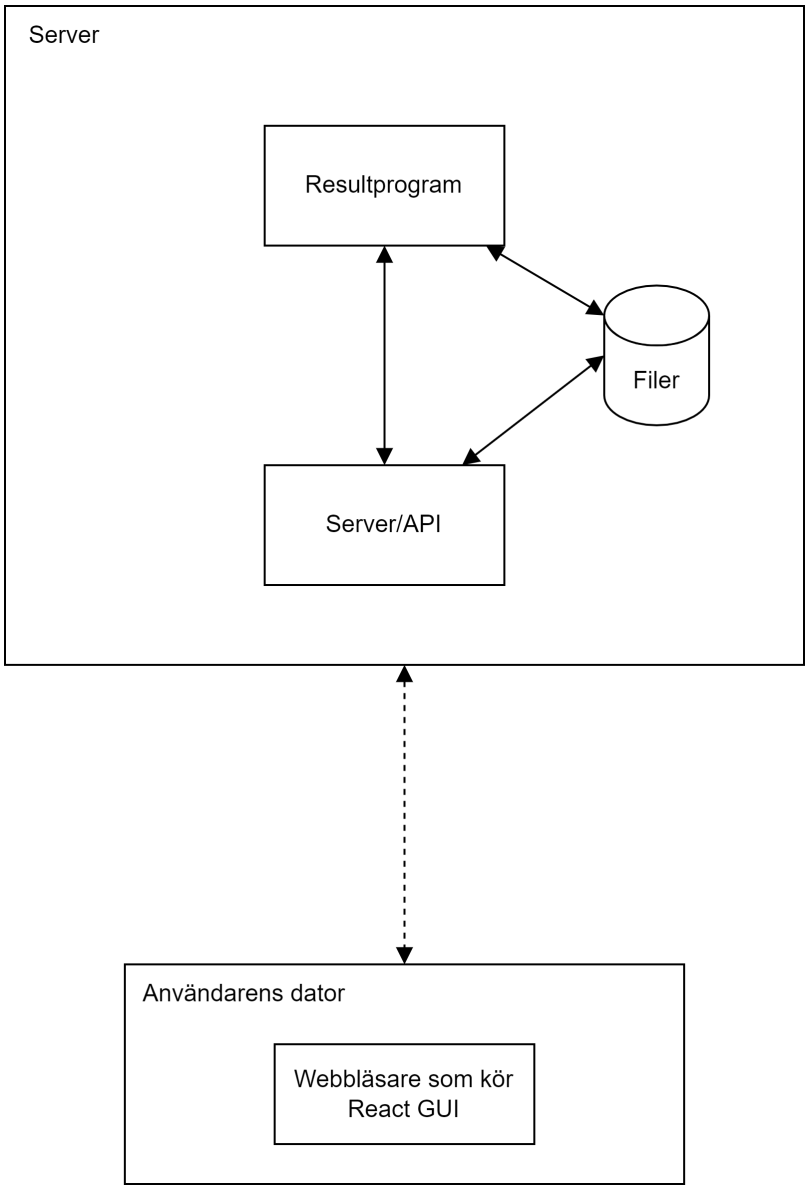
Server/API applikation skrivet i Node.js och Express.js med syftet att hosta React GUI samt sköta kommunikationen med resultatprogrammet i Java. Applikationen skapar namnfiler samt indata-filer till resultatprogrammet och hämtar ut resultfilen som resultatprogrammet genererat beroende på API-anrop. Följande api endpoints finns.

GET /result - Hämtar resultatet av tävlingen.
GET /drivers - Hämtar alla förrare som är registrerade.
POST /drivers - Registrerar en ny förare.
GET /start - Hämtar alla starttider som är registrerade.
POST /start - Registrerar ny starttid.
GET /end - Hämtar alla måltider som är registrerade.
POST /end - Registrerar ny måltid.

2.4.3 React GUI

Ett React.js web GUI för registrering av förares namn samt start-, varv- och måltider för olika startnummer.

2.5 Arkitektur



1. Användaren besöker användargränssnittet genom webbläsaren. Hemsiden läses in från Server/API.

2. Användaren matar in vilka förare som ska vara en del av loppet. Dessa skickas till Server/API som lagrar datat.

3. Användaren laddar upp en konfigurationsfil som beskriver loppet. Filen skickas till Server/API.

4. Flera användare kan sedan parallellt besöka hemsidan och registrera start-, varv- och måltider. Data lagras i Server/API i filer.

5. När tävlingen är slut kan användare generera resultatfiler via hemsidan. Anropet triggar en start av resultatprogrammet i Server/API med alla indata-filer samt konfigurationsfilen Server/API har lagrat.