

Git code management

Improving the Grand Prix experience
for F1 viewers at home

S8 Graduation FHICT

4 Sept 2023 - 16 Jan 2024

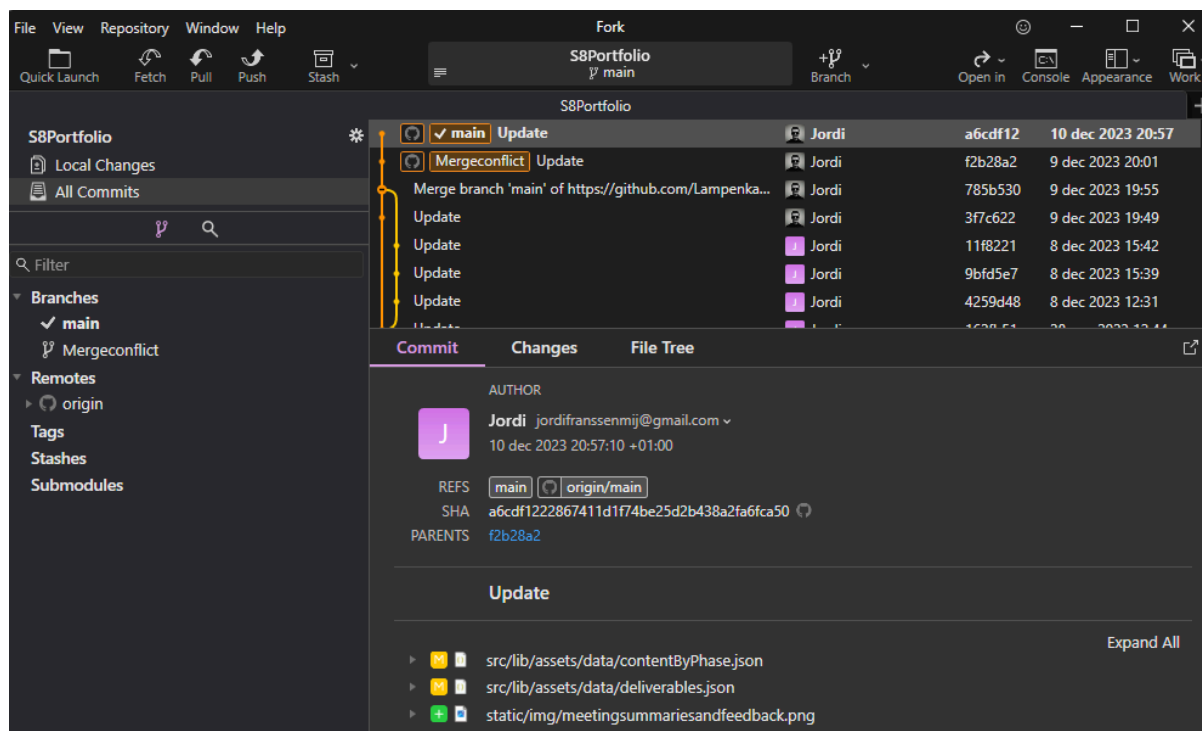
By Jordi Franssen

Introduction	3
Git Fork	3
S8 Portfolio	4
Development API	5
Team Jumbo-Visma Rebrand	5
RacingNews365-Liveblog	6
Summary	7
Learning Outcome Clarification	8

Introduction

I used Git as code management for the deliverables I've been working on during my internship. This document provides an overview of all the deliverables where I applied Git and how I applied it. I haven't used Git for all my code related deliverables. The Yuka.js POC's were relatively small and these deliverables were only used to validate if something would work. They are not used during the build of the project itself. I also used Git to synchronize my work across my laptop for working at the office, and my desktop pc when working from home.

Git Fork

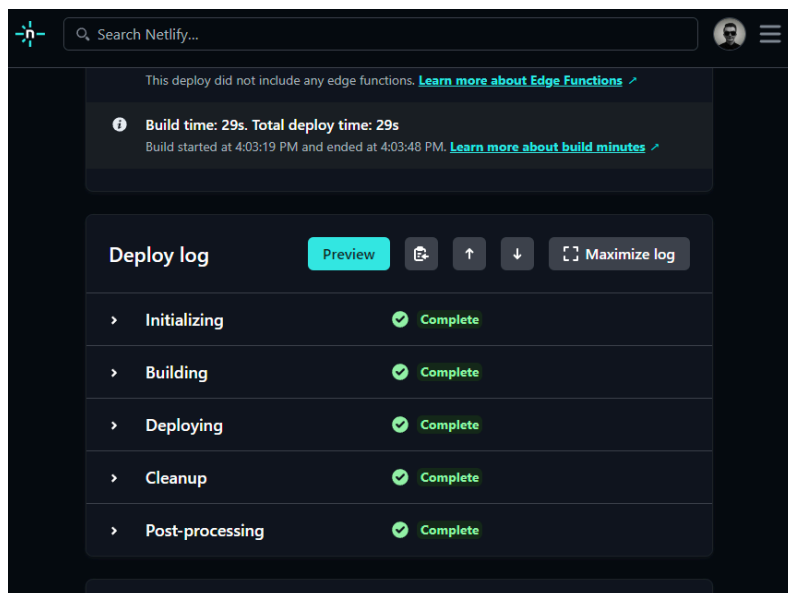


If possible, I used Git Fork as my Git UI client. I learned to use Git Fork during my first internship at E-Sites. Git Fork is still my favorite Git client as it's free, not too

complicated, but still has all the features you need. However, there was a problem using Git Fork for projects of TDE itself. To run the projects of TDE, I have to work in a Linux environment. On my Windows laptop, I use WSL. Unfortunately, Git Fork can't run inside WSL. This results in problems with EOL (End Of Line) sequences as Git Fork detects a change in the file if a different EOL sequence is used than the CLRF standard Windows uses. Therefore, I had to use the Git CLI, or the Git functionality inside VSCode, which is able to run inside WSL and is therefore able to work with the correct EOL sequence used in the project.

S8 Portfolio

The first deliverable where I used Git is my own portfolio website for this semester. For this website, I implemented continuous deployment from the Github repository to deploy my website in Netlify. This makes life easy as updates on my portfolio will be automatically deployed when I push changes to the master branch in Github. Therefore, I don't have to manually upload new files to make updates to my portfolio.



Development API

The next deliverable is the development API. This deliverable is important to the project as I have to use this API to build the new liveblog. The development API has seen some updates and changes during its use. These updates were managed in Git.

Team Jumbo-Visma Rebrand

Team Jumbo-Visma is a project of TDE. As explained in the introduction, I learned to work with Git in WSL as the projects of TDE run in a Linux environment. I found that TDE uses Git in a relatively simple way compared to E-Sites, where I did my first internship. At E-Sites, a ticket system was used for each project. Every ticket had described a task or change to the project. In Git, each ticket would have its own branch, with the ticket ID as name. Before each branch was merged with the development branch, a code review was done. This all caused a lot of complexity and resulted in problems like merge conflicts. Therefore, I highly prefer the more simple approach TDE is using for Git.

RacingNews365-Liveblog

I built the new liveblog directly in the RN365 repository. First I avoided the Git repository as I wanted to build my project completely separate from the live project by TDE. Therefore, I downloaded and extracted a version of the repository from TDE's Gitlab. My plan was to run my project in a completely separate repository. However, my company coach and I agreed to build the project in the main repository after all, and commit my changes in a separate branch. I had already made some progress in the build, but it turned out to be quite difficult to add the remote repository to my local installation I downloaded as a zip file. Therefore, I learned that it's always best to initiate a remote repository from the beginning, before you start building.

Summary

I used Git as code management for the deliverables I've been working on during my internship. I haven't used Git for all my code related deliverables. The Yuka.js POC's were relatively small and these deliverables were only used to validate if something would work. They are not used during the build of the project itself. I also used Git to synchronize my work across my laptop for working at the office, and my desktop pc when working from home.

Git Fork is still my favorite Git client. Although I wasn't able to use it in WSL for projects from TDE itself. Instead I used the Git CLI and the Git functionality in VSCode. I also used Git for continuous deployment from Github in Netlify where I host my portfolio. Next, I used Git for the development API to keep track of constant modifications. I collaborated with other colleagues in Git in the Team Jumbo-Visma rebrand side project, and of course I used Git for the build of my own project in the RN365 repository. I initially made a mistake by downloading a zip file from the repository and made it almost impossible to add a remote repository to the changes I had already made. But with some tips from John I was able to still add my project to a completely separate branch in the RN365 repository in TDE's Gitlab.

Learning Outcome Clarification

- Learning Outcome 1: Professional Duties
- Learning Outcome 2: Situation-Orientation
- Learning Outcome 6: Targeted Interaction

Learning outcome 1 applies to this deliverable because this is a professional duty on a bachelor level in the activity of Manage&Control and is in line with the IT area User Interaction.

Learning outcome 2 applies to this deliverable because I applied previously acquired knowledge about Git and adapted to the processes and way of working of the company.

Learning outcome 6 applies to this deliverable because I collaborated with colleagues in Git to work on the Team Jumbo-Visma side project.