

Checklista att tänka på för Git

Om du börjar känna dig nöjd med din kod och vill skicka upp den för granskning (dvs lägga en PR) så:

- 1) Gå alltid till main-branschen och kör **git pull** (så du läser in senaste filerna till main branschen)
- 2) Gå tillbaka till din lokala-bransch (den du jobbar i) och skriv i terminalen: **git merge main**

Sen är det bara köra git add + git commit -m "meddelande" + git push -u origin dinbransch exempelvis.

Varför gör vi detta? Under tiden du jobbat så kan uppdateringar skett på main-branschen så du vill säkerställa att du har senaste filerna och läst in detta i din lokala bransch innan du skickar PR. Då undviker vi merge-konflikter.

Jag står som kodgranskare och ska granska en PR, vad bör jag tänka på?

- 1) Gå in i Github i vårt repo under fliken "Pull requests", klicka dig in på berörd PR (du står som reviewer till höger)
- 2) Läs commit-message och description (förhoppningsvis ska det framgå tydligt där vad den berör). Gå därefter in på fliken "Files changes" så ser du allt tillägg som gjorts/även borttag i koden.
- 3) Känns inte ovan tillräckligt (?) Säg att du kanske vill testa funktionaliteten lokalt på din dator innan du godkänner den. Då kan du klona ned den lokalt (git clone), se instruktion: <https://docs.github.com/en/github/collaborating-with-pull-requests/reviewing-changes-in-pull-requests/checking-out-pull-requests-locally>

Så kan du testa av funktionaliteten innan du godkänner PR.

- 3) Om du godkänner PR är det alltid kodaren (den som lagt PR) som ska mergea in den i main.

Att tänka på vid merge (kodare), din PR är nu godkänd av granskaren och nu ska jag mergea in den i main.

Finns det inga konflikter så står det att "This branch has no conflicts with the base branch" och då är det bara mergea in den.

<https://docs.github.com/en/github/collaborating-with-pull-requests/incorporating-changes-from-a-pull-request/merging-a-pull-request>

Alternativen:

Merge pull request (mergar in alla dina commits på den branschen)

Squash and merge (du kombinerar alla dina commits till en commit när du skickar in den i main)

Rebase and merge (har jag själv inte använt men kan läsas om ovan eller fråga Fredrik vad denna innebär)

Hantera merge-konflikter

Finns det merge konflikter så måste detta lösas (du kommer tydlig se detta, då knappen är utgråad för merge och du kan exakt se vad som står i konflikt direkt i webbgränsnittet).

Man kan lösa konflikterna direkt i webbläsaren (kan vara krångligt med mycket kod) eller lösa konflikterna direkt i VS Code.

Instruktion:

<https://docs.github.com/en/github/collaborating-with-pull-requests/addressing-merge-conflicts/resolving-a-merge-conflict-on-github>

Övrigt så finns allt ni behöver veta i docs.github.com om hur ni arbetar med Git:

<https://docs.github.com/en/github/collaborating-with-pull-requests>

Finns förklaringar och instruktioner för princip allt ni undrar om ☺

Har ni några frågor om Git så kan jag försöka bistå men är själv lite ringrostig (men har hyffast bra koll).

//Robin Karlsson