

Verklegt Námskeið 1

T-113-VLN1 2023

29. Nóvember, 2024

Háskólinn í Reykjavík



Handbók GitHub og GitHub Desktop

Hópur 14

Rakel Eir Magnúsdóttir

Regína Sjöfn Sveinsdóttir

Vigdís Hervör Guðjónsdóttir

Vigdís Klara Stefánsdóttir

Tinna Sigurrós Traustadóttir

Þorbjörg Ísold Sigurjónsdóttir

Dr. Gylfi Þór Guðmundsson

Efnisyfirlit

Inngangur GitHub og GitGub Desktop

Uppsetning reikninga – Setja upp aðgang

Grunnaðgerðir: “Cloning”, “Commits”, “Push”, “Pull”

“Pull requests” og “code reviews”

“Branching” og “Merging”

“Branching strategies” eins og t.d. “Git flow”

Lausn árekstra: Útfæra einfalt dæmi um hvernig á að leysa úr sameiningarárekstrum

Algeng vandamál og lausnir

Inngangur að GitHub og GitHub Desktop

GitHub er sá staður þar sem hægt er að vinna í kóða sem einnig hægt er að vinna hann saman í teymi á einfaldan máta. Þar sem hægt er að sjá allar breytingar sem hver og einn setur inn í kóðann. Þar er haldið utan um kóðabreytingar hvers og eins.

Setja upp aðgang

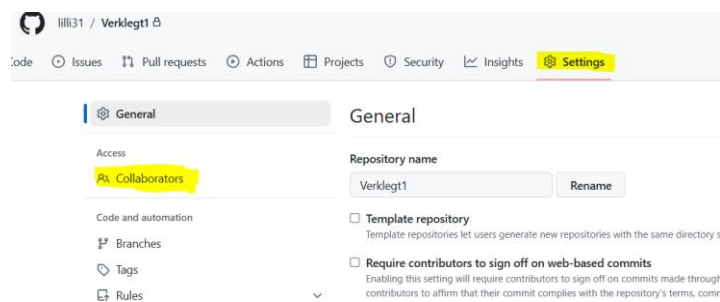
Skref 1 -

Stofna aðgang á <https://github.com/>

Einn í hópnum stofnar repository (“geymslu”) inn á github aðgangi sínum, nefnt er verkefnið Verklegt 1.

Sá sem býr til Repository (geymsluna) þarf að bæta við aðgang fyrir teymið sitt.

Þar er farið inn á *settings* og þaðan inn á *collaborations* sem hægt er að bæta hinum við.

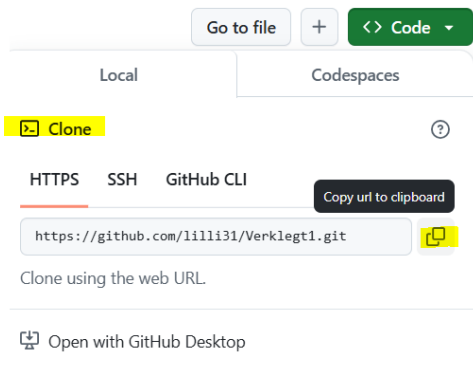


Skref 2-

Niðurhala þarf *GitHub Desktop*, þar sem hægt er að sjá allar breytingar sem gerðar eru í kóðanum frá hverjum og einum.

Þar er valið “File” og síðan “Clone Repository“. Til þess að setja verkefnið okkar þar inn.

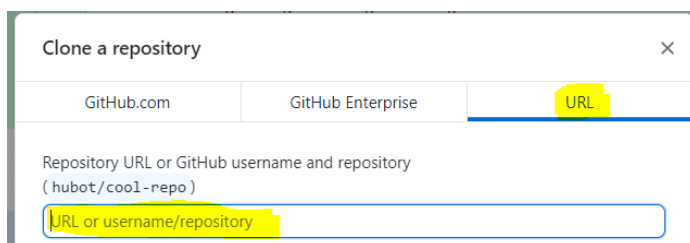
Farið er á GitHub.com og ýtt á code og þaðan er afritað URL fyrir verkefnið



Skref 3 –

Farið er inn á GitHub Desktop þar sem valið er File og Clone repository

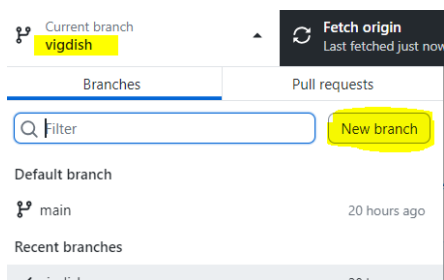
Valið er URL og þaðan er settur inn linkurinn af Github.com sem þýðir að nú er verkefnið okkar tengt þar inn á.



Skref 4-

Það er komið að búa til branch

Farið er inn á Branch inn á GitHub Desktop og þaðan er ýtt á “New branch” þar sem hver og einn í teyminu býr til sitt eigið branch með nafninu sínu. Þetta er í stuttu máli hver og einn að búa til sitt eigið geymslusvæði fyrir kóðann sinn.



Það þarf að hafa eitt branch sem heitir main þar sem aðalverkefnið er geymt.

Það er æskilegt að alltaf að vera á sínu branchi þar sem ef upp koma villur að þá veit sá einstaklingur hvar og hvenær hann gerði sá villu á sínu branchi áður en sett er kóðann inn á main, svo að það séu engar villur í kóðanum.

Hver er tilgangur Branch og Merge?

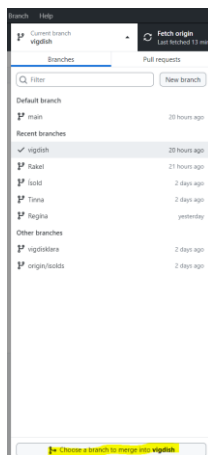
Branch er ákveðin útgáfa af verkefninu þínu þar sem hver og einn vinnur sjáfstætt að kóðanum á sínu svæði.

Aðgerðin Merge er þegar verið er að sameina kóða sinn t.d frá sínu branchi (svæði).

Þegar einstaklingur er búinn að breyta/bæta við kóðann á sínu branchi (svæði) þá getur hann mergeað sitt branch(svæði) við main(aðalverkefnið). Í öðrum orrðum sameinað kóðann á sínu svæði við aðalverkefnið.

En hvernig geri ég það?

Smelltu á "Branch" > "Choose a branch to merge into" í GitHub Desktop til að sameina sitt branch við t.d main ef það á að sameina verkefnið sitt við aðalverkefnið (main).



Mikilvægt að gera í byrjun dags

Þar sem main er aðal branchið þá vilja allir vera up to date með kóðann sinn í samræmi við teymið. Þá eru allir að vinna í sínum kóða á sínu branchi og geta síðan alltaf sett inn sinn kóða á main.

Svo í byrjun dags þá er **MJÖG MIKILVÆGT** að mergea sitt branch við main til þess að hafa updateaða verkefnið (hafa nýjustu breytingarnar á sínu svæði).

Valið er sitt branch og ýtt á "Choose a branch to merge into" og smellt er á "main".

Grunnaðgerðir í GitHub Desktop

Hvað þýða þessi nýju hugtök í GitHub Desktop?

Klóna (Cloning)

Sú aðgerð sem afritar geymslu kóðans fyrir verkefnið (Clone Repository) eins og við gerðum hér að ofan í skrefi 2.

Commits

Á Github Desktop þar sem hægt er að bæta við breytingum með því að ýta á commit.

Farið er inn á changes, sem er við hliðina history

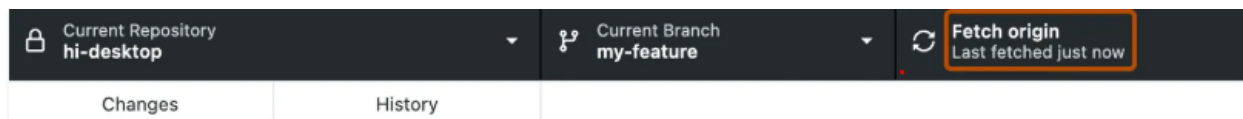
Þegar gerð er breyting inn á kóðanum frá einstakling og hann vill setja sinn kóða í aðalverkefnið sem heitir “main“ þá er farið inn á changes og skrifað inn í summary dálkinn lýsingu á kóðanum sem verið er að setja inn. Þar er síðan ýtt á Commit to xxx. Þá er kóðinn kominn inn á þitt branch og þaðan síðan merge við main (aðal verkefnið. Þetta einfaldar fyrir hinum í teyminu að sjá hvaða þýðingu með hvað kóðinn gerir sem einsaklingurinn setti inn.

Svo er einnig hægt að sjá í history varðandi breytingu hvers og eins í kóðanum og kl hvað breytingin var gerð.

Það **græna** er það sem var bætt inn í kóðann og það **rauða** er það sem var eytt úr kóðanum.

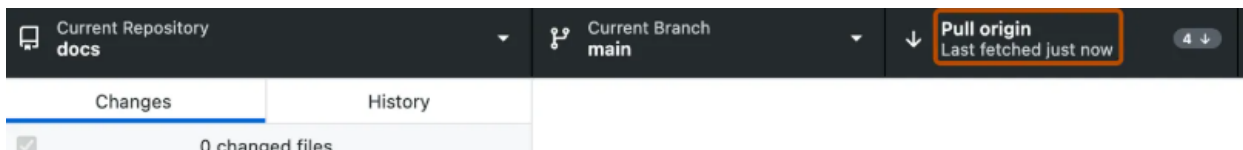
Fetch origin

Er sú aðgerð að ná í og downloada kóðanum á verkefninu og þá strax updateað local verkefnið til að matcha það content. Líka til þess að gá að hvort það var einhver í teyminu sem breytti/bætti við kóðann sem þú þarft að sækja.



Pull origin

Er sú aðgerð notuð til þess að ná í þær breytingar (commits) sem hafa verið gerðar.



Push origin

Það að setja kóðann sem þú breyttir/bættir við, á þitt branch.

Pull Requests og Codereviews

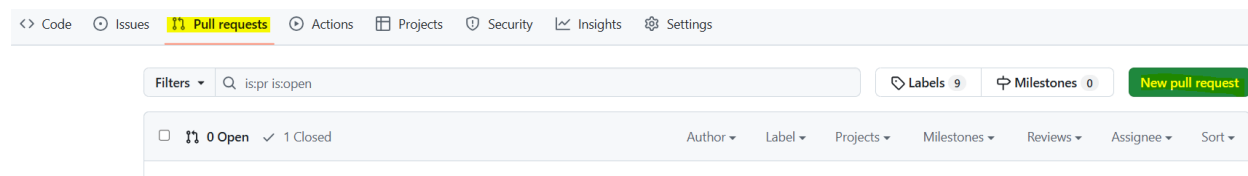
Pull request

Pull Request er notað til að leggja til breytingar áður en þær eru sameinaðar í main branch.

Farið er á Github.com

Ýtt er á "Pull Requests" og þaðan ýtt á "New Pull Request".

Þar er hægt að bera saman breytingar og skrifa lýsingu við kóðann sem verið er að pulla.



Code Reviews

Þar er hægt að fara yfir kóðann þegar búið er að ýta á Pull request

Þetta er mjög hjálplegt þegar villur koma upp, þar sem hægt er að leysa þær þar (conflict)

Hægt er að samþykja breytingarnar

Branching strategies -Git Flow

Feature Branches:

Nota branches fyrir nýja eiginleika. Er afritun af main kóðanum þar sem teymið getur unnið að nýrri útgáfu af kóðanum þar til hún er komin.

Release Branches:

Búnað til fyrir stöðuga útgáfu. Sem leyfir þér að vinna í branchinu þínu án þess að hafa áhrif á kóðann í main.

Hotfix Branches

Notaðir til að laga villur í framleiðslu. Eru til þess að laga villur í prod release.

Lausn árekstra (Conflicts)

Dæmi um árekstur

Þegar tvö branches reyna að gera breytingu í sömu línu í skrá og þegar reynt er að sameina þær skrár þá verður árekstur (conflict file).

Leið til að leysa úr árekstur

Upp kemur "Conflict file" og hvað skal gera þá..

Þegar komið er að því að leysa árekstur er farið á GitHub Desktop og ýtt á "Resolve conflicts".

Þá þarf að breyta skránni og þaðan er hægt að bera saman breytingarnar í kóðanum í vscode. Samsagt bera saman þær breytingar sem valda villunni (conflict) í kóðanum.

Síðan þegar búið er að breyta línunni í kóðanum sem villa var í, þá er breytt kóðanum "Mark as resolved" og síðan í GitHub Desktop "Commit merge".

Einnig er hægt að fara inn á Github.com og fara inn á Pull request og bera saman branch og sjá hvaða skrár í kóðanum er að valda villunni (conflict).

Þar er hægt að bera saman kóðana og hægt að leysa það með að ýta á "resolve" og síðan þegar búið er að gera breytingar og það er ekki lengur conflict file þá er merkt sem resolved.

