

طريقة التعاون بين الفريق بإستخدام الـ git و الـ GitHub

البداية

قائد الفريق

يقوم بإنشاء Repo على الـ GitHub مع الخيارات التالية :

- اعطاء اسم المشروع بصيغة PascalCase
- اختيار تكون public
- اختيار اضافة README
- اختيار اضافة gitignore من نوع python

بعد عملية الـ Clone على جهازه ، يقوم بإعداد المشروع للإستخدام الأولي (إنشاء المشروع ، إنشاء التطبيقات اللازمة مبدئياً، الخ) .

في ملف الـ README.md يقوم باضافة معلومات المشروع ، اسم المشروع ، وصف المشروع ، اعضاء فريق المشروع ، روابط الملفات المطلوبة (User Stories, UML, Wireframe) <

بعدها يقوم بمزامنة التعديلات المحلية على جهازه مع الريبو على الـ GitHub .

من ثم يشارك اعضاء الفريق الرابط . و يكون المطلوب من كل فرد منهم يعمل Fork ، و بعدها يتم عمل Clone و البدء بالعمل.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * / Repository name *

T-PYTHON-JULY-23 / GroupProjectt

✔ GroupProjectt is available.

Great repository names are short and memorable. Need inspiration? How about **ideal-telegram** ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: Python

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None

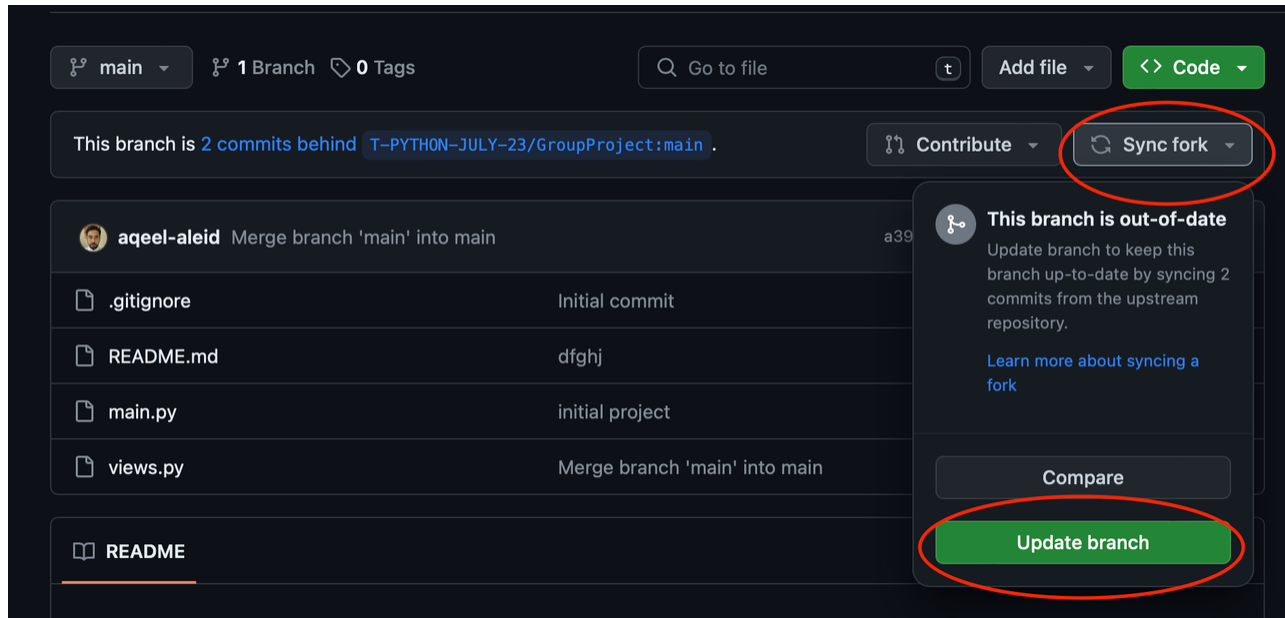
دورة العمل و التعاون

عضو الفريق

بعد ان يتم انجاز عمل ما ، يقوم عضو الفريق بإعتماد التعديلات ، من ثم مزامنة التعديلات المحلية مع الريبو . بعدها يقوم بإرسال pull request

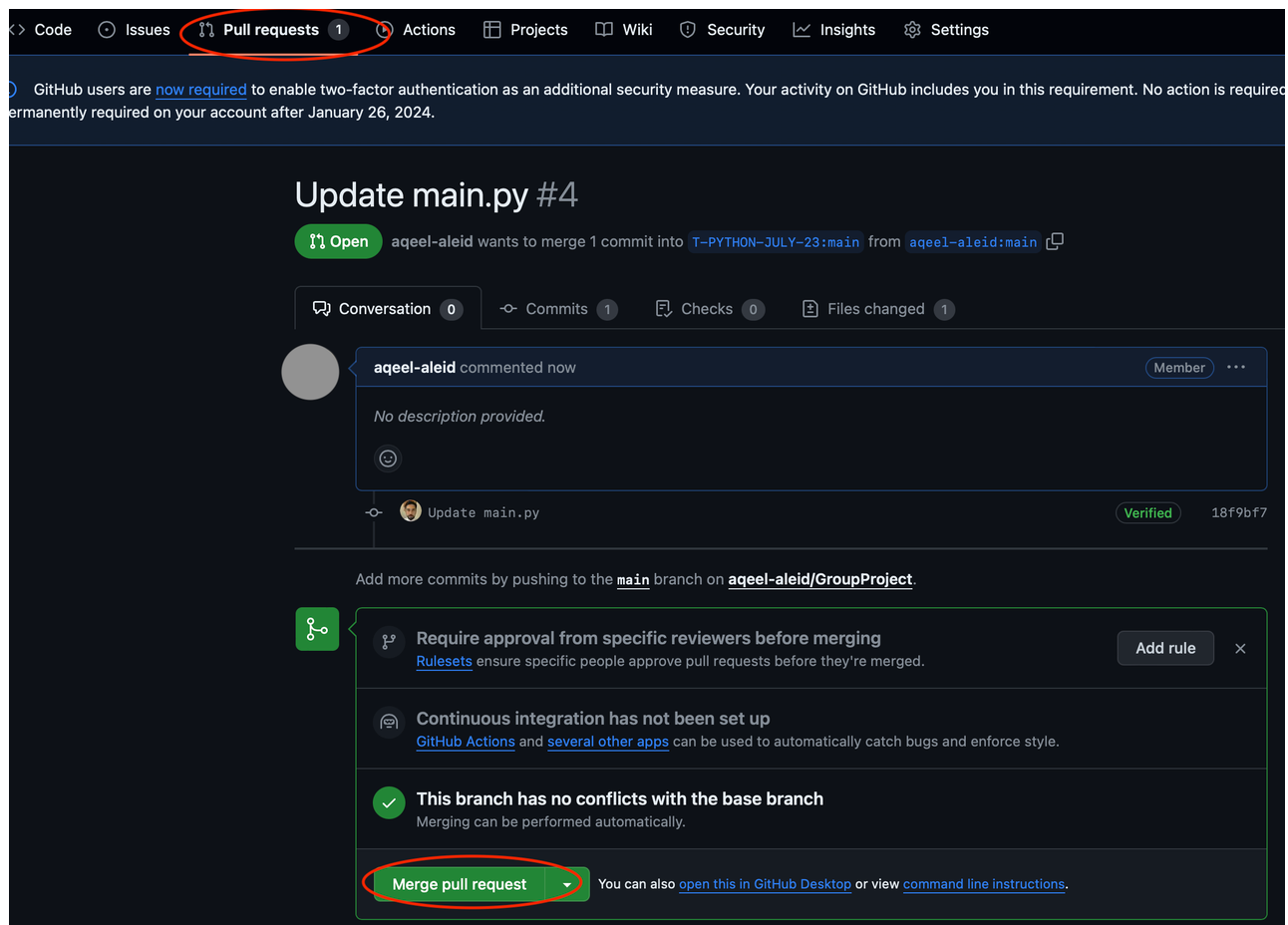
- اعتمد التعديلات المحلية (على جهازك) و زامننها Commit & Sync
- انشئ طلب Create Pull Request
- اخبر قائد الفريق بإرسال تعديلات له.

- انتظر خبر من قائد الفريق انه تم اعتماد التعديلات ودمجها .
- قم بفتح صفحة الريبو الخاص بك (الفورك) - حدث الصفحة اذا احتجت - .
- في اعلى الشاشة (كما هو موضح بالصورة بالأسفل) اضغط على زر Sync Fork من ثم اضغط على زر Update branch.
- من ثم ارجع للريبو المحلية على جهازك و اعمل مزامنة. و بعدها يمكنك متابعة العمل.



القائد

- اعتمد التعديلات المحلية (على جهازك) و زامنهما Commit & Sync
- اذهب الى صفحة الريبو الاساسية على الـ GitHub - حدثها اذا احتجت -
- استعرض طلبات الـ Pull Requests
- من ثم قم بالدخول على احدها ، و اضغط زر الـ Merge pull request
- بعد انتهاء عمليات الـ merge لجميع الـ pull requests الموجودة لديك ، قم بمزامنة التعديلات محليا على جهازك.
- اخبر باقي اعضاء الفريق بانتهاء عمليات الدمج و ان بإمكانهم الحصول على التعديلات الجديدة.



في حال وجود تصادمات - حل التصادمات Resolving Conflicts


اثناء عملية محاولة دمج الـ pull request قد لا يستطيع الـ Github دمجها مباشرة وذلك لوجود تصادمات يتوجب حلها. التصادمات في محتوى الملف تحدث عندما يعمل أكثر من عضو على نفس الملف ، بالتالي يجب حلها يدويا و اختيار المحتوى المطلوب اعتماده .


اتبع المثال التالي

- عند عملية الدمج ، اذا حصل تصادمات ، لن يظهر لك زر Merge pull request الأخضر ، و انما سيكون غير فعال. لكن على يمين سيظهر معك زر اسمه Resolve Conflicts .
- اضغط على زر Resolve Conflicts .
- قم بحل التصادمات و اختيار المحتوى المطلوب اعتماده (ستظهر لك عدة ملفات او ملف على حسب الملفات التي فيها تصادمات) حلها كلها.
- بعد اختيار المحتوى المطلوب اضغط على زر Mark as resolved
- من ثم Commit Merge و تابع معه زر I understand continue updating main

- من ثم سيرجع بك الى صفحة الـ pull request و سيتفعل معك الزر باللون الأخضر من جديد Merge pull request.

Add more commits by pushing to the `main` branch on `aqeel-aleid/GroupProject`.



**This branch has conflicts that must be resolved**

Use the [web editor](#) or the [command line](#) to resolve conflicts.

Conflicting files

`main.py`

Resolve conflicts


Merge pull request

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Update main.py #4

Resolving conflicts between `aqeel-aleid:main` and `T-PYTHON-JULY-23:main` and committing changes → `aqeel-aleid:main`

1 conflicting file

`main.py`
main.py


```
1 print("hello world")
2
3 <<<<<< main
4 #some comment
5 >>>>>>
6 print("hello world again")
7 >>>>>> main
8
```

1 conflict Prev Next Mark as resolved

Update main.py #4

Resolving conflicts between `aqeel-aleid:main` and `T-PYTHON-JULY-23:main` and committing changes → `aqeel-aleid:main`

1 conflicting file

`main.py`
main.py

```
1 print("hello world")
2
3 |
4 #some comment
5
6 print("hello world again")
7
8
```

1 conflict Prev Next Mark as resolved

Update main.py #4

Resolving conflicts between `ageel-aleid:main` and `T-PYTHON-JULY-23:main` and committing changes → `ageel-aleid:main`

Commit merge

1 conflicting file

main.py

main.py



main.py

```
1 print("hello world")
2
3
4 #some comment
5
6 print("hello world")
7
8
```

✓ Resolved

⚠ Heads up, this will commit to `main`.

- ➡ Commit updates to the `main` branch.
- 🔗 Create a new branch and commit updates. Your pull request will be updated automatically.

I understand, continue updating main