GitExcely: Version control for Excel via Git

GitHub: https://github.com/hellofromtheothersky/Excel-data-version-control-via-Git

In this demo, we will walk through these points:

- Create a new project
- Edit Excel and discover its text version after being parsed
- How to deal with conflict situation

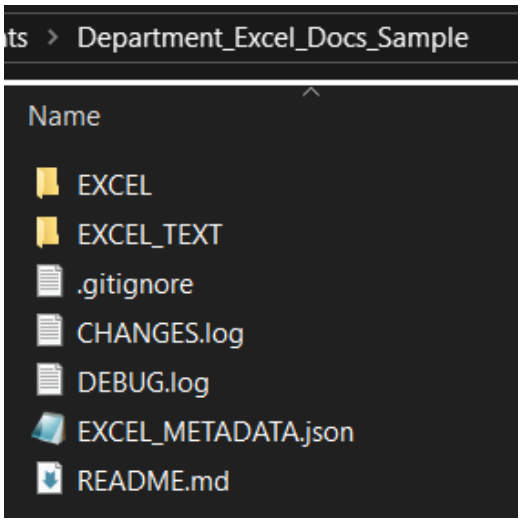
Demo images:

1) Init project

Create a repo on Git to manage an Excel file named **Department_Excel_Docs_Sample.xlsx**
Here are command lines to install gitexcely and create the project:

```
pip install gitexcely
gitexcely init --path Department_Excel_Docs_Sample
```

Then, we will get a local repo with a pre-set layout to work with

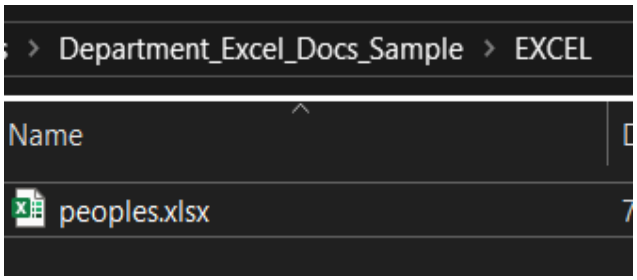


Below are meanings of some important file/folder of the tool

```
. Project Name
├── EXCEL/
│   ├── (Store all *.xlsx here, subfolder accepted)
├── EXCEL_TEXT/
│   ├── (Excel data will be parsed into text here)
├── EXCEL_METADATA.json/
│   (mannually setup header line and keys in Excel sheets)
├── CHANGES.log/
│   Changes summary of Excel when before running git push (automatically removed before pushing)
├── DEBUGS.log/
│   Show jobs running when run git commit (automatically removed before pushing)
```

2) First commit

Create a new simple excel file named `peoples.xlsx` file in the `EXCEL/` folder

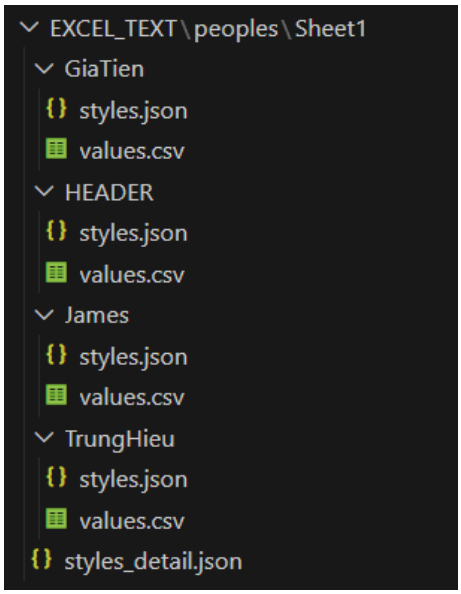


	A	B	C
1	name	age	city
2	Trung Hieu	22	Quy Nhon
3	Gia Tien	28	Ho Chi Minh city
4	James	25	London
5			

I want to specify some features (the header row, index/key columns) of my sheet, the **`EXCEL_METADATA.json`** is used for this purpose

Note: even if I don't specify any features in the files, the Excel file will still be parse in its natural way *(A, B, C, D... for the column name instead of the header row, and number line count instead of the key columns)*

And after create commit action, a pre-commit action from git will check to find the changed Excel files and parse it into text. Then, we get this after creating `peoples.xlsx`.



Here is the structure of path to store data in text:

$$\text{EXCEL\_TEXT/\{EXCEL\_FILE\_NAME\}/\{SHEET\_NAME\}/\{ROW\_NAME\}/} \frac{\text{values.csv}}{\text{styles.json}}$$

ROW_NAME is default count by line (“L1”, “L2”, “L3” ...), but if the key columns of data are declared, it will replace by the very data of key columns in this row (“James”, “TrungHieu”, “GiaTien”), if duplicate occurs, the row name will come with a prefix is the line count (“L6_James”, “L9_James”).

***values.csv*, and *styles.json* will store data of only one row.**

> ***“I choose this approach because, for most of the Excel use cases, we store row-based data. And splitting a whole table into rows and store it many files help us control version better, easily track, change focally for a single row”***

After committing, we also have a change log file to view changes summary before pushing:

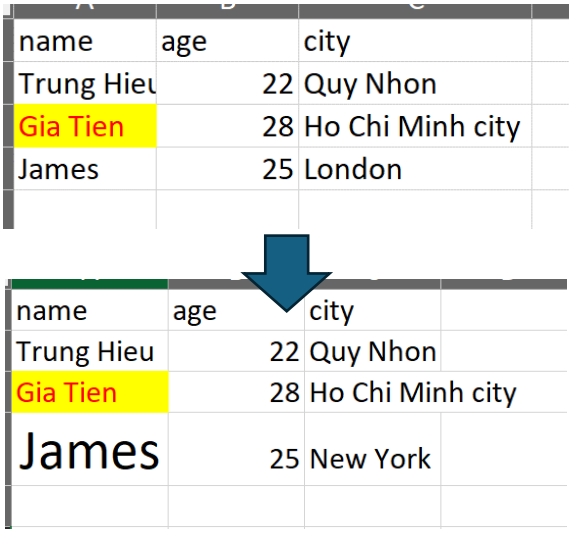
CHANGES.log

```
1  Changes:
2  - Excel: "peoples":
3      - Sheet: "Sheet1":
4          - 4 value rows CREATED (4 style rows CREATED)
5
```

3) Make some changes

Now, I will make some changes on “James” row,

- Change “London” to “New York”
- Make the font size of column “name” bigger:

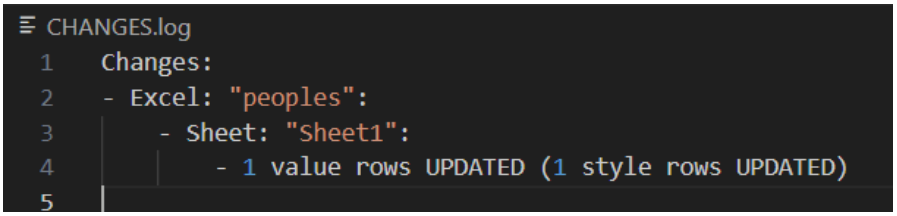


name	age	city
Trung Hieu	22	Quy Nhon
Gia Tien	28	Ho Chi Minh city
James	25	London

name	age	city
Trung Hieu	22	Quy Nhon
Gia Tien	28	Ho Chi Minh city
James	25	New York

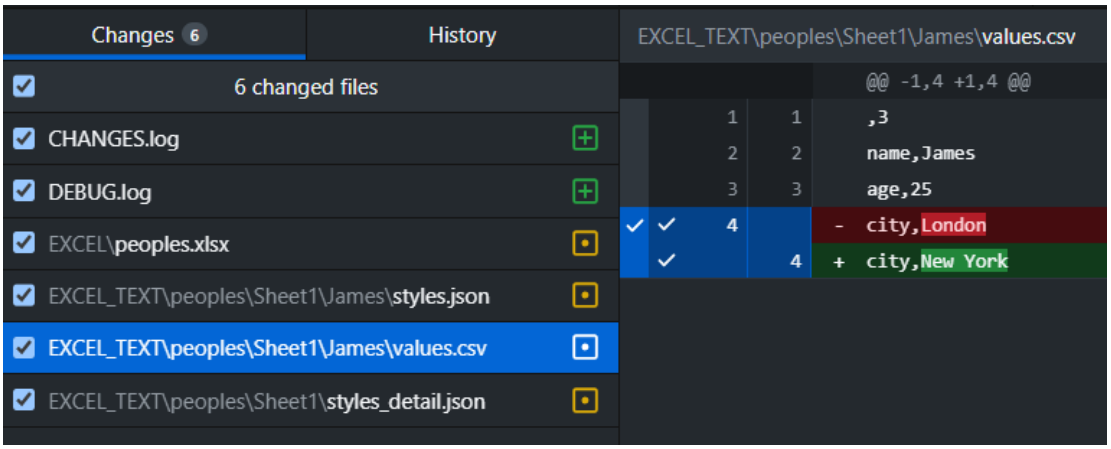
Commit again, and we get:

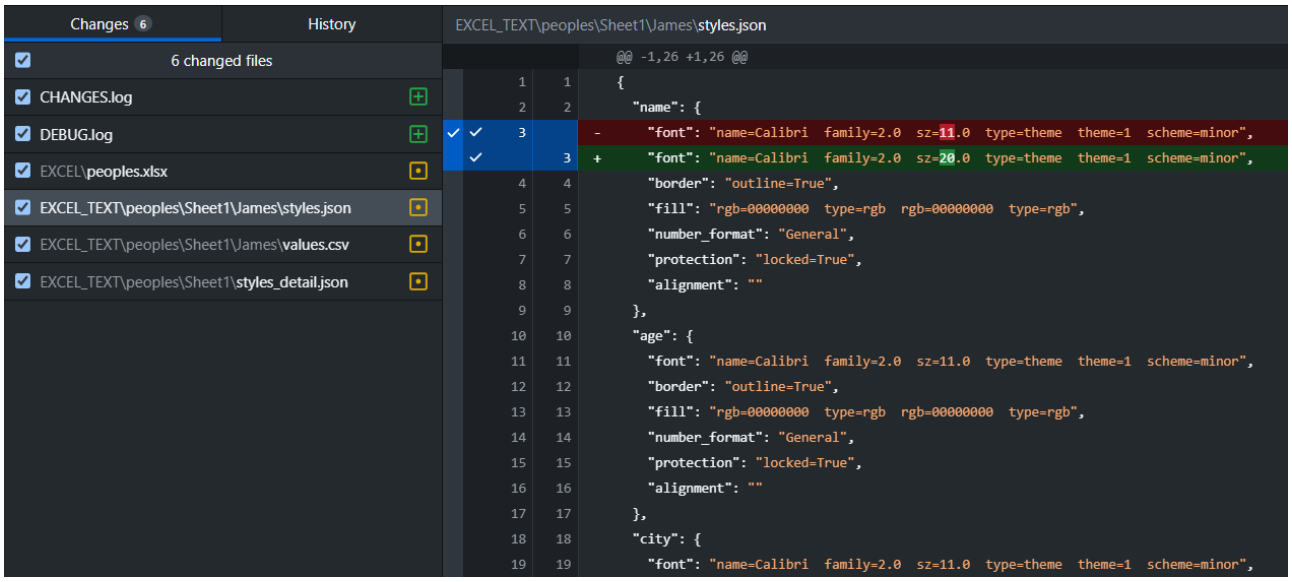
Changes summary:



```
CHANGES.log
1  Changes:
2  - Excel: "peoples":
3      - Sheet: "Sheet1":
4          - 1 value rows UPDATED (1 style rows UPDATED)
5
```

Changes detail in files:



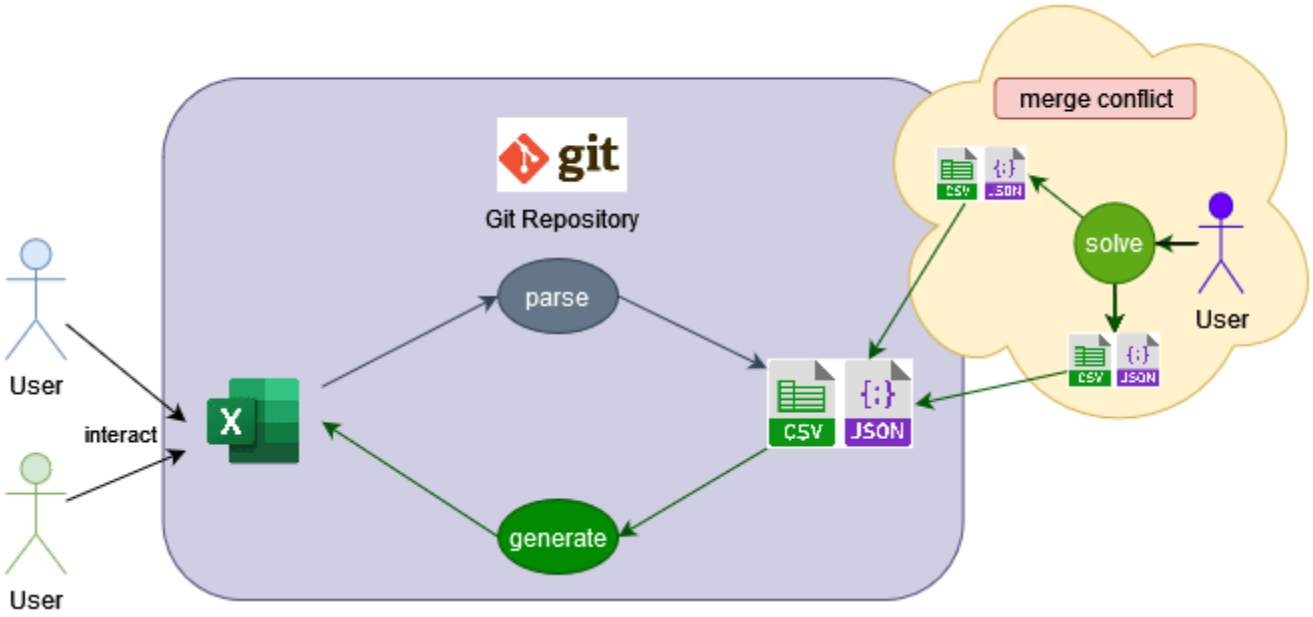


Parse Excel elements into text and utilize current git diff, we can now check the changes more easily

4) Solve conflict

The big difference between current Excel different tracking system (MS Version History or others) and GitExcely is the ability to work and collaborate in the real Git style including create branch, merge and solve conflict.

And beside the parsing function from Excel into text for us to check diff of a binary file through text, GitExcely also have generating function Excel from text which help user who found conflict from text, and solve directly on text to utilize current git support (like what do we with code version control)



For example, I will create a feature branch basing on main branch. Make a change on the same position to create conflict when merging and then trying to solve the conflict

On the feature branch:

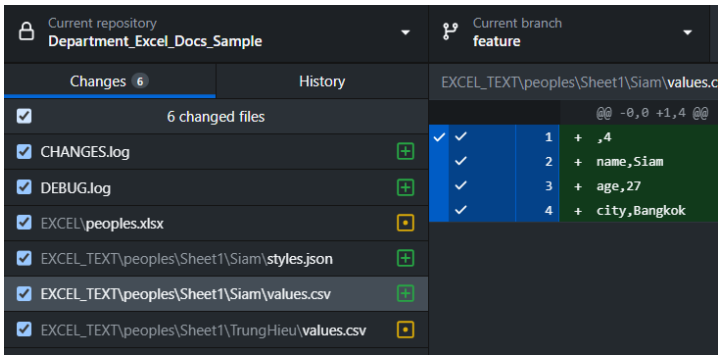
- Add a new row with key “Siam”
- On “TrungHieu” row, change “Quy Nhon” to “Ha Noi”

name	age	city
Trung Hieu	22	Quy Nhon
Gia Tien	28	Ho Chi Minh city
James	25	New York

name	age	city
Trung Hieu	22	Ha Noi
Gia Tien	28	Ho Chi Minh city
James	25	New York
Siam	27	Bangkok

Commit data

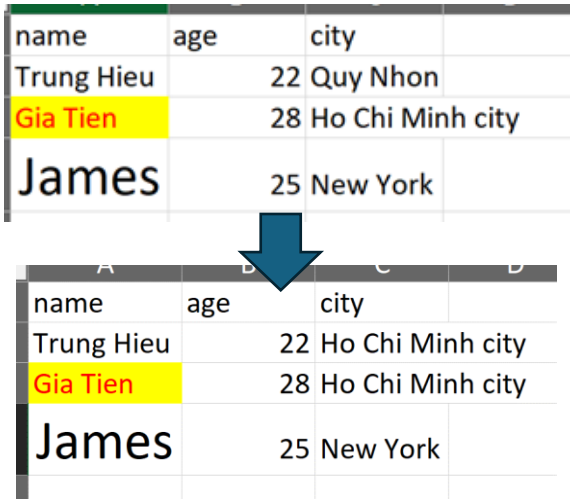
```
Changes:
- Excel: "peoples":
    - Sheet: "Sheet1":
        - 1 value rows CREATED (1 style rows CREATED)
        - 1 value rows UPDATED (0 style rows UPDATED)
```



Two new files (Siam\styles.json and Siam\values.csv) for the new row, and a modified file for the updated row (TrungHieu\values.csv)

On the main branch:

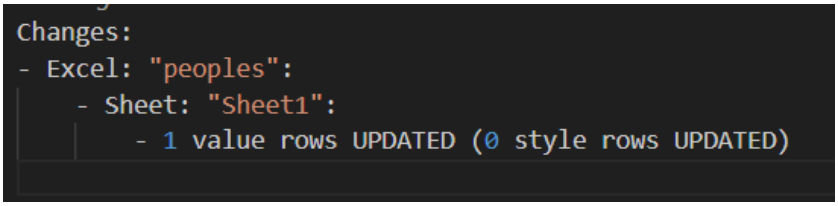
Still on “TrungHieu” row, change “Quy Nhon” to “Ho Chi Minh city”



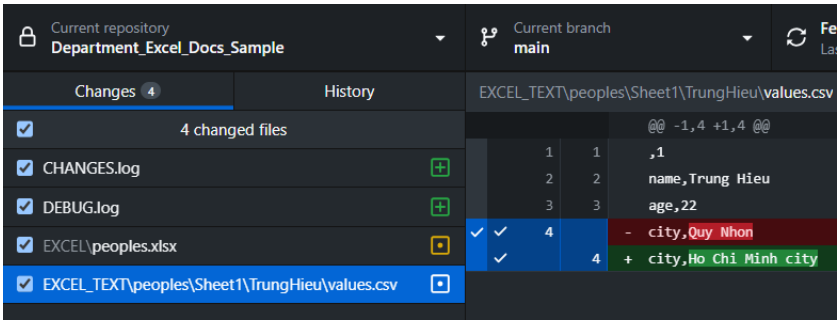
name	age	city
Trung Hieu	22	Quy Nhon
Gia Tien	28	Ho Chi Minh city
James	25	New York

name	age	city
Trung Hieu	22	Ho Chi Minh city
Gia Tien	28	Ho Chi Minh city
James	25	New York

Commit data:



```
Changes:
- Excel: "peoples":
    - Sheet: "Sheet1":
        - 1 value rows UPDATED (0 style rows UPDATED)
```

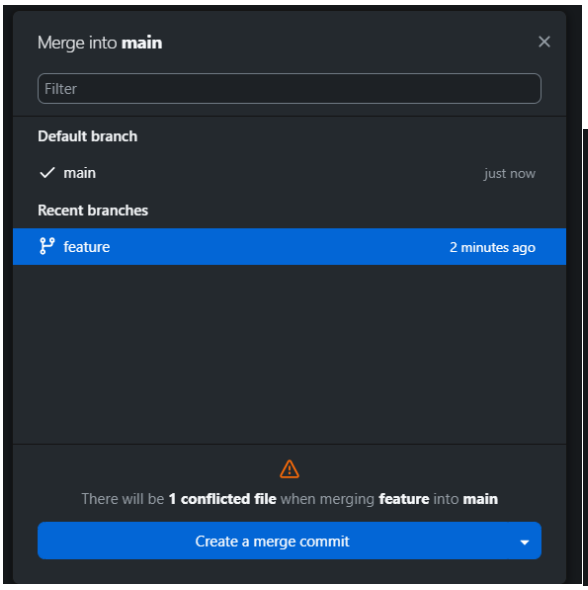


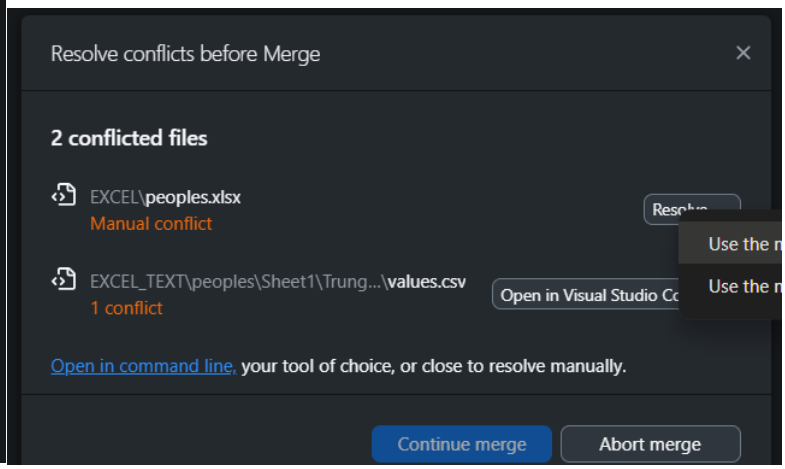
Merge feature into main branch:

In this case, conflict occur in two files:

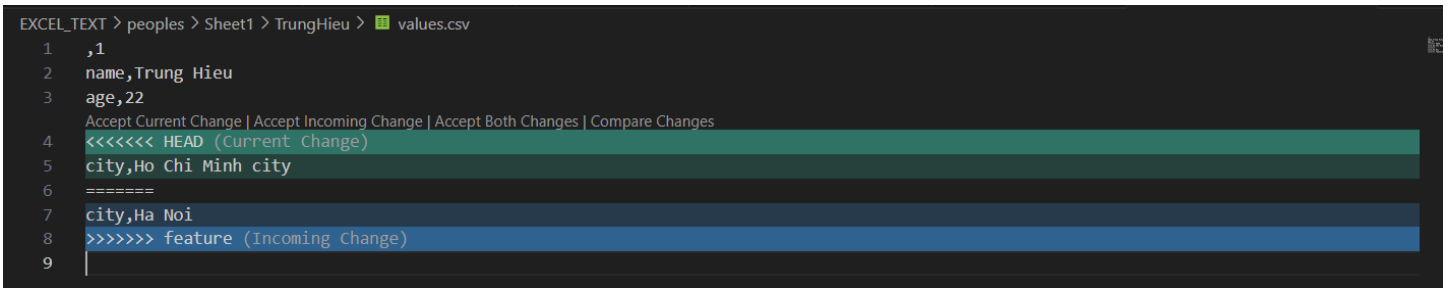
- .xlsx file, which is a binary file, git cannot help us find the conflict parts
- EXCEL_TEXT/peoples/Sheet1/TrungHieu/values.csv file

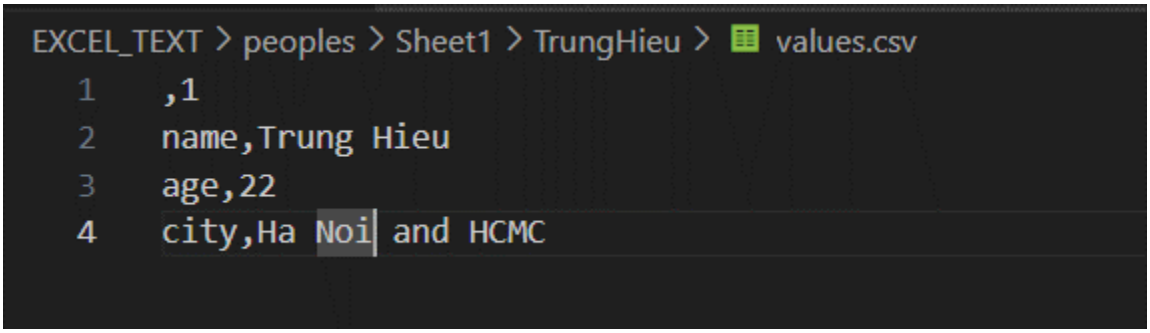
To solve the conflict, we will manipulate on the text version instead, and then when commit, the Excel will be generated again to mimic its text version



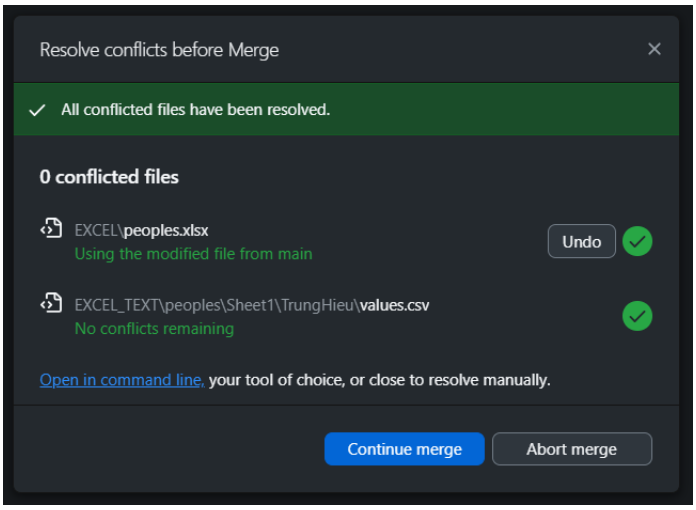


Solve conflict on the file – which is row data that have confict

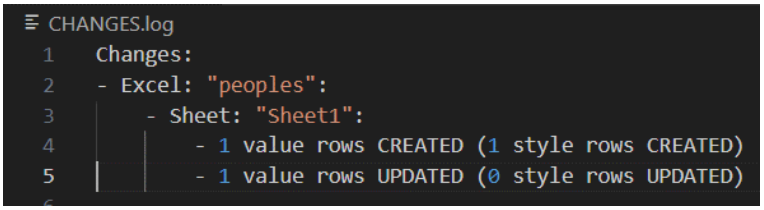


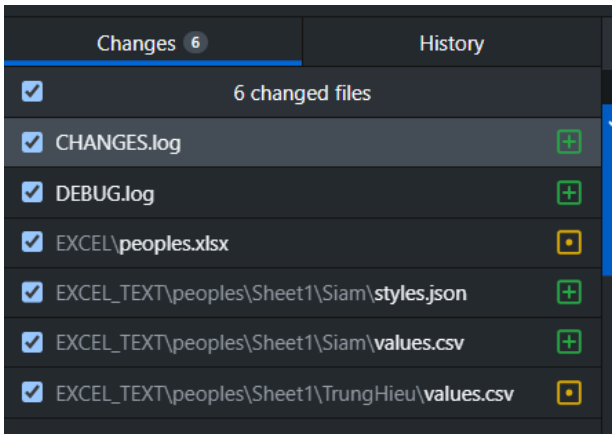


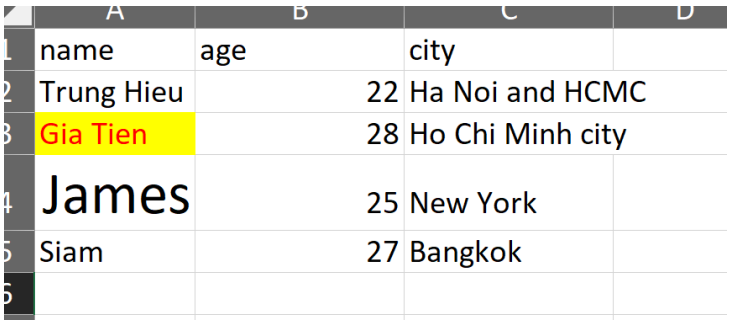
Merge



Checking the changes







	A	B	C	D
1	name	age	city	
2	Trung Hieu	22	Ha Noi and HCMC	
3	Gia Tien	28	Ho Chi Minh city	
4	James	25	New York	
5	Siam	27	Bangkok	
6				

And finally, the conflict was solved, excel file was updated right the same as text version