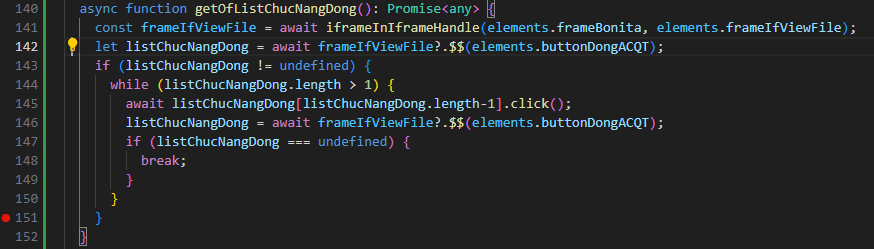


* Khai báo getOf ListChucNangDong gán vô biến (any) > biến này sẽ cân đc mọi kiểu ( như string, number và boolean,…)
* Khởi tạo or gọi iframe ra
* Gọi ra 1 object ListChucNangDong và gọi ra elements buttonDongACQT
* Thực hiện câu lệnh IF > nếu list đó NOT NULL thì sẽ thực hiện dòng dưới CÒN NULL thì hệ thống sẽ dừng lại và kết thúc vòng lặp IF.
* ĐK IF NOT NULL >> thì sẽ thực hiện tiếp vòng FOR: khai báo list và sẽ chạy hết các element của ListChucNangDong.
* Trong quá trình chạy vòng FOR thì đồng thời sẽ thực hiên thao tác click vào list đó.

( cái await thêm vỗ để nó gán câu click vào đồng bộ còn nếu k có thì nó sẽ chạy loạn)



* Khai báo getOf ListChucNangDong gán vô biến (any) > biến này sẽ cân đc mọi kiểu ( như string, number và boolean,…)
* Khởi tạo or gọi iframe ra
* Gọi ra 1 object ListChucNangDong và gọi ra elements buttonDongACQT
* Thực hiện câu lệnh if > nếu list đó ko undefined thì sẽ thực hiện dòng dưới >< Trường hợp nó undefined thì sẽ chuyển xuống dòng if (ListChucNangDong === undefined) và đóng vòng chạy
* Dùng vòng chạy while ( kèm thêm ĐK là (ListChucNangDong.length > 1) tức là đếm số lượng của giá trị của 1 mảng ý [a,b,c] > là 3 giá trị
* Còn dòng 145 thì sẽ xác định index bằng length -1 > sẽ ra bằng index

1. Câu này sẽ chạy là khi xem đk là List đó k undefined >> thì sẽ chuyển xuống điều kiện giá trị phải lớn hơn 1 thì sẽ chạy tiếp
2. Và khi chạy qua list đó sẽ trừ đi 1 >> 2 và hệ thống sẽ chạy lại dòng 146 để loading lại giá trị
3. Lúc này trong list ý sẽ trừ đi 1 [a,b] length bằng 2 và chạy lại dòng 1 và 2

Lúc này trong list ý sẽ trừ đi 1 [a] length bằng 1 và 1 ko thể nào lớn hơn 1 nên sẽ đừng lại vòng while

***Trường hợp làm xong gửi lên trên:***

node: trước khi đẩn code lên git thì chọn: (+) > Split Teminal >Git bash

1. git checkout master //switch sang branch master

2. git pull origin // kéo code mới nhất trên master về local

3. git checkout -b [branch\_name] // tạo ra nhánh mới và switch qua nhánh đó

4. viết code + test passed local >> phần viết code của mình

5. add những phần thay đổi vào staged changes (kiểm tra từng phần thay đổi ở Changes, nếu đúng thì click dấu + để add vào Staged changes) ( tại phần SOURCE CONTROL )

6. git commit -m '[message commit]' // commit code lên vùng nhớ tạm

7. git push origin [branch\_name] // đẩy code lên git web

8. git fetch // download những sự thay đổi trên git web so với máy pc cá nhân

9. git rebase origin/master // merge những cái mới nhất ở nhánh maseter vào nhánh của mình

10. git push origin [branch\_name] -f // đẩy phần rebase lên git web

11. vào git và tạo merge request

***Trường hợp update xong gửi lên trên:***

1. add những phần thay đổi vào staged changes (kiểm tra từng phần thay đổi ở Changes, nếu đúng thì click dấu + để add vào Staged changes) ( tại phần SOURCE CONTROL )

(denghitiepquy\_kq16 - write new script for De nghi tiep quy \_16)

6. git commit -m '[message commit]' // commit code lên vùng nhớ tạm

7. git push origin [branch\_name] // đẩy code lên git web

8. git fetch // download những sự thay đổi trên git web so với máy pc cá nhân

9. git rebase origin/master // merge những cái mới nhất ở nhánh maseter vào nhánh của mình

10. git push origin [branch\_name] -f // đẩy phần rebase lên git web

**( git checkout [branch\_name] // về nhánh cũ của mình )**

**Trường hợp**

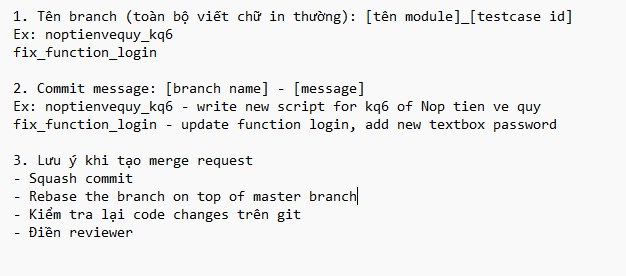
**git rebase origin/master >> mà bị comfix thì:**

+ B1: thực hiện chữa comfix all

+ B2: + vô phần mới và commit

+ B3 thực hiện git push origin [branch\_name] -f // đẩy phần rebase lên git web

Khi nào git rebase origin/master > thì chúng ta mới thực hiện git push origin … -f



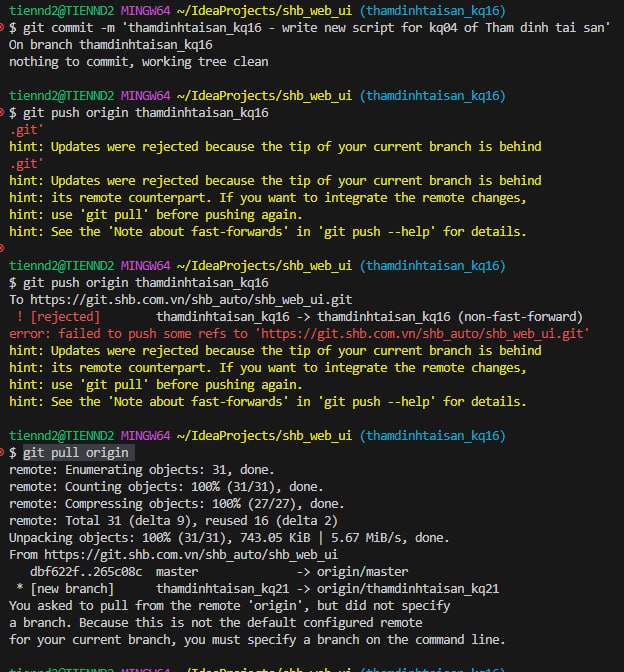
Khi A sửa và update trên nhánh của mình thì mình sẽ thực hiện câu lệnh:

git checkout master

git branch -D congdichvuketoan\_kq15

git checkout congdichvuketoan\_kq15

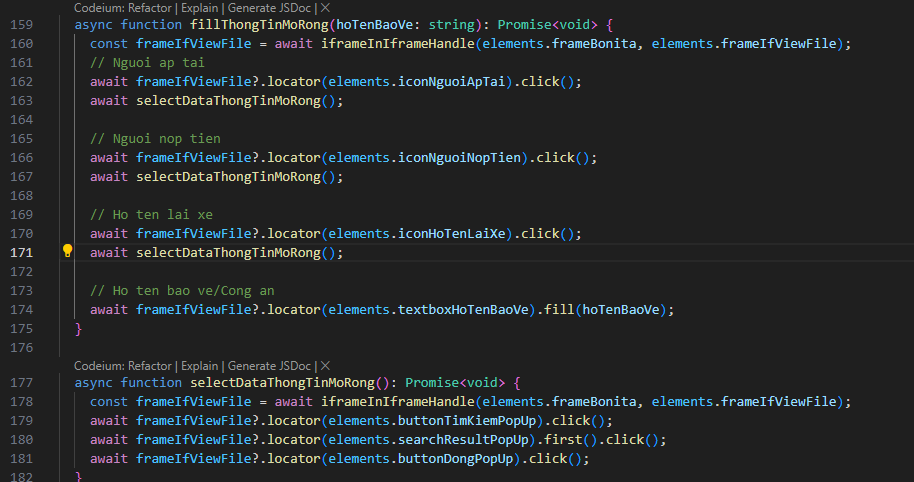
git pull origin



Khi chuyển qua 1 nhánh khác thành công >> thì mình thực hiện câu lệnh ( git pull origin )

Làm vậy thì hệ thống sẽ hiểu mình đang kéo code mới về để đồng bộ với nhau > nếu không dùng câu này trong trường hợp đã đẩn code của nhánh A lên r mà đẩn lại thì hệ thống sẽ báo lỗi và ko cho đẩn code lên

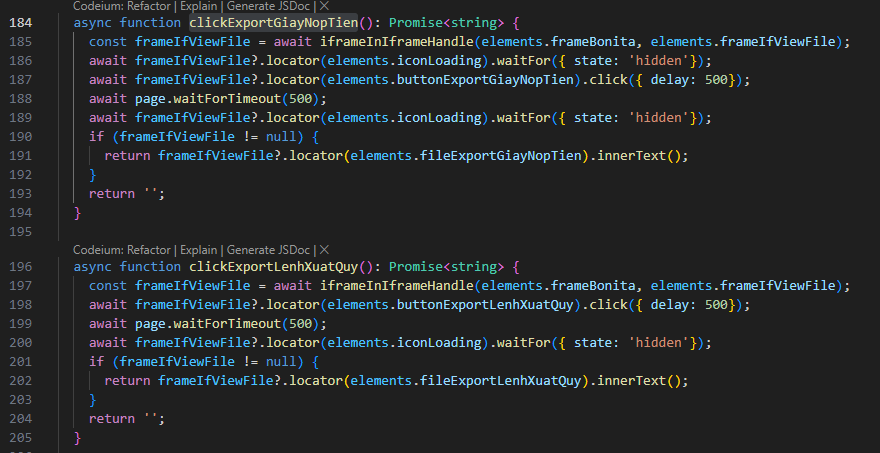
🡺 nên phải pull về thì mới được đẩn lên



* Câu lệnh này sẽ bắt đầu chạy theo tuần tự từ ( Người Áp Tải 🡺 Họ tên Bảo Vệ Công An)
* Trong quá trình chay tuần tự thì sẽ có chung các thao tác như:

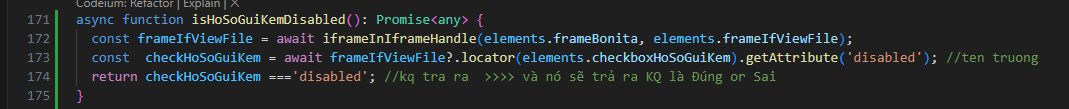
+ Click chọn vô element ( 162, 166, 170 )

+ Thực hiện câu lệnh selectDataThongTinMoRong: Click Tìm Kiếm > Chọn data đầu tiên của Bảng thông qua câu **first()** > Click Đóng

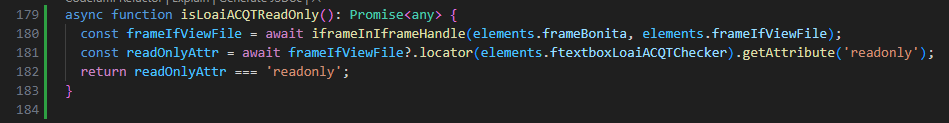


Mục đích của 2 câu này là gọi ra element từ đó get text > để kiểm tra hiển thị text đó tên màn hình innerText()

* Vòng IF > nếu element = NULL thì hệ thống sẽ trả ra KQ return = ‘ ‘ tức là ko có gì >< ngược lại kết quả trả ra bằng NOT NULL thì hệ thống thực hiện tiếp câu lệnh dưới
* ở câu lênh dưới ND chính sẽ là lấy ra element và TEXT giá trị ý ra

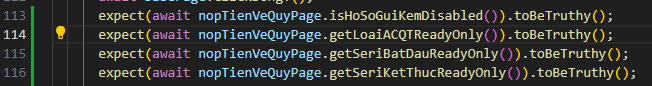


**OR**



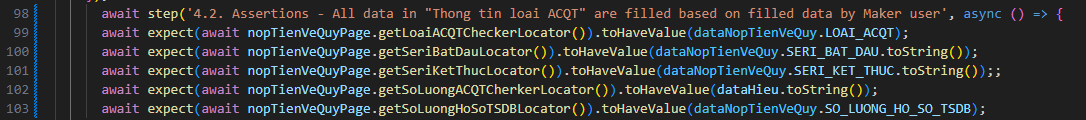
* Nội Dung 2 câu code này: để xác định 2 element hiển thị với trạng thái readonly và disabled
* Thì sẽ dùng hàm getAttribute () để KT ( tạo ra 1 biến tên là readOlyAttr)
* Với giá trị trong getAttribute (…….) > là câu đầu của phần tử ( tại dòng 173)
* Còn dòng 174, 182 là kết quả trả ra từ biến readOlyAttr phải bằng readonly và disabled



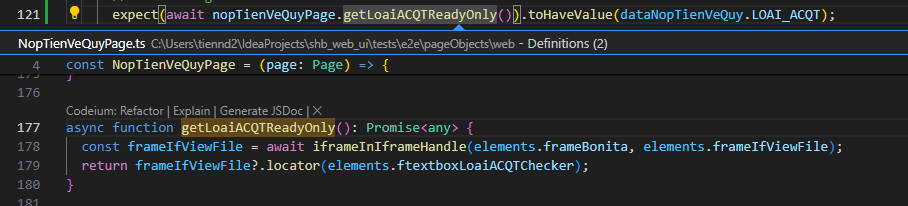


Nội dung đoạn code này mục đích để kiểm tra trường hiển thị trên màn hình > không cho phép nhận giá trị thông quá câu lệnh **toBeTruthy()**

KẾT HỢP từ code trên và code đưới thành 1 câu hoàn chỉnh

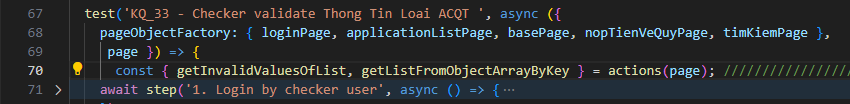


* Nội dung đoạn code trên để so sánh giá trị cùng 1 trường tại 2 roll khác nhau > xem data có bằng nhau k thông qua câu lệnh **toHaveValue() >>>>>** câu lệnh ý chỉ dùng được với kiểu dữ liệu là string
* Nếu data là kiểu khác string thì phải ép kiểu cho nó bằng câu **.toString()**

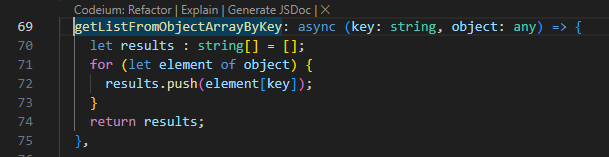


177 -> 179: thực hiện khai báo locater > để thực hiện gán cho hàm toHaveValue vì hàm này chỉ cho phép chạy bằng locater và giá trị sau dataNop TienVeQUy >> phải là string

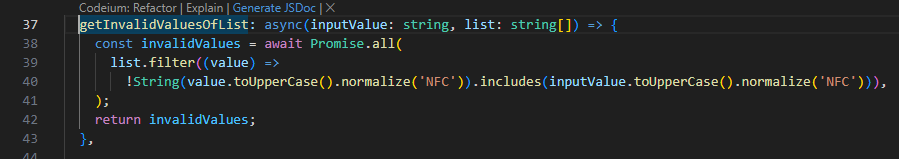
Còn k phải thì thực hiện câu lệnh .toString()

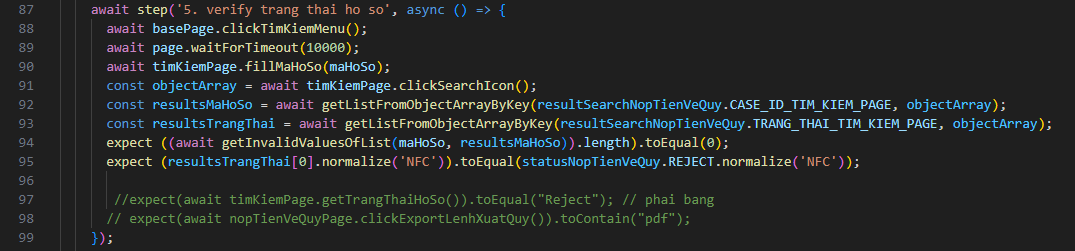


Khai báo câu lệnh (getInvalidValuesOfList và getListFromObjectArrayByKey)



Phần này sẽ kiểu gom nhóm list lại các giá trị lại vào 1 object > thông qua key





Mục tiêu: verify trạng thái của hồ sơ sau khi Checker thực hiện thao tác hủy luồng

91 > Tạo biến **objecArray**: với mục đích để thực hiện Click tìm kiếm

92 > Tạo biến **resultsMaHoSo**: gom nhóm và thực hiện thao tác tìm mã hồ sơ đó bằng 3 giá trị trong ngoặc

93 > Tạo biến **resultsMaHoSo**: gom nhóm và thực hiện thao tác tìm trạng thái của mã hồ sơ đó bằng 3 giá trị trong ngoặc

94 > là xác nhận : **NFC** kia là chuẩn hóa chữ cho ng và AI dễ đọc

**Hàm parseInt()** chuyển đổi chuỗi sang số nguyên.

**Hàm parseFloat()** chuyển đổi chuỗi sang số thực (số có phần thập phân).

**toBeTruthy():** Kiểm tra hiển thị giá trị + chức năng + trường có diseble

**toBeVisible():** Kiểm tra hiển thị trên màn hình

**toEqual():** Phải bằng

Team mình thống nhất với nhau 1 xíu nhé

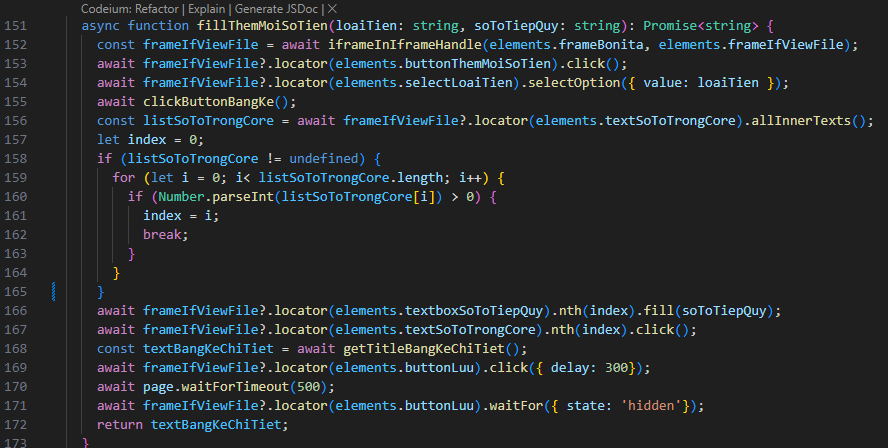
Để đảm bảo format code ok hơn thì

1. Trước khi mn push code lên git thì chạy **lệnh Shift+Alt+F** --> format code

2**. alt+ z** --> auto wrap nếu dòng bị dài quá

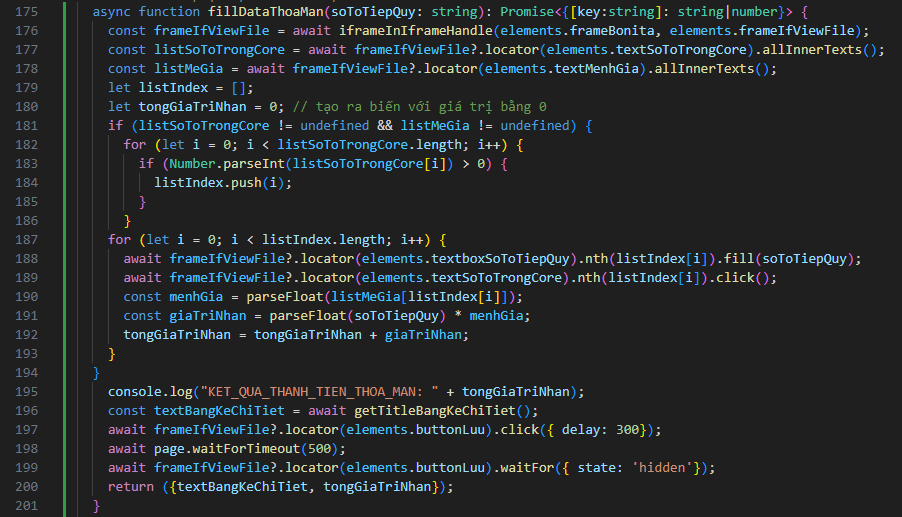
A screenshot of a computer

Description automatically generated



Mục đích của đoạn code này là thao tác fill data + mở đóng

* Nhấn chọn Thêm mới số tiền > Chọn data Loại tiền > Click Bảng kê
* 156 > tạo ra 1 object listSoToTrongCore: gọi element Số Tờ Và get all text trong list ý bằng câu lệnh **allInerTexts()**
* Khai báo biến index = 0 >> chỉ là bằng 0
* Nếu IF listSoToTrongCore = UNDEFINED > thì sẽ đến thẳng câu lệnh 162 là đừng >< còn NOT UNDEFINED thì thực hiện tiếp vòng FOR
* Trong vòng FOR khai báo let I = 0 (tức là đặt câu lệnh chạy từ 0) tiếp đến là ĐK đừng là I << hơn độ dài của listSoToTrongCore và sau mỗi lần chạy vòng FOR thì I sẽ tự động tăng lên 1
* Dòng 160 là khi chạy vòng FOR thì thực hiện điều kiện listSoToTrongCore data phải > 0 và được ép kiểu là number.parseInt
* Dòng 166 sẽ thực hiện fill data vô soToTiepQuy được khai báo ở trên là kiểu string và fill vào các dòng mà vòng FOR đã tìm thấy bằng câu lệnh **nth(index)**
* và 167 > thực hiện click ra ngoài để dữ liệu của dòng 166 ăn và kích theo vòng FOR đã tìm thấy bằng câu lệnh **nth(index)**



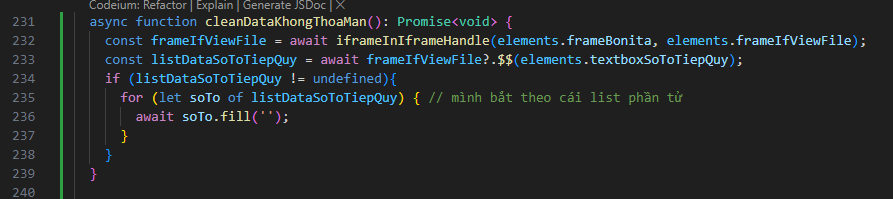
Mục đích của đoạn code này là thao tác fill data + mở đóng kèm theo tính tổng giá trị Thành tiền

* Nhấn chọn Thêm mới số tiền > Chọn data Loại tiền > Click Bảng kê
* 156 > tạo ra 1 object listSoToTrongCore: gọi element Số Tờ Và get all text trong list ý bằng câu lệnh **allInerTexts()**
* Dòng 180 khai báo tongGiaTriNhan = 0
* Khai báo IF listMenh + listSoToTrongCore != UNDEFINED
* Trong vòng FOR khai báo let I = 0 (tức là đặt câu lệnh chạy từ 0) tiếp đến là ĐK đừng là I << hơn độ dài của listSoToTrongCore và sau mỗi lần chạy vòng FOR thì I sẽ tự động tăng lên 1
* Dòng 183 là khi chạy vòng FOR thì thực hiện điều kiện listSoToTrongCore data phải > 0 và được ép kiểu là number.parseInt
* Dòng 187 khai báo vòng FOR để chạy và điền dữ liệu thỏa mãn vào các index đã tìm được
* Dòng 188 sẽ thực hiện fill data vô soToTiepQuy được khai báo ở trên là kiểu string và fill vào các dòng mà vòng FOR đã tìm thấy bằng câu lệnh **nth(index)**
* và 189 > thực hiện click ra ngoài để dữ liệu của dòng 188 ăn và kích theo vòng FOR đã tìm thấy bằng câu lệnh **nth(index)**
* 190 khai báo object listMenhGia > và tại dòng này sẽ lấy ra các Mệnh giá thỏa mãn ĐK đồng thời ép kiểu nó thành số thực thông qua câu lệnh **parseFloat()**
* Dòng 191 thực hiện tính toán đồng thời ép kiểu cho SoToTiepQuy
* Dòng 192 thực hiện thao tác tính tổng: VD KQ phần Thành Tiền là 10, 20, 30 > thì sẽ tính theo công thức sau:
* tongGiaTriNhan(0) + thanhTien(10) = tongGiaTriNhan(10)
* tongGiaTriNhan(10) + thanhTien(20) = tongGiaTriNhan(30)
* tongGiaTriNhan(30) + thanhTien(30) = tongGiaTriNhan(60)
* .... nếu Đk trên tìm đc thêm giá trị của phần thành tiền

A computer screen with many colorful text

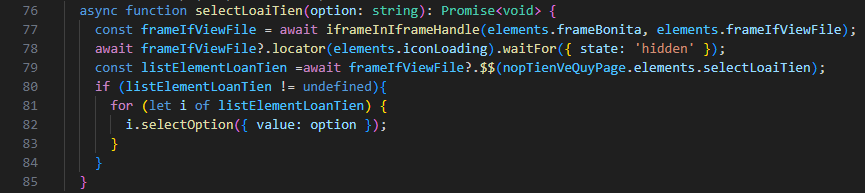
Description automatically generated

Giống trên chỉ khác ĐK là tìm giá trị nhỏ hơn < 0 thôi



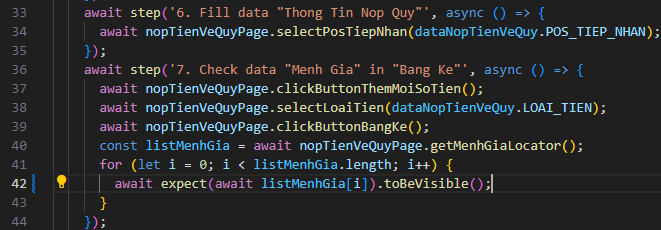
Mục đích: thực hiện clean data của tất cả các giá trị trong element

* 235 > khai báo soto của listDataSoToTiepQuy + vì k có ĐK nên nó sẽ thực hiện chạy hiết các giá trị của element đó
* 236 > thực hiện fill soto mà vòng FOR đã chạy >> đúng ra gọi là clean dữ liệu hiển thị trên trường
* Chỉ áp dụng được cho giá trị tĩnh k thay đổi or fix cứng



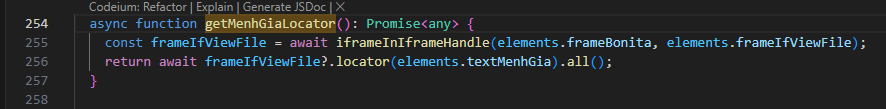
Mục Đích: Thực hiện fill toàn bộ dữ liệu data vào từng trường Loại tiền

* Dòng 79 > thực hiện khai báo list elements Loại tiền > sử dụng dấu $$ để lấy ra list
* Dòng 80 > nếu list elements khác undefined > thì sẽ thực hiện dòng 81 là chạy vòng for cho từ giá trị của list đó
* Dòng 82 là sẽ gán giá trị vào I cho từng giá trị elements chạy đc

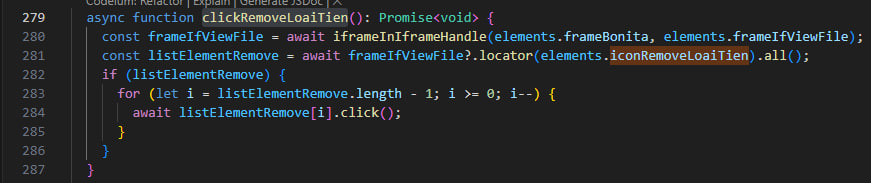


Mục đích câu lệnh này là thực hiện kiểm tra hiển thị của trường Loại tiền trên màn hình có hiển thị không

* Dòng 40 thực hiện gọi ra list mệnh giá và thực hiện vòng for tại dòng 42 > ở đây I sẽ bắt đầu đếm từ 0 và điều kiện I < độ dài của list đó thì đừng + mỗi lần chạy I sẽ đc + thêm 1
* Và trong mỗi lần chạy đó sẽ kèm thêm kiểm tra hiển thị mệnh giá bằng câu toBeVisible()

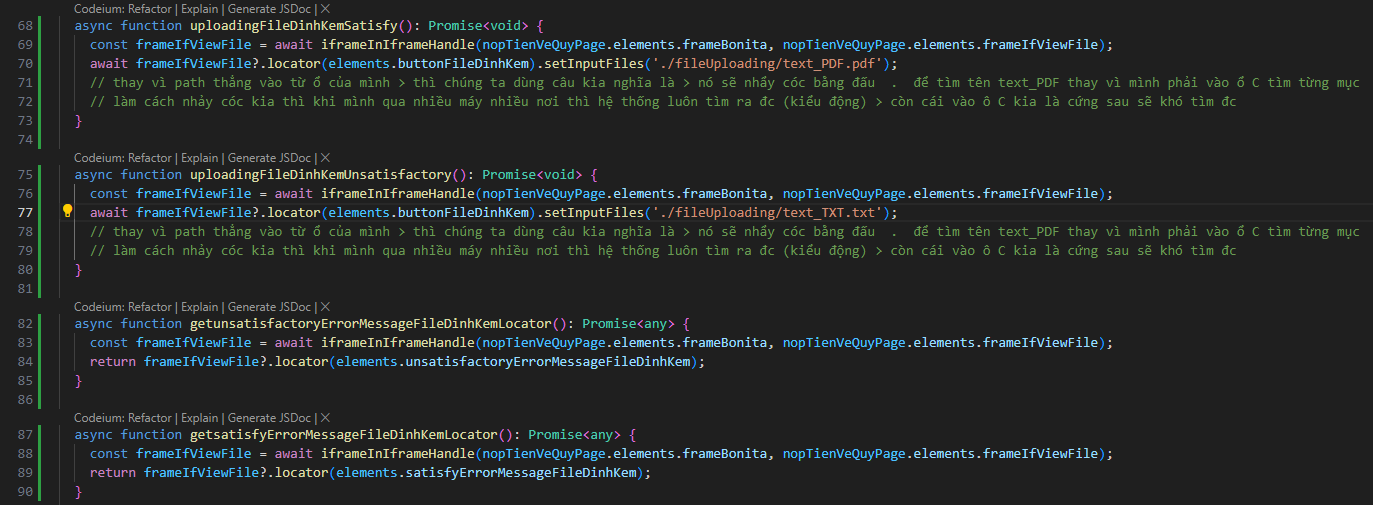


Mục đích câu lệnh này là gọi tất cả các elemest của text mệnh giá ra bằng câu lệnh .all()



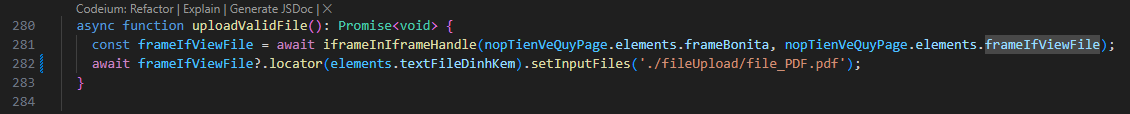
Mục đích câu lệnh này sẽ thực hiện xóa giá trị từ dưới lên

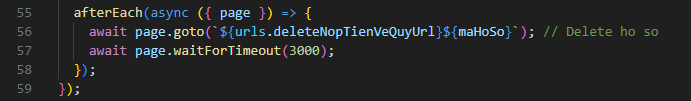
* 281 thực hiện gọi hết tất cả các elememts bằng . all()
* 283 thì gán I bằng list element – 1 tức là sẽ gán cho giá trị cuối cùng (còn I = 0 là giá trị đầu tiên)
* Sau mỗi lần chạy thì I sẽ đc trừ đi 1 và gặp điều kiện I >= 0 thì sẽ đừng lại
* 284 là điều kiện khi mỗi lần chạy vòng for thì hệ thống sẽ thực hiện click



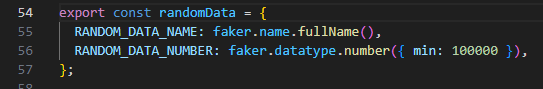
🡺 Chúng ta không được điền đầy đủ path của file được lưu trong máy mình > vì làm vậy hệ thống chỉ tìm đc path khi đang ở máy mình > còn khi chạy code tại máy khác thì không tìm đc dẫn đến lỗi

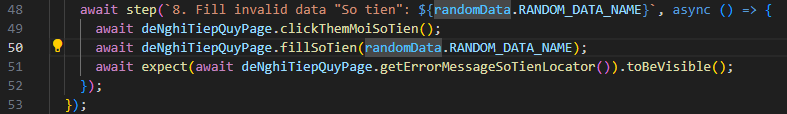
* Chính vì vậy cần điền khoảng để hệ thống thực hiện thao tác tìm > dù có đi đến đâu thì hệ thống vẫn sẽ tìm đc file mà mình muốn update lên





Mục đích câu lệnh này là thực hiện xóa hồ sơ khi thực hiện chạy xong dù đúng hay sai

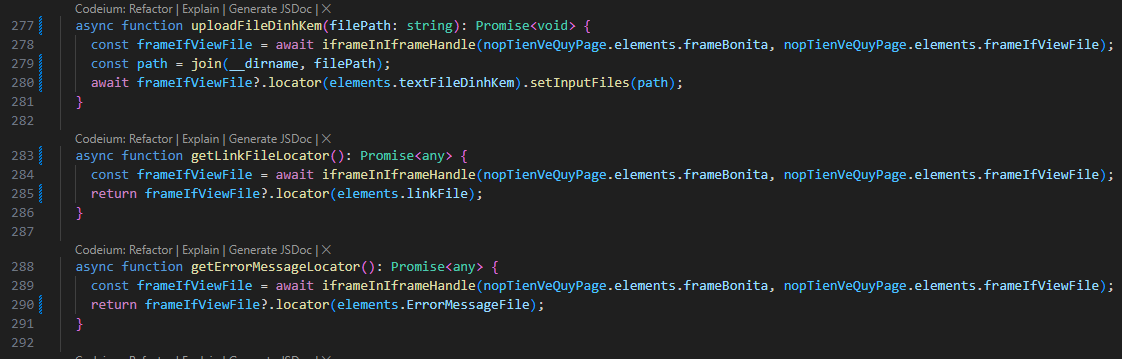




Câu lệnh vể radom



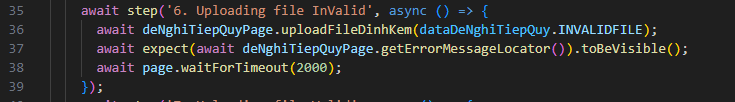
Lưu ý: cái path kia ‘../../fileUpload/file\_TXT.txt’ >> tức là lần đầu ../ nó sẽ thoát ra 1 vùng và lần t2 cx vậy > để tìm ra giá trị của path



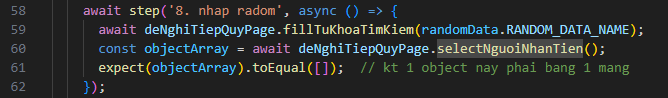
Mục đích dòng 227 đến 281 là thực hiện khao báo file

279 > tạo 1 object là path > sẽ là thực hiện khai báo join filePath đc khai báo là kiểu string

280 > gọi elemets trường để upload file ý vào > và thự hiện câu lệnh setInputFile(path)

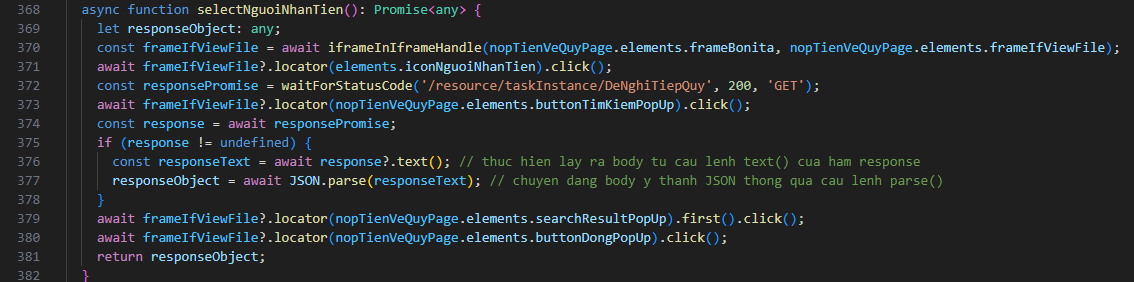


Mục địch để upload file và trường và kiểm tra hiển thị bt



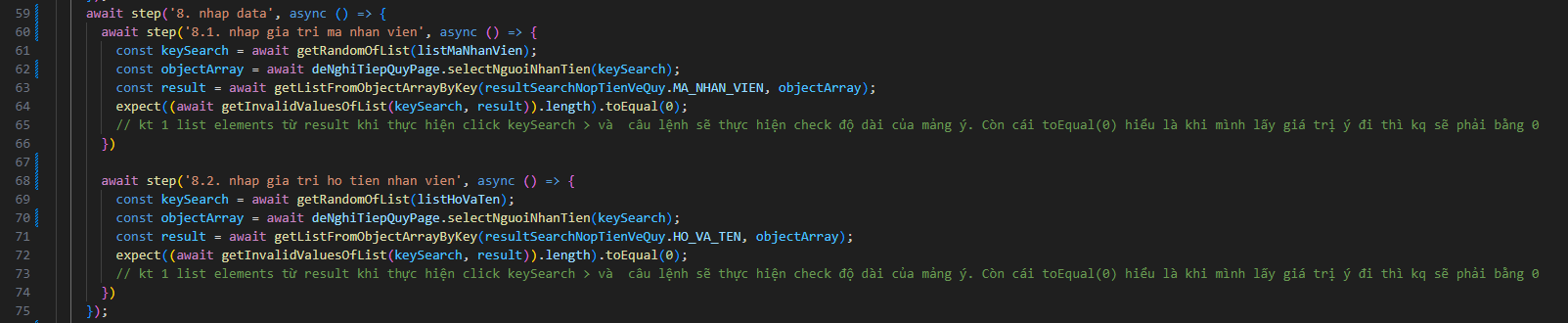
Mục đích dòng 61 > kiểm tra hiển thị kết quả trả ra phải là 1 mảng

VD như mình nhập giá trị random vô 1 trường + Tìm kiếm = 1 list DL >> thì ở đây verify KQ trả là 1 list



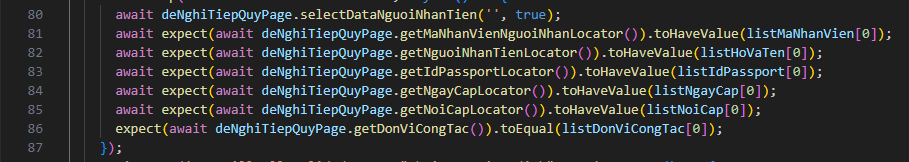
Mục địch đoạn code này để gọi API của hệ thống về ( ở trường hợp này là 1 list danh sách)

* Dòng 369 là khai báo 1 biến responseOblect là kiển any
* Dòng 372 > tạo 1 responsePromise trong đó chứa data liên kết vs API thông qua câu waitForStatusCode ( chứa URL + status + Methor )
* Dòng 374 thì hiểu đơn giả là khai báo response sẽ là responsePromise
* Vòng if sẽ là nếu response là k undefined thì sẽ thực hiện tiếp câu lệnh
* Dòng 376 tạo 1 object responsetext = ở đây sẽ thực hiện lấy data text từ body của API thông qua câu lệnh response.text()
* Dòng 377 sẽ lấy biến responseOblect = ở đây sẽ chuyển dạng body text thành JSON thoog nqua câu lệnh parse
* Dòng 381 KQ trả ra sẽ là responseOblect

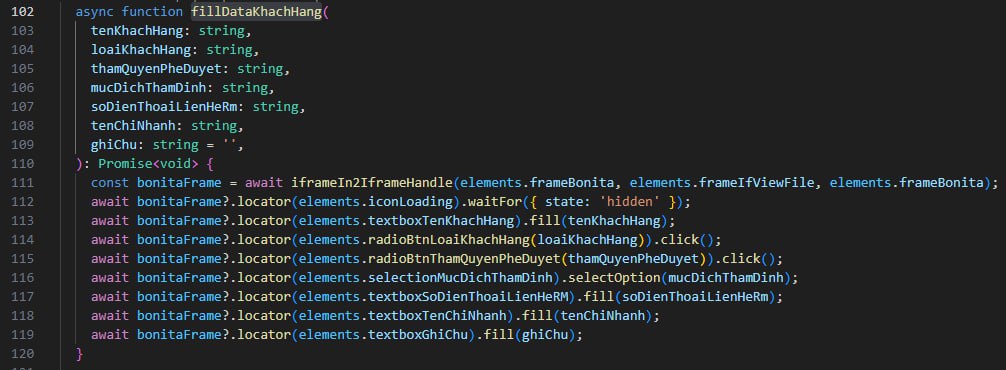




Mục đích sử dùng URL cho nhiều câu lệnh



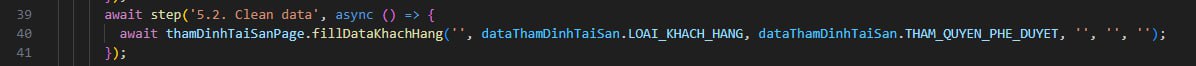
Dùng toHavalue thì phải cần await còn dùng câu lệnh toEqual (phải bằng) thì k cần await



A screen shot of a computer

Description automatically generated

Khi đã có quá nhiều câu lệnh cần xử lý trên 1 màn hình thì chúng ta thực hiện gộp tất cả vào để chạy 1 câu lênh như hình trên > tối ưu hóa dòng lệnh



Mục đính fill tất cả các trường là data rỗng thì chúng ta thực hiện là ‘’

A screenshot of a computer program

Description automatically generated

Theo như chị Lan Anh chia sẻ thì nên bắt locator vì như vậy sẽ tìm ra dễ hơn và chính xác > hạn chế tối đa lỗi vặt

 🡺 đây là cách viết của “id” and “name”

A screen shot of a computer

Description automatically generated

A screen shot of a computer code

Description automatically generated

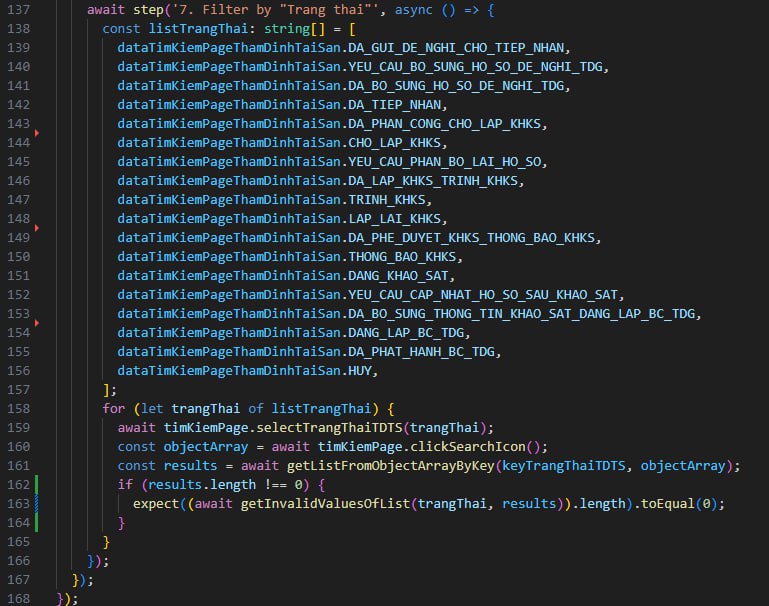
Mục đích kiểm tra phần ghi chú của acc B phải bằng acc A

Thao tác expect kiểm tra giá trị phải bằng thông qua câu lệnh toEqual()



Quan tâm dòng 91 đến 93 > vì giá trị của nó là (ngày + giờ)

+ ở đây chúng ta dùng câu lệnh split ( ‘ ‘)[0] > tức là tác giá trị ngày và giờ theo khoảng cách và lấy phần tử đầu tiên là ngày bằng [0]



Mục đích con này là search từng kết quả trạng thái và kiểm tra kq thôi

Giải thích dòng 163 hơi khó

A computer screen shot of a program code

Description automatically generated

Hiểu đơn giản là Void thì sẽ k trả ra KQ

A screen shot of a computer code

Description automatically generated

Phần fill ( userCVDTSTiep nhận ) >> đây sẽ là phần data truyền từ bên ngoài vào trong code bằng string

**trim(); Loại bỏ khoảng trắng ở đầu và cuối câu**

**getAttribute: lấy ra giá trị bên tay phải của 1 locator >> **

**>> ví Dụ lấy ra mode**

**A computer screen with colorful lines

Description automatically generated**

Mục đích đoạn function này là lấy ra value theo lable truyền vào

420 > đây mục đích gọi locator ra và lấu theo giá trị label > và in ra ( phần này để xem lấy đúng lable k )

442 > đây mục đích gọi locator ra và lấu theo giá trị lable Kèm dùng câu lệnh getAttribute(‘value’) > lấy ra value

Ở đây nếu k có vòng if thì hệ thống sẽ thông báo rằng cái getvalue option kia không sách định được vì chưa thỏa mãn 2 đk là != và != undefine

A computer screen with many small letters

Description automatically generated with medium confidence

Dòng 189 là cách dùng của nó

A close up of text

Description automatically generated

Mục đích câu này nếu mã nhân viên được phép nhập thì thực hiện fill data(maNhanVienNguoiDeNghi);

Nếu maNhanVienDeNghi là redonly !== redonly nếu maNhanVienDeNghi khác redonly thì

A screen shot of a computer screen

Description automatically generated

Mục đích gọi ra list data Mã hồ sơ + Tên công việc + Họ và tên người lao động + Chức danh >> từ đó thực hiện chạy vòng for với điều kiện “Mã hồ sơ + Tên công việc + Họ và tên người lao động + Chức danh != rỗng”

Và khi lấy ra đc các hồ sơ không rỗng thì thực hiện map với các điều kiện tên công việc + họ và tên ng lao động + chức danh

Kiểu lấy ra hồ sơ đạt đủ 3 tiêu trí trên k null > tìm đc phát thực hiện câu lệnh break thoát ra khỏi vòng lặp