Contents

[**I. Cấu hình chạy auto test api workspace – jmeter.** 2](#_Toc153271656)

[II. Giải thích các Element sử dụng trong project. 6](#_Toc153271657)

[2.1. User Variable Define. 6](#_Toc153271658)

[2.2. Thread group 7](#_Toc153271659)

[2.3. Property file reader 8](#_Toc153271660)

[2.4. Http request 10](#_Toc153271661)

[2.6. Pre procesors – JSR223 PreProcessor 11](#_Toc153271662)

[2.7. PostProcessor - JSR223 PostProcessor. 12](#_Toc153271663)

[2.8. If controller. 13](#_Toc153271664)

[III. Cách dựng api auto test. 13](#_Toc153271665)

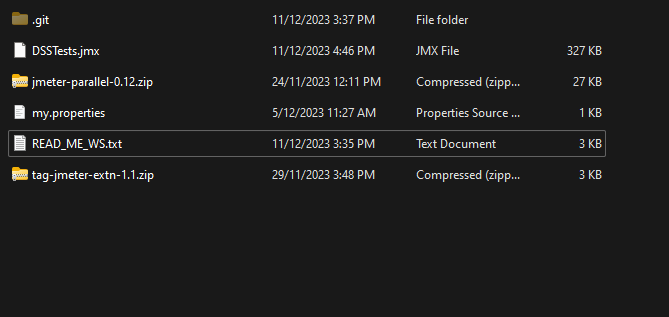
[VI. Các lưu ý khác. 14](#_Toc153271666)

# **I. Cấu hình chạy auto test api workspace – jmeter.**

**1.1. Clone project**

- Link github: <https://github.com/mduc279/DSSTestCase.git>

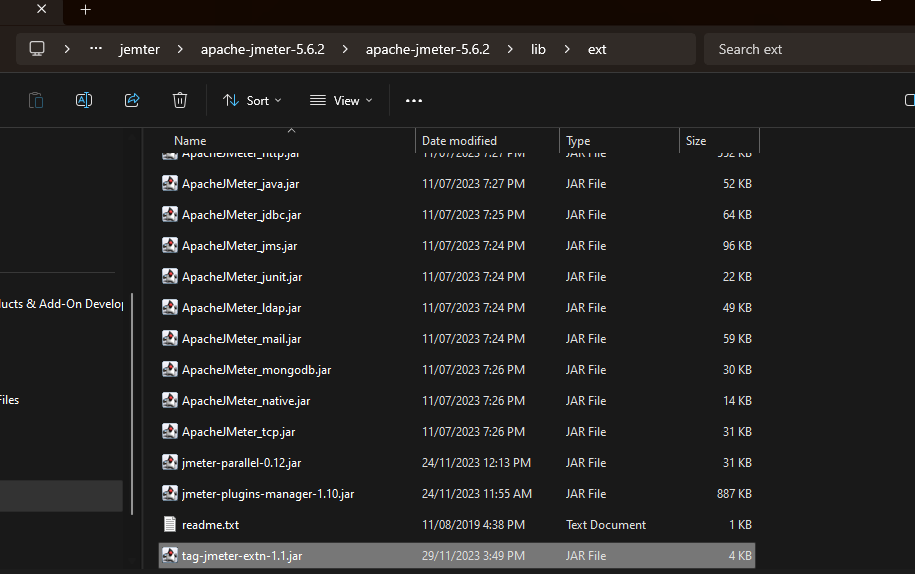
- Sau khi clone về, ta có folder như hình.



2. Cài Property file reader cho jemter.

- Giải nén file tag-jmeter-extn-1.1.zip đc file tag-jmeter-extn-1.1.jar

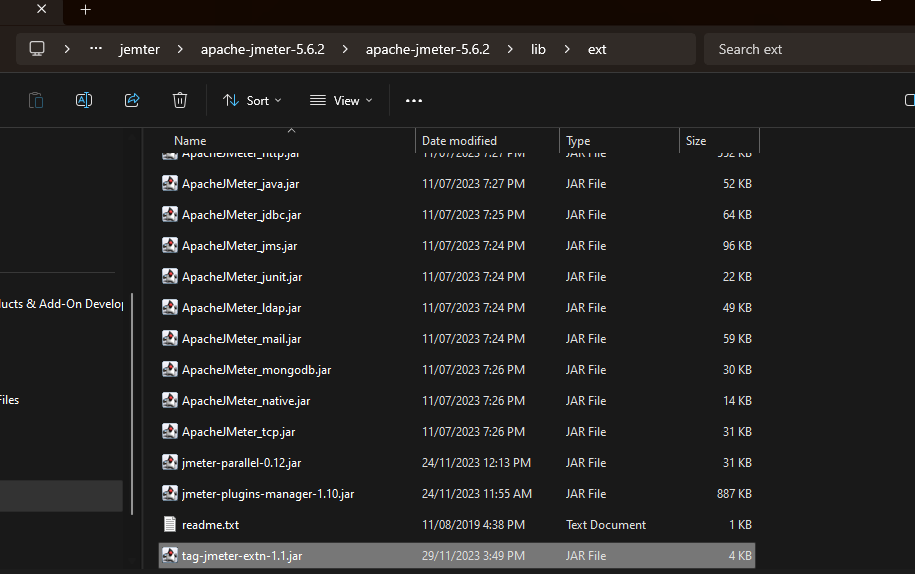
- Copy file tag-jmeter-extn-1.1.jar vào folder cài jemter với đường dẫn chi tiết: …/ lib/ext.



3. Cài Parallel Controller cho jemter.

- Giải nén file jmeter-parallel-0.12.zip đc file jmeter-parallel-0.12.jar

- Copy file jmeter-parallel-0.12.jar vào folder cài jemter với đường dẫn chi tiết: …/ lib/ext.



4. Tắt đi bật lại jemter.

5. Cấu hình giá trị cho các biến sử dụng khi chạy api - config: workspace\_variable:

- TokenDelegatedPermission - token do mobile truyền lên

- LoginEmailArray - chuỗi các email đăng nhập khi sử dụng workspace, format: email1,email2,email3,...

- LoginUserIDArray - chuỗi userID tương ứng với từng email trong chuỗi LoginEmailArray, format: userid1,userid2,.....

- SubjectUpdateArray - chuỗi các subject có thể được chọn để update appoinment, format: subject1,subject2,subject3....

- TimeSpan - Độ dài thời gian của các appointment - phút

- Is365 - Mode Online or On-Premse

- SearchResource\_Text - Text tìm kiếm resource - vd: room301 -> search ra 1 tập room 3010 -> 3019

- rowsPerPage - Số bản ghi load mỗi trang

- FindColleague\_Text - Text tìm kiếm Colleague -> vd: int -> search full intune

- LoginEmailArrayBase64 - Chuỗi các email đăng nhập tương ứng với chuỗi LoginEmailArray nhưng được mã hóa base64

- SearchResource\_ResourceType - id resource type tìm kiếm

- SearchAttendees\_Text - Text tìm kiếm attendee lúc thêm appointment

6.Các cầu hình trên server:

- Cấu hình push notification cho workspace.

- Cấu hình các room có tiền tố giống nhau, khoảng 10 room: vd room3010 -> 3019.

- Cấu hình > 2 floor plan, chia và vẽ full các room vào các floorplan.

- Có thể add thêm Category, Find Colleague Option để xem kết quả trả về của api liên quan.

- Có thể add thêm Confirm reservation vài resource + parameter ShowQRCode để test case chạy thêm các api liên quan đến confirm reservation.

7.Chạy phần test cho workspace.

- Ctrl + left click vào 3 thread group: "Search,Create - WS", "List,Detail,Master,Update,Confirm - WS", "FindColleague, Get List Appointment, Get Detail - WS".

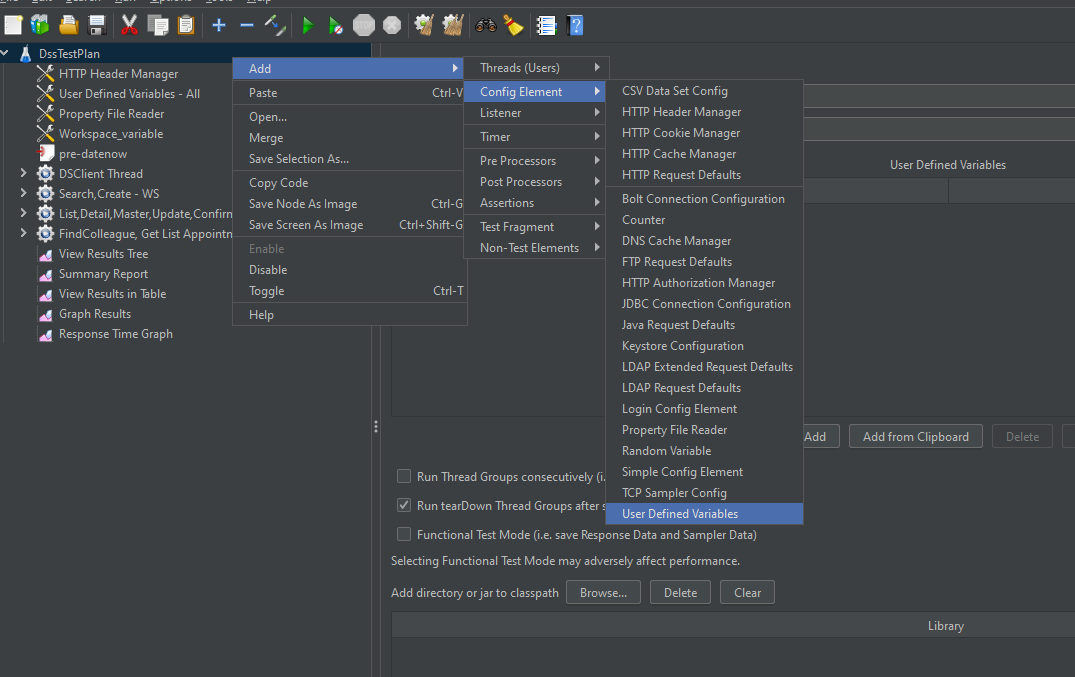
- Right click -> start.

A screenshot of a computer

Description automatically generated

# ­II. Giải thích các Element sử dụng trong project.

## 2.1. User Variable Define.



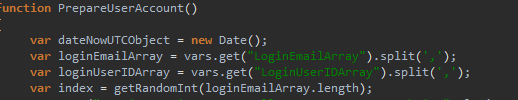
A screenshot of a computer

Description automatically generated

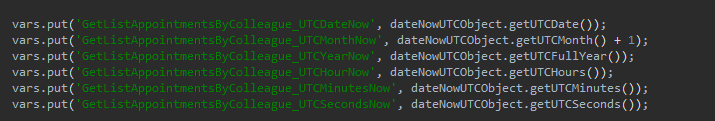
* Element giúp định nghĩa các biến sử dụng trong một luồng (Thread group) giả lập gọi api của 1 user.
* Các biến này sẽ là độc lập giữa các luồng.
* Cách gọi biến để sử dụng trong config: ${tên biến}.



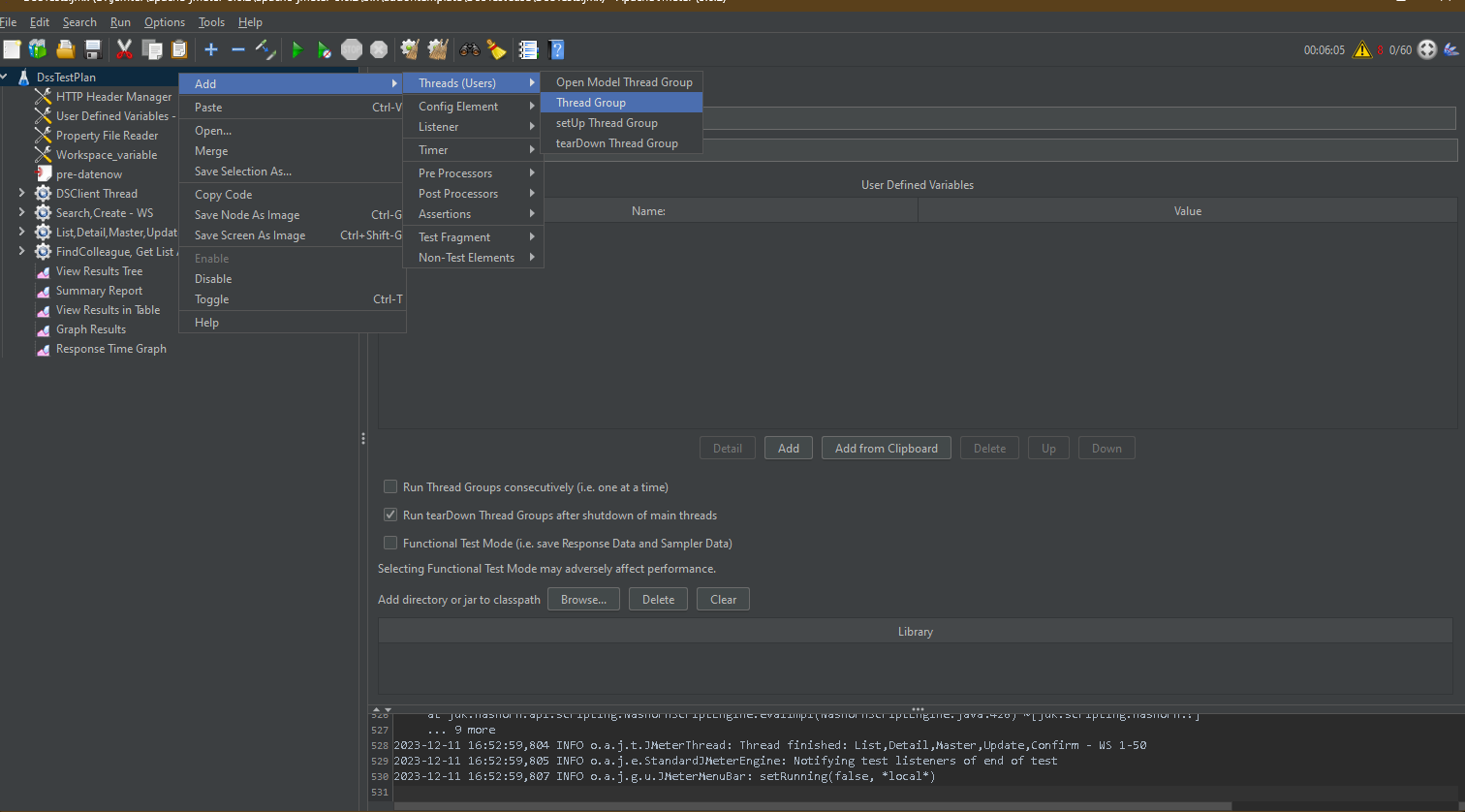
- Cách gọi biến trong code: vars.get(“Tên biến”).



- Cách gán giá trị cho biến: vars.put(“Tên biến”, giá trị).



## 2.2. Thread group

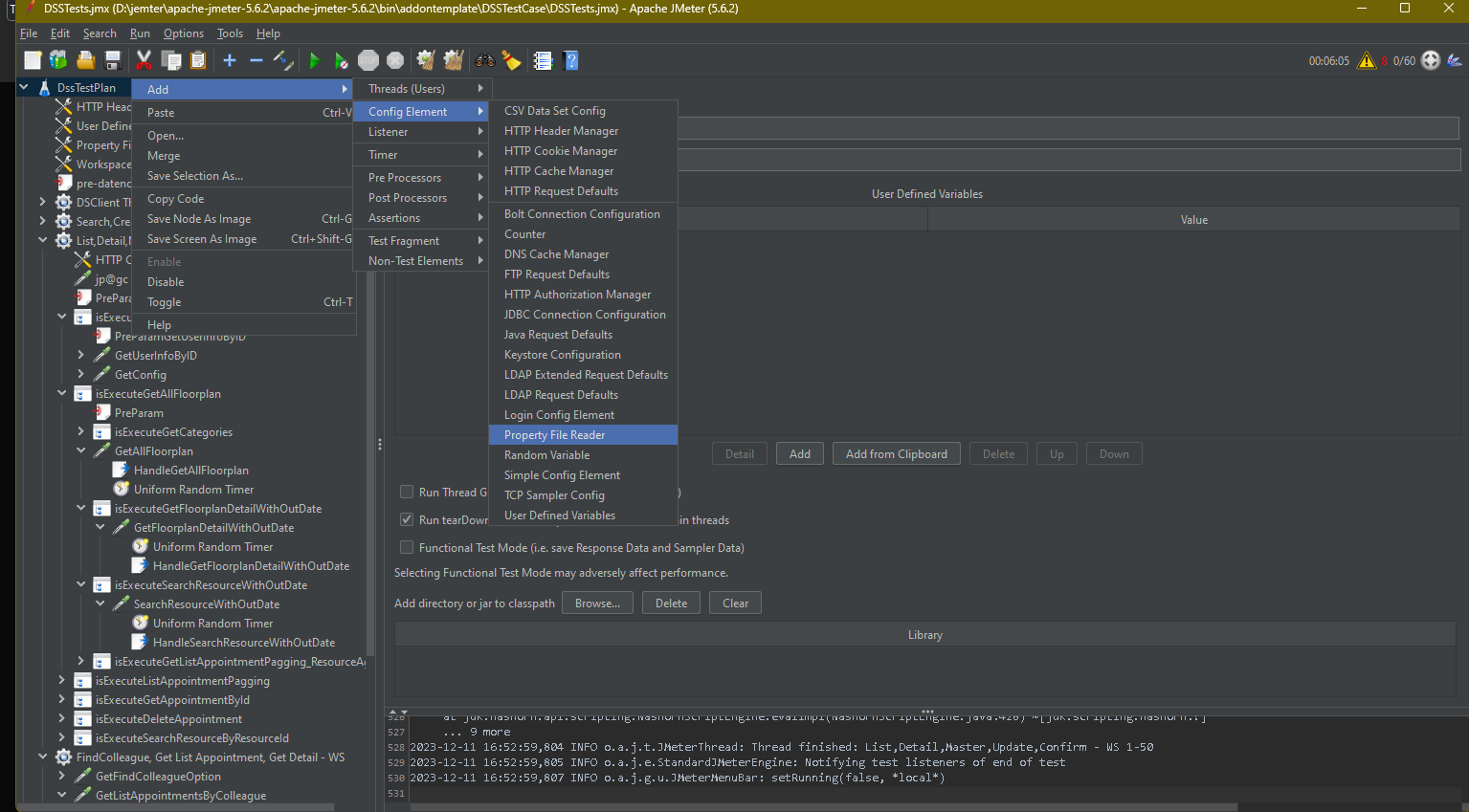


A screenshot of a computer

Description automatically generated

* Chứa kịch bản giả lập luồng gọi api của user. Vd luồng: gọi api list -> gọi api detail -> gọi api edit….
* Có thể cấu hình số lượng user đồng thời thực hiện luồng này, thời gian delay cho đến khi bắt đầu chạy, số lần chạy luồng.

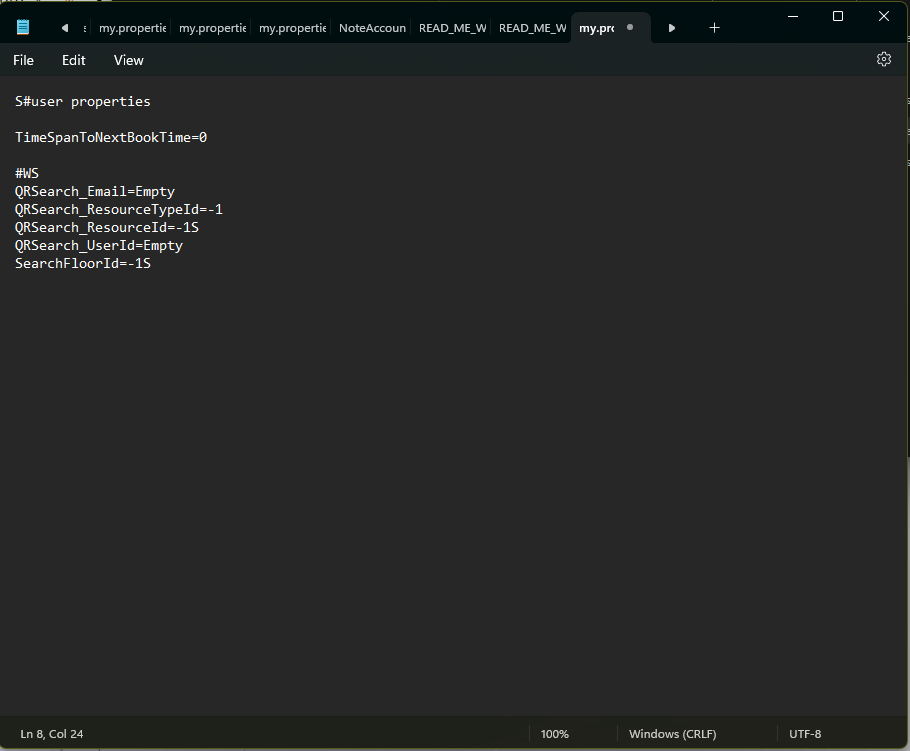
## 2.3. Property file reader



A screen shot of a computer

Description automatically generated

* Element này giúp đọc cái biến từ file cấu hình ngoài.
* Các biến được đọc từ Element này được chia sẻ, sử dụng và cập nhật trên tất cả các luồng.
* Cấu hình biến và giá trị trong file.



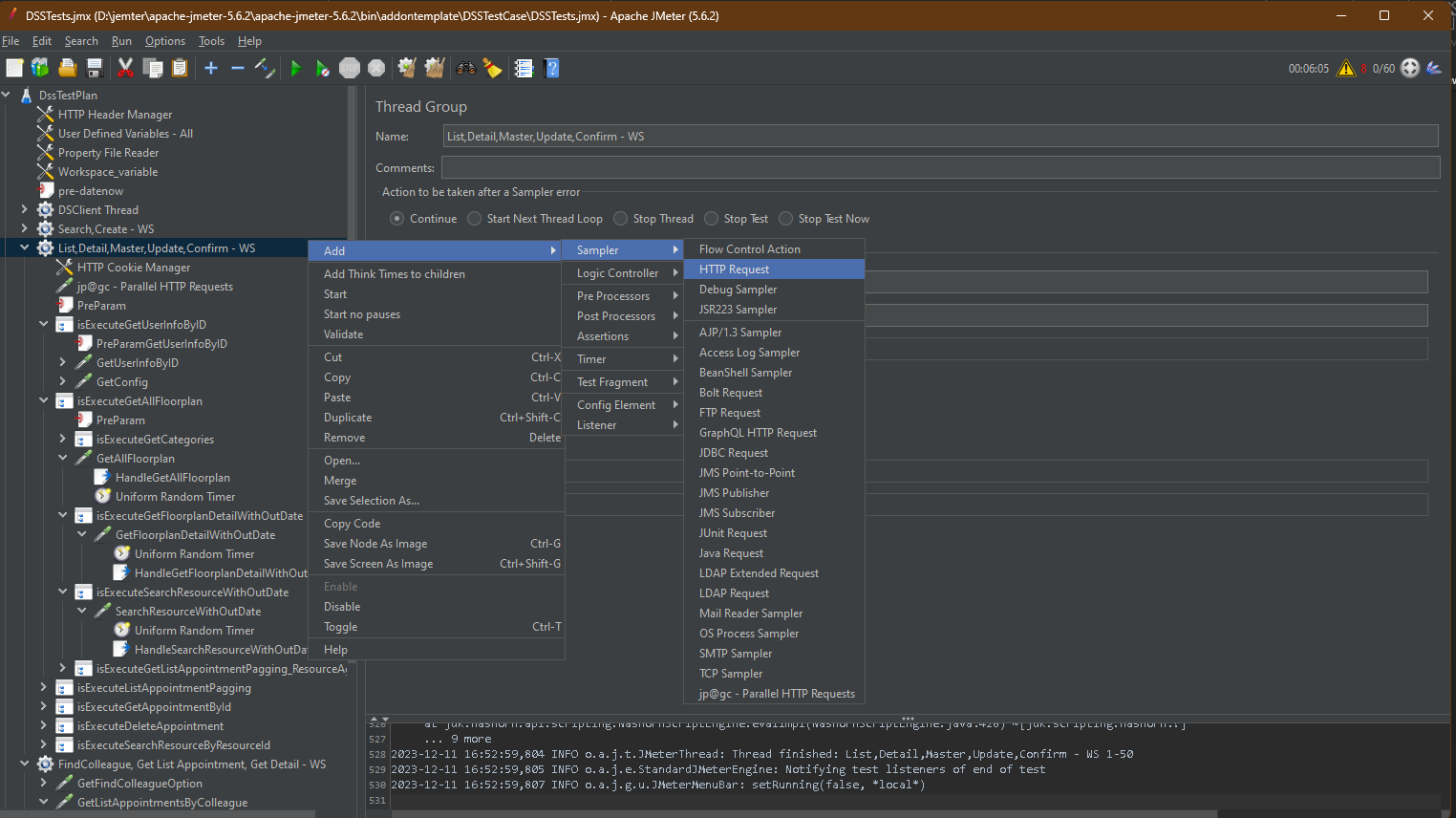
- Cách lấy biến ra sử dụng trong code: props.get(“Tên biến”).



- Cách gán gia trị cho biến: props.put(“Tên biến”, Giá trị).



## 2.4. Http request

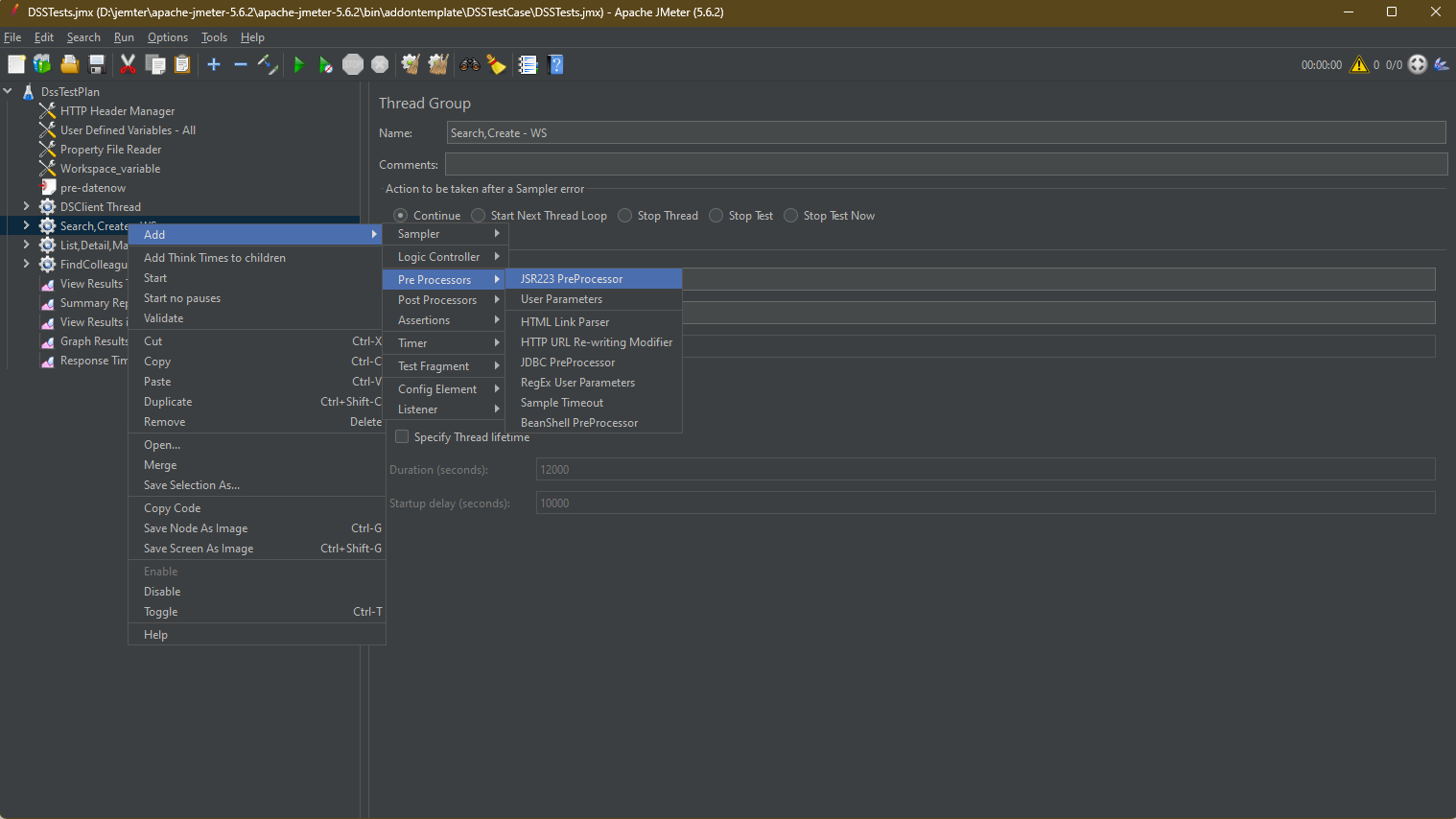


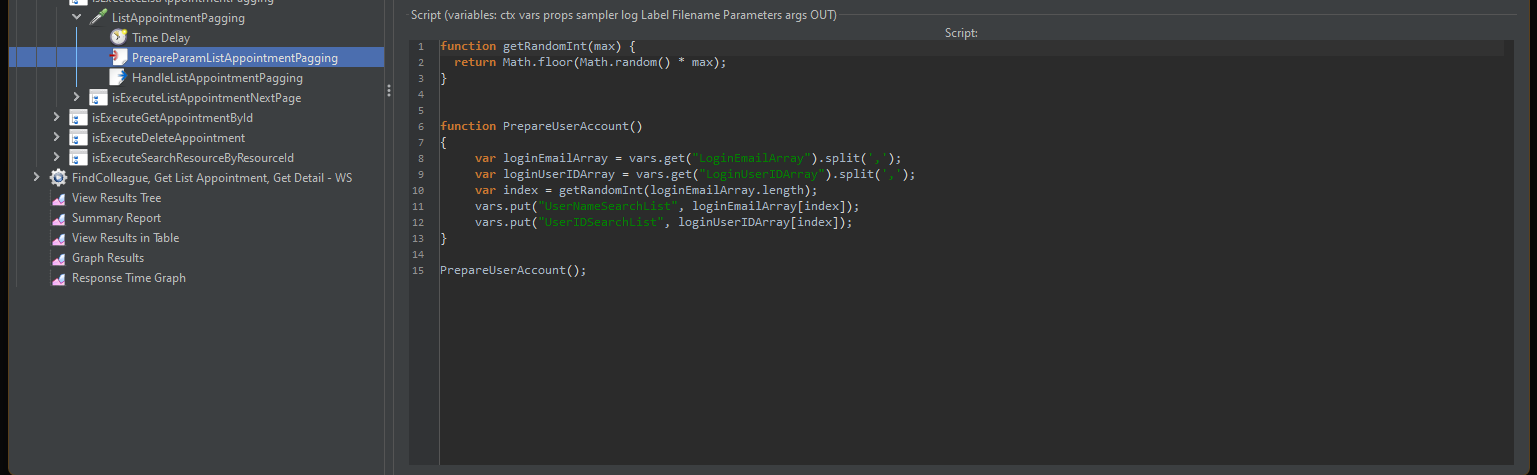
A screenshot of a computer

Description automatically generated

* Element thực hiện gọi api.

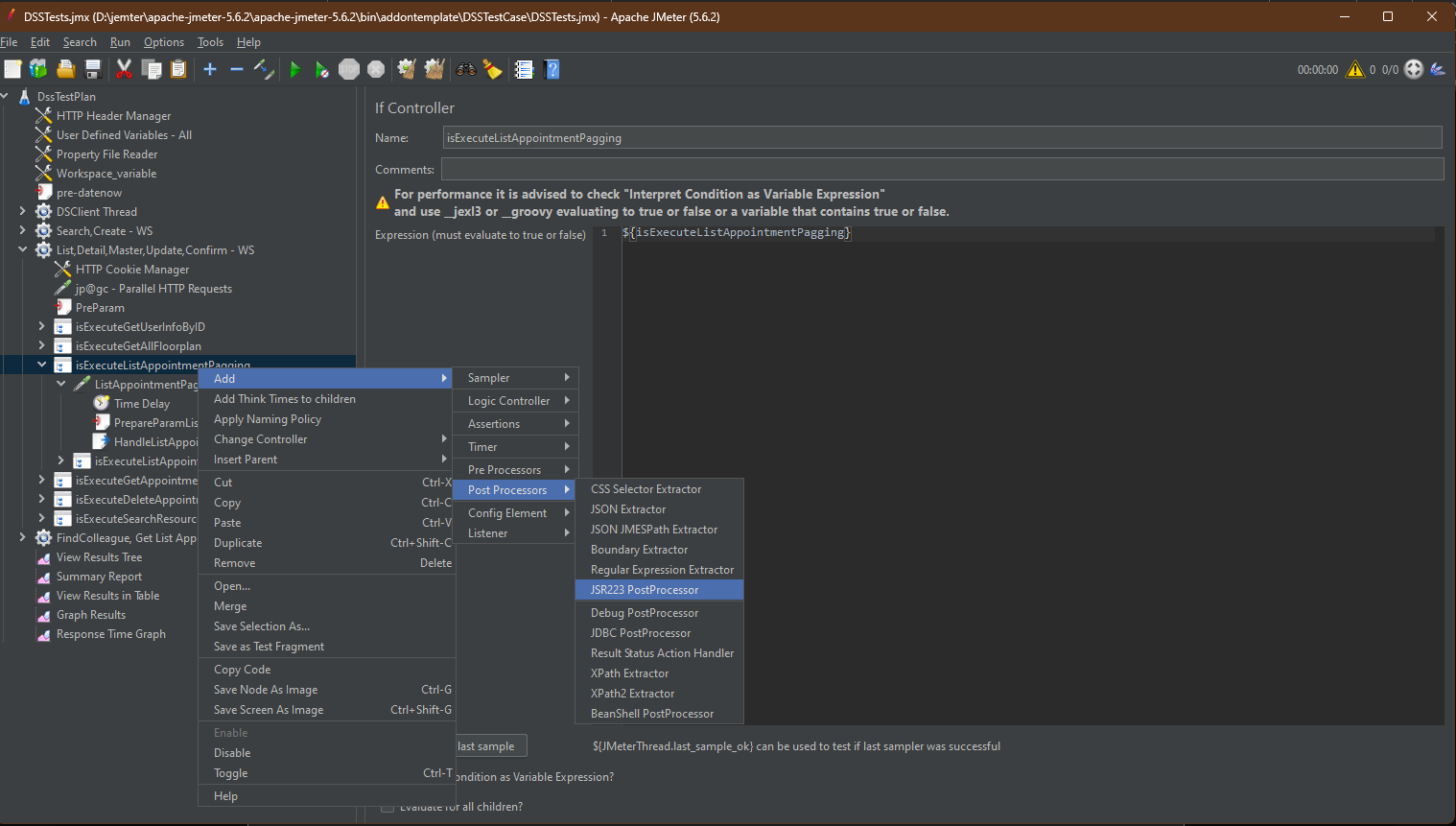
## 2.6. Pre procesors – JSR223 PreProcessor





* Element này có thêm cho các element khác như: “Test plan”, “Thread group”, “Controller”, “Sample”.
* Element được đính kèm PreProcessor sẽ chạy các tác vụ trong PreProcessor trước rồi mới thực hiện tác vụ của mình.
* Ví dụ với ảnh trên, trước khi thực hiện gọi api ListAppointmentPagging thì chương trình sẽ thực hiện PreProcessor HandleListAppointmentPagging trước để chuẩn bị trước một số biến được sử dụng trong lần gọi api sau đó.
* Với PreProcessros JSR223 hỗ trợ code js, python, java…

## 2.7. PostProcessor - JSR223 PostProcessor.

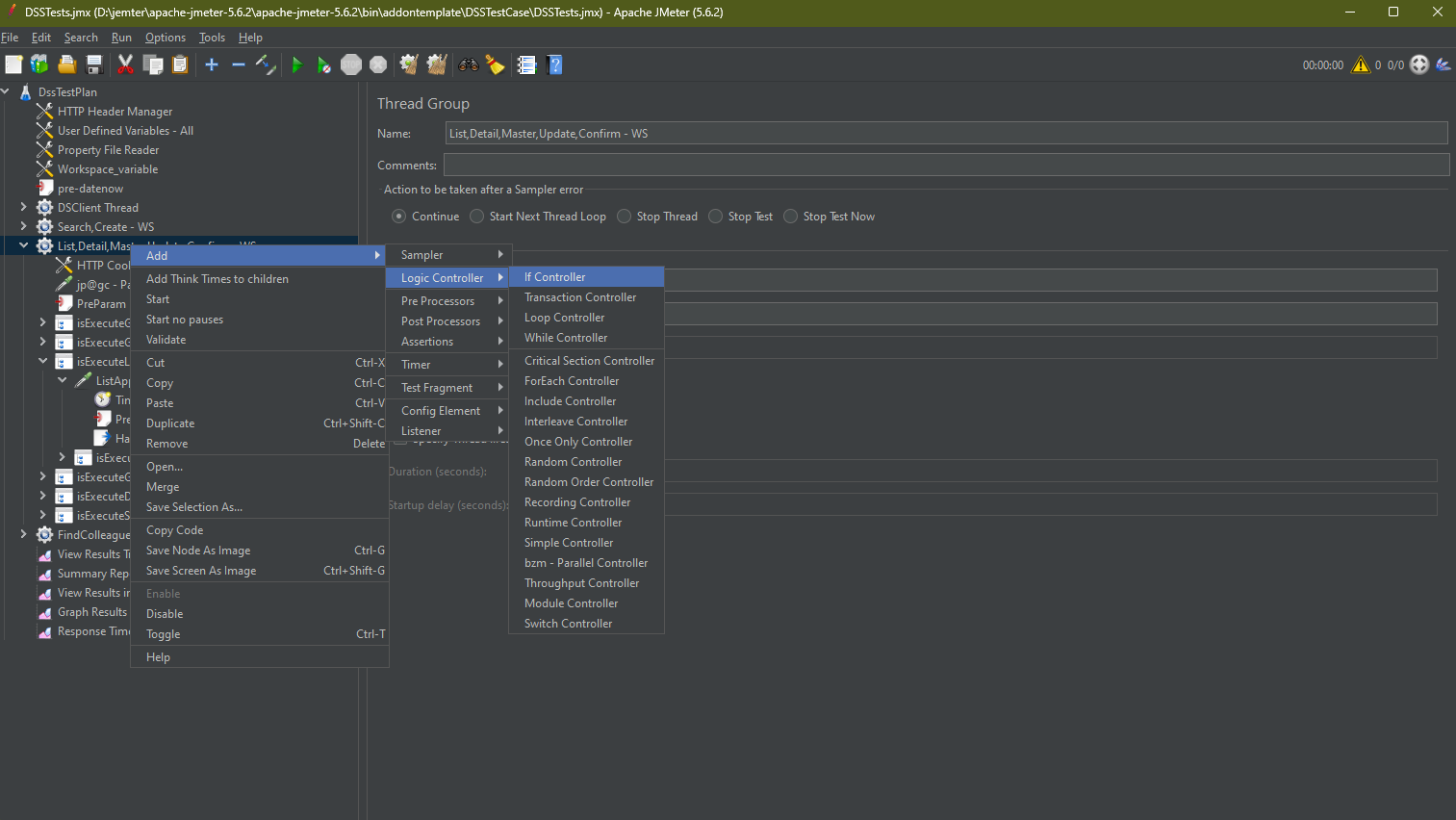


A screenshot of a computer

Description automatically generated

* Tương tự như PreProcessor nhưng PostProcessor thực hiện sau tác vụ chính.

## 2.8. If controller.



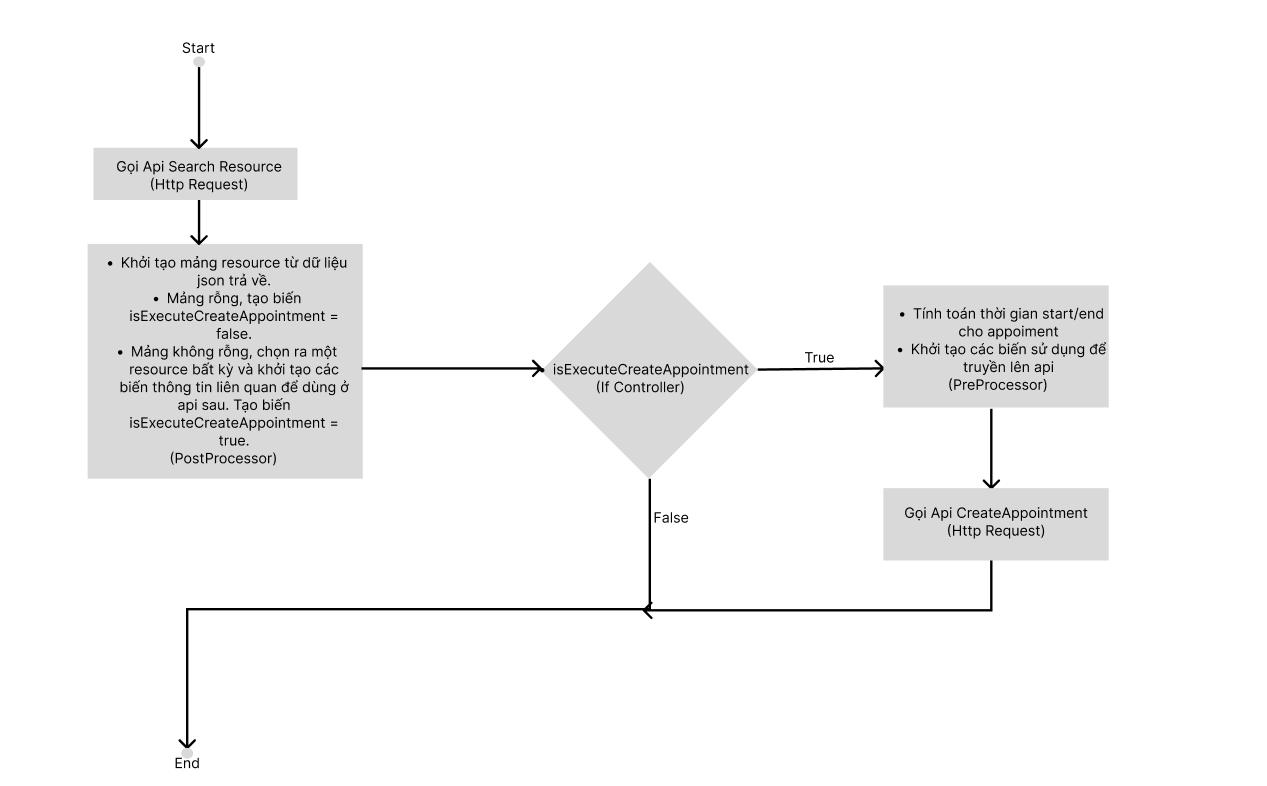
A screenshot of a computer

Description automatically generated

* Element quyết định việc có thực thi các tác vụ dưới cấp mình hay không.
* Element dựa vào biểu thức hoặc biến có kết quả là true hoặc false để quyết định việc thực thi.

# III. Cách dựng api auto test.

* Cố gắng dựng kịch bản gọi api giống nhất với kịch bản sử dụng của user.
* Kết quả của api trước sẽ làm param đẩy lên cho api sau.
* Sử dụng if controller để điều hướng luồng, kịch bản chạy theo nhiều case khác nhau.
* Ví dụ:



# VI. Các lưu ý khác.

4.1. Lệnh lấy kết quả trả về dạng string json: prev.getResponseDataAsString().

4.2. Chọn ngôn ngữ compile trong các processors.

