JIRAE, Vol. 4, No. 2, October 2019, 40-45
e-ISSN 2407-7259

DOI: 10.9744/jirae.4.2.40-45

Keyword Recorder for Robot Framework

Amelia Kezia Gunawan
Petra Christian University, Jl. Siwalankerto 121-131, Surabaya 60236, Idonesia
ameliakezia.g@gmail.com

Abstract. In this era where Agile's popularity is growing fast in the IT industry, test automation becomes more and more crucial for Agile. However, test automation is time-consuming. For this reason, the author creates a tool named Keyword Recorder to create and generate test scripts — the most time-consuming part of test automation — intending to find a solution to speed up the test automation process. Keyword Recorder is a standalone program for web testing and is made to generate scripts for Robot Framework. It consists of two components: a browser and a recorder. The browser is where the user does the web testing and the recorder's job is to capture all the user actions along with the web elements during web testing and generate a Robot Framework's test script based on the recorded actions.

Keywords: Web testing, robot framework, test automation, test scripts.

1. Introduction

Agile is booming. It revolutionized the world in such a way that most IT companies acknowledge they need to be more Agile to survive. But Agile is not complete without test automation. To maximize the effectiveness and speed of Agile while ensuring the software quality, test automation is necessary. Creating test scripts for test automation, however, is time-consuming.

The goal of this project is to create a prototype called Keyword Recorder for web testing, which is capable of creating and developing test scripts for Robot Framework.

The author will do research to find the best possible solution for the prototype, implement the solution for the prototype and test it. The second section explains the research, such as the research question and the answer. The third section explains about the project progress and results. The fourth section explains the conclusion.

2. Research

The following are sub-questions for "How to create a keyword recorder for Robot Framework?"

2.1 What is Testing?

To know what is testing, the author wants to know the reason of why testing is necessary in the first place. Results shows that testing is needed because software failures happen, and human error is one of the main reasons it happened. In order to avoid the software failures, we either avoid making errors or find the error and fix them. Testing can contribute to both of these processes.

But how does testing really works in practice? To make it easy to understand, let's compare testing and debugging side by side.

From the comparison, we know that the goal of testing is to find defects. Thus, it is safe to assume that a test is considered good if it finds a defect when there is a defect present. A test that does not find defects will not increase the software quality, yet a test that finds defect creates a chance to increase the software quality.

After the author know what testing is and why testing is necessary, the author wants to know how testing is done in Agile Software Development Model.

Table 1. Testing vs Debugging [1]

Testing	Debugging
Performed by Tester	Performed by the developer or programmer
The purpose of testing is to find and report defects	The purpose of debugging is to find the cause of bugs or defects in code
Didn't correct defects	Correct defects
Check whether changes made by developer affect the other part of the system or component	Check whether the changes are resolved on the system where the correction was made
Can be automated	Can't be automated
Gives confidence that the software meets the requirements	Doesn't give confidence that the software meets the requirement completely
Starts with the predefined condition and expected results	Starts with the unknown condition and unexpected results
The testing process can and should be planned, designed and scheduled	The process can't be constrained
Can be done by an outsider	Can only be done by an insider
Can be done without design knowledge	Can't be done without proper design knowledge

Agile Software Development Model

Agile, or cyclical model, separate the project into multiple 'sprints', the sprints should have the same duration and the duration ranges from 2–8 weeks. The whole development process is done, from defining requirements to development and testing, within every sprint. At the end of each sprint, or, at the end of the software development process, a prototype is delivered, and a meeting will be conducted to create a plan the next sprint.

Because the development processes are repeated multiple times in a project, testing is done from the beginning of software development life cycle (1st sprint) until the end of the software development life cycle (last sprint).

As explained before, in order to make an impact to the error, testing must be done from the very beginning until the end of software development process. This type of software models makes it possible for a test to find a defect early, making it cheaper to fix.

2.2 Why Test Automation is Necessary?

To know why test automation is necessary, first, the author needs to know what text automation exactly is. What is the difference in automation testing and manual testing? What are the advantages and disadvantages of each testing method?

As you can guess, manual testing means that test cases are manually executed without using an automation tool. This is usually done when the project is still at the beginning or early phase of the software development life cycle because testers need to ensure that the test case does not fail by testing it manually first before automating it. It is also commonly used to test GUI or when writing test scripts for automation testing takes more time than executing it manually.

Manual testing advantages is that it is practical to use if the test cases only need to be executed once or twice. It is also cheap and faster if used in short term project. However, Manual testing is time-consuming when doing regression testing, and the test result is not accurate because there is a chance human error will happen.

Automation testing executes the test cases with the help of an automation tool. It is usually used to automate repetitive test. However, test scripts need to be created first in order to do test automation, and the process itself is time-consuming.

The advantages of automation testing are that the result will be accurate, as there is no human error. It is faster in term of test execution, not when writing test scripts. It is used to automate the repetitive test case, meaning it will be cheaper than manual testing in long term project. The test scripts are also reusable. However, the disadvantages are that it will be expensive on the initial investment, and it requires a lot of maintenance if the project requirement were to change continuously. It also cannot be used to automate certain test cases, such as captcha [2].

After the author knows what test automation is, along with the advantages and disadvantages, the author wants to know why test automation is necessary.

Continuous delivery is one of the agile methodology's principles. It means that the developer keeps on making and delivering the changes in the program. To give confidence in the program quality each time it is delivered, continuous testing is necessary.

For every change made in the program, testers need to make sure that the change did not affect other parts of the programs. Meaning that they need to do regression testing and retesting; testing the old test cases repetitively which will be time-consuming if it were to be done with manual testing. Therefore, test automation is necessary, especially in Agile methodology [3].

2.3 What is Robot Framework?

We know that automation tool is required for test automation, but why does the automation framework is created? As explained, test scripts need to be created first before doing test automation. For example, what if multiple scenarios need to be tested, using one same script, but with different kinds of data? That means we need to create multiple copies of the same script, each with different data. Moreover, what if we need to change the object property of the scripts? We need to go through all the test scripts to change it. The script is indeed re-usable, but hard to maintain. Copy pasting and manually changing the data takes a lot of time. Not only that, but the test automation process should also be planned to get an effective result.

This is why test automation framework is created. It helps us organize the tests scripts, making it maintainable by providing us the ability to re-use components or code in multiple test scripts. It stores the test data in a different location but accessible by all the test script, so we can change the test data in that location instead of changing every single test script.

Not only that, its other advantages are that it improves the test efficiency, has a low maintenance cost, has a minimal manual testing intervention, and provides maximum test coverage.

There are different types of automation framework, which is [4]:

- Linear
- Modular based
- Data Driven
- Library Architecture
- Keyword Driven
- Hybrid

Robot Framework itself is a keyword driven framework which is originally created by Nokia [5]. It was first used to test the Nokia product, but version 2 was open sourced in 2004, followed by the release of version 3 in 2007. Robot Framework is utilized for acceptance testing and uses Python and java as its programming language.

The documentation and source code are released on GitHub under the Apache 2.0 license. The project itself has more than 80 contributors and 80 releases.

2.4 How to write tests in Robot Framework?

To make a prototype which generate scripts for Robot Framework, first the author needs to know how to make a test script in Robot Framework itself.

This research is done in windows environment, so the installation and setting up the environment manual is for windows.

After reading the user manual for Robot Framework, the author found out that Robot Framework's test data is divided into different sections, which is [6]:

Settings

The section where test libraries, resource files and variable files are imported. Metadata for test suites and test cases is also defined in this section. The accessible settings are: documenttation, tags, setup and teardown, test templates and timeouts.

Variables

The section to define the variables which will be used in test data.

Test Cases

This section is used to create test cases from keywords

Keywords

This section is used to create customized keywords from lower level keywords.

2.5 How to automate tests in Robot Framework

The prototype needs to be able to record the user action when they're doing web testing and to generate a test script in Robot Framework format. But Robot Framework itself is not build for web testing. So how do we create a test script that can be used for web testing if the Robot Framework built in keywords does not support web testing?

After some research, the author found out that web testing is possible in Robot Framework using selenium library. For this reason, author will make the prototype to be able to recognize selenium library's keywords. So, the author will add this library into the setting section for the test scripts.

2.6 Are there Recorder Tools available for Robot Framework?

The author then researches recorder tools that are already available in the market, which is made especially for Robot Framework. There are only one recorder tools so far that is made exclusively for Robot Framework, so the author compares it with another popular recorder tools. The results are:

- RobotCorder, a browser extension, generate scripts into Robot Framework format [7]
- Katalon Recorder, a browser extension, scripts are generated in selenese, and it needs to be changed into selenium library keywords of Robot Framework [8]
- Katalon Studio, a desktop application, scripts are not generated in Robot Framework [9]

The author examines the capability of the 3 tools based on the following questions:

- Can it record the change if the tab address is changed using the address bar?
- Can it record multiple tabs?
- Can it record tabs in multiple browser windows?
- Does it have a play, pause, and stop button?
- Can it record pop up alert on websites?
- What does it record?
- Can the tools support more than one test scripts?
- What other functionality do the tools have?

Based on this research, the author determines the main features to be added and also the additional features which are good to have.

2.7 Extension or Standalone program?

The author then needs to know what type of prototype to make, is it going to be an extension program like the two recorder tools (Katalon Recorder and RobotCorder) or a standalone program like Katalon Studio?

Based on the research in 2.6, if the author were to make an extension, then it will only work on the browser which the author makes the extension for. For example, if the author makes a chrome extension, the prototype will only work on chrome, and the author needs to create another prototype for Mozilla Firefox. Moreover, based on the research result in 2.6, the extensions are not able to record if the user changed the page address in the address bar and they cannot test multiple tabs. It only records the tab where the extension is opened and did not record other tabs.

However, if the author were to make a standalone program, like Katalon Studio, it can record multiple tabs, supports multiple browsers without having to create a different application for different browsers.

Based on the results above, the author chooses to make a standalone program because the author needs the prototype to be able to test multiple tabs, and possibly, multiple browsers.

2.8 What Programming Language is best suited for the Prototype?

The author needs to know which programming language to use. For that reason, the author researches top 3 most used programming languages in GitHut. [10]

Results show that JavaScript is in the first place. It is the most used programming language and is supported by multiple browsers. It is also implemented on most web applications to interact with web elements.

Java came in second place. Code is written using static type in this programming language, which makes the error to be detected in compile time rather than run time and the program to perform faster.

Python is in third place. This programming language uses dynamic typing, which makes it flexible and simple. They also have a wide variety of libraries dedicated to data science, machine learning, and others.

Because the program needs to interact with web elements, the author chooses JavaScript as the programming language. Python is the second choice because of its data processing capabilities.

2.9 Full JavaScript program or JavaScript + Python program?

Based on 2.8, the option is to make either a full JavaScript program or a combination of Python and JavaScript program.

If the author were to make a full JavaScript program, the design would be more modern and user-friendly because the program's design can be written in CSS and HTML. Electron and NW.js, for example, are some of the frameworks available for creating a desktop application with JavaScript, HTML, and CSS. Both of them uses chromium engine and node.js.

However, if the author were to make a JavaScript + Python program, the author can use Python's library for the data processing and as a backend. Then use JavaScript for retrieving the web elements from the browser. However, there are no frameworks or programs available to create this type of program, so the author needs to do additional steps to add another programming language inside a programming language.

Based on the results, JavaScript + Python program is a better option as the author needs to do data processing with

Python for the web element that is retrieved from the web page using JavaScript. However, it is considered illegal to inject JavaScript external websites.

An extension can insert the JavaScript internally because it is opened from the browser itself. But desktop apps are considered as an external source for browsers.

After discussing this issue with an expert, the expert's suggestion is to create the program to be a wrapper of the web page that is going to be recorded. By opening the web page inside of the program instead of opening the web page with an external browser, the author can insert JavaScript internally into the program.

Based on the suggestion, the author decided to create a browser. Meaning that the prototype will be a combination of browser and recorder itself, which means that the websites are opened internally inside the prototype and not on another browser and the user can start the record function inside the prototype itself.

With this solution, the author chooses to make a full JavaScript program instead of JavaScript + Python program. The reason is that the frameworks which are used to create a JavaScript program, such as Electron and NW.js, utilizes Chromium Browser, meaning that its functions are similar to a browser.

3. Implementation

After the research is completed, the author makes the prototype according to the research results. The author uses Electron to build the desktop application and Visual Studio Code as the text editor.

The author then learns JavaScript and Node.js in order to create an application in electron. In addition, the author reads the electron documentation for further understanding.

The author uses Python version 3.7.2 and the latest version of Robot Framework, which is v3.1.1. The author use Eclipse as the IDE and uses RED Robot Editor which can be found in Eclipse Marketplace.

Afterwards, the author creates the prototype based on the research results.

The features which is implemented are:

1. Prototype has the basic browser functionalities.
   - The author does a research on browser's functionalities. Afterwards, the author creates a list of browser functions based on research result and client's feedback.
   - The following are the contents of the list containing the required browser function:
     - The browser can have multiple tabs
     - The browser has an address bar for the user to enter URL
     - If the user does not enter URL in address bar, the browser will open a google page with the user input
     - The user can add tab by pressing an add tab button or choosing "New Tab" options from the application menu
     - The user can reload the tab by pressing reload button
     - The user can go back from the tab by pressing the back button or choosing "Go Back" from the right click menu
     - The user can go forward from the tab by pressing the forward button or choosing "Go Forward" from the right click menu
     - The user can do inspect element by pressing the "Inspect Element" options from the right click menu
     - The user can delete the tab by clicking on the close tab button
     - The user can change the tab's page by inserting URL or text which will be searched by google in the address bar
   - All of the required browser functions are implemented
2. User can start recording anytime in the prototype's browser
   - This feature is requested by the client. The idea is that the user can start the recorder anytime when using the browser. For example, the user can open the prototype's browser, navigate to the page which the user wants the action to be recorded, and press the record button.
   - The extensions have a "record" button that the user can press to start the recording anytime within the page.
   - The author decided to create a record button inside the browser and makes the browser to record user action only after the button is pressed.
3. Prototype can record multiple tabs
   - The author has studied the electron documentation; therefore, the author knows how to make a communications between the tabs to maintain the recording status even if the tab is switched
   - The communication process is implemented, and the prototype can now maintain the recording state even if the tab is different.
4. Prototype can record if the user changes the page from the address bar
   - The idea of this function is that the recorder can record whenever the page URL is changed from the address bar.
   - Because the student knows how to make a communication between tabs, this functionality is implemented successfully
5. Hover on HTML Element will show the element's XPath during recording
   - The idea of this function is to show a red border on the hovered html element and shows the hovered element's XPath on top of the page
   - The author does a research on internet on how to add HTML and CSS style programmatically
   - After the author knows how to insert CSS, the author does a research on how to display a border for every mouse hover which will not cause the html elements to shift or change position
   - The author decides to use outlines instead of border because it does not cause the HTML properties to move
   - The author then does a research on internet about how to write Relative XPath. Therefore, the author can generate XPath for every hovered element.
   - This function is implemented successfully.
6. Recording process can be paused or stopped.
   - The idea of this function is that if the user pauses the recording, then the tool will not record anything until user start the recording again. If the user stopped the

recording and then start the recording again, the old table of recorded contents will be closed and replaced with a new one.
   - This functionality is implemented
7. The table of recorded items shows keyword, object (XPath) and value.
   - The idea of this function is that if the user starts the recorder, a new program will open up and it will contain a table for the recorded items. The table will be updated on every user action and contains the keywords, object and value for each recorded action.
   - This functionality is implemented
8. The XPath of the recorded items is a Rel XPath and can be re-used.
   - The author has already implemented this function when hovering on HTML Element will show Relative XPath
9. User actions on recording process is recorded and mapped as a selenium library's keywords on the table of recorded items.
   - The author has studied the selenium library keywords during research phase. As a result, the student can categorize or map the user action based on the keywords.
   - This tool can record most of the selenium library keywords, not all of it. Because there are some keywords that can not be categorized by the user Actions.
   - The user can also create an item manually inside the table of recorded items if they want to use the keywords which can not be recorded by pressing the add new button on top of the table.
   - This functionality is implemented successfully
10. Show keyword list on table cell if double click
    - The idea is that if the user double clicks on the table cell inside the keyword column, the cell is editable and shows a list of keywords. If the user type "Click" in the table cell, then a list of keyword suggestion such as "Click Element" and "Click Link" will be shown
    - The author searches for solution in the internet on how to show a list of keyword suggestion based on the user input
    - The author then implements the solution successfully
11. Modifiable table.
    - The idea is that the user can change any table cells value by double clicking the cell. The user can also delete a certain table row if the user presses the delete button in that row
    - For editing the cell, the author has already implemented this function before for showing keyword suggestion. Therefore, the author knows how to make the rest of the cells editable.
    - The author then adds a delete button for every row.
    - This function is implemented successfully
12. The table of recorded items can be converted into Robot Framework script

- The idea is that the user can get a test script for Robot Framework based on the table of recorded contents.
- The author has studied how to make Robot Framework test script. Therefore, the author knows which sections is needed to generate a fully functional test script.
- The author implements this function successfully and the generated test script can be used for Robot Framework

The extra functionalities to be implemented are:

- The recorder's able to record multiple window
- The browser can save username and passwords
- The browser has a history of tabs opened
- Can restore browser's session if browser is closed and opened again.
- Using an algorithm (i.e. Deep Learning) for extracting the XPath to get more accurate result
- User can create script inside the recorder

3.1 Testing

The author has done two types of test:

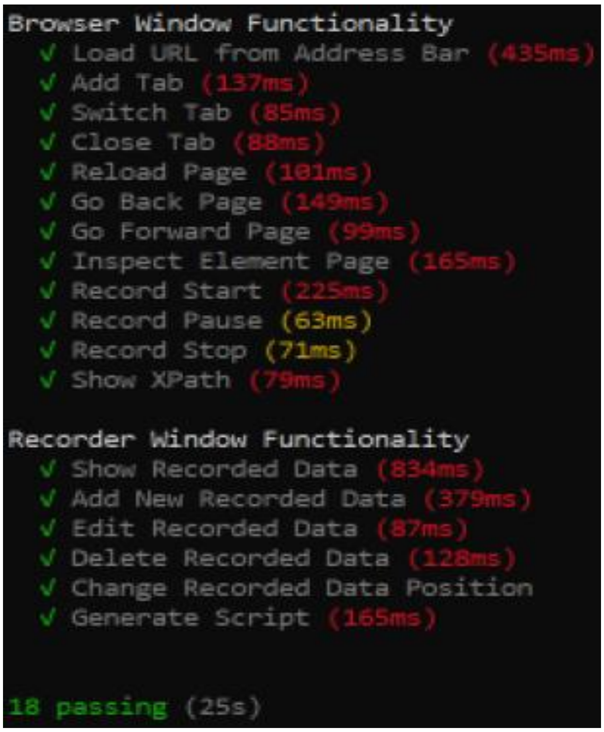
1. Unit Test, to test the functionality with certain input and then check if the output is the same as expected output
2. Acceptance test, to ensure that the prototype meets the user requirements. This is tested by the stakeholder.

Table 2. Features tested

Features	Description
Tab can load URL from address bar	The user can open a URL by entering the URL in the address bar. If the user did not enter a URL, the browser will load a google page for the user input.
Add Tab	The user can add a new tab by pressing the add button. The user can add a new tab by choosing File→"New Tab" in the application menu
Switch Tab	The user can switch tab by clicking on the tab they want to open
Close Tab	The user can close a tab by pressing the close tab button which will appear if the user hovers a tab
Reload Page	The user can reload the page by pressing the reload button
Go Back Page	The user can make the page go back by pressing the back button
Go Forward Page	The user can make the page go forward by pressing the back button
Inspect Element Page	The user can open inspect element of the page by choosing "Inspect Element" from the right click menu
Record Start	The user can start recording by pressing the play button anytime within the browser. Recording window will appear where the user can see the recording data on the table of recorded items. If the user has pressed a stop button before, the opened recorder window will be closed, and a new recorder window will be opened.
Record Pause	The user can pause the recording process by pressing the pause button
Record Stop	The user can stop the recording process by pressing the stop button
Show XPath	The user can see the hovered element's XPath while recording
Show recorded data	The user can see the table of recorded data in the recorder window
Add new recorded data	The user can manually add keyword, locator, and value by pressing add new button on the recorder window
Edit recorded data	The user can change the keyword, locator, or value by double clicking the table cell

Features	Description
Delete recorded data	The user can delete a row of recorded data by clicking the delete button which is located in the same row
Change recorded data position	The user can change the position of the recorded data by clicking and dragging on the table
Generate Script	The user can generate a script by clicking the script tab in the recorder window

The functionalities above passed the acceptance test. Below are the results of the unit test:



```
Browser Window Functionality
  √ Load URL from Address Bar (435ms)
  √ Add Tab (137ms)
  √ Switch Tab (85ms)
  √ Close Tab (88ms)
  √ Reload Page (101ms)
  √ Go Back Page (149ms)
  √ Go Forward Page (99ms)
  √ Inspect Element Page (165ms)
  √ Record Start (225ms)
  √ Record Pause (63ms)
  √ Record Stop (71ms)
  √ Show XPath (79ms)

Recorder Window Functionality
  √ Show Recorded Data (834ms)
  √ Add New Recorded Data (379ms)
  √ Edit Recorded Data (87ms)
  √ Delete Recorded Data (128ms)
  √ Change Recorded Data Position
  √ Generate Script (165ms)


18 passing (25s)
```

Figure 1. The result of the unit test.

4. Conclusion & Recommendation

Nowadays, more and more companies start adopting Agile. To get the maximum results from Agile, test automation is needed. However, test automation is time-consuming because the tester needs to write test scripts.

Accenture has a large client base which implements Agile and uses Robot Framework for the Test Automation. The problem is that most of tester's time is spent on doing test automation, especially in writing test scripts. This can cause the tester unable to keep up with the developer's pace as there is a lot of changes to be tested and there is not enough time to test it all.

The goal of this graduation project is to create a prototype called Keyword Recorder to solve the problem mentioned above, to create and develop a test script. The project development uses Agile Scrum method, in which the projects are distributed into a sprint of two weeks and the author needs to show the progress of the research/prototype to stakeholder/company mentor at the end of each sprint.

The prototype consists of two programs, browser and recorder, and has been proven to be able to create test scripts faster than the manual method of tester writing the test scripts. The functionalities which are implemented are:

- The prototype's browser has the basic functionalities
- Recording can be started, paused, and stopped anytime within the browser and it can record multiple tabs
- The prototype's recorder can record if the user switch tab or change page using address bar
- The prototype's browser can record the user's actions in the browser and map it to the selenium library keywords
- The prototype's browser will show XPath on hovered HTML element during recording
- The table of recorded items is modifiable and contains keyword, object ID / locator, and value.
- The XPath is a relative XPath and can be reused
- The table of recorded items can be converted into Robot Framework script

All of the main functionalities are implemented and tested while the extra functionalities are not implemented due to lack of time.

Recommendations for future developments are:

- The recorder's able to record multiple window
- The browser can save username and passwords
- The browser has a history of tabs opened
- Can restore browser's session if browser is closed and opened again.
- Using an algorithm (i.e. Deep Learning) for extracting the XPath to get more accurate result
- User can create script inside the recorder

References

1. Difference between Testing and Debugging. (2018, July 14). Retrieved July 11, 2019, from https://www.softwaretestingclass.com/difference-between-testing-and-debugging/
2. Automation Testing Vs. Manual Testing: What's the Difference? (2019, May 24). Retrieved July 11, 2019, from https://www.guru99.com/difference-automated-vs-manual-testing.html
3. How Important is Test Automation in a Software Project? (2017, August 14). Retrieved July 11, 2019, from https://www.seguetech.com/important-test-automation/
4. Types of Test Automation Frameworks | Software Testing Material. (2019, February 2). Retrieved July 11, 2019, from https://www.softwaretestingmaterial.com/types-test-automation-frameworks/
5. Robot Framework. (n.d.). Retrieved July 11, 2019, from https://robotframework.org/
6. Robot Framework User Guide Version 3.1.2. (n.d.). Retrieved from https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html
7. Sohwendy/Robotcorder. (n.d.). Retrieved July 11, 2019, from https://github.com/sohwendy/Robotcorder
8. Katalon-studio/katalon-recorder. (n.d.). Retrieved July 11, 2019, from https://github.com/katalon-studio/katalon-recorder
9. Free automation testing solution developed by Katalon LLC. (2019, June 19). Retrieved July 11, 2019, from https://en.wikipedia.org/wiki/Katalon_Studio
10. GitHut - Programming Languages and GitHub. (n.d.). Retrieved July 11, 2019, from https://githut.info/
11. 5 Best JavaScript Frameworks for Desktop Apps. (2019, February 14). Retrieved July 11, 2019, from https://brainhub.eu/blog/javascript-frameworks-for- desktop-apps/