

Tool Support for Testing

Software Testing: INF3121 / INF4121

Summary:

Types of test tools

Tool support for testing | Test tool classification

Tools for ...

Test management / Static testing / Test specification / Execution and logging

Performance and monitoring / Specific testing needs

Effective use of test tools

Benefits and risks | Special considerations for tools

Introducing a test tool into an organisation



Part I: Close-ended questions

Question 1

Which **tools** help support **static testing**?

- a. Static analysis tools and test execution tools
- b. Review process support tools, static analysis tools and coverage measurement tools
- c. Dynamic analysis tools and modelling tools
- d. Review process support tools, static analysis tools and modelling tools



Question 2

Which **test activities** are **supported** by **test harness** or **unit test framework** tools?

- a. Test management and control
- b. Test specification and control
- c. Test execution and control
- d. Performance and monitoring



Question 3

What are the potential **benefits** from **using tools** in general to **support testing**?

- a. Greater quality of code, reduction in the number of testers needed, better objectives for testing
- b. Greater repeatability of tests, reduction in repetitive manual work, objective assessment
- c. Greater responsiveness of users, reduction of tests run, objectives not necessary
- d. Greater quality of code, reduction in paperwork, fewer objectives to the tests

Question 4

What is a potential **risk** in **using tools** to **support testing**?

- a. Unrealistic expectations, expecting the tool to do too much
- b. Insufficient reliance on the tool, i.e. still doing manual testing when a test execution tool has been purchased
- c. The tool may find defects that are not there
- d. The tool will repeat exactly the same thing it did the previous time



Question 5

Which of the following are **advanced scripting techniques** for **test execution** tools?

- a. Data-driven and keyword-driven
- b. Data-driven and capture-driven
- c. Capture-driven and keyhole-driven
- d. Playback-driven and keyword-driven



Question 6

Which of the following would **NOT** be **done** as part of **selecting a tool** for an **organisation**?

- a. Assess the organisational maturity, strengths and weaknesses
- b. Roll out the tool to as many users as possible within the organisation
- c. Evaluate the tool features against clear requirements and objective criteria
- d. Identify internal requirements for coaching and mentoring in the use of the tool

Question 7

Which of the following is a **goal** for a **pilot phase** of **introducing** a new **tool** to an **organisation**?

- a. Decide which tool to acquire
- b. Decide the main objectives and requirements for this type of tool
- c. Evaluate the vendor including training, support, and commercial aspects
- d. Decide on standard ways of using, managing, storing, and maintaining the tool and the test assets

Question 8

Pair the following **testing tools** with the main **activity** they support

Tools for test execution and logging	Used for traceability of tests, test results and incidents. Used to connect tests with their originating documents, such as requirements specifications.
Tools for static testing	Enable tests to be executed automatically using stored inputs and expected outcomes.
Tools for performance and monitoring	Used for testing the structure and dependencies of the code. Used to measure code coverage with tests.
Tools for test management	They simulate a load on: - An application / A database / A system environment

Question 9

Test comparators are used when the executed **test generates a lot of output**. In order to **validate** the output against an **oracle**, one **needs** to use a **test tool**.

E.g. Send SMS with less than 10 special characters to 20.000 users

- a. True
- b. False



Question 10

A potential _____ of **using** a test **tool** is the **reduced repetitive manual work**.

E.g. When running regression tests, re-entering the same input data, etc.



Question 11

Which of the following are **benefits** and which are **risks** of **using tools to support testing**?

1. Over-reliance on the tool
2. Greater consistency and repeatability
3. Objective assessment
4. Unrealistic expectations
5. Underestimating the effort required to maintain the tool
6. Ease of access to information about tests or testing
7. Repetitive work is reduced

Question 11

Which of the following are **benefits** and which are **risks** of **using tools to support testing**?

- a. Benefits: 3, 4, 6 and 7. Risks: 1, 2, and 5
- b. Benefits: 1, 2, 3 and 7. Risks: 4, 5, and 6
- c. Benefits: 2, 3, 6 and 7. Risks: 1, 4, and 5
- d. Benefits: 2, 3, 5 and 6. Risks: 1, 4, and 7

Question 12

Which **test activities** are **supported** by **test data preparation** tools?

- a. Test management and control
- b. Test specification and control
- c. Test execution and control
- d. Performance and monitoring



Question 13

Consider the following **types of tools**:

1. Test management tools
2. Static analysis tools
3. Modelling tools
4. Dynamic analysis tools
5. Performance testing tools



Question 13

Which of the following **tools** are most **likely** to be **used** by **developers**?

- a. Static analysis tools, modelling tools, and dynamic analysis tools
- b. Test management tools, dynamic analysis tools, and performance testing tools
- c. Test management tools, static analysis tools, and performance testing tools
- d. Modelling tools, dynamic analysis tools, and performance testing tools

Question 14

Which **success factors** are **required** for good **tool support** within an **organisation**?

- a. Acquiring the best tool and ensuring that all testers use it
- b. Adapting processes to fit with the use of the tool and monitoring tool use and benefits
- c. Setting ambitious objectives for tool benefits and aggressive deadlines for achieving them
- d. Adopting practices from other successful organisations and ensuring that initial ways of using the tool are maintained

Question 15

What kind of **interface** can be **used** to **automate** tests?

- a. API – Application programming interface
- b. GUI – Graphical user interface
- c. Both API and GUI
- d. None of the above



Question 16

Which of the following are **advantages of test automation**?

- a. Tests run faster and can be more complex
- b. Tests are run by machines and the results are interpreted by humans
- c. Data sets used in testing can be very simple
- d. The results of running the tests is always the same



Question 17

Which of the following is a **limitation** of **test automation**?

- a. Tests can be very simple
- b. Tests need to be complex in order to be considered for automation
- c. One cannot automate all tests
- d. Data sets used in testing are not stored, therefore tests are not always reproducible



Question 18

Pair the following **approaches** to **automated testing** with their corresponding **description**:

Capture and Replay	The test inputs are extracted or generated with scripts. To automate testing, we reuse one main script together with this data to implement a number of tests.
Data-driven approach	The automated test scripts are built by putting together reusable smaller scripts, name keywords.
Keyword-driven approach	Tools are used to capture interactions with the system under test (SUT) while performing the sequence of actions as defined by a test procedure.

Question 19

Which of the following **factors** must be **considered** when **transitioning** from **manual** to **automated** testing?

1. Frequency of use of the tested feature
2. The upcoming release date
3. How complex it is to automate the test
4. The current cyclomatic complexity of the code

Question 20

A test **manager** does **not need** to take into account **re-educating** the team when **preparing** to go from **manual** to **automated** testing

- a. True
- b. False



Part II: Exercises and Open-ended questions

Exercise 1: Test Automation Tools

Browse the internet to find an **example** of a **tool** used for **test automation**.

Explain briefly how the tool **works**.



Exercise 1: Test Automation Tools

Example: **Selenium IDE**

Integrated Development Environment for testing

Record, **Edit**, and **Debug** tests

Firefox extension

Features

Record and **playback** of test **scripts**

Intelligent **field selection**

Walkthrough of test runs

Save **tests** as HTML, Ruby scripts, other **formats**



Exercise 1: Test Automation Tools

Example: Selenium IDE

Presentation at start-up

Four important sections

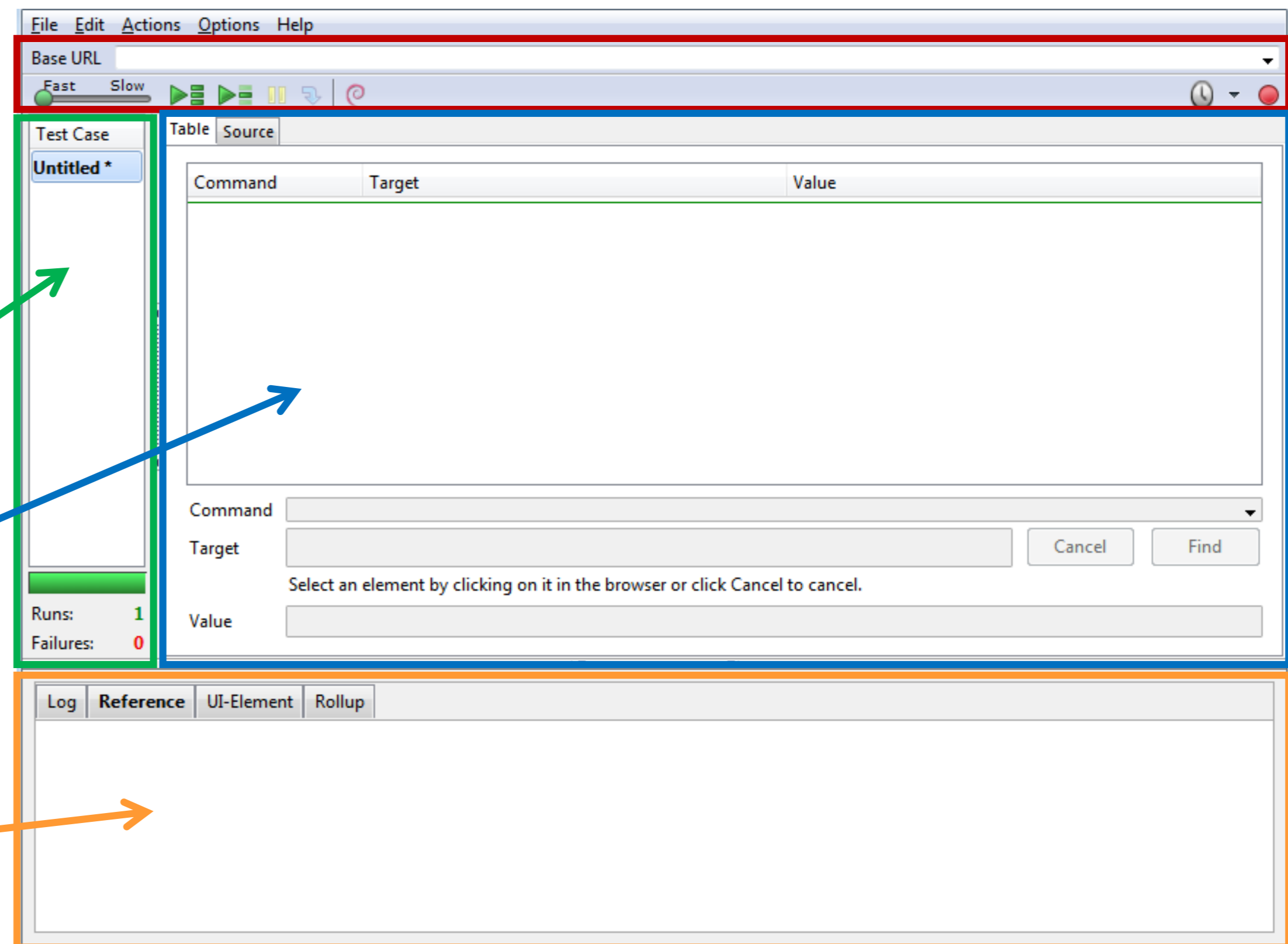
Components

1. Test control

2. Test suite tool

3. Test editor

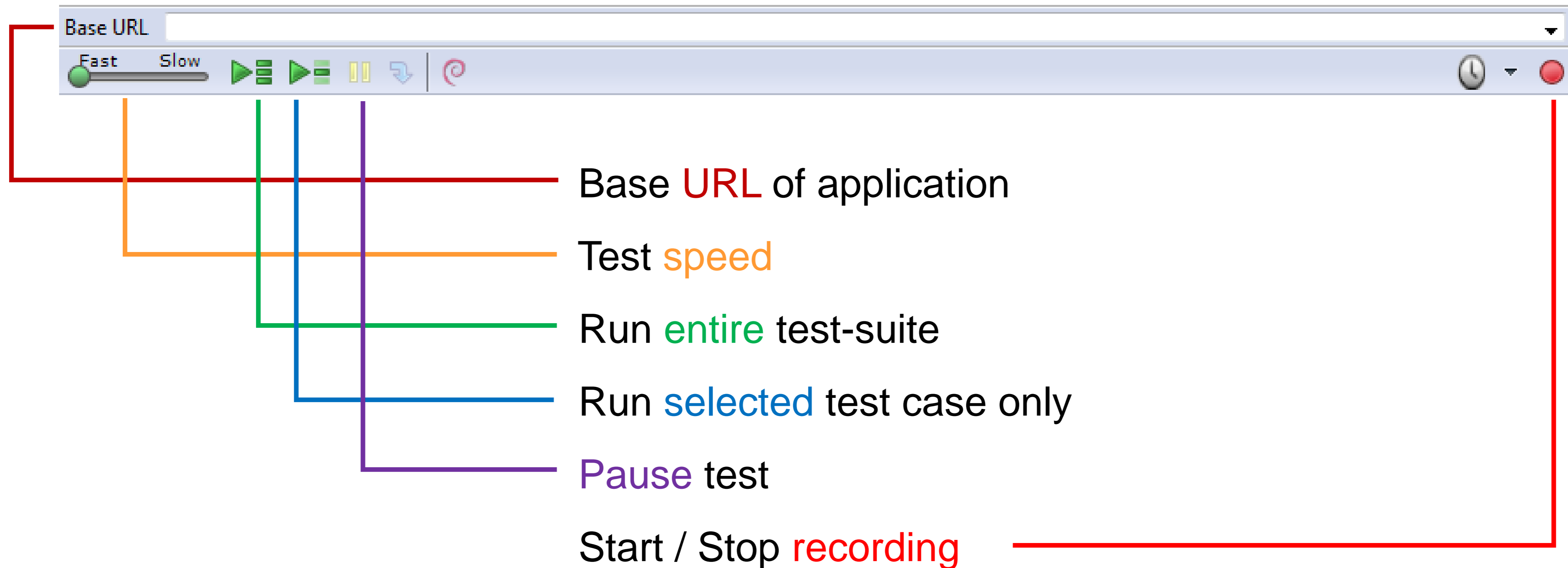
4. Tool panel



Exercise 1: Test Automation Tools

Example: Selenium IDE

Test control



Exercise 1: Test Automation Tools

Example: Selenium IDE

Test suite tool

Shows **all tests** in a test suite

List of all test cases

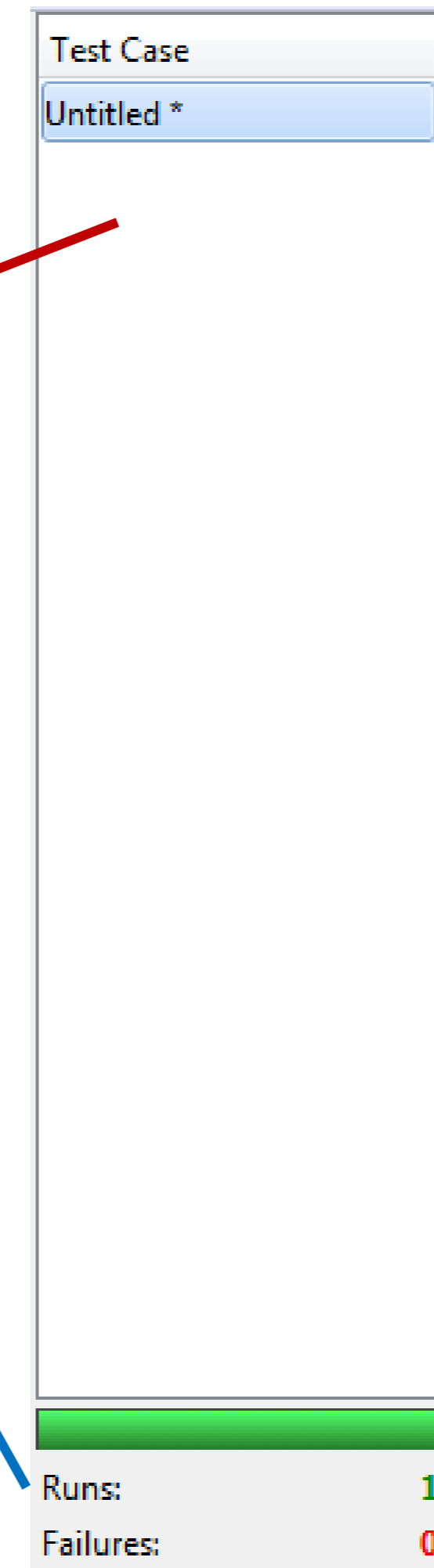
Can be given unique names

Shows **results** of running the tests

Number passed / failed

Green = pass

Red = fail



Exercise 1: Test Automation Tools

Example: Selenium IDE

Test editor

The test **steps**

Command of current step

Locator argument

Find button

Highlights target of locator on page

Value argument

The screenshot shows the Selenium IDE interface. At the top, there are two tabs: 'Table' and 'Source'. Below the tabs is a table with three columns: 'Command', 'Target', and 'Value'. The table is currently empty. Below the table, there is a form for editing a step. It has three input fields: 'Command', 'Target', and 'Value'. The 'Command' field is a dropdown menu. The 'Target' field is a text input field. The 'Value' field is a text input field. To the right of the 'Target' field are two buttons: 'Cancel' and 'Find'. Below the 'Target' field, there is a text label: 'Select an element by clicking on it in the browser or click Cancel to cancel.'

Command	Target	Value
---------	--------	-------

Command

Target

Select an element by clicking on it in the browser or click Cancel to cancel.

Value

Exercise 1: Test Automation Tools

Example: Selenium IDE

Test panel

Execution log of current tests

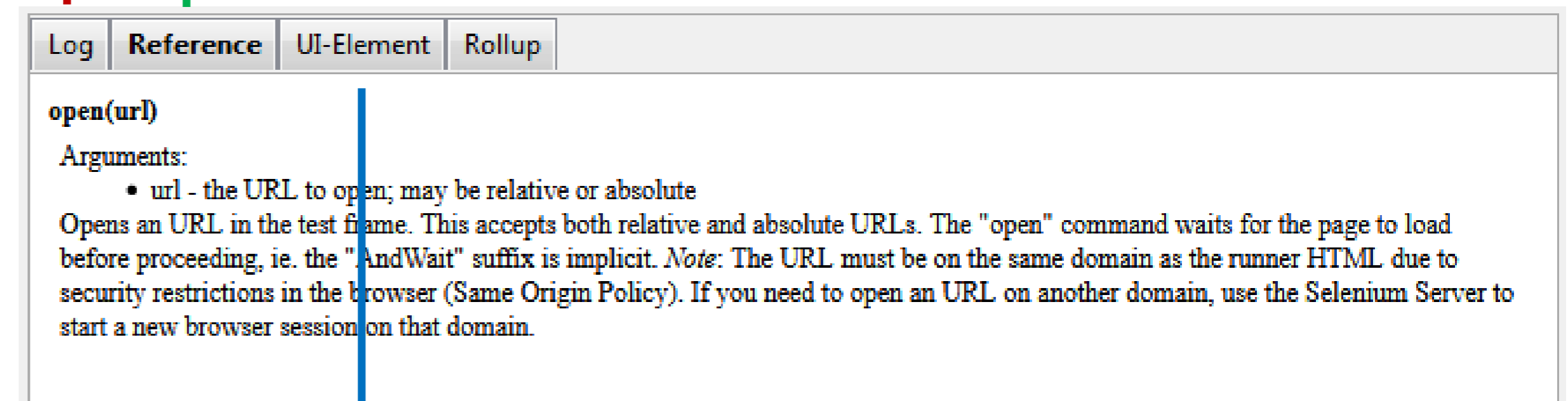
Displays errors

Reference

Documentation of command

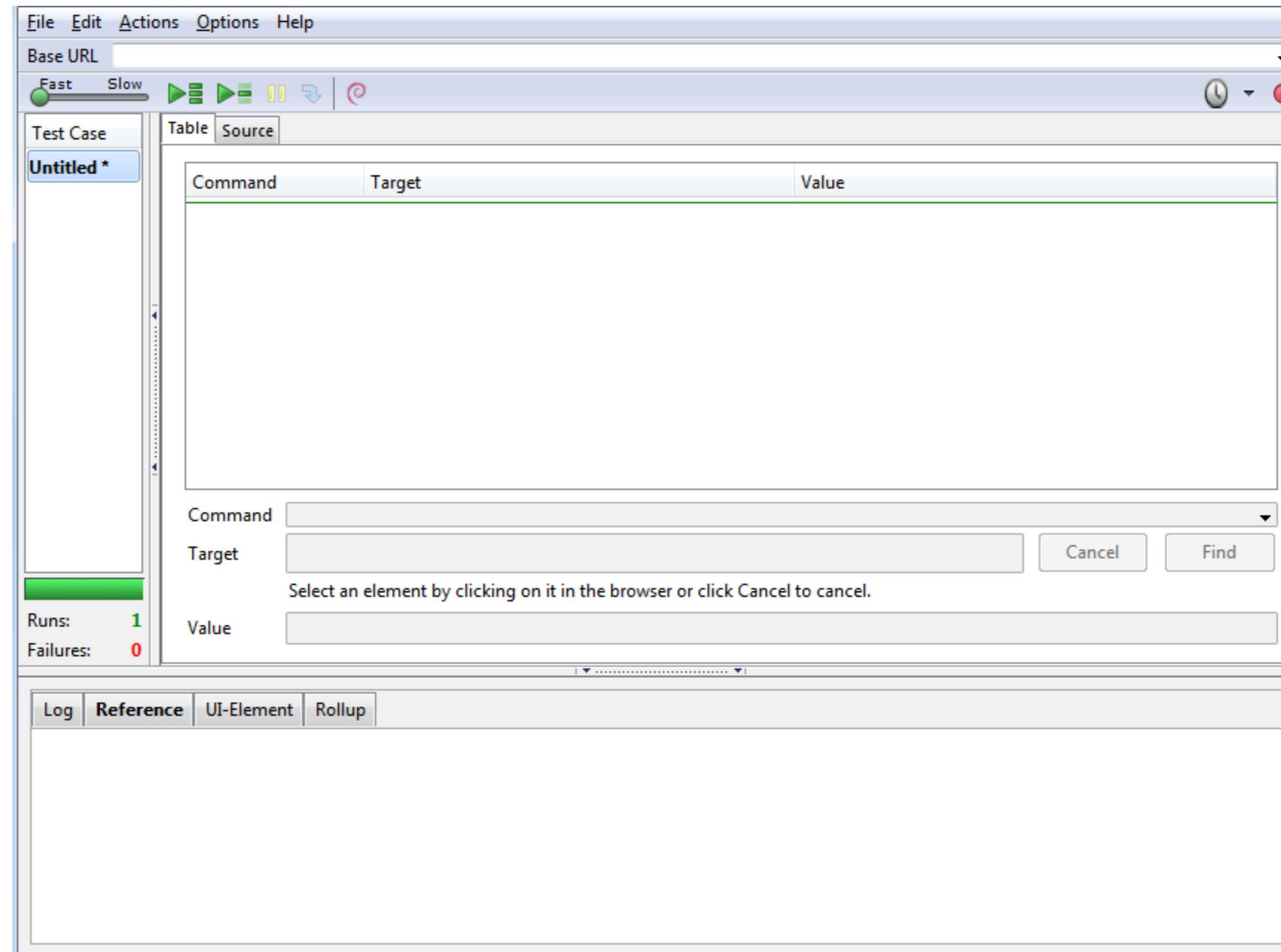
UI-element

Displays UI-element in use



Exercise 1: Test Automation Tools

Example: **Selenium IDE** → All parts combined



Exercise 1: Selenium IDE

Example: Simple Login Page

We can now use Selenium for test automation

Have created a simple login page for this purpose

Location: <http://inf3121-login-example.bitballoon.com/>

Simple Login Page

Username : Password :

How to use: Please provide a username and password.

Exercise 1: Selenium IDE

Example: Simple Login Page

How can we test the **test login** functionality?

Write down the login **procedure**

1. Go to **site**
2. Type in **username**
3. Type in **password**
4. Click "**Login**" button

We have been given a **list** of valid **usernames** and **passwords**

>> Know which to accept / reject

Simple Login Page

Username : Password :

How to use: Please provide a username and password.

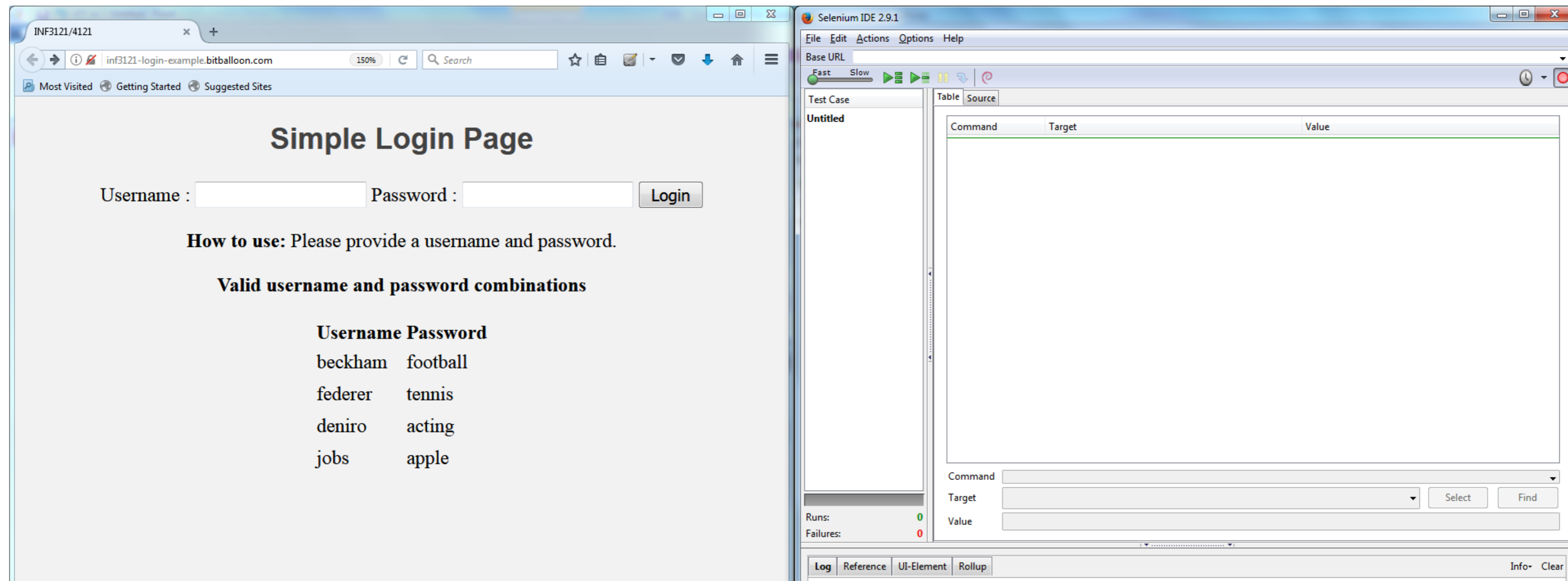
Username	Password
beckham	football
federer	tennis
deniro	acting
jobs	apple

Exercise 1: Selenium IDE

Example: Simple Login Page

1. Go to the login site and open Selenium IDE

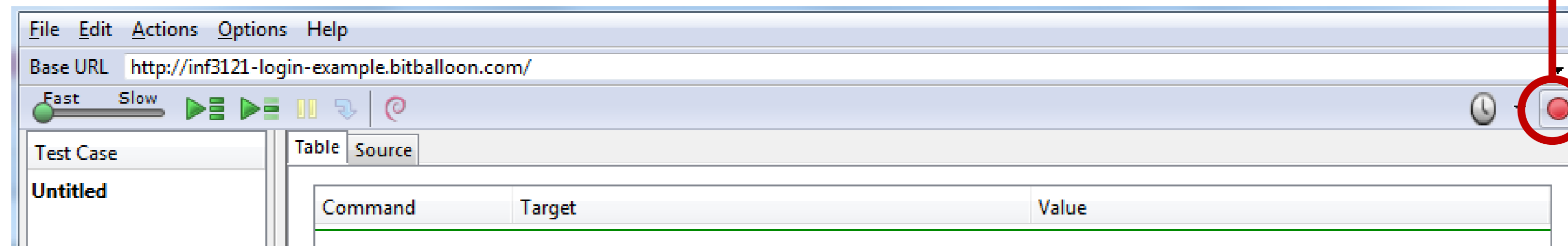
Tip: Have the windows side by side



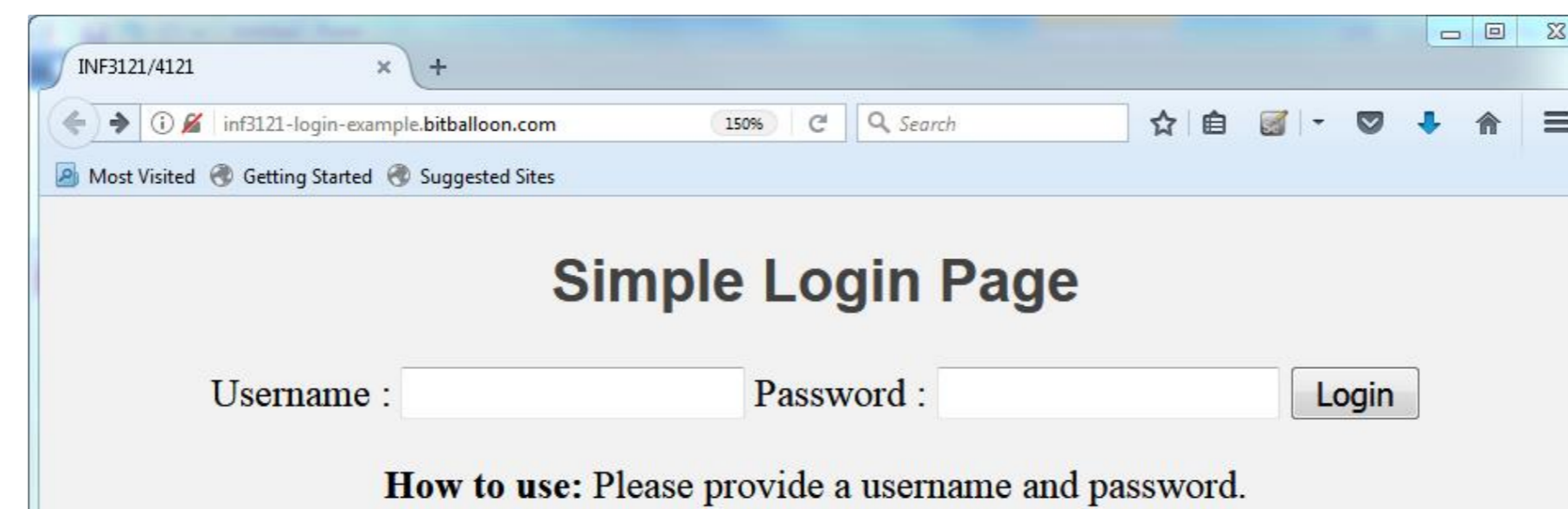
Exercise 1: Selenium IDE

Example: Simple Login Page

2. In Selenium: Click on the **record** button



3. Switch to the **login** site



Exercise 1: Selenium IDE

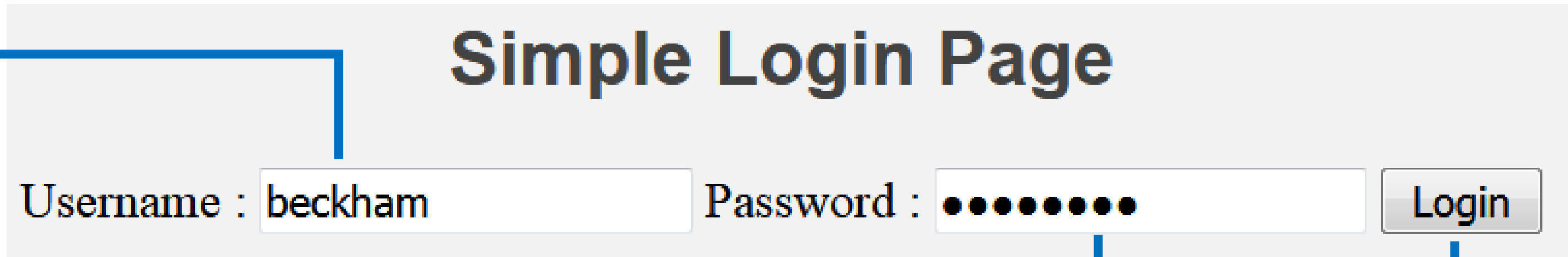
Example: Simple Login Page

4. Follow the login procedure for a valid user

i. Username

ii. Password

iii. Click "Login"



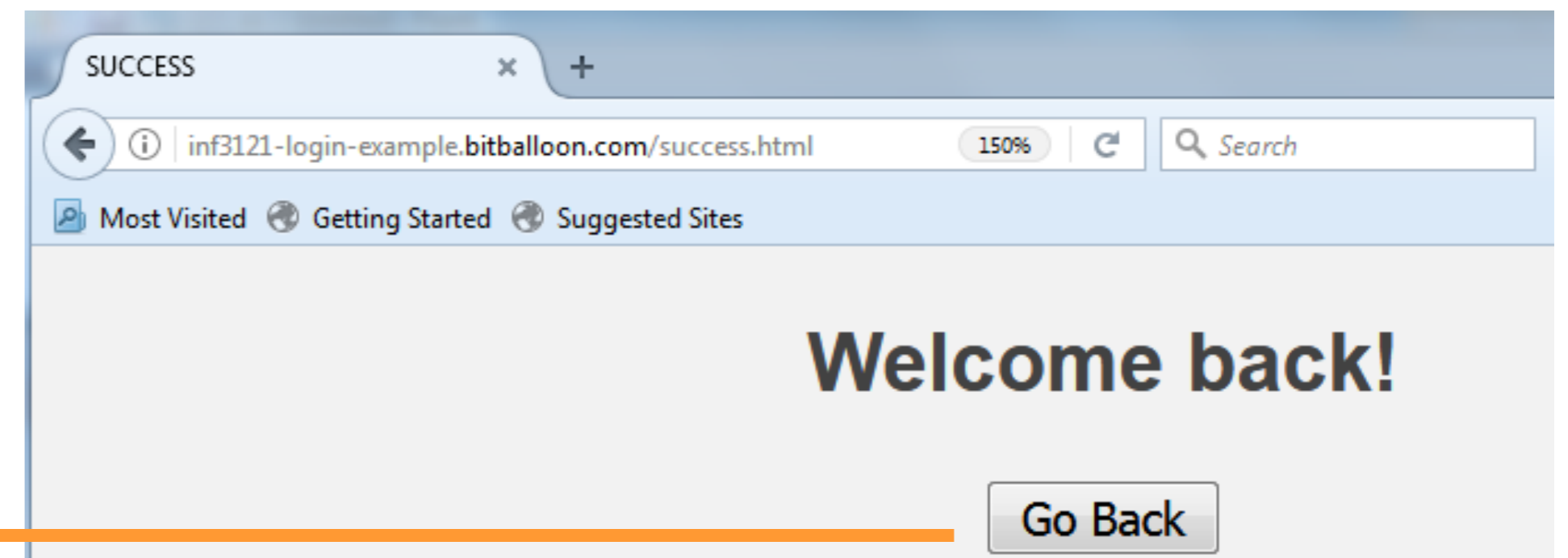
The screenshot shows a web form titled "Simple Login Page". It contains two input fields: "Username : beckham" and "Password : ●●●●●●". A "Login" button is located to the right of the password field. Blue lines connect the text labels on the left to the corresponding elements in the form: "Username" to the first input field, "Password" to the second input field, and "Click 'Login'" to the "Login" button.

5. Once login is approved

You are directed to the page 'success.html'

Page title can later be used to verify access

6. Click on the "Go Back" button



The screenshot shows a browser window with the title "SUCCESS". The address bar displays "inf3121-login-example.bitballoon.com/success.html". The page content includes a "Welcome back!" message and a "Go Back" button. An orange line connects the "Go Back" button in the screenshot to the text "Click on the 'Go Back' button" in the previous block.

Exercise 1: Selenium IDE

Example: Simple Login Page

7. In Selenium: **Stop** recording by clicking the **record** button
8. You now have an **automated test** for logging in

The tool **recorded** each **step** of the **procedure**

The tool **captures** and stores **data values**

Command	Target	Value
open	/	
type	id=username	beckham
type	id=password	football
clickAndWait	id=submit	
clickAndWait	css=button	

Go to site (points to 'open' row)

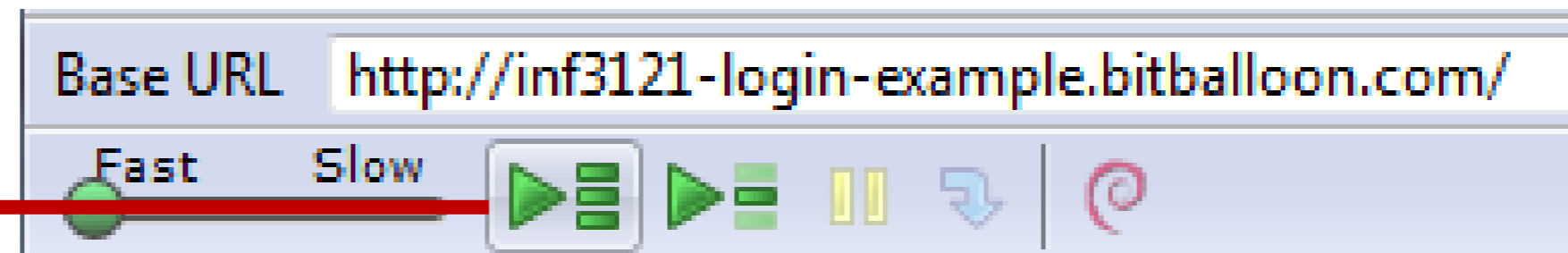
Username, password (points to 'type' rows)

Buttons: Login / Back (points to 'clickAndWait' rows)

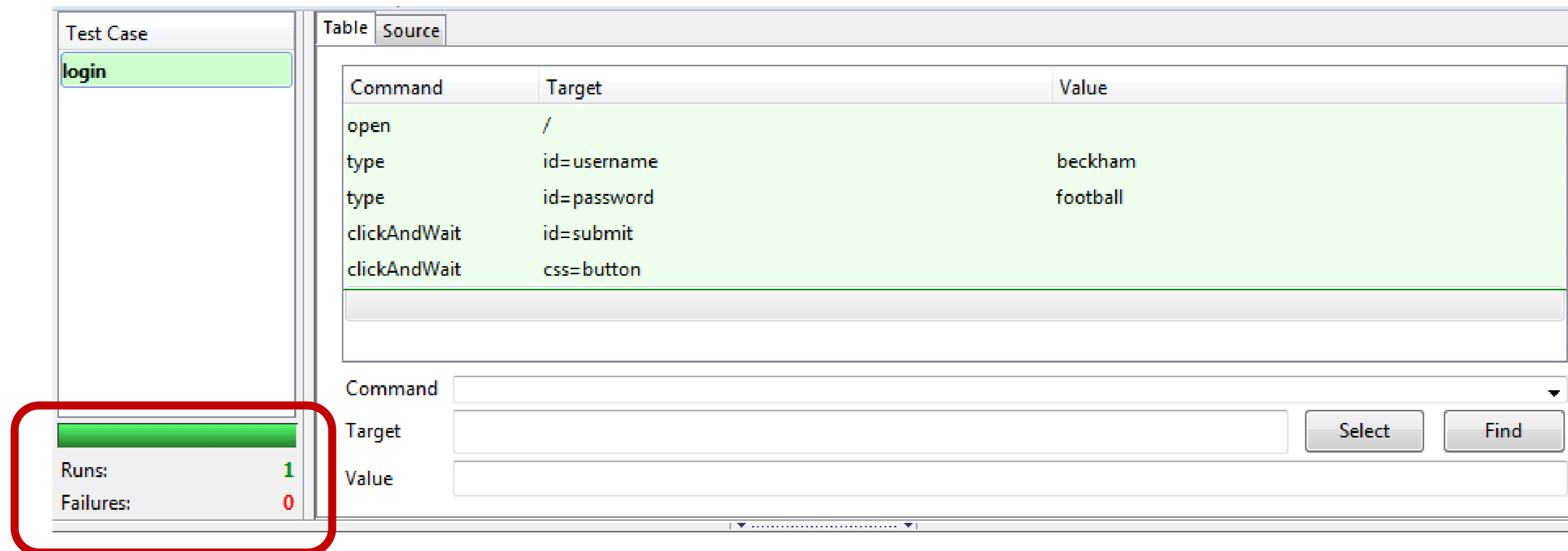
Exercise 1: Selenium IDE

Example: Simple Login Page

9. In Selenium: **Click** to play entire test suite



10. **Wait** for the test to **run**, and **verify** that it runs **without failures**



A screenshot of the Selenium IDE interface showing the details of a test case named 'login'. The 'Test Case' panel on the left shows 'login' selected. The 'Table' panel on the right displays the test steps:

Command	Target	Value
open	/	
type	id=username	beckham
type	id=password	football
clickAndWait	id=submit	
clickAndWait	css=button	

Below the table, there are input fields for 'Command', 'Target', and 'Value', along with 'Select' and 'Find' buttons. At the bottom left, a summary box shows 'Runs: 1' and 'Failures: 0', which is highlighted with a red box.

Exercise 1: Selenium IDE

Summary

We have now **created** a **simple automated test** using Selenium IDE

Testing login procedure for valid username and password combination

Selenium offers a **variety** of additional **features**

Explore Selenium to see if you can:

1. Write / record the remaining tests for **valid** users
2. Write / record an automated test for an **invalid** user
3. Write a Selenium test that **logs into** your **Facebook** account

Exercise 2: Benefits and Limitations

Discuss the **advantages** and **limitations** of **automated testing**.



The seminar slides are made by

Yulai Fjeld

ydfjeld @ uio.no

Master student

Department of Informatics

University of Oslo

Previously taught courses

Systemutvikling (INF1050), Universitet i Oslo

Software Testing (INF3121/4121), Universitetet i Oslo

Systemutvikling (ADSE2200), Høgskolen i Oslo og Akershus

