

CMPE3008

Software Engineering Testing

Assignment

Tanaka Chitete (20169321), Scott Berryman
(19747176), and Daniel Whitehead (20232430)

Contents

| | |
|--|-----------|
| 1 Source Code Analysis | 3 |
| 1.1 Background | 3 |
| 1.2 Coverage and Testing | 4 |
| 1.2.1 Implementation Method and Test Method 1 | 4 |
| 1.2.1.1 Source Code | 4 |
| 1.2.1.1.1 Implementation Method | 4 |
| 1.2.1.1.2 Test Method | 5 |
| 1.2.1.2 Graph Conversion | 6 |
| 1.2.1.3 Prime Path Coverage | 6 |
| 1.2.1.3.1 Prime Paths | 6 |
| 1.2.1.3.2 Prime Path Coverage Test Cases | 7 |
| 1.2.1.4 Base Choice Analysis | 7 |
| 1.2.1.4.1 Characteristics and Corresponding Blocks | 7 |
| 1.2.1.4.2 Constraints | 7 |
| 1.2.1.4.3 Base Choice | 8 |
| 1.2.1.4.4 Test Cases | 8 |
| 1.2.1.5 Test Method Comparison | 9 |
| 1.2.2 Implementation Method and Test Method 2 | 10 |
| 1.2.2.1 Source Code | 10 |
| 1.2.2.1.1 Implementation Method | 10 |
| 1.2.2.1.2 Test Methods | 10 |
| 1.2.2.2 Graph Conversion | 12 |
| 1.2.2.3 Prime Path Coverage | 12 |
| 1.2.2.3.1 Prime Paths | 12 |
| 1.2.2.3.2 Prime Path Coverage Test Cases | 13 |
| 1.2.2.4 Base Choice Analysis | 13 |
| 1.2.2.4.1 Characteristics and Corresponding Blocks | 13 |
| 1.2.2.4.2 Constraints | 13 |
| 1.2.2.4.3 Base Choice and Test Cases | 14 |
| 1.2.2.5 Test Method Comparison | 14 |
| 2 Testing Tool Investigation | 16 |
| 2.1 Overview | 16 |
| 2.2 Getting Started | 16 |
| 2.2.1 Setup | 16 |
| 2.2.2 Recording a Test | 17 |
| 2.2.3 Organising Tests | 19 |
| 2.2.3.1 Tests | 19 |
| 2.2.3.2 Suites | 21 |
| 2.2.3.2.1 Creating a Suite | 21 |
| 2.2.3.2.2 Adding a Test to a Suite | 22 |
| 2.2.4 Saving Work | 24 |
| 2.2.5 Playback | 25 |
| 2.2 Relevant Considerations | 26 |
| 2.2.1 Advantages | 26 |
| 2.2.1.1 Cross-Platform Support | 26 |
| 2.2.1.2 Extensibility with Plugins | 26 |
| 2.2.1.3 Popularity | 26 |
| 2.2.2 Disadvantages | 26 |
| 2.2.2.1 Reporting Capabilities | 26 |
| 2.2.2.2 Test Maintenance | 27 |
| 2.2.2.3 Standards | 27 |
| 2.3 Verdict | 27 |
| References | 28 |
| Appendices | 29 |

1 Source Code Analysis

1.1 Background

Our chosen code comes from a public GitHub repository. The Java project relates to various networking functionality, including configurations of network protocols. The main repository can be found at: https://github.com/LucidProject/platform_frameworks_base. Our 2 methods both come from the file “WifiConfiguration.java”, which can be found at:
https://github.com/LucidProject/platform_frameworks_base/blob/d6d4abf090ba3982542fb6ca88c795f2bcda7177/wifi/java/android/net/wifi/WifiConfiguration.java.

Their respective test methods are in the same repository, but in a different subdirectory. The test file “WifiConfigurationTest.java” can be found at:
https://github.com/LucidProject/platform_frameworks_base/blob/d6d4abf090ba3982542fb6ca88c795f2bcda7177/wifi/tests/src/android/net/wifi/WifiConfigurationTest.java.

1.2 Coverage and Testing

1.2.1 Implementation Method and Test Method 1

1.2.1.1 Source Code

1.2.1.1.1 Implementation Method

The implementation method, “getKeyIdForCredentials()”, can be found at line 2082 of “WifiConfiguration.java”.

```
2082     /**
2083      * Get an identifier for associating credentials with this config
2084      * @param current configuration contains values for additional fields
2085      *          that are not part of this configuration. Used
2086      *          when a config with some fields is passed by an application.
2087      * @throws IllegalStateException if config is invalid for key id generation
2088      * @hide
2089      */
2090     public String getKeyIdForCredentials(WifiConfiguration current) {
2091         String keyMgmt = "";
2092
2093         try {
2094             // Get current config details for fields that are not initialized
2095             if (TextUtils.isEmpty(SSID)) SSID = current.SSID;
2096             if (allowedKeyManagement.cardinality() == 0) {
2097                 allowedKeyManagement = current.allowedKeyManagement;
2098             }
2099             if (allowedKeyManagement.get(KeyMgmt.WPA_EAP)) {
2100                 keyMgmt += KeyMgmt.strings[KeyMgmt.WPA_EAP];
2101             }
2102             if (allowedKeyManagement.get(KeyMgmt.OSEN)) {
2103                 keyMgmt += KeyMgmt.strings[KeyMgmt.OSEN];
2104             }
2105             if (allowedKeyManagement.get(KeyMgmt.IEEE8021X)) {
2106                 keyMgmt += KeyMgmt.strings[KeyMgmt.IEEE8021X];
2107             }
2108             if (allowedKeyManagement.get(KeyMgmt.SUITE_B_192)) {
2109                 keyMgmt += KeyMgmt.strings[KeyMgmt.SUITE_B_192];
2110             }
2111
2112             if (TextUtils.isEmpty(keyMgmt)) {
2113                 throw new IllegalStateException("Not an EAP network");
2114             }
2115
2116             return trimStringForKeyId(SSID) + "_" + keyMgmt + "_" +
2117                 trimStringForKeyId(enterpriseConfig.getKeyId(current != null ?
2118                     current.enterpriseConfig : null));
2119         } catch (NullPointerException e) {
2120             throw new IllegalStateException("Invalid config details");
2121         }
2122     }
```

1.2.1.1.2 Test Method

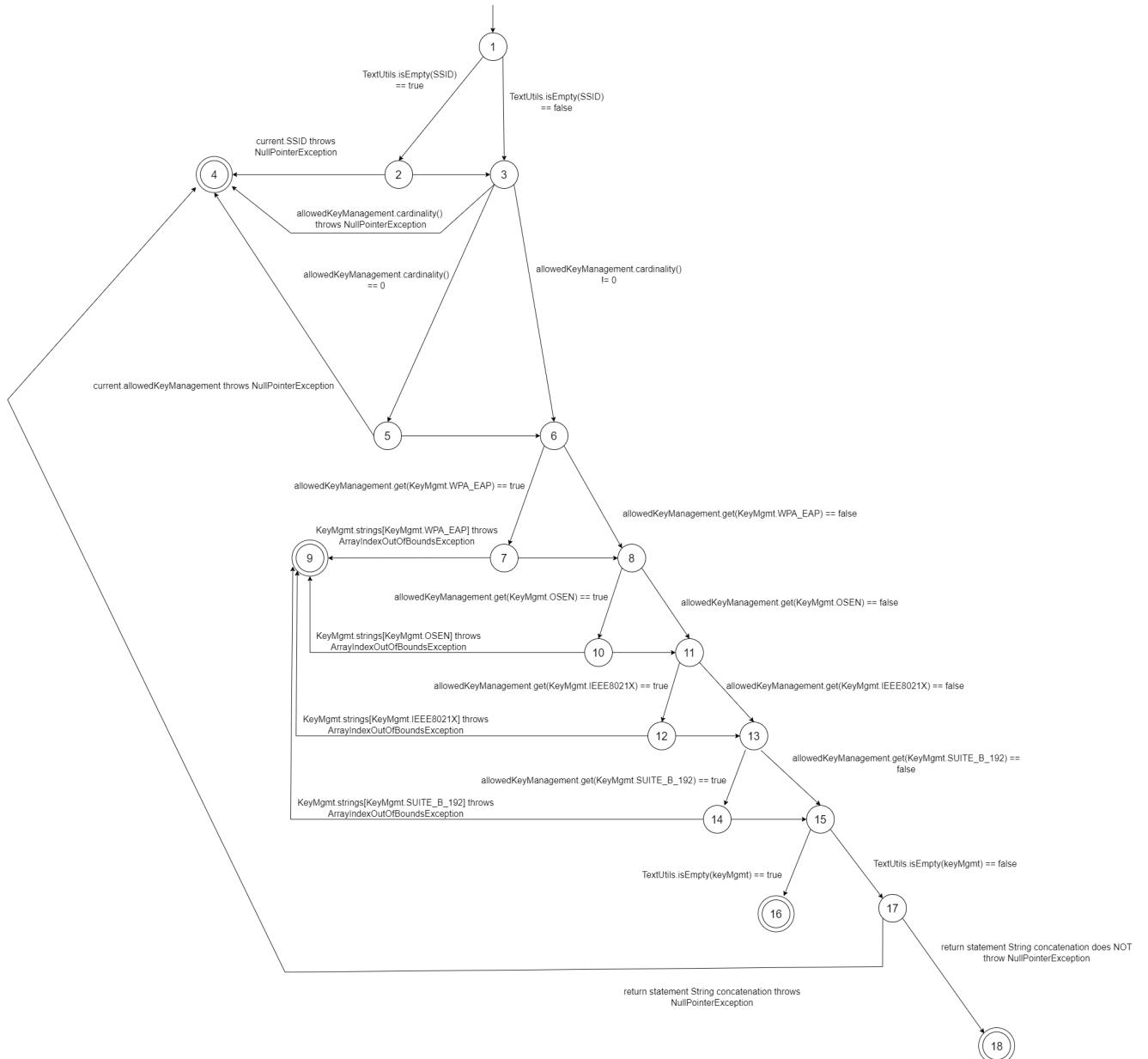
The test method, “testGetKeyIdForCredentials()”, can be found at line 298 of “WifiConfigurationTest.java”.

```
298     /**
299      * Verifies that getKeyIdForCredentials returns the expected string for Enterprise networks
300      * @throws Exception
301      */
302     @Test
303     public void testGetKeyIdForCredentials() throws Exception {
304         WifiConfiguration config = new WifiConfiguration();
305         final String mSsid = "TestAP";
306         config.SSID = mSsid;
307
308         // Test various combinations
309         // EAP with TLS
310         config.allowedKeyManagement.set(KeyMgmt.WPA_EAP);
311         config.enterpriseConfig.setEapMethod(WifiEnterpriseConfig.Eap.TLS);
312         config.enterpriseConfig.setPhase2Method(WifiEnterpriseConfig.Phase2.NONE);
313         String keyId = config.getKeyIdForCredentials(config);
314         assertEquals(keyId, mSsid + "_WPA_EAP_TLS_NULL");
315
316         // EAP with TTLS & MSCHAPv2
317         config.allowedKeyManagement.set(KeyMgmt.WPA_EAP);
318         config.enterpriseConfig.setEapMethod(WifiEnterpriseConfig.Eap.TTLS);
319         config.enterpriseConfig.setPhase2Method(WifiEnterpriseConfig.Phase2.MSCHAPV2);
320         keyId = config.getKeyIdForCredentials(config);
321         assertEquals(keyId, mSsid + "_WPA_EAP_TTLS_MSCHAPV2");
322
323         // Suite-B 192 with PWD & GTC
324         config.allowedKeyManagement.clear();
325         config.allowedKeyManagement.set(KeyMgmt.SUITE_B_192);
326         config.enterpriseConfig.setEapMethod(WifiEnterpriseConfig.Eap.PWD);
327         config.enterpriseConfig.setPhase2Method(WifiEnterpriseConfig.Phase2.GTC);
328         keyId = config.getKeyIdForCredentials(config);
329         assertEquals(keyId, mSsid + "_SUITE_B_192_PWD_GTC");
330
331         // IEEE8021X with SIM
332         config.allowedKeyManagement.clear();
333         config.allowedKeyManagement.set(KeyMgmt.IEEE8021X);
334         config.enterpriseConfig.setEapMethod(WifiEnterpriseConfig.Eap.SIM);
335         config.enterpriseConfig.setPhase2Method(WifiEnterpriseConfig.Phase2.NONE);
336         keyId = config.getKeyIdForCredentials(config);
337         assertEquals(keyId, mSsid + "_IEEE8021X_SIM_NULL");
338
339         // Try calling this method with non-Enterprise network, expect an exception
340         boolean exceptionThrown = false;
341         try {
342             config.allowedKeyManagement.clear();
343             config.allowedKeyManagement.set(KeyMgmt.WPA2_PSK);
344             config.preSharedKey = "TestPsk";
345             keyId = config.getKeyIdForCredentials(config);
346         } catch (IllegalStateException e) {
347             exceptionThrown = true;
348         }
349         assertTrue(exceptionThrown);
350     }
```

1.2.1.2 Graph Conversion

Note: An “IndexOutOfBoundsException” can be thrown by the get() methods at nodes {6, 8, 11, 13}. This should be represented as a 3rd edge from each of these nodes that goes to a new end node.

Note: This graph was originally going to be attached as an additional .png, however the single .pdf turnitin submission prevented this. It should still be legible by zooming in.



1.2.1.3 Prime Path Coverage

1.2.1.3.1 Prime Paths

Prime path coverage for this graph resulted in a lot of paths. As such, the prime paths for this graph can be found in the Appendices.

1.2.1.3.2 Prime Path Coverage Test Cases

As all prime paths for this graph start at the initial node and end at an end node, the test cases needed to cover prime path coverage are the same as the list of prime paths.

1.2.1.4 Base Choice Analysis

1.2.1.4.1 Characteristics and Corresponding Blocks

1. current is null
 - a. Yes
 - b. No
2. allowedKeyManagement is null
 - a. Yes
 - b. No
3. allowedKeyManagement has WPA_EAP == true
 - a. Yes
 - b. No
4. allowedKeyManagement has OSEN == true
 - a. Yes
 - b. No
5. allowedKeyManagement has IEEE8021X == true
 - a. Yes
 - b. No
6. allowedKeyManagement has SUITE_B_192 == true
 - a. Yes
 - b. No
7. enterpriseConfig is null
 - a. Yes
 - b. No

1.2.1.4.2 Constraints

Each of 3a, 4a, 5a and 6a must be paired with 2b.

i.e for allowedKeyManagement to have the given values == true, allowedKeyManagement must not be null.

1.2.1.4.3 Base Choice

| | | | | | | | |
|--------------|--------------------|----------------------------------|--------------------------------------|-----------------------------------|--|--|-----------------------------|
| | 1: Current is null | 2: allowed KeyManagement is null | 3: allowed KeyManagement has WPA_EAP | 4: allowed KeyManagement has OSEN | 5: allowed KeyManagement has IEEE8021X | 6: allowed KeyManagement has SUITE_B_192 | 7: enterpriseConfig is null |
| Base Choice: | b: no | b: no | a: yes | b: no | b: no | b: no | b: no |

These blocks were chosen to be the base choice as they assume the method is being used properly (without null values). The other blocks were chosen based on the comment about the declaration of “allowedKeyManagement”. This comment states that “Defaults to WPA-PSK WPA-EAP”. As such, we have assumed that the most common scenario will be some slight variation of this. We have decided the base case should use an instance of allowedKeyManagement that has WPA-EAP, but not the other values. Note that this method doesn’t reference the other mentioned default (WPA-PSK).

1.2.1.4.4 Test Cases

| | | | | | | | |
|-----------------------|--------------------|----------------------------------|---|----------------------------------|--|--|-----------------------------|
| | 1: Current is null | 2: allowed KeyManagement is null | 3: allowedKeyManagement has WPA_EAP | 4: allowedKeyManagement has OSEN | 5: allowed KeyManagement has IEEE8021X | 6: allowed KeyManagement has SUITE_B_192 | 7: enterpriseConfig is null |
| Test 1: (Base Choice) | b: no | b: no | a: yes | b: no | b: no | b: no | b: no |
| Test 2: | a: yes | b: no | a: yes | b: no | b: no | b: no | b: no |
| Test 3: | b: no | a: yes | b: no (Changed from base case due to constraint) | b: no | b: no | b: no | b: no |
| Test 4: | b: no | b: no | b: no | b: no | b: no | b: no | b: no |
| Test 5: | b: no | b: no | a: yes | a: yes | b: no | b: no | b: no |
| Test 6: | b: no | b: no | a: yes | b: no | a: yes | b: no | b: no |
| Test 7: | b: no | b: no | a: yes | b: no | b: no | a: yes | b: no |
| Test 8: | b: no | b: no | a: yes | b: no | b: no | b: no | a: yes |

1.2.1.5 Test Method Comparison

This test method works by creating a real WifiConfiguration object (the class of the method being tested). This constructor in turn creates a real WifiEnterpriseConfig object (another external class), and assigns this as a class field. The lack of dependency injection prevents a mocked WifiEnterpriseConfig from being used, which means the results of the test method depend on both the WifiConfiguration and WifiEnterpriseConfig classes (as the enterpriseConfig is used in the return statement of the method being tested).

The test method only uses 5 test cases. The first 4 test cases update the WifiConfiguration's "allowedKeyManagement" and "enterpriseConfig" fields; then call the method that is being tested, then use an "assertEquals" statement to compare the actual result with the expected result. The final test case updates the "allowedKeyManagement" to only include a value not contained in the original method to test whether or not the expected exception is thrown.

The test cases chosen only cover a tiny fraction of the prime paths. This is completely understandable, as the number of prime paths for what is such a simple method is far too large to be worth the time or effort.

They also do not cover many of the combinations described by base choice coverage (though the base case itself is tested by the first test case). The test cases chosen only ever have 1 boolean in "allowedKeyManagement" set to true. I believe that more combinations of allowedKeyManagement should be tested to ensure the expected results for cases where multiple allowedKeyManagements elements are true.

The tests also appear to be more concerned with the combination of a single allowedKeyManagement with an enterpriseConfig. This combination is not something that has much importance in prime path coverage or base choice coverage as the enterpriseConfig is only ever referenced by the original method in the return statement.

Due to the class's tight coupling with other classes (WifiEnterpriseConfig, KeyMgmt), it is difficult to test many of the error cases in isolation. These include instances where certain elements are null, or where directly accessing another class's array throws an ArrayIndexOutOfBoundsException.

In conclusion, I believe the test cases used are lacking in coverage, however this could be justified with domain knowledge. Further testing could prove difficult due to tight class coupling.

1.2.2 Implementation Method and Test Method 2

1.2.2.1 Source Code

1.2.2.1.1 Implementation Method

The implementation method, “getOrCreateRandomizedMacAddress()”, can be found at line 1050 of “WifiConfiguration.java”.

```
/*
 * @hide
 * Returns Randomized MAC address to use with the network.
 * If it is not set/valid, creates a new randomized address.
 * If it can't generate a valid mac, returns the default MAC.
 */
public @NonNull MacAddress getOrCreateRandomizedMacAddress() {
    int randomMacGenerationCount = 0;
    while (!isValidMacAddressForRandomization(mRandomizedMacAddress)
            && randomMacGenerationCount < MAXIMUM_RANDOM_MAC_GENERATION_RETRY) {
        mRandomizedMacAddress = MacAddress.createRandomUnicastAddress();
        randomMacGenerationCount++;
    }

    if (!isValidMacAddressForRandomization(mRandomizedMacAddress)) {
        mRandomizedMacAddress = MacAddress.fromString(WifiInfo.DEFAULT_MAC_ADDRESS);
    }
    return mRandomizedMacAddress;
}
```

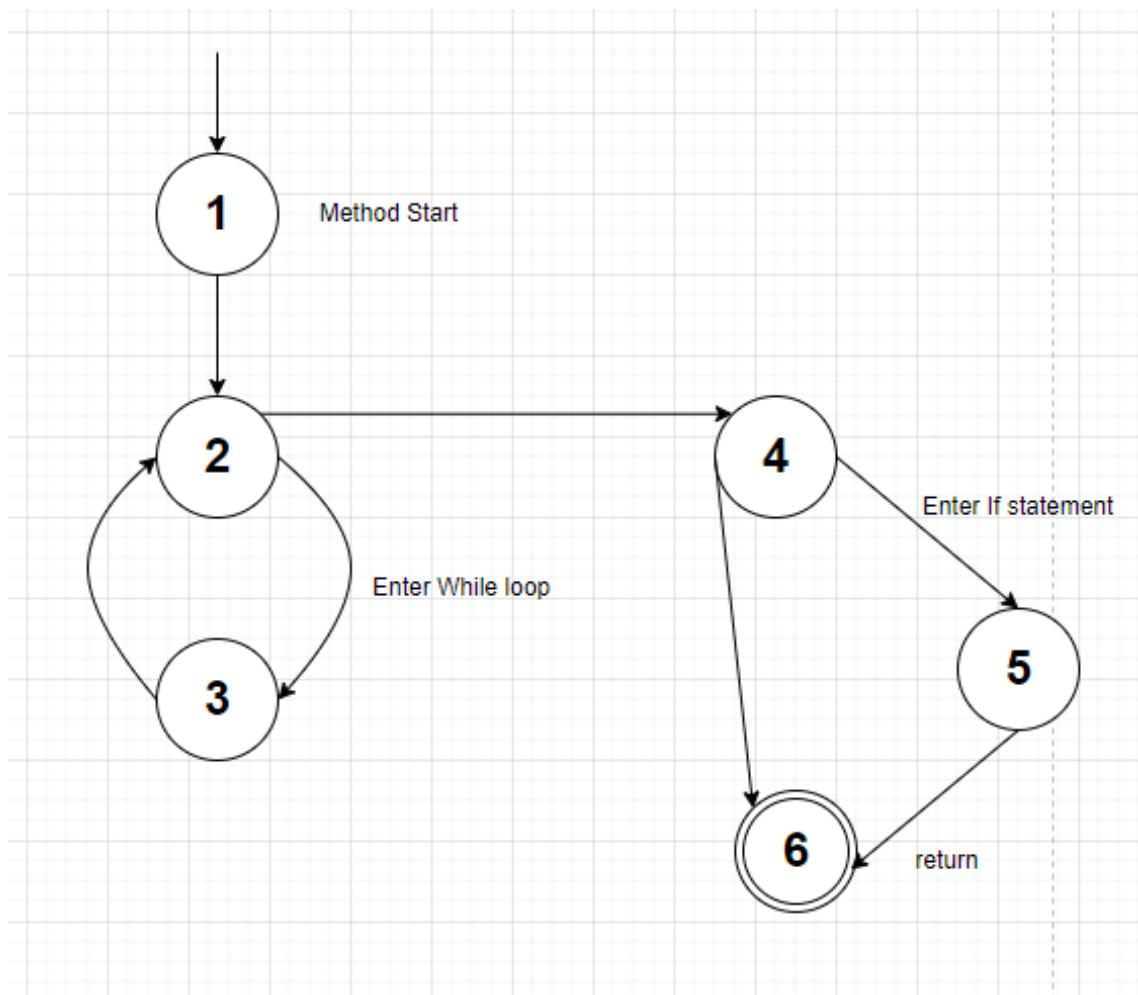
1.2.2.1.2 Test Methods

The test methods, “testGetOrCreateRandomizedMacAddress_SavesAndReturnsSameAddress()” and “testGetOrCreateRandomizedMacAddress_ReRandomizesInvalidAddress()”, can be found at lines 195 and 221 of “WifiConfigurationTest.java”, respectively.

```
195     @Test
196     public void testGetOrCreateRandomizedMacAddress_SavesAndReturnsSameAddress() {
197         WifiConfiguration config = new WifiConfiguration();
198         MacAddress defaultMac = MacAddress.fromString(WifiInfo.DEFAULT_MAC_ADDRESS);
199         assertEquals(defaultMac, config.getRandomizedMacAddress());
200
201         MacAddress firstMacAddress = config.getOrCreateRandomizedMacAddress();
202         MacAddress secondMacAddress = config.getOrCreateRandomizedMacAddress();
203
204         assertNotEquals(defaultMac, firstMacAddress);
205         assertEquals(firstMacAddress, secondMacAddress);
206     }
```

```
221     @Test
222     public void testGetOrCreateRandomizedMacAddress_ReRandomizesInvalidAddress() {
223         WifiConfiguration config = new WifiConfiguration();
224
225         MacAddress defaultMac = MacAddress.fromString(WifiInfo.DEFAULT_MAC_ADDRESS);
226         MacAddress macAddressZeroes = MacAddress.ALL_ZEROS_ADDRESS;
227         MacAddress macAddressMulticast = MacAddress.fromString("03:ff:ff:ff:ff:ff");
228         MacAddress macAddressGlobal = MacAddress.fromString("fc:ff:ff:ff:ff:ff");
229
230         config.setRandomizedMacAddress(null);
231         MacAddress macAfterChange = config.getOrCreateRandomizedMacAddress();
232         assertEquals(macAfterChange, null);
233
234         config.setRandomizedMacAddress(defaultMac);
235         macAfterChange = config.getOrCreateRandomizedMacAddress();
236         assertEquals(macAfterChange, defaultMac);
237
238         config.setRandomizedMacAddress(macAddressZeroes);
239         macAfterChange = config.getOrCreateRandomizedMacAddress();
240         assertEquals(macAfterChange, macAddressZeroes);
241
242         config.setRandomizedMacAddress(macAddressMulticast);
243         macAfterChange = config.getOrCreateRandomizedMacAddress();
244         assertEquals(macAfterChange, macAddressMulticast);
245
246         config.setRandomizedMacAddress(macAddressGlobal);
247         macAfterChange = config.getOrCreateRandomizedMacAddress();
248         assertEquals(macAfterChange, macAddressGlobal);
249     }
250 }
```

1.2.2.2 Graph Conversion



1.2.2.3 Prime Path Coverage

1.2.2.3.1 Prime Paths

- [1, 2, 3]
- [2, 3, 2]
- [3, 2, 3]
- [1, 2, 4, 6]
- [1, 2, 4, 5, 6]
- [3, 2, 4, 6]
- [3, 2, 4, 5, 6]

1.2.2.3.2 Prime Path Coverage Test Cases

[1, 2, 4, 6]

[1, 2, 4, 5, 6]

[1, 2, 3, 2, 4, 5, 6]

[1, 2, 3, 2, 3, 2, 4, 6]

1.2.2.4 Base Choice Analysis

1.2.2.4.1 Characteristics and Corresponding Blocks

1. mRandomizedMacAddress is null
 - a. True
 - b. False
2. mRandomizedMacAddress is valid
 - a. True
 - b. False
3. mRandomizedMacAddress is invalid and not null
 - a. True
 - b. False
4. MAXIMUM_RANDOM_MAC_GENERATION_RETRY
 - a. Less than 0
 - b. 0
 - c. 1
 - d. Greater than 1

1.2.2.4.2 Constraints

Between characteristics 1, 2 and 3 only 1 block can be true as they are variations of the single mRandomizedMacAddress input.

1.2.2.4.3 Base Choice and Test Cases

| | 1: mRandomized MacAddress is null | 2: allowedKeyMa nagement is nullmRandomi zedMacAddre ss is valid | 3: mRandomized MacAddress is invalid and not null | 4: MAXIMUM_R ANDOM_MAC _GENERATIO N_RETRY |
|-----------------------|--|---|---|---|
| Test 1: (Base Choice) | b: False | a: True | b: False | d: Greater than 1 |
| Test 2: | a: True | b: False | b: False | d: Greater than 1 |
| Test 3: | b: False | b: False | a: True | d: Greater than 1 |
| Test 4: | b: False | a: True | b: False | a: Less than 0 |
| Test 5: | b: False | a: True | b: False | b: 0 |
| Test 6: | b: False | a: True | b: False | c: 1 |

The test selected as the base case is chosen as it represents what would be expected use of the method under normal circumstances.

1.2.2.5 Test Method Comparison

Test 1

@Test

```
public void testGetOrCreateRandomizedMacAddress_SavesAndReturnsSameAddress()
```

This test is designed to check that the method correctly reassigned the mac address value upon receiving a valid randomised mac address. In doing this it essentially is the base case shown in the tests above as by default the mac generation retry is set as 3 and therefore greater than 1.

Test 2

@Test

```
public void testGetOrCreateRandomizedMacAddress_ReRandomizesInvalidAddress()
```

This test is designed to ensure that the method has expected behaviour when given invalid randomised mac addresses. The test checks for null, default mac, multicast address, all 0 address and the global mac address. None of these are valid for randomised mac addresses.

Overall the test methods cover test cases 1, 2 and 3. While this is only half the generated test cases from the base choice coverage I believe that it is a good test suite for this particular method. The reason for this is that there are a number of types of invalid mac addresses for randomisation accounted for in the testing that my coverage does not include.

On top of this while the tests do not account for cases where the retry attempts are not greater than 1 this is actually an unimportant metric as in the program this input comes from a static int that is set to 3, this means that under all circumstances it will be greater than 1.

2 Testing Tool Investigation

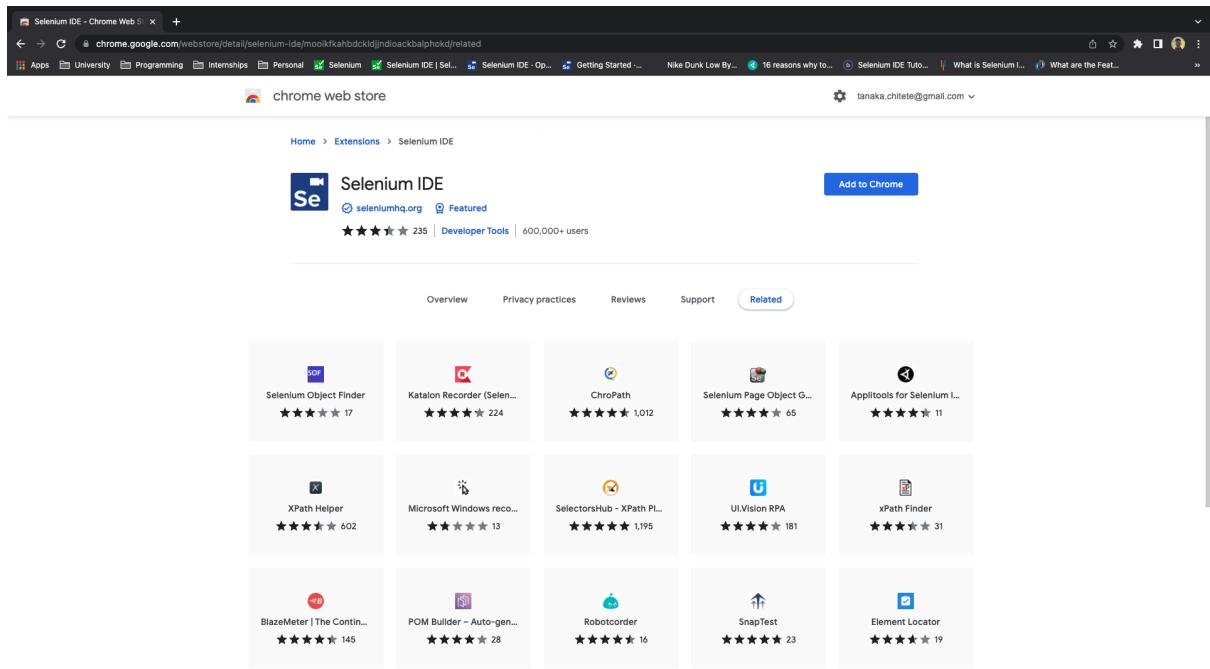
2.1 Overview

In this section, we will explore Selenium IDE—an integrated development environment (IDE) which allows test case developers to record and playback user actions for test automation (Selenium, n.d.) on web applications. Selenium IDE allows a user or a test case developer to write test cases and construct test suites, run individual tests or entire suites, and make edits deemed necessary by their requirements. The IDE is available as extensions for Google Chrome and Mozilla Firefox.

2.2 Getting Started

2.2.1 Setup

To get started, you need to download the Selenium IDE extension for Google Chrome from [Google Chrome Web Store](#) or for Mozilla Firefox from [Add-ons for Firefox](#). In this walkthrough, we are going to be using the version for Google Chrome and we will be creating a test for the Google Search page. This is a specific example, but feel free to apply the same steps to your own project.

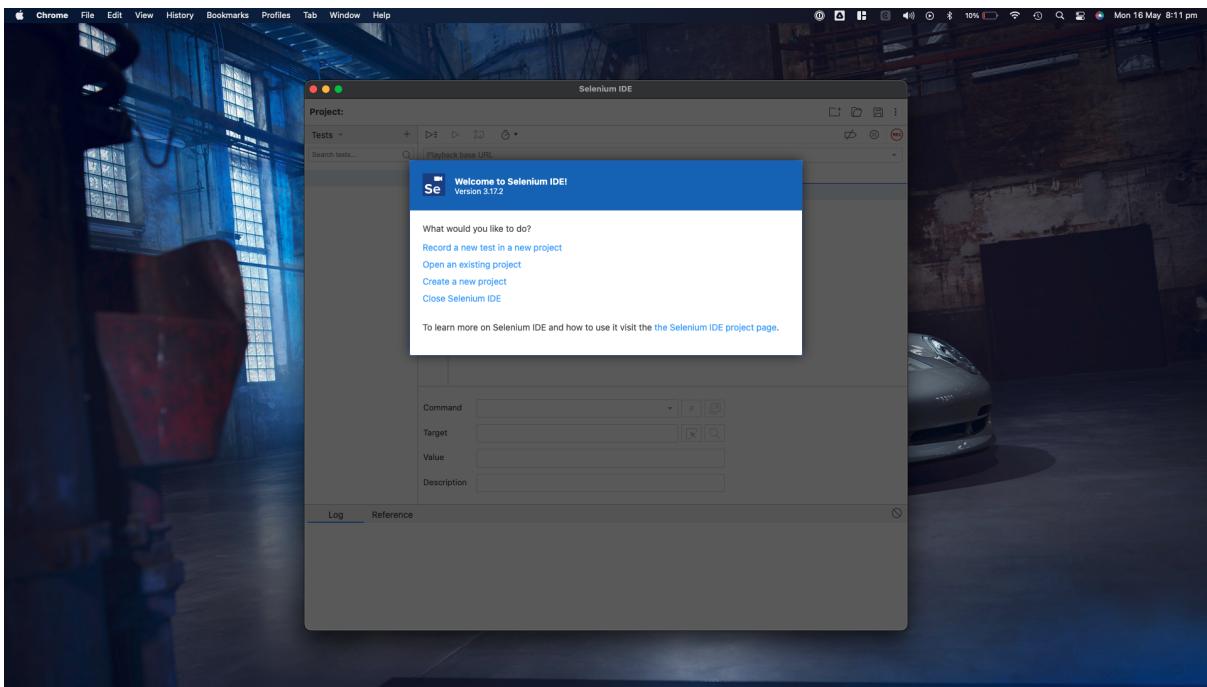


After installing the plugin, launch the IDE by clicking its icon found to the right side of the menu bar.

A welcome dialog will present itself on the screen, providing you—the test case developer—with convenient access to:

- Record a test in a new project

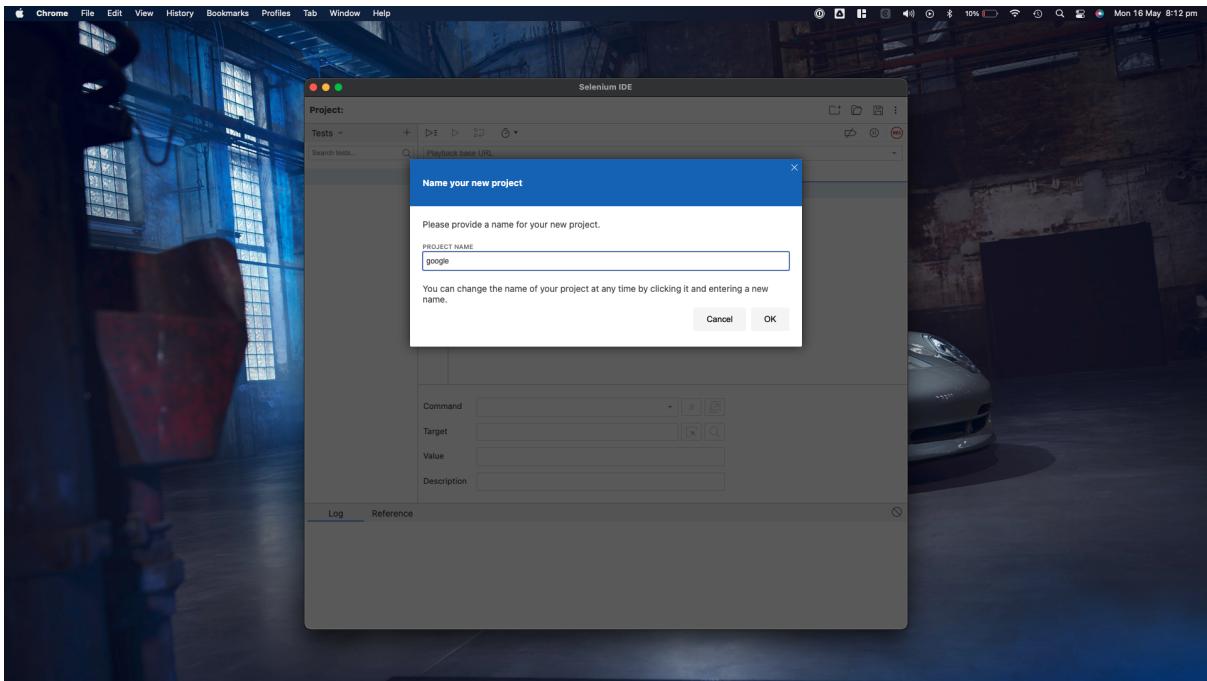
- Create a project
- Open a project
- Quit the IDE



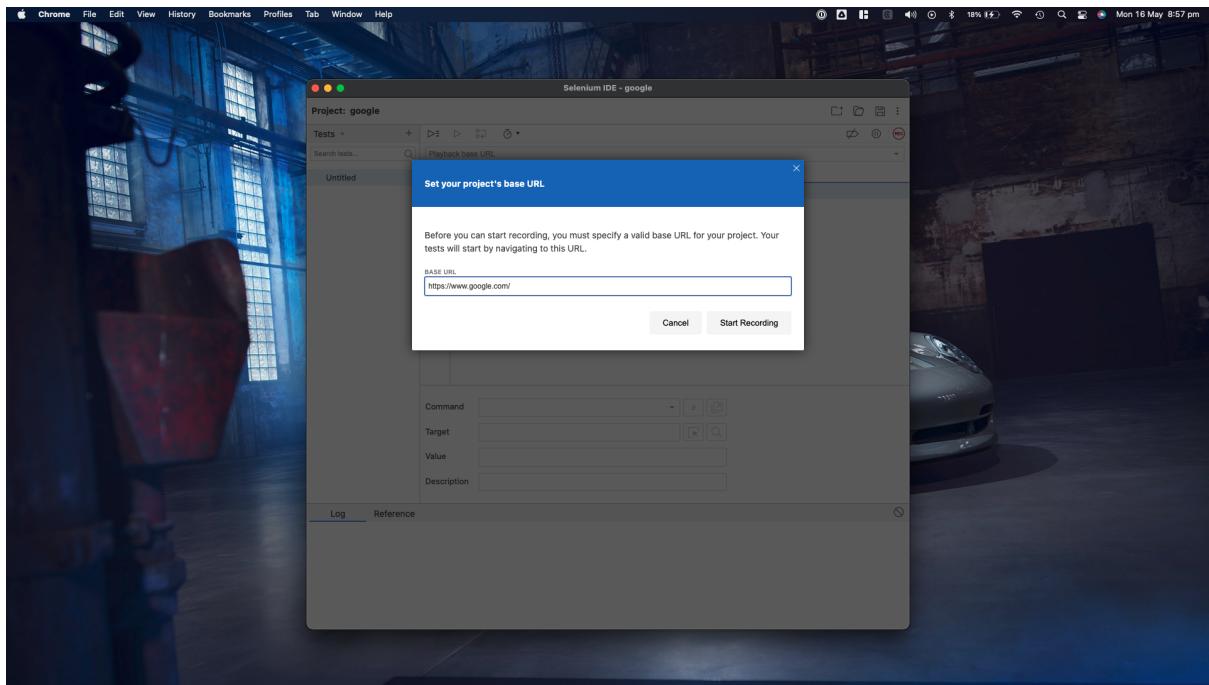
For the purposes of the walkthrough, select “Record a new test in a new project”.

2.2.2 Recording a Test

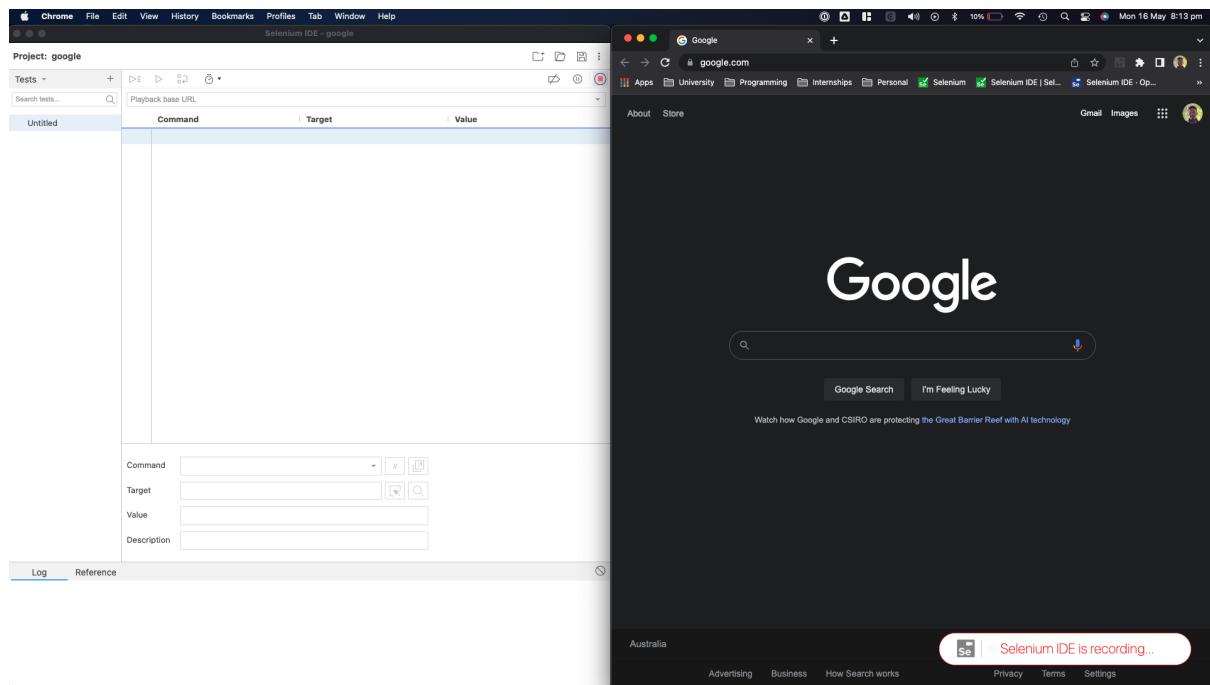
At the “PROJECT NAME” prompt, enter “google”.



At the subsequent “BASE URL” prompt, enter “<https://google.com/>”. The base URL is defined as the URL at which the application to be tested is hosted.



Shortly afterwards, a new browser window will open to “<https://google.com/>” and begin recording.



Observe that when interacting with the page each of your actions is recorded in the IDE.

The screenshot shows the Selenium IDE interface on the left and a browser window on the right. The browser window displays the Google search results for the query "Hello, world!". The Selenium IDE window shows a test named "Untitled*" with four recorded steps:

| Command | Target | Value |
|-----------------|----------|---------------|
| open | / | |
| set window size | 960x1055 | |
| type | name:q | Hello, world! |
| send keys | name:q | \${KEY_ENTER} |

If you want to stop recording, click the recording icon found on the IDE window.

The screenshot shows the Selenium IDE interface on the left and a browser window on the right. The browser window displays the Google search results for the query "Hello, world!". The Selenium IDE window shows a test named "Untitled*" with four recorded steps:

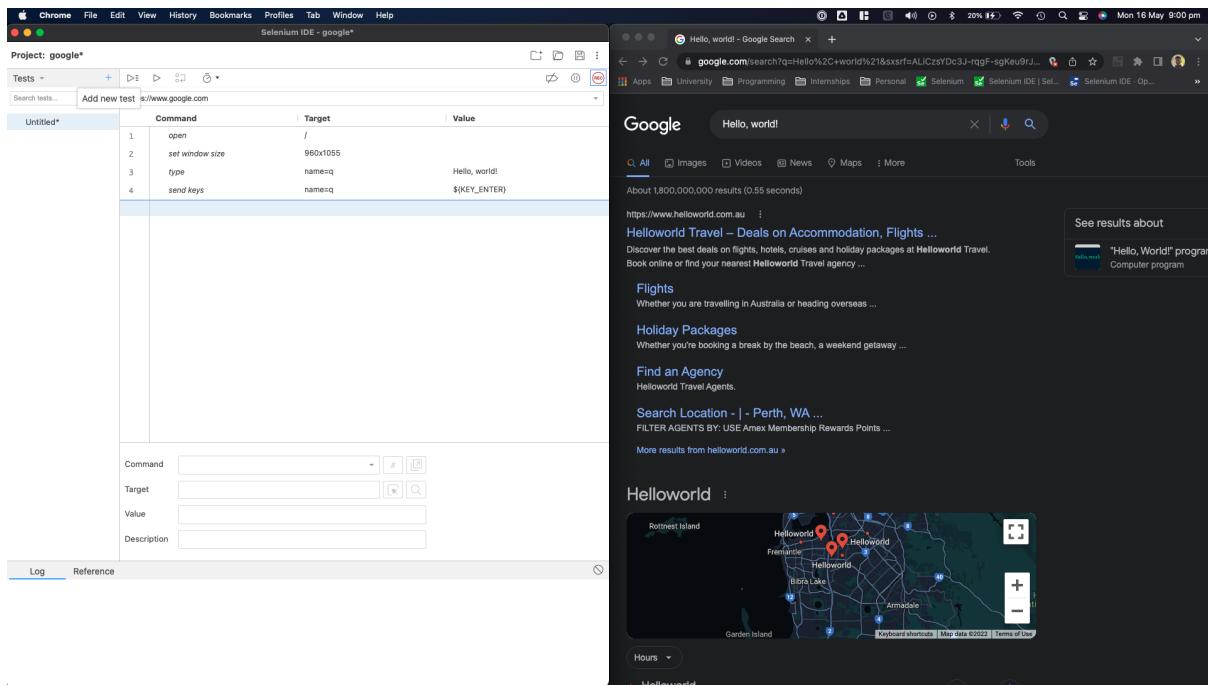
| Command | Target | Value |
|-----------------|----------|---------------|
| open | / | |
| set window size | 960x1055 | |
| type | name:q | Hello, world! |
| send keys | name:q | \${KEY_ENTER} |

A red box highlights the "Stop recording" button in the top right corner of the IDE window. A red callout bubble on the right side of the browser window says "Selenium IDE is recording...".

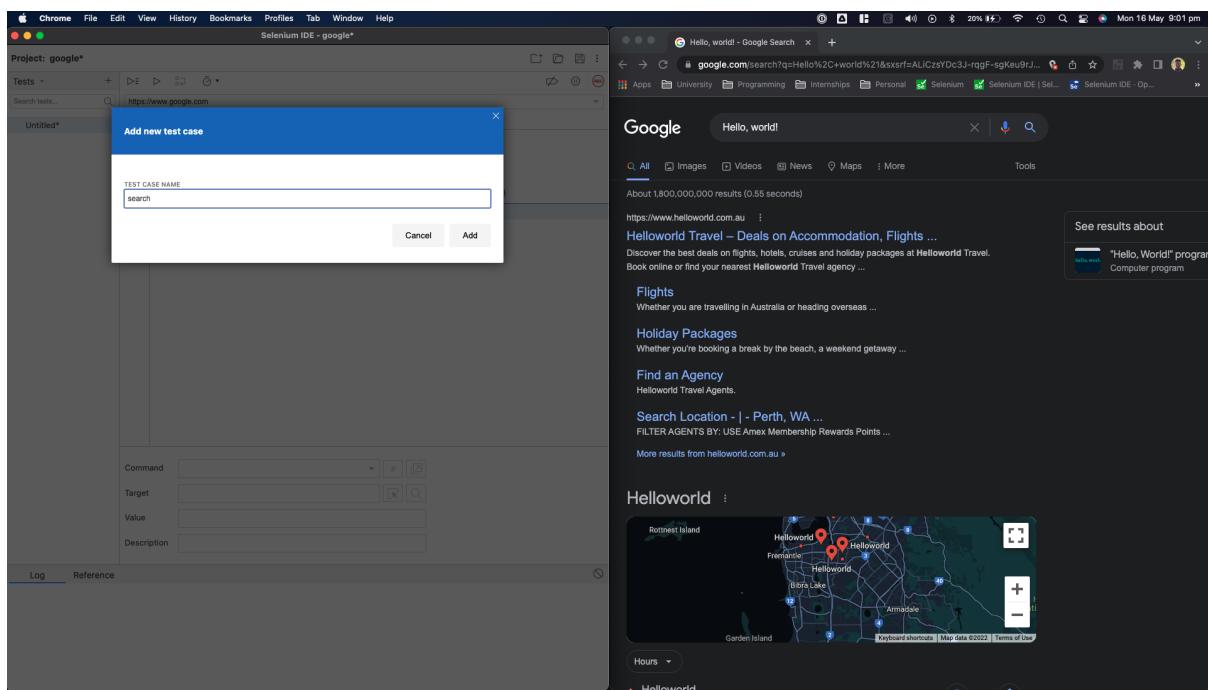
2.2.3 Organising Tests

2.2.3.1 Tests

To add a new test, click the "+" icon found to the right of "Tests".



Afterwards, at the “TEST CASE NAME” prompt, enter “search”, and then click “Add”.



Upon doing so, you can record the test manually by writing commands or automatically by clicking the record icon and executing the actions from which you want to form the test.

The screenshot shows the Selenium IDE interface on the left and a web browser window on the right. The browser window displays the Google search results for "Hello, world!". The Selenium IDE script contains the following commands:

| Command | Target | Value |
|-----------------|----------|---------------|
| open | / | |
| set window size | 960x1055 | |
| type | name:q | Hello, world! |
| send keys | name:q | \$(KEY_ENTER) |

2.2.3.2 Suites

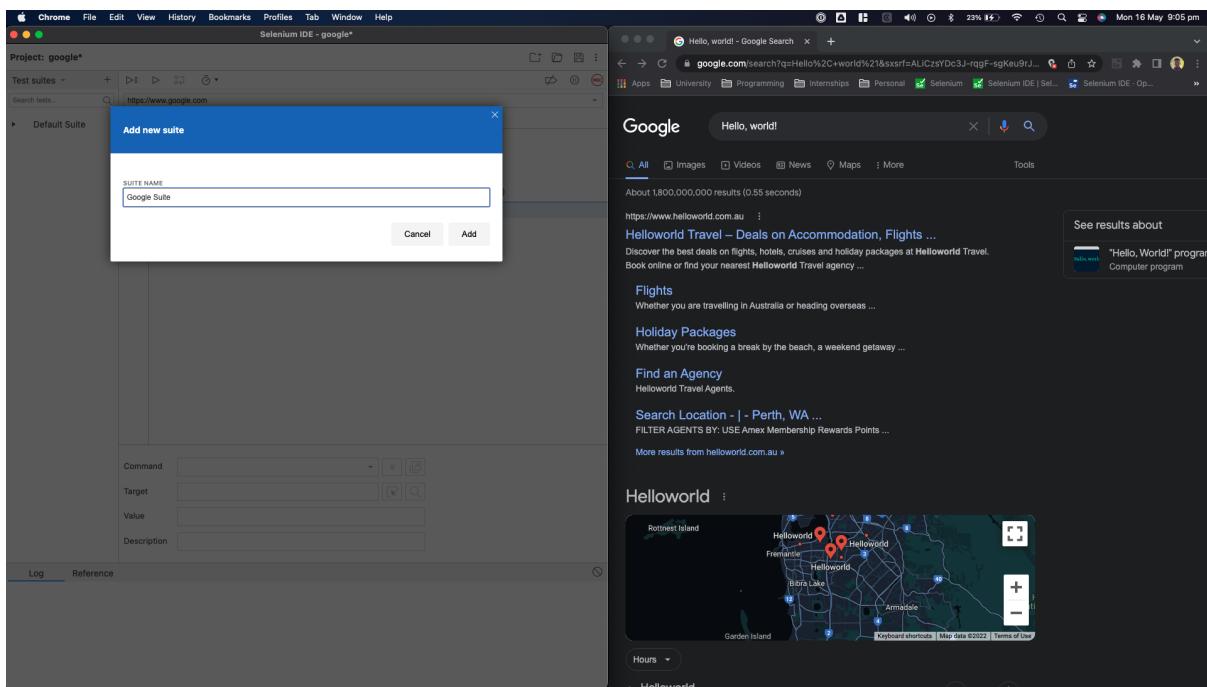
After creating a project, a test suite called “Default Suite” is created and your first test is added to it automatically. For suite creation and management, navigate to the “Test Suites” panel.

The screenshot shows the Selenium IDE interface on the left and a web browser window on the right. The browser window displays the Google search results for "Hello, world!". The Selenium IDE script is now part of a suite named "Google Suite".

| Command | Target | Value |
|-----------------|----------|---------------|
| open | / | |
| set window size | 960x1055 | |
| type | name:q | Hello, world! |
| send keys | name:q | \$(KEY_ENTER) |

2.2.3.2.1 Creating a Suite

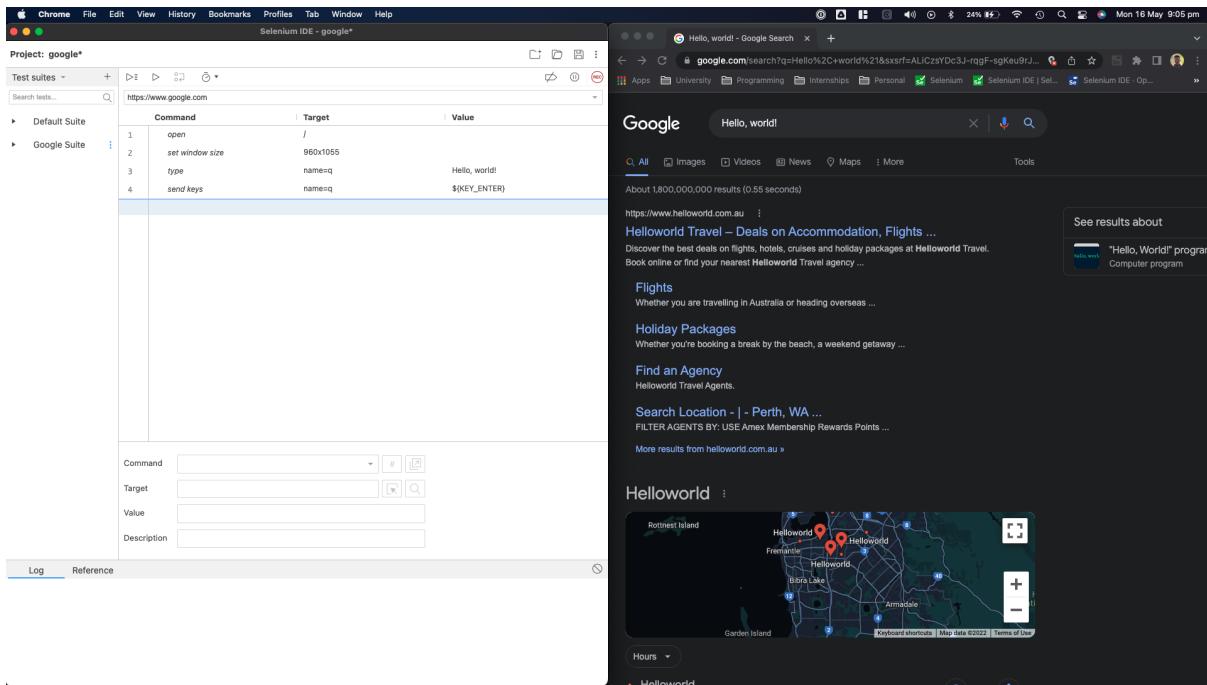
Click the “+” icon to the right of “Test suites”. Then, at the “Suite Name” prompt, enter “Google Suite”, and click “Add”.



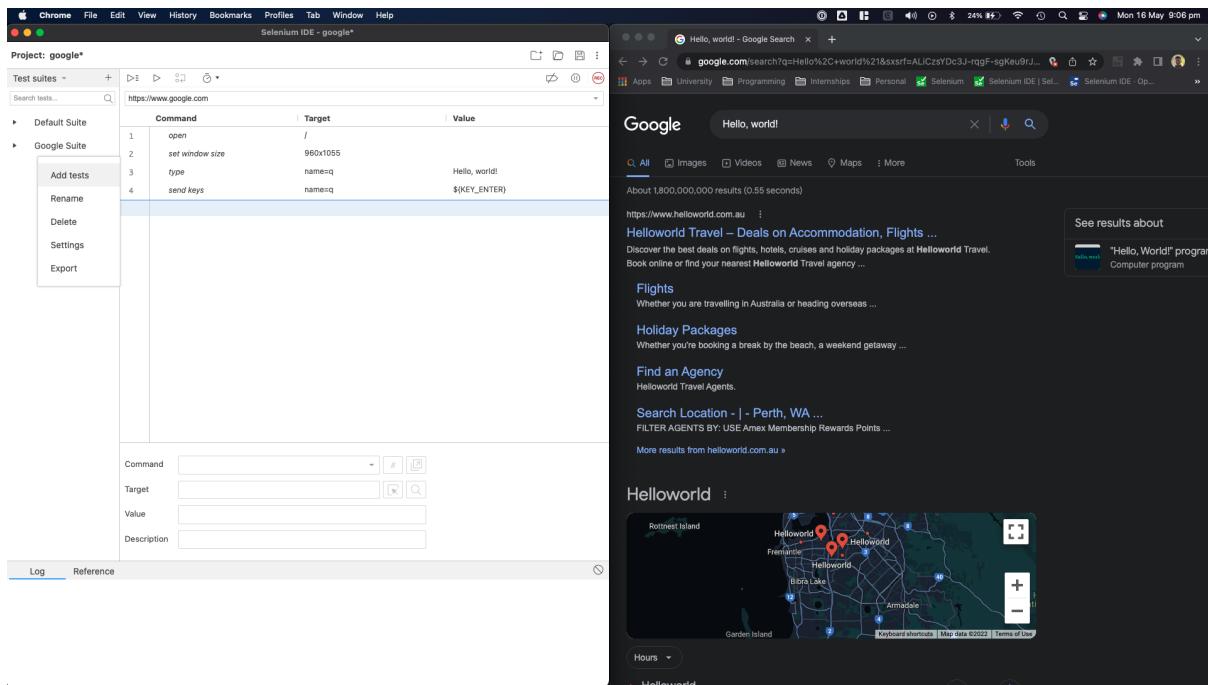
2.2.3.2.2 Adding a Test to a Suite

Hover over “Google Suite”, and then do the following:

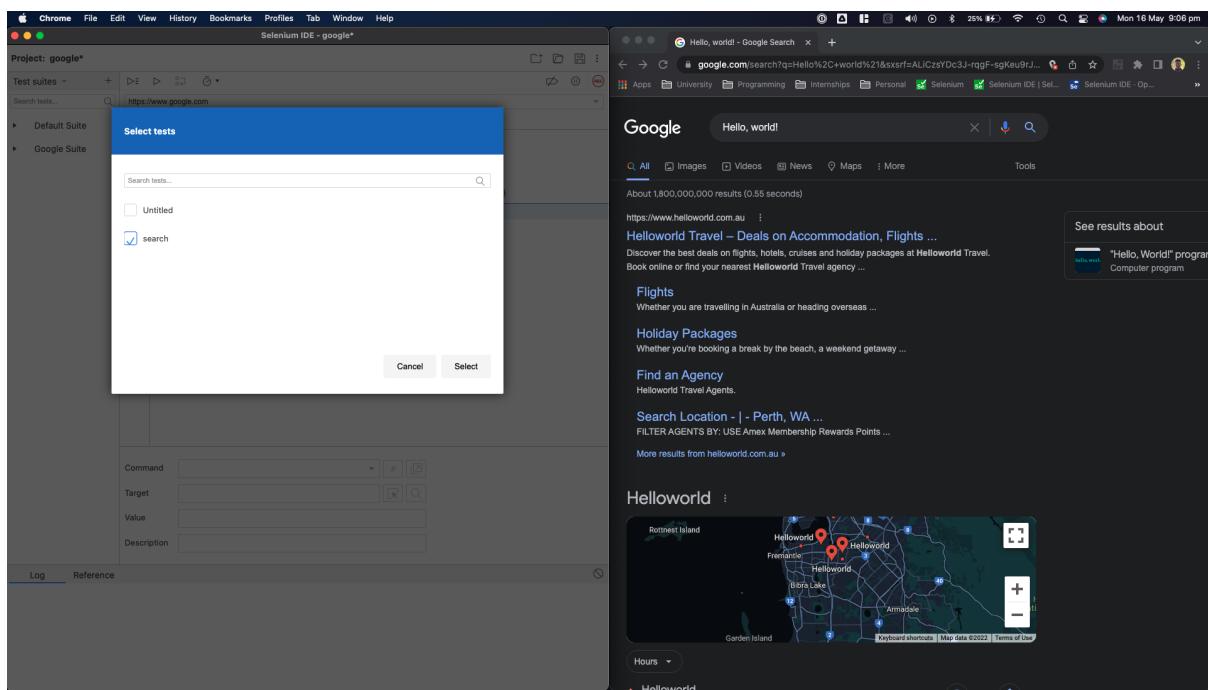
1. Click the “?” icon



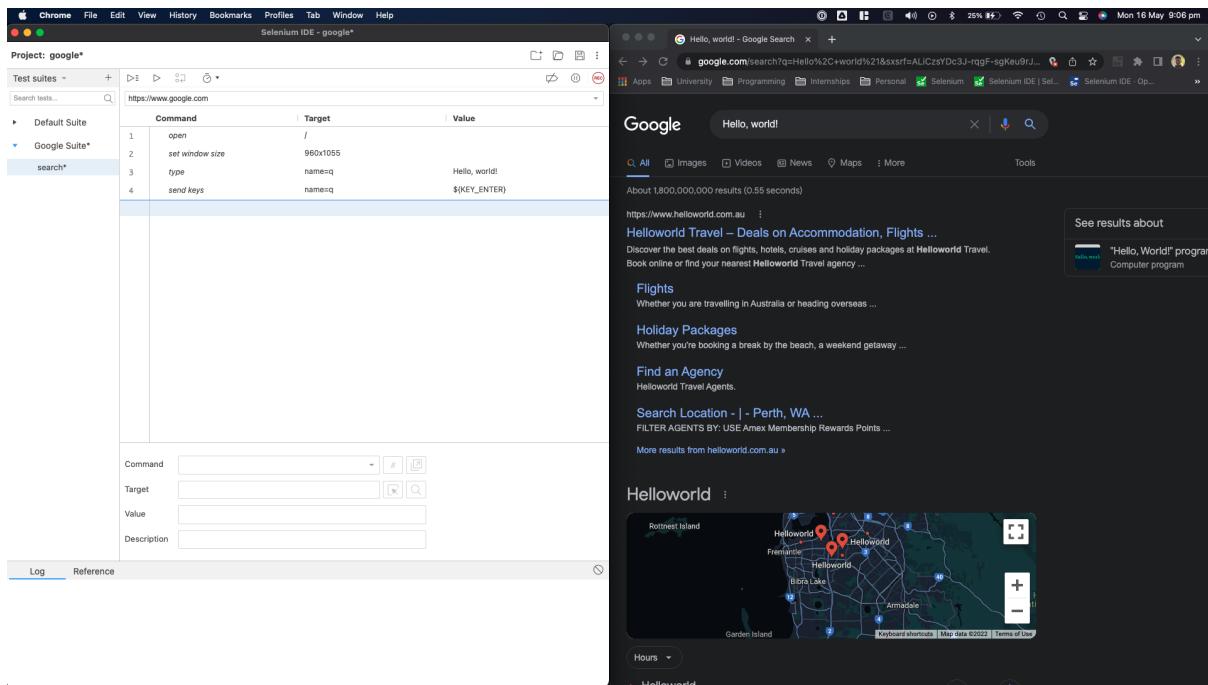
2. Click “Add tests”



3. Select “search”.

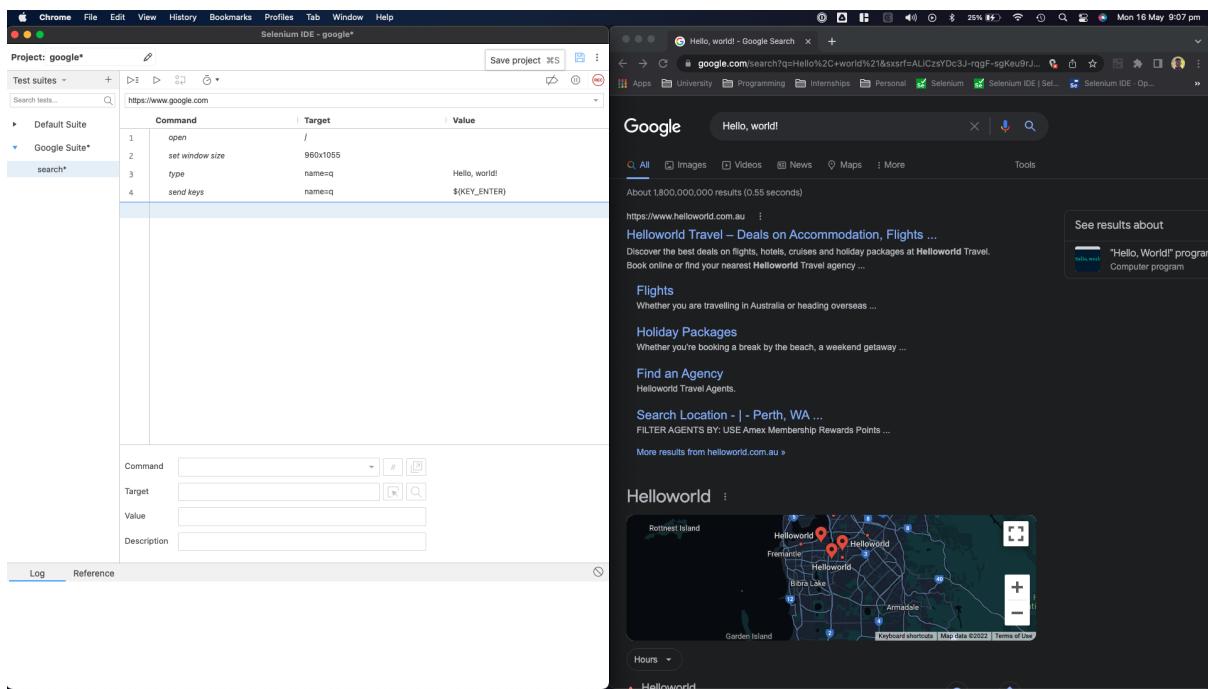


4. Click “Select”



2.2.4 Saving Work

To save everything done in the IDE thus far, click the save icon in the top-right corner of the IDE window.

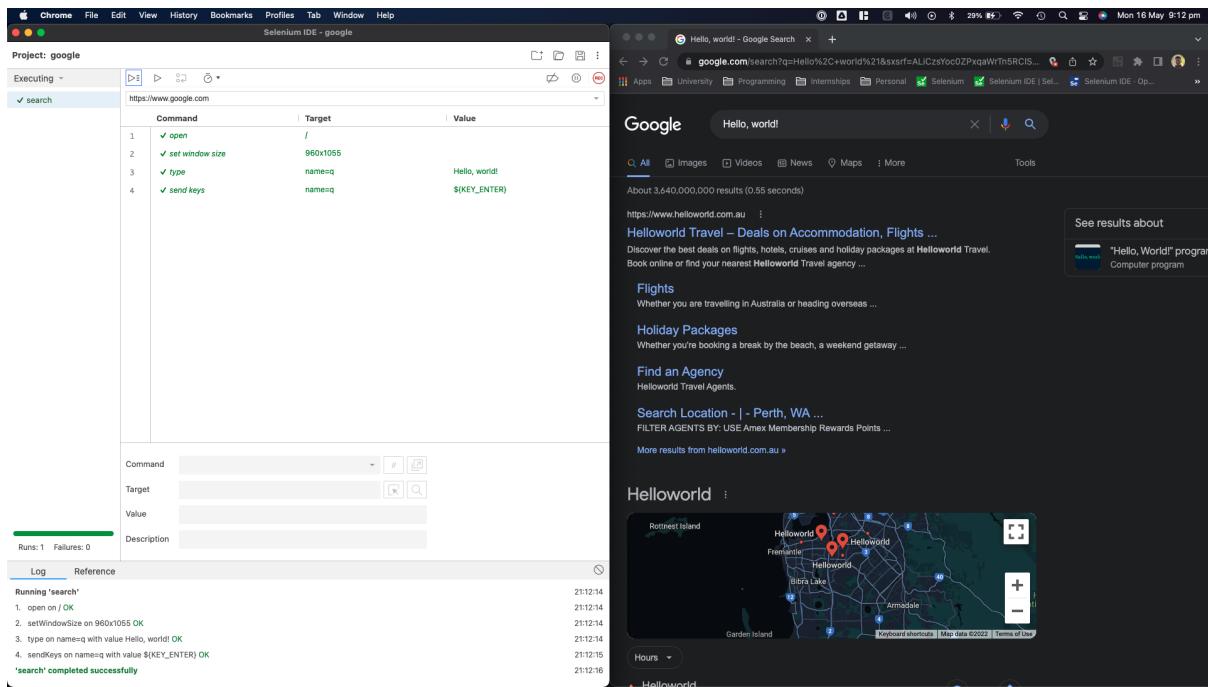


At the project filename prompt, enter “google” and specify the location at which you wish to save the project.

2.2.5 Playback

Select “Google Suite” and then click the play button in the menu bar above the test editor.

Thereafter, the tests contained within the selected suite—in our case, just “search”—will play back in the browser.



2.2 Relevant Considerations

2.2.1 Advantages

2.2.1.1 Cross-Platform Support

Selenium IDE allows test-case developers to record test cases on macOS, Linux and Windows; record tests on Firefox and Google Chrome; and export these tests to Java-, Python- and JavaScript-based test libraries (altexsoft 2021).

2.2.1.2 Extensibility with Plugins

Unlike previous versions of Selenium IDE, test case developers are now able to tailor the IDE to their requirements or the requirements of their organisation via the use of third-party plugins (Battat 2021).

2.2.1.3 Popularity

Selenium occupies a 27.48 percent market share of the software testing tools market. Its next closest competitor, Apache Jmeter, takes less than half of the market space, coming in at a mere 10 percent (Battat 2021).

2.2.2 Disadvantages

2.2.2.1 Reporting Capabilities

Since Selenium IDE lacks automatically generated reports, in order to capture test failures, test case developers are essentially required to take screenshots of the output of failing tests (Battat 2021).

2.2.2.2 Test Maintenance

Due to the fact that tests rely on a single element identifier, any slight modifications to these identifiers can cause tests to fail. This can ultimately derail future releases since developers will need to debug and rerun the necessary tests (testim 2020).

2.2.2.3 Standards

The open-source nature of Selenium IDE has resulted in great variation in the usage of various builtin features. In order for a test case developer to accomplish anything complicated, they will likely need to dedicate considerable time familiarising themselves with the usage of different commands (Yarn 2016).

2.3 Verdict

Ultimately, we strongly recommend Selenium IDE for individuals developing smaller-scale web apps, as its out of the box functionality allows for relative ease in recording, editing and executing tests. However, for teams with more stringent requirements—namely, those working on larger-scale applications—we would not recommend Selenium IDE, unless, there exists a set of plugins that (i) appropriately meets your requirements, and (ii) your team is willing and able to learn.

References

- altexsoft. 2021. "The Good and the Bad of Selenium Test Automation Software". altexsoft. <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-selenium-test-automation-tool/>
- Battat, Michael. 2021. "16 reasons why to use Selenium IDE in 2022 (and 1 why not)". appliTools. <https://appliTools.com/blog/why-selenium-ide-2019/>
- LucidProject. 2020a. "LucidProject/platform_frameworks_base". LucidProject. https://github.com/LucidProject/platform_frameworks_base
- LucidProject. 2020b. "WifiConfiguration.java". LucidProject. https://github.com/LucidProject/platform_frameworks_base/blob/d6d4abf090ba3982542fb6ca88c795f2bcda7177/wifi/java/android/net/wifi/WifiConfiguration.java
- LucidProject. 2020c. "WifiConfigurationTest.java". LucidProject. https://github.com/LucidProject/platform_frameworks_base/blob/d6d4abf090ba3982542fb6ca88c795f2bcda7177/wifi/tests/src/android/net/wifi/WifiConfigurationTest.java
- Selenium. n.d. "Selenium IDE". Selenium. <https://www.selenium.dev/documentation/ide/>
- testim. 2020. "A Detailed Look at Selenium Pros and Cons". testim. <https://www.testim.io/blog/selenium-pros-and-cons/>
- Yarn, Jared. 2016. "Selenium IDE: The Good, The Bad, and the Ugly". Lucidchart. <https://www.lucidchart.com/techblog/2016/09/13/selenium-ide-the-good-the-bad-and-the-ugly/>

Appendices

Method 1 Prime paths:

[1, 2, 4]

[1, 2, 3, 4]

[1, 3, 4]

[1, 2, 3, 5, 4]

[1, 3, 5, 4]

[1, 2, 3, 5, 6, 7, 9]

[1, 3, 5, 6, 7, 9]

[1, 2, 3, 6, 7, 9]

[1, 3, 6, 7, 9]

[1, 2, 3, 5, 6, 7, 8, 10, 9]

[1, 3, 5, 6, 7, 8, 10, 9]

[1, 2, 3, 6, 7, 8, 10, 9]

[1, 3, 6, 7, 8, 10, 9]

[1, 2, 3, 5, 6, 8, 10, 9]

[1, 3, 5, 6, 8, 10, 9]

[1, 2, 3, 6, 8, 10, 9]

[1, 3, 6, 8, 10, 9]

[1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 9]

[1, 3, 5, 6, 7, 8, 10, 11, 12, 9]

[1, 2, 3, 6, 7, 8, 10, 11, 12, 9]

[1, 3, 6, 7, 8, 10, 11, 12, 9]

[1, 2, 3, 5, 6, 8, 10, 11, 12, 9]

[1, 3, 5, 6, 8, 10, 11, 12, 9]

[1, 2, 3, 6, 8, 10, 11, 12, 9]

[1, 3, 6, 8, 10, 11, 12, 9]

[1, 2, 3, 5, 6, 7, 8, 11, 12, 9]

[1, 3, 5, 6, 7, 8, 11, 12, 9]

[1, 2, 3, 6, 7, 8, 11, 12, 9]

[1, 3, 6, 7, 8, 11, 12, 9]

[1, 2, 3, 5, 6, 8, 11, 12, 9]

[1, 3, 5, 6, 8, 11, 12, 9]

[1, 2, 3, 6, 8, 11, 12, 9]

[1, 3, 6, 8, 11, 12, 9]

[1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 13, 14, 9]

[1, 3, 5, 6, 7, 8, 10, 11, 12, 13, 14, 9]

[1, 2, 3, 6, 7, 8, 10, 11, 12, 13, 14, 9]

[1, 3, 6, 7, 8, 10, 11, 12, 13, 14, 9]

[1, 2, 3, 5, 6, 8, 10, 11, 12, 13, 14, 9]
[1, 3, 5, 6, 8, 10, 11, 12, 13, 14, 9]
[1, 2, 3, 6, 8, 10, 11, 12, 13, 14, 9]
[1, 3, 6, 8, 10, 11, 12, 13, 14, 9]
[1, 2, 3, 5, 6, 7, 8, 11, 12, 13, 14, 9]
[1, 3, 5, 6, 7, 8, 11, 12, 13, 14, 9]
[1, 2, 3, 6, 7, 8, 11, 12, 13, 14, 9]
[1, 3, 6, 7, 8, 11, 12, 13, 14, 9]
[1, 2, 3, 5, 6, 8, 11, 12, 13, 14, 9]
[1, 3, 5, 6, 8, 11, 12, 13, 14, 9]
[1, 2, 3, 6, 8, 11, 12, 13, 14, 9]
[1, 3, 6, 8, 11, 12, 13, 14, 9]
[1, 2, 3, 5, 6, 7, 8, 10, 11, 13, 14, 9]
[1, 3, 5, 6, 7, 8, 10, 11, 13, 14, 9]
[1, 2, 3, 6, 7, 8, 10, 11, 13, 14, 9]
[1, 3, 6, 7, 8, 10, 11, 13, 14, 9]
[1, 2, 3, 5, 6, 8, 10, 11, 13, 14, 9]
[1, 3, 5, 6, 8, 10, 11, 13, 14, 9]
[1, 2, 3, 6, 8, 10, 11, 13, 14, 9]
[1, 3, 6, 8, 10, 11, 13, 14, 9]
[1, 2, 3, 5, 6, 7, 8, 11, 13, 14, 9]
[1, 3, 5, 6, 7, 8, 11, 13, 14, 9]
[1, 2, 3, 6, 7, 8, 11, 13, 14, 9]
[1, 3, 6, 7, 8, 11, 13, 14, 9]
[1, 2, 3, 5, 6, 8, 11, 13, 14, 9]
[1, 3, 5, 6, 8, 11, 13, 14, 9]
[1, 2, 3, 6, 8, 11, 13, 14, 9]
[1, 3, 6, 8, 11, 13, 14, 9]

[1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16]
[1, 3, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16]
[1, 2, 3, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16]
[1, 3, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16]
[1, 2, 3, 5, 6, 8, 10, 11, 12, 13, 14, 15, 16]
[1, 3, 5, 6, 8, 10, 11, 12, 13, 14, 15, 16]
[1, 2, 3, 6, 8, 10, 11, 12, 13, 14, 15, 16]
[1, 3, 6, 8, 10, 11, 12, 13, 14, 15, 16]
[1, 2, 3, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16]
[1, 3, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16]
[1, 2, 3, 6, 7, 8, 11, 12, 13, 14, 15, 16]
[1, 3, 6, 7, 8, 11, 12, 13, 14, 15, 16]
[1, 2, 3, 5, 6, 8, 11, 12, 13, 14, 15, 16]
[1, 3, 5, 6, 8, 11, 12, 13, 14, 15, 16]
[1, 2, 3, 6, 8, 11, 12, 13, 14, 15, 16]
[1, 3, 6, 8, 11, 12, 13, 14, 15, 16]
[1, 2, 3, 5, 6, 7, 8, 10, 11, 13, 14, 15, 16]

[1, 3, 5, 6, 7, 8, 10, 11, 13, 14, 15, 16]
[1, 2, 3, 6, 7, 8, 10, 11, 13, 14, 15, 16]
[1, 3, 6, 7, 8, 10, 11, 13, 14, 15, 16]
[1, 2, 3, 5, 6, 8, 10, 11, 13, 14, 15, 16]
[1, 3, 5, 6, 8, 10, 11, 13, 14, 15, 16]
[1, 2, 3, 6, 8, 10, 11, 13, 14, 15, 16]
[1, 3, 6, 8, 10, 11, 13, 14, 15, 16]
[1, 2, 3, 5, 6, 7, 8, 11, 13, 14, 15, 16]
[1, 3, 5, 6, 7, 8, 11, 13, 14, 15, 16]
[1, 2, 3, 6, 7, 8, 11, 13, 14, 15, 16]
[1, 3, 6, 7, 8, 11, 13, 14, 15, 16]
[1, 2, 3, 5, 6, 8, 11, 13, 14, 15, 16]
[1, 3, 5, 6, 8, 11, 13, 14, 15, 16]
[1, 2, 3, 6, 8, 11, 13, 14, 15, 16]
[1, 3, 6, 8, 11, 13, 14, 15, 16]
[1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 13, 15, 16]
[1, 3, 5, 6, 7, 8, 10, 11, 12, 13, 15, 16]
[1, 2, 3, 6, 7, 8, 10, 11, 12, 13, 15, 16]
[1, 3, 6, 7, 8, 10, 11, 12, 13, 15, 16]
[1, 2, 3, 5, 6, 8, 10, 11, 12, 13, 15, 16]
[1, 3, 5, 6, 8, 10, 11, 12, 13, 15, 16]
[1, 2, 3, 6, 8, 10, 11, 12, 13, 15, 16]
[1, 3, 6, 8, 10, 11, 12, 13, 15, 16]
[1, 2, 3, 5, 6, 7, 8, 11, 12, 13, 15, 16]
[1, 3, 5, 6, 7, 8, 11, 12, 13, 15, 16]
[1, 2, 3, 6, 7, 8, 11, 12, 13, 15, 16]
[1, 3, 6, 7, 8, 11, 12, 13, 15, 16]
[1, 2, 3, 5, 6, 8, 11, 12, 13, 15, 16]
[1, 3, 5, 6, 8, 11, 12, 13, 15, 16]
[1, 2, 3, 6, 8, 11, 12, 13, 15, 16]
[1, 3, 6, 8, 11, 12, 13, 15, 16]
[1, 2, 3, 5, 6, 7, 8, 10, 11, 13, 15, 16]
[1, 3, 5, 6, 7, 8, 10, 11, 13, 15, 16]
[1, 2, 3, 6, 7, 8, 10, 11, 13, 15, 16]
[1, 3, 6, 7, 8, 10, 11, 13, 15, 16]
[1, 2, 3, 5, 6, 8, 10, 11, 13, 15, 16]
[1, 3, 5, 6, 8, 10, 11, 13, 15, 16]
[1, 2, 3, 6, 8, 10, 11, 13, 15, 16]
[1, 3, 6, 8, 10, 11, 13, 15, 16]
[1, 2, 3, 5, 6, 7, 8, 11, 13, 15, 16]
[1, 3, 5, 6, 7, 8, 11, 13, 15, 16]
[1, 2, 3, 6, 7, 8, 11, 13, 15, 16]
[1, 3, 6, 7, 8, 11, 13, 15, 16]
[1, 2, 3, 5, 6, 8, 11, 13, 15, 16]
[1, 3, 5, 6, 8, 11, 13, 15, 16]
[1, 2, 3, 6, 8, 11, 13, 15, 16]
[1, 3, 6, 8, 11, 13, 15, 16]

[1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 17, 4]
[1, 3, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 17, 4]
[1, 2, 3, 6, 7, 8, 10, 11, 12, 13, 14, 15, 17, 4]
[1, 3, 6, 7, 8, 10, 11, 12, 13, 14, 15, 17, 4]
[1, 2, 3, 5, 6, 8, 10, 11, 12, 13, 14, 15, 17, 4]
[1, 3, 5, 6, 8, 10, 11, 12, 13, 14, 15, 17, 4]
[1, 2, 3, 6, 8, 10, 11, 12, 13, 14, 15, 17, 4]
[1, 3, 6, 8, 10, 11, 12, 13, 14, 15, 17, 4]
[1, 2, 3, 5, 6, 7, 8, 11, 12, 13, 14, 15, 17, 4]
[1, 3, 5, 6, 7, 8, 11, 12, 13, 14, 15, 17, 4]
[1, 2, 3, 6, 7, 8, 11, 12, 13, 14, 15, 17, 4]
[1, 3, 6, 7, 8, 11, 12, 13, 14, 15, 17, 4]
[1, 2, 3, 5, 6, 8, 11, 12, 13, 14, 15, 17, 4]
[1, 3, 5, 6, 8, 11, 12, 13, 14, 15, 17, 4]
[1, 2, 3, 6, 8, 11, 12, 13, 14, 15, 17, 4]
[1, 3, 6, 8, 11, 12, 13, 14, 15, 17, 4]
[1, 2, 3, 5, 6, 7, 8, 10, 11, 13, 14, 15, 17, 4]
[1, 3, 5, 6, 7, 8, 10, 11, 13, 14, 15, 17, 4]
[1, 2, 3, 6, 7, 8, 10, 11, 13, 14, 15, 17, 4]
[1, 3, 6, 7, 8, 10, 11, 13, 14, 15, 17, 4]
[1, 2, 3, 5, 6, 8, 10, 11, 13, 14, 15, 17, 4]
[1, 3, 5, 6, 8, 10, 11, 13, 14, 15, 17, 4]
[1, 2, 3, 6, 8, 10, 11, 13, 14, 15, 17, 4]
[1, 3, 6, 8, 10, 11, 13, 14, 15, 17, 4]
[1, 2, 3, 5, 6, 7, 8, 11, 13, 14, 15, 17, 4]
[1, 3, 5, 6, 7, 8, 11, 13, 14, 15, 17, 4]
[1, 2, 3, 6, 7, 8, 11, 13, 14, 15, 17, 4]
[1, 3, 6, 7, 8, 11, 13, 14, 15, 17, 4]
[1, 2, 3, 5, 6, 8, 11, 13, 14, 15, 17, 4]
[1, 3, 5, 6, 8, 11, 13, 14, 15, 17, 4]
[1, 2, 3, 6, 8, 11, 13, 14, 15, 17, 4]
[1, 3, 6, 8, 11, 13, 14, 15, 17, 4]
[1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 13, 15, 17, 4]
[1, 3, 5, 6, 7, 8, 10, 11, 12, 13, 15, 17, 4]
[1, 2, 3, 6, 7, 8, 10, 11, 12, 13, 15, 17, 4]
[1, 3, 6, 7, 8, 10, 11, 12, 13, 15, 17, 4]
[1, 2, 3, 5, 6, 8, 10, 11, 12, 13, 15, 17, 4]
[1, 3, 5, 6, 8, 10, 11, 12, 13, 15, 17, 4]
[1, 2, 3, 6, 8, 10, 11, 12, 13, 15, 17, 4]
[1, 3, 6, 8, 10, 11, 12, 13, 15, 17, 4]
[1, 2, 3, 5, 6, 7, 8, 11, 12, 13, 15, 17, 4]
[1, 3, 5, 6, 7, 8, 11, 12, 13, 15, 17, 4]
[1, 2, 3, 6, 7, 8, 11, 12, 13, 15, 17, 4]
[1, 3, 6, 7, 8, 11, 12, 13, 15, 17, 4]
[1, 2, 3, 5, 6, 8, 11, 12, 13, 15, 17, 4]
[1, 3, 5, 6, 8, 11, 12, 13, 15, 17, 4]

[1, 2, 3, 6, 8, 11, 12, 13, 15, 17, 4]
[1, 3, 6, 8, 11, 12, 13, 15, 17, 4]
[1, 2, 3, 5, 6, 7, 8, 10, 11, 13, 15, 17, 4]
[1, 3, 5, 6, 7, 8, 10, 11, 13, 15, 17, 4]
[1, 2, 3, 6, 7, 8, 10, 11, 13, 15, 17, 4]
[1, 3, 6, 7, 8, 10, 11, 13, 15, 17, 4]
[1, 2, 3, 5, 6, 8, 10, 11, 13, 15, 17, 4]
[1, 3, 5, 6, 8, 10, 11, 13, 15, 17, 4]
[1, 2, 3, 6, 8, 10, 11, 13, 15, 17, 4]
[1, 3, 6, 8, 10, 11, 13, 15, 17, 4]
[1, 2, 3, 5, 6, 7, 8, 11, 13, 15, 17, 4]
[1, 3, 5, 6, 7, 8, 11, 13, 15, 17, 4]
[1, 2, 3, 6, 7, 8, 11, 13, 15, 17, 4]
[1, 3, 6, 7, 8, 11, 13, 15, 17, 4]
[1, 2, 3, 5, 6, 8, 11, 13, 15, 17, 4]
[1, 3, 5, 6, 8, 11, 13, 15, 17, 4]
[1, 2, 3, 6, 8, 11, 13, 15, 17, 4]
[1, 3, 6, 8, 11, 13, 15, 17, 4]

[1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 17, 18]
[1, 3, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 17, 18]
[1, 2, 3, 6, 7, 8, 10, 11, 12, 13, 14, 15, 17, 18]
[1, 3, 6, 7, 8, 10, 11, 12, 13, 14, 15, 17, 18]
[1, 2, 3, 5, 6, 8, 10, 11, 12, 13, 14, 15, 17, 18]
[1, 3, 5, 6, 8, 10, 11, 12, 13, 14, 15, 17, 18]
[1, 2, 3, 6, 8, 10, 11, 12, 13, 14, 15, 17, 18]
[1, 3, 6, 8, 10, 11, 12, 13, 14, 15, 17, 18]
[1, 2, 3, 5, 6, 7, 8, 11, 12, 13, 14, 15, 17, 18]
[1, 3, 5, 6, 7, 8, 11, 12, 13, 14, 15, 17, 18]
[1, 2, 3, 6, 8, 10, 11, 12, 13, 14, 15, 17, 18]
[1, 3, 6, 8, 10, 11, 12, 13, 14, 15, 17, 18]
[1, 2, 3, 5, 6, 7, 8, 11, 12, 13, 14, 15, 17, 18]
[1, 3, 5, 6, 8, 11, 12, 13, 14, 15, 17, 18]
[1, 2, 3, 6, 8, 10, 11, 12, 13, 14, 15, 17, 18]
[1, 3, 6, 8, 10, 11, 12, 13, 14, 15, 17, 18]
[1, 2, 3, 5, 6, 7, 8, 10, 11, 13, 14, 15, 17, 18]
[1, 3, 5, 6, 7, 8, 10, 11, 13, 14, 15, 17, 18]
[1, 2, 3, 6, 7, 8, 10, 11, 13, 14, 15, 17, 18]
[1, 3, 6, 7, 8, 10, 11, 13, 14, 15, 17, 18]
[1, 2, 3, 5, 6, 8, 10, 11, 13, 14, 15, 17, 18]
[1, 3, 5, 6, 8, 10, 11, 13, 14, 15, 17, 18]
[1, 2, 3, 6, 8, 10, 11, 13, 14, 15, 17, 18]
[1, 3, 6, 8, 10, 11, 13, 14, 15, 17, 18]
[1, 2, 3, 5, 6, 7, 8, 11, 13, 14, 15, 17, 18]
[1, 3, 5, 6, 7, 8, 11, 13, 14, 15, 17, 18]
[1, 2, 3, 6, 7, 8, 11, 13, 14, 15, 17, 18]
[1, 3, 6, 7, 8, 11, 13, 14, 15, 17, 18]

[1, 2, 3, 5, 6, 8, 11, 13, 14, 15, 17, 18]
[1, 3, 5, 6, 8, 11, 13, 14, 15, 17, 18]
[1, 2, 3, 6, 8, 11, 13, 14, 15, 17, 18]
[1, 3, 6, 8, 11, 13, 14, 15, 17, 18]
[1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 13, 15, 17, 18]
[1, 3, 5, 6, 7, 8, 10, 11, 12, 13, 15, 17, 18]
[1, 2, 3, 6, 7, 8, 10, 11, 12, 13, 15, 17, 18]
[1, 3, 6, 7, 8, 10, 11, 12, 13, 15, 17, 18]
[1, 2, 3, 5, 6, 8, 10, 11, 12, 13, 15, 17, 18]
[1, 3, 5, 6, 8, 10, 11, 12, 13, 15, 17, 18]
[1, 2, 3, 6, 8, 10, 11, 12, 13, 15, 17, 18]
[1, 3, 6, 8, 10, 11, 12, 13, 15, 17, 18]
[1, 2, 3, 5, 6, 7, 8, 11, 12, 13, 15, 17, 18]
[1, 3, 5, 6, 7, 8, 11, 12, 13, 15, 17, 18]
[1, 2, 3, 6, 7, 8, 11, 12, 13, 15, 17, 18]
[1, 3, 6, 7, 8, 11, 12, 13, 15, 17, 18]
[1, 2, 3, 5, 6, 8, 11, 12, 13, 15, 17, 18]
[1, 3, 5, 6, 8, 11, 12, 13, 15, 17, 18]
[1, 2, 3, 6, 8, 11, 12, 13, 15, 17, 18]
[1, 3, 6, 8, 11, 12, 13, 15, 17, 18]
[1, 2, 3, 5, 6, 7, 8, 10, 11, 13, 15, 17, 18]
[1, 3, 5, 6, 7, 8, 10, 11, 13, 15, 17, 18]
[1, 2, 3, 6, 7, 8, 10, 11, 13, 15, 17, 18]
[1, 3, 6, 7, 8, 10, 11, 13, 15, 17, 18]
[1, 2, 3, 5, 6, 8, 10, 11, 13, 15, 17, 18]
[1, 3, 5, 6, 8, 10, 11, 13, 15, 17, 18]
[1, 2, 3, 6, 8, 10, 11, 13, 15, 17, 18]
[1, 3, 6, 8, 10, 11, 13, 15, 17, 18]
[1, 2, 3, 5, 6, 8, 11, 13, 15, 17, 18]
[1, 3, 5, 6, 8, 11, 13, 15, 17, 18]
[1, 2, 3, 6, 8, 11, 13, 15, 17, 18]
[1, 3, 6, 8, 11, 13, 15, 17, 18]