

# 实验报告

课程名称：软件测试

实验名称：黑盒测试工具

Selenium IDE 使用练习

专业班级：软件工程 16 级四班

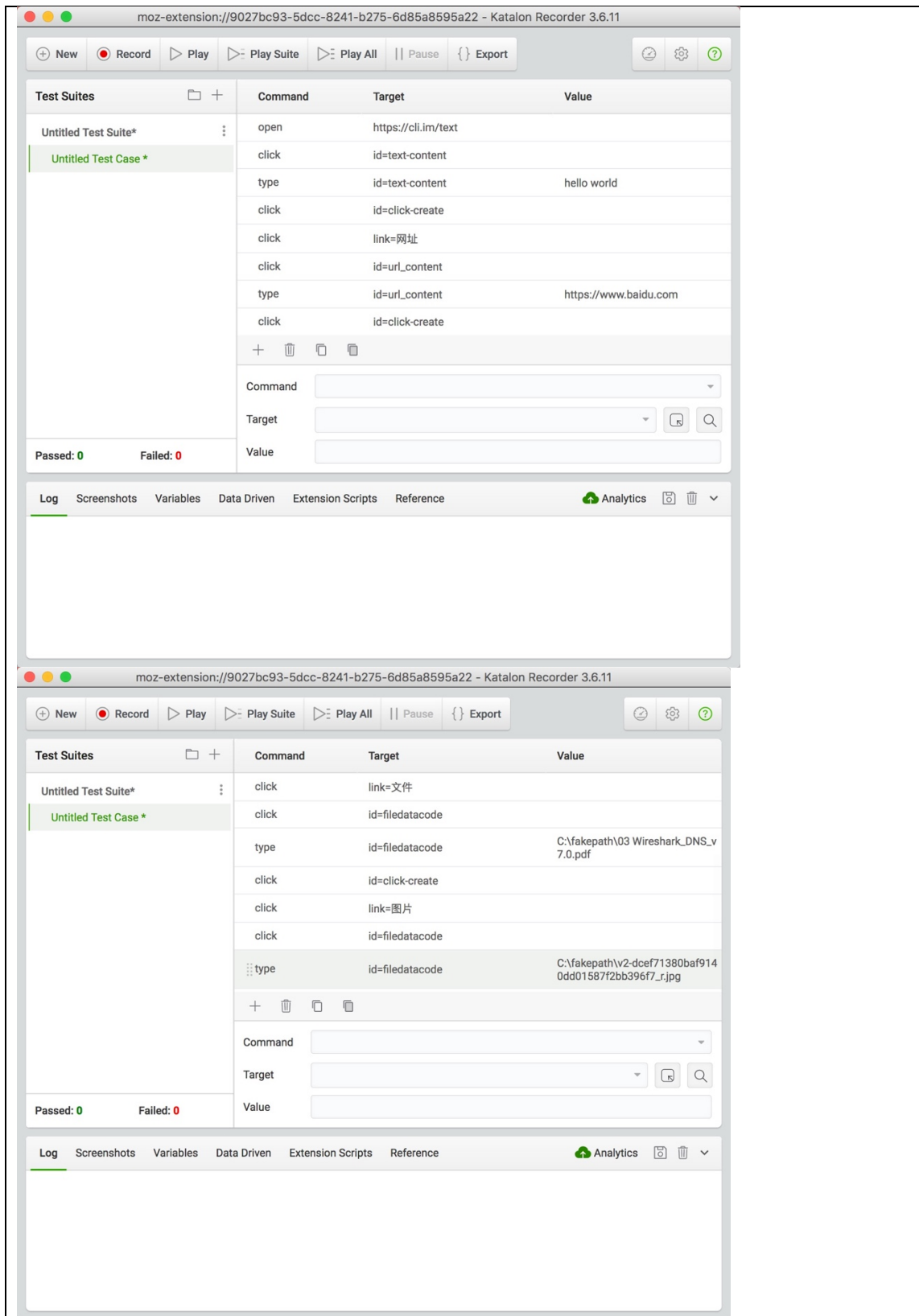
学 号：1611736

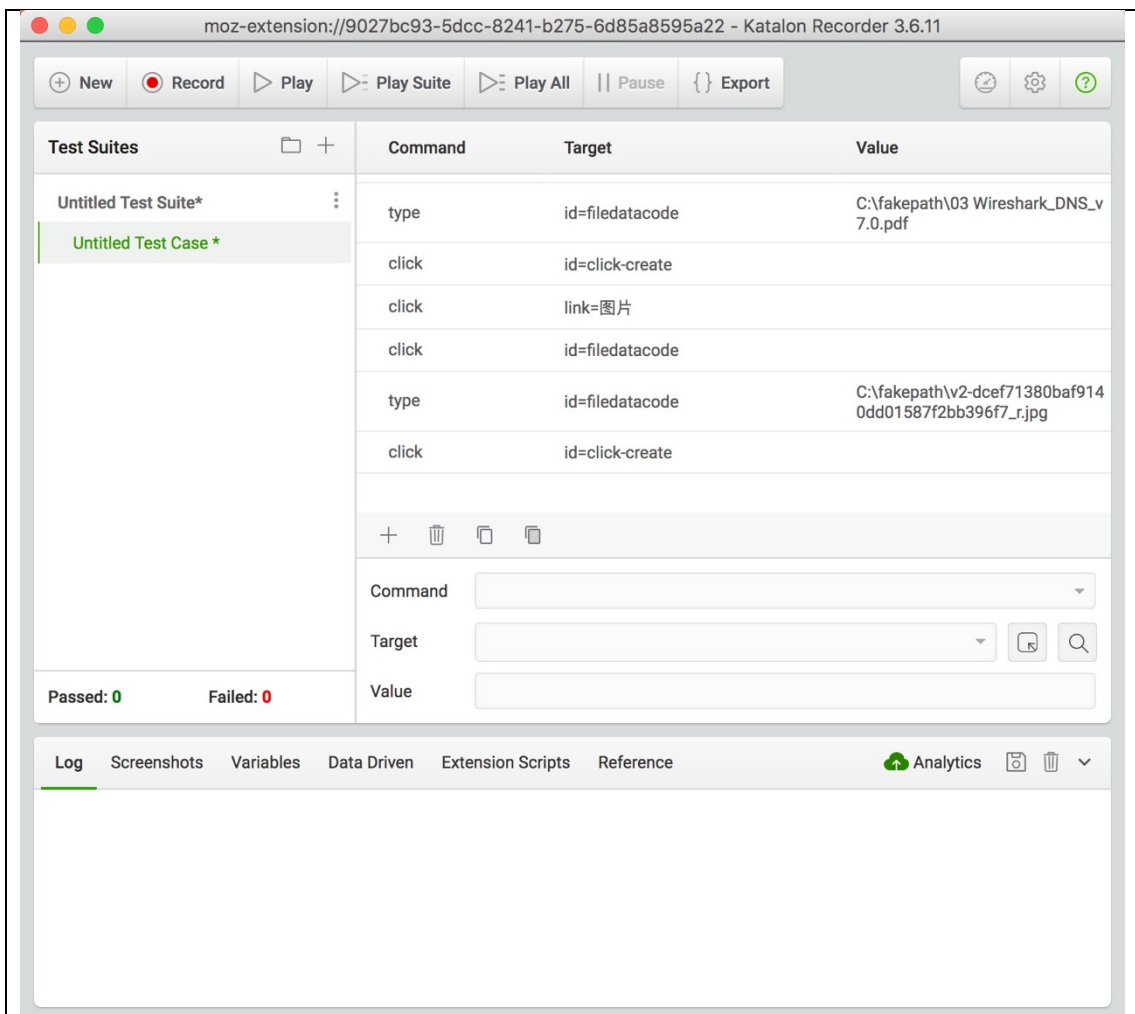
姓 名：钟腾

2018 年 11 月 6 日

## 实验一

实验名称	黑盒测试工具 Selenium IDE 使用练习		
实验地点	泰达 5 区 105	实验时间	2018/11/6
实验目的和要求			
安装火狐浏览器和 Selenium IDE，熟悉 Selenium 的使用，在线测试“草料二维码”生成二维码情况，要求录制操作过程并提交脚本			
实验环境			
Firefox 56.0 Katalon Recorder			
实验过程			
<ol style="list-style-type: none"><li>1. 打开火狐浏览器，添加扩展程序 Katalon Recorder，打开该扩展程序，点击 Record 按钮开始录制</li><li>2. 在浏览器输入 <a href="https://cli.im">https://cli.im</a> 进入草料二维码网站，点击文本并输入，生成二维码，再依次测试网址、文件、图片的上传和生成二维码功能。</li></ol>			





3. 点击 Katalon Recorder 中的 Record 按钮停止录制，点击 Export 生成 java 脚本文件。以下为生成的 java 脚本。

```
package com.example.tests;

import java.util.regex.Pattern;
import java.util.concurrent.TimeUnit;
import org.testng.annotations.*;
import static org.testng.Assert.*;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;

public class UntitledTestCase {
    private WebDriver driver;
    private String baseUrl;
    private boolean acceptNextAlert = true;
    private StringBuffer verificationErrors = new StringBuffer();

    @BeforeClass(alwaysRun = true)
```

```

public void setUp() throws Exception {
    driver = new FirefoxDriver();
    baseUrl = "https://www.katalon.com/";
    driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
}

@Test
public void testUntitledTestCase() throws Exception {
    driver.get("https://cli.im/text");
    driver.findElement(By.id("text-content")).click();
    driver.findElement(By.id("text-content")).clear();
    driver.findElement(By.id("text-content")).sendKeys("hello world");
    driver.findElement(By.id("click-create")).click();
    driver.findElement(By.linkText("网址")).click();
    driver.findElement(By.id("url_content")).click();
    driver.findElement(By.id("url_content")).clear();

    driver.findElement(By.id("url_content")).sendKeys("https://www.baidu.com");
    driver.findElement(By.id("click-create")).click();
    driver.findElement(By.linkText("文件")).click();
    driver.findElement(By.id("filedatacode")).click();
    driver.findElement(By.id("filedatacode")).clear();
    driver.findElement(By.id("filedatacode")).sendKeys("C:\\fakepath\\03
Wireshark_DNS_v7.0.pdf");
    driver.findElement(By.id("click-create")).click();
    driver.findElement(By.linkText("图片")).click();
    driver.findElement(By.id("filedatacode")).click();
    driver.findElement(By.id("filedatacode")).clear();
    driver.findElement(By.id("filedatacode")).sendKeys("C:\\fakepath\\v2-
dcef71380baf9140dd01587f2bb396f7_r.jpg");
    driver.findElement(By.id("click-create")).click();
}

@AfterClass(alwaysRun = true)
public void tearDown() throws Exception {
    driver.quit();
    String verificationErrorString = verificationErrors.toString();
    if (!"".equals(verificationErrorString)) {
        fail(verificationErrorString);
    }
}

private boolean isElementPresent(By by) {
    try {
        driver.findElement(by);
        return true;
    } catch (NoSuchElementException e) {
        return false;
    }
}
}

```

```
private boolean isAlertPresent() {
    try {
        driver.switchTo().alert();
        return true;
    } catch (NoAlertPresentException e) {
        return false;
    }
}

private String closeAlertAndGetItsText() {
    try {
        Alert alert = driver.switchTo().alert();
        String alertText = alert.getText();
        if (acceptNextAlert) {
            alert.accept();
        } else {
            alert.dismiss();
        }
        return alertText;
    } finally {
        acceptNextAlert = true;
    }
}
```

## 心得体会

这次实验熟悉了火狐浏览器中黑盒测试工具 Selenium IDE 和 Katalon Recorder 的使用，一开始在使用 Selenium IDE 生成脚本时出现了问题，后来换了工具后成功解决了该问题，从中学习到要学会灵活采用多种方式来解决问题。这节课学到了新的软件测试工具使用方法，对此我感到兴奋。