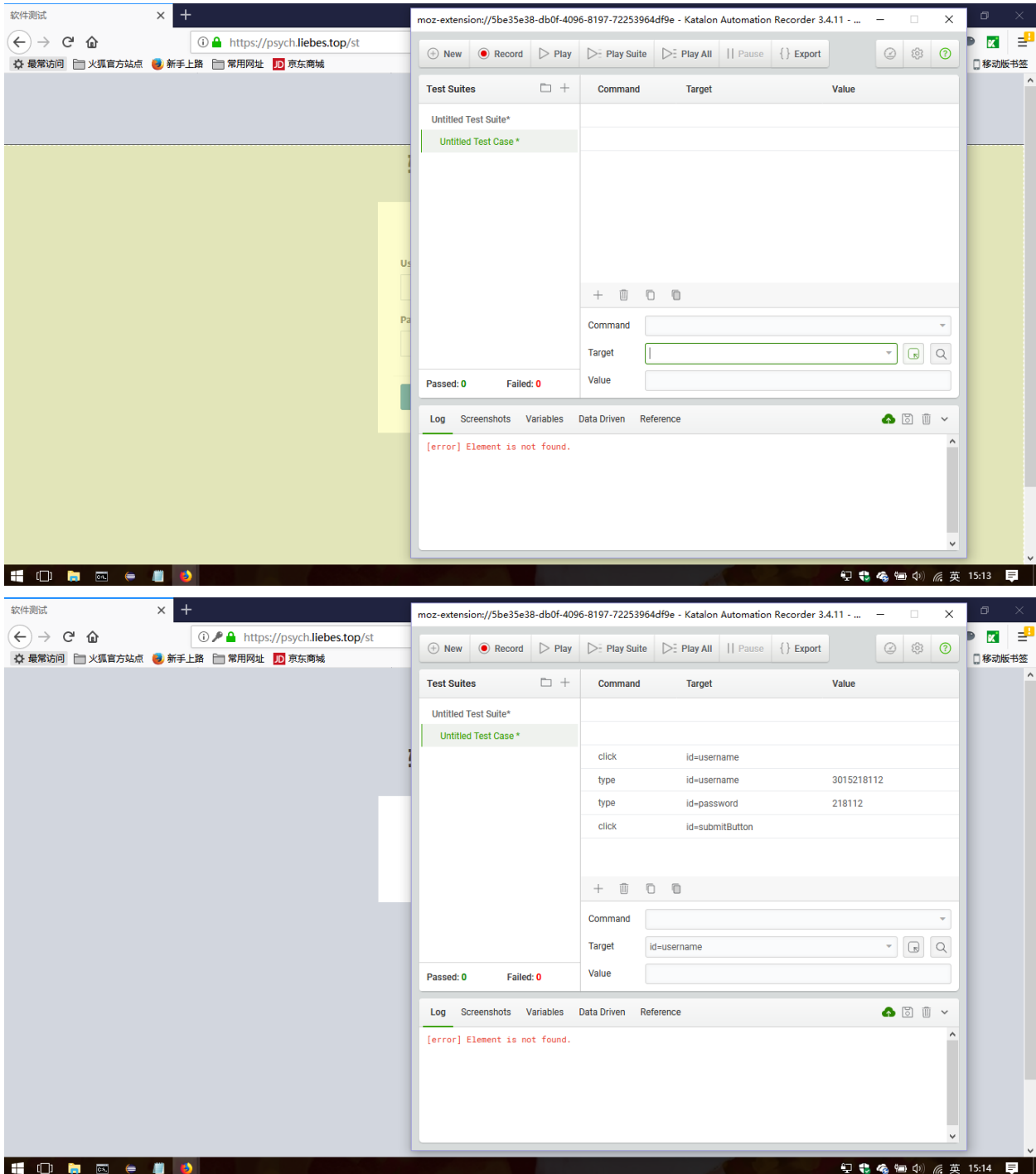


## 1、安装 Selenium IDE 插件

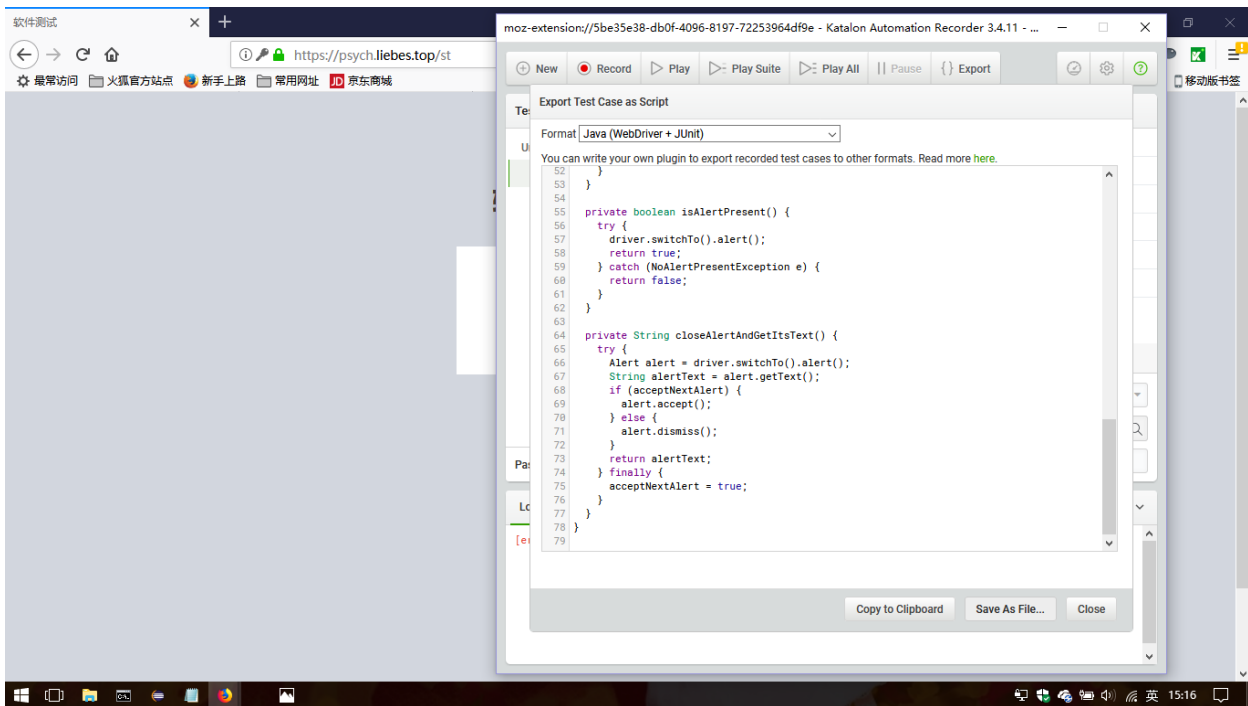
选择菜单 - 附加组件 - 插件，搜索 Selenium IDE 并安装

## 2、使用 Selenium IDE 录制脚本与导出脚本

录制脚本：点击录制，在页面进行需要录制的操作，然后点击停止录制



导出脚本：点击导出，选择目标语言，然后选择保存为文件



导出文件如下：

```
package com.example.tests;
```

```
import java.util.regex.Pattern;
import java.util.concurrent.TimeUnit;
import org.junit.*;
import static org.junit.Assert.*;
import static org.hamcrest.CoreMatchers.*;
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();
```

@Before

```
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 {
    // ERROR: Caught exception [unknown command []]
    // ERROR: Caught exception [unknown command []]
    driver.findElement(By.id("username")).click();
```

```

driver.findElement(By.id("username")).clear();
driver.findElement(By.id("username")).sendKeys(""); //username
driver.findElement(By.id("password")).clear();
driver.findElement(By.id("password")).sendKeys(""); //password
driver.findElement(By.id("submitButton")).click();
}

```

@After

```

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;
    }
}
}

```

### 3、编写 Selenium Java WebDriver 程序，测试用户信息是否对应正确

测试环境为 Windows 10 Pro（32-bit）与 Internet Explorer 11。测试策略为：判断删去空白符的输入地址是否与网站提供的地址一致。

使用如下代码进行测试：

```
import jxl.Sheet;
import jxl.Workbook;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.ie.InternetExplorerDriver;
import org.openqa.selenium.remote.DesiredCapabilities;

import org.junit.*;
import org.junit.runner.RunWith;
import org.junit.runners.Parameterized;
import org.junit.runners.Parameterized.*;

import java.io.File;
import java.util.Collection;
import java.util.List;
import java.util.ArrayList;

import static junit.framework.Assert.assertTrue;

class User {
    private String username;
    private String url;

    public User(String username, String url) {
        this.username=username;
        this.url=url;
    }

    public String getUsername() {return username;}
    public String getPassword() {return username.substring(4);}
    public String getUrl() {return url;}

    @Override
    public String toString() {
        return username +
            ' ' +
            url;
    }
}
```

```

@RunWith(Parameterized.class)
public class Test {
    private static List<User> information;
    private static WebDriver driver;

    private static boolean isValid(String obj) {
        if (obj==null) return true;
        if ("".equals(obj.trim())) return true;
        return false;
    }

    private static boolean check(String ori, String obj) {
        if (ori==null || obj==null) return false;

        String x=ori.trim();
        String y=obj.trim();

        if (x.equals(y)) return true;
        return false;
    }

    public static List<User> readFile(String path) {
        List<User> re=new ArrayList<>();
        try {
            Workbook workbook=Workbook.getWorkbook(new File(path));
            Sheet sheet=workbook.getSheet(0);

            for (int i=0; i<sheet.getRows(); ++i) {
                String id=sheet.getCell(0,i).getContents();
                String url=sheet.getCell(1,i).getContents();
                re.add(new User(id,url));
            }

            workbook.close();
        } catch (Exception e) {
            e.printStackTrace();
        }

        return re;
    }

    private String username;
    private String password;
    private String url;

    public Test(String username, String password, String expected) {
        this.username=username;
        this.password=password;
        this.url=expected;
    }
}

```

```

@BeforeClass
public static void init() {
    String driver_path = System.getProperty("user.dir")+"/IEDriverServer.exe";
    System.setProperty("webdriver.ie.driver", driver_path);

    DesiredCapabilities ieCapabilities = DesiredCapabilities.internetExplorer();

ieCapabilities.setCapability(InternetExplorerDriver.INTRODUCE_FLAKINESS_BY_IGNORING_SECURITY_DOMAINS, true);

    driver = new InternetExplorerDriver(ieCapabilities);
    //driver.manage().timeouts().implicitlyWait(15000, TimeUnit.MILLISECONDS);
}

@org.junit.Test
public void setUp() {
    driver.get("https://psych.liebes.top/st");

    WebElement username_element=driver.findElement(By.id("username"));
    WebElement password_element=driver.findElement(By.id("password"));
    WebElement submit=driver.findElement(By.id("submitButton"));

    assertTrue(!isInvalid(url));

    username_element.sendKeys(username);
    password_element.sendKeys(password);
    submit.click();

    try {
        Thread.sleep(1000);
    } catch (InterruptedException e) {}

    assertTrue(username,check(driver.findElement(By.xpath("//p[@class='login-box-msg']")).getText(),url));
}

@AfterClass
public static void cleanup() {
    driver.close();
}

@Parameters
public static Collection<Object[]> provide() {
    information=readFile(System.getProperty("user.dir")+"/input.xls");
    Collection<Object[]> re=new ArrayList<>();
    for (User x : information) {
        re.add(new Object[]{x.getUsername(),x.getPassword(),x.getUrl()});
    }
    return re;
}
}

```

测试结果如下：

The screenshot displays the Eclipse IDE interface with the following components:

- Package Explorer:** Shows the project structure with a tree view of test cases. A summary indicates "Runs: 97/97", "Errors: 0", and "Failures: 6". A specific test case, `setUp[18]`, is highlighted with a duration of 3.093 s.
- Testjava Editor:** Displays the source code for `Testjava`. The code includes Selenium WebDriver setup, element finding, and assertions. The current line of focus is `assertTrue(!isInvalid(url));` at line 106.
- Task List:** Located on the right, it shows a list of tasks, including `readFile(String)` and `init()`.
- Outline:** Also on the right, it provides a hierarchical view of the code structure, showing classes like `Test(String, String, String)` and methods like `init()` and `setUp()`.
- Console:** At the bottom, it shows the output of the test run, including the Java version (1.8.0\_91) and the Selenium server status (listening on port 10853).