

Szakdolgozat

1.Összefoglaló

A szakdolgozat célja egy olyan fordító alkalmazás elkészítése, amely a munkahelyemen ahol jelenleg dolgozom, a meglévő Selenium keretrendszerben íródott teszt állományokat képes olyan formára alakítani, hogy a már említett eszköz leváltására és modernizálására készült Katalon Recorder teszt eszközhöz kompatibilis, egy rögtön importálható és futtatható formátumú fájl, fájlokat hoz létre. A fordító későbbi fejlesztésben ezeket az állományokat további eszközökre is tudja alakítani így alkalmazható lesz a Sideex teszt eszközön és a Selenium IDE 3 eszközön is. Ezen felül az eszköz struktúrája miatt könnyen bővíthető lehet a jövőben egyéb eszközökre is. Későbbiekben kitérnék az említett teszt eszközök bemutatására és illetve arra , hogy miért van szükség a most használatban lévő Selenium IDE 2 kiváltására.

2.Bevezetés

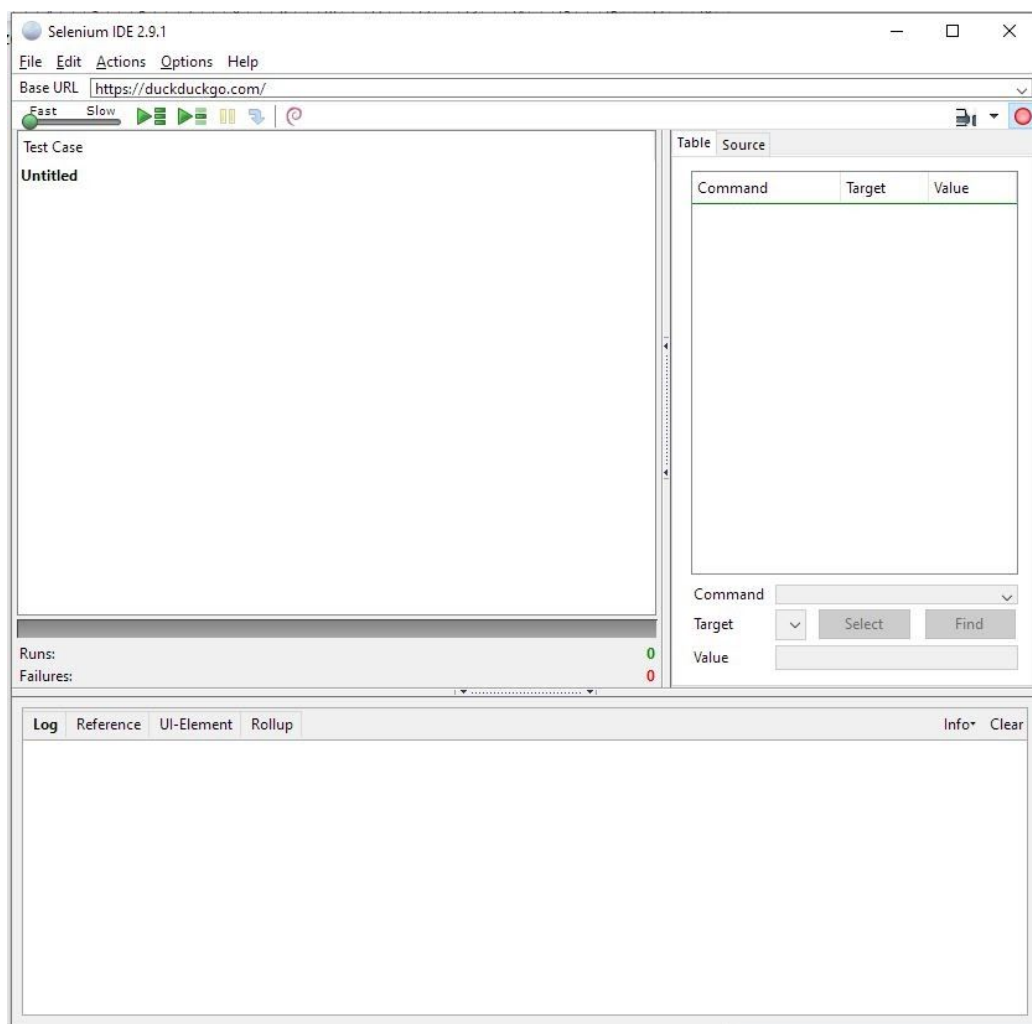
//

3.Selenium,Selenium IDE 2

A Selenium egy webes alkalmazások tesztelésére fejlesztett nyílt forráskódú keretrendszer.Ez a keretrendszer lehetőséget biztosít funkcionális és nem funkcionális tesztek készítésére anélkül a hogy el kellene sajátítani a teszt szkriptnyelvét. Továbbá biztosít egy tesztterület specifikus nyelvet (Selenese) amelynek köszönhetően számos népszerű programozási nyelvben lehet tesztek implementálni például C#,Groovy,Ruby,Java,PHP vagy Scala. A tesztek ezután a legtöbb modern böngészővel lehet futtatni és összehasonlítani az általuk produkált eredményeket.A Selenium nem rendszerfüggő ezért a nagyobb operációs rendszereken mint a Windows, macOS vagy a Linux is használható.

A Selenium IDE 2 egy integrált fejlesztési környezet(*integrated development environment*) amely Selenium tesztek egyszerű futtatását és létrehozását segíti elő. A környezet egy

kiegészítőként telepíthető a Firefox böngészőre .Célja, hogy megpróbálja reprodukálni a felhasználó által a böngészőben végrehajtott műveletek sorozatát. Ez azt jelenti hogy egyes gombokra úgy kattint ahogy a felhasználó tenné,szimulálni tudja a felhasználó által beírt adatokat mintha gépelne , ki tudja választani a legördülő listákban található megadott értékeket ,jelölőnégyzeteket tud kipipálni . A Selenium IDE 2 főbb tulajdonságai közé tartozik , hogy a használata nagyon könnyen elsajátítható ezért könnyen fejleszthetőek benne a különböző tesztesetek. Ez annak köszönhető , hogy található a felületen egy recorder gomb amely felveszi , a felhasználó által kattintott mezőket, ezzel is gyorsítva a tesztek létrehozását.



3.1 Selenium IDE 2 felülete

Ahogy a képen is látható a Selenium IDE 2 felületén találhatóak különböző funkciógombok ,sorrendben az első a sebesség csúszka amely a nevéből is adódóan a végrehajtandó parancsok sebességet tudja befolyásolni. Mellette található gomb arra hivatott , hogy a Test Case alatt lévő, általunk létrehozott vagy betöltött teszteseteket egymás után végrehajtsa. Ha

tovább haladunk akkor a következő indító gomb pedig csak éppen az aktív tesztet fogja végrehajtani. Mellette található a Pause gomb amellyel meg tudjuk állítani az éppen futtatott Test Case-ünket egy bizonyos állapotban. A jobb felső sarokban pedig a már említett record gomb található. Felépítését tekintve a Selenium IDE 2 és a későbbiekben bemutatásra kerülő eszközök is megjelenésük.

Egy parancs kiadására a <<Command>><<Target>> <<Value>> mezőket használjuk.

Körülbelül 100 és 120 között vannak az IDE -ben megtalálható parancsok, azért nem pontos mert a különböző verziókban más és más parancsok kerültek be illetve kerültek ki.

Néhány jellemző parancs amely szinte minden felületen előfordul:

click, type, waitForElementPresent, select, sendKeys, refresh, pause, store, open, assertElement, assertValue, altKeyDown, checkandWait, echo.

A Target részhez kerülnek azok a locatorok, útvonalak amelyek az adott oldalon egyértelműen azonosítják azt a mező amire a parancsot szeretnénk végrehajtani. Ilyen például:

“link=Home” , ”//select [contains(@id, 'Tesztmező')]" . Ebbe a mezőbe megadhatóak xpath útvonalak, css jellemzők illetve konkrét értékek is . Az eszköz képes javascript parancsok végrehajtására is a storeEval, getEval parancsokkal, ebben az esetben a Targethez kerülnek a javascriptes parancsok.

A Value mező pedig változóneveket tartalmazhat vagy konkrét értékeket amelyeket egy adott mezőbe szeretnénk kiválasztani , vagy beleírni.

Egyéb funkcióként a különböző parancsoknál elhelyezhetőek breakpointok a B billentyű lenyomására , illetve startpointok az S billentyű lenyomására. Első esetben a teszt futása addig a sorig fog tartani ameddig el nem ér a breakpointot tartalmazó sorig, a másik esetben pedig a testcase futása a startpointot tartalmazó sortól fog kezdődni.

Az eszköz egy hasznos tulajdonsága , hogy ha futtatunk egy tesztet és valahol hibába ütközünk, nem jó parancsot adunk meg , vagy a teszt nem találja az adott mezőt akkor megállíthatjuk ott a tesztet és javíthatjuk a parancsot, amit utána egyből végre is tudunk hajtani. Nem kell újrakezdeni az egész teszt futtatást ennek az egy parancsnak az ellenőrzéséhez. Ezen funkció hiánya néhány tesztautomatizáló eszközöknél hatalmas hiány. Hasznosnak mondható az is , hogy ha már létrehoztunk egy Test Case-t akkor többször is felhasználhatjuk a tesztünk során például , egy hosszabb folyamat esetén ha kétszer érintünk egy oldalt amelyhez tartozó Test Case-t már megírtuk akkor elég csak újra meghívni . Továbbá ha van egy olyan tesztünk amelyet több száz tesztben is használunk , például egy weboldal megnyitást és az az oldal amire a parancsokat kiadtuk megváltozik akkor nem kell minden tesztben átírni az oldal url-jét hanem elég csak egy helyen és az mindenhol változni fog.

Sajnos a Selenium IDE 2 futása közben nem nagyon tudunk mást tevékenységet folytatni mert nagyon érzékenyen reagál ha véletlenül az egér egy olyan mezőre kerül ahová és nem kellene és emiatt könnyen elbukhatnak az általunk futtatott tesztek. Hátrányként mondható az

is , hogy eléggé instabil így például a munkahelyem is használt rendszerek esetén sűrűn előfordul, hogy egymás után többször lefuttatva nem ugyanazt az eredményt kapjuk. Parancsok tekintetében is hiánynak mondható , hogy nincsenek benne beépített vezérlési szerkezetet így egy egyszerű feltétel vizsgálathoz egy külső plugin kell beimportálni az eszközbe és csak utána válnak elérhetővé az alap while és if funkciók. Párhuzamos futtatásra sincs lehetőségünk mivel egyszerre csak egy test Suite-ot tudunk beimportálni és futtatni és mellé nem tudjuk újra megnyitni az eszközt. A Selenium IDE 2 nem böngészőfüggetlen és verziófüggetlen ezért csak a Firefoxon működik és annak is csak az 55 ös verziójáig. Ennek az oka az lehet hogy viszonylag régi eszközről van szó mert 2006-ban lett elérhető és 2017-ben megszűnt a fejlesztése, feltehetően a támogatottság hiánya miatt. 2018-ban azonban egy közösségi indíttatás hatására újraindult a fejlesztése és Google Chrome -ra is elérhetővé vált az eszköz újabb verziója a Selenium IDE 3 .

4.Selenium IDE 3

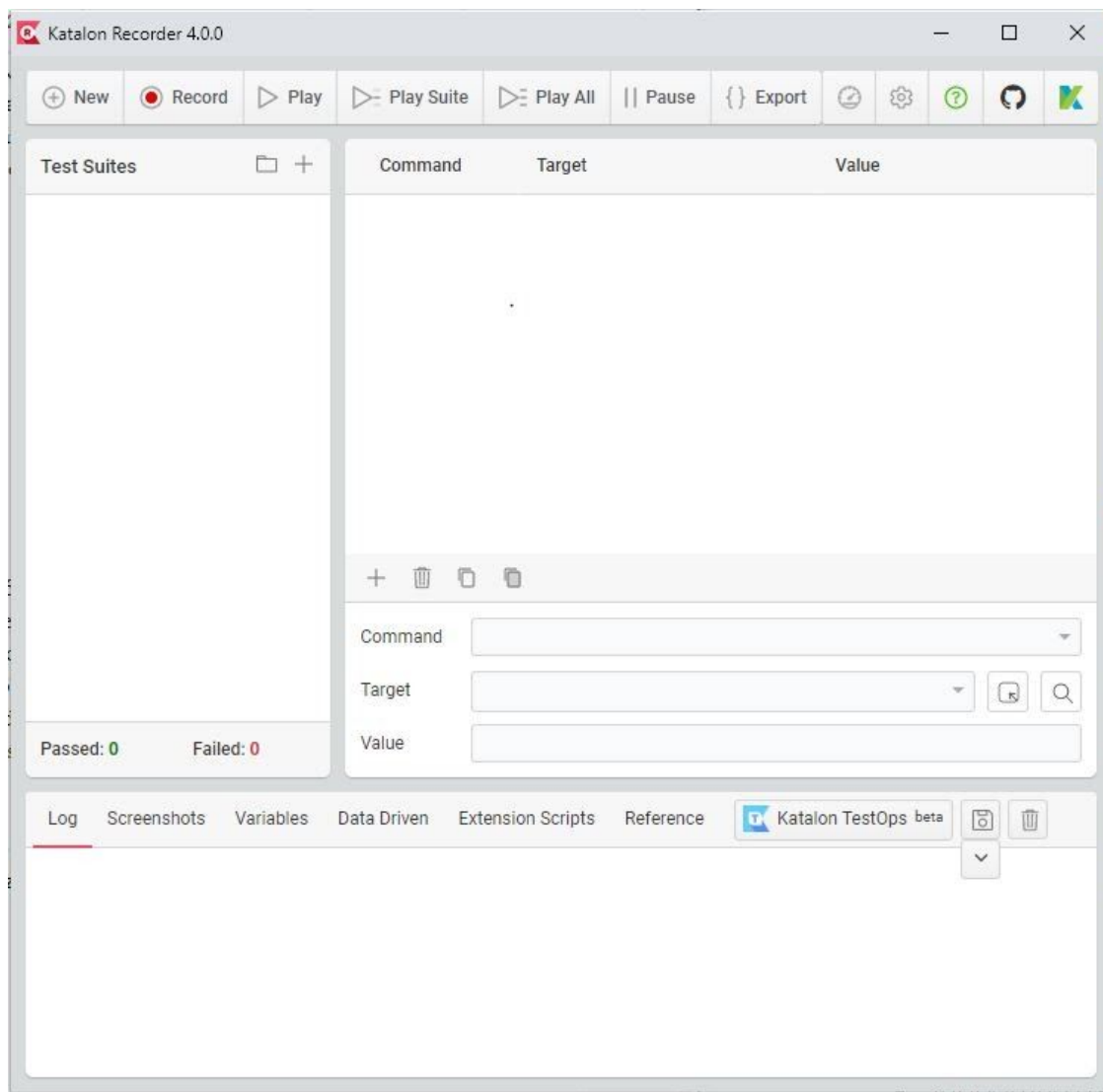
Kidolgozása később...

5.Katalon Recorder

A Katalon Recorder a Selenium IDE-hez hasonlóan egy böngésző kiegészítő amelyet a Katalon Studio csapat fejleszt és ugyan csak a felhasználói interakciók rögzítését hajtja végre a különböző webalkalmazásokban. A Katalon Recorder támogatja a nagyobb böngészők közül a Mozilla Firefox és a Google Chrome legfrissebb verzióját . A recorder hasonlóan működik mint a Katalon Studio rögzítő segédprogramja , de a felvétel mellett végre is tudja hajtani a teszt lépéseket amelyeket exportálni lehet számos nyelvre mint például C# ,JAVA vagy Python.

Ezen is eszközön is megtalálható a rögzítő gomb amely elősegíti a gyors teszt esetek gyártását és könnyen kezelhetővé teszi ezeknek az teszt eseteknek a lejátszását , szerkesztését. Felülete hasonlóan épül fel mint a már bemutatott eszköz mind a két verziója . A parancsok <<Command>><<Target>> <<Value>>formát követik és itt is rengeteg paranccsal találkozhatunk . Minden Selenese parancs megtalálható ezen az eszközhöz. Az itt létrejövő teszt eseteket a már bemutatott eszközön megtalálható nyelveken kívül Robot Framework és Katalon studio általi formátumra is exportálni tudjuk . Támogatja az CSV fájlok importálást ezért elősegíti az adatvezérelt tesztek írását és futtatását. Külső kiegészítőket is támogat így akár saját actionokat és egyedi locatorokat is kezelhetünk vele. További előnye , hogy a tesztek futása során nagyon szép és letisztult logokat láthatunk ezen felül ahol egy teszt elbukott ott az eszköz egy screenshotot is készít. Megemlítenéd , hogy az

eszköznek van egy beépített riport generáló funkciója amely a tesztelők számára tud különböző statisztikákat gyártani. Az instabilabb rendszereket is nagyon jól kezeli tehát a tesztet rögzítése után a visszajátszásnál nem fog egyes oldal betöltéseket vagy ajax hívásokat figyelmen kívül hagyni , emiatt egy megbízható eszköznek nevezhetjük. Beépített funkcióként található meg benne a sűrűn használt vezérlési szerkezetek például , if,while vagy for ciklus.Ezen eszközzel képesek vagyunk egy gépen egy böngészővel egyszerre több példányban is Katalon Recordert futtatni. Legnagyobb hátránya , hogy ugyan az eszközben lehet összerakni több teszt suitot is,de a suite-ok a teszteseteket külön tartalmazzák. Amennyiben két suite-ba is beillesztettük ugyanazon tesztesetet, és az egyik suiteban az egyik tesztesetet módosítjuk, más suiteban nem módosul annak tartalma, mivel külön fájlokra generálódnak a suite-ok.



5.1 Katalon Recorder felülete