

GESTIONE CASINÒ | Diario di lavoro - 15.03.2019

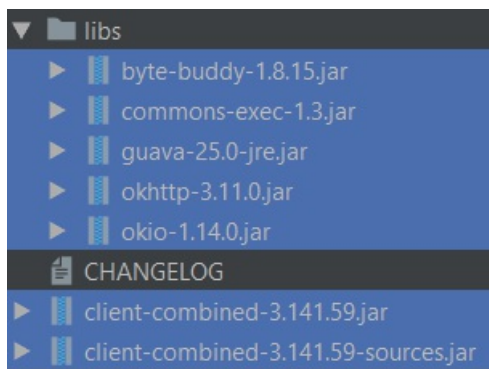
Matan Davidi, Thor Döblin, Matteo Forni, Carlo Pezzotti, Mattia Toscanelli

Trevano, 15 marzo 2019

Lavori svolti

Data l'assenza di Thor, oggi Matan si è occupato di prendere in mano il suo lavoro e portarlo avanti. Come prima cosa, quindi, si è informato velocemente su come funzionasse Selenium, poi ha creato un'applicazione di test usando la libreria di Selenium WebDriver. L'applicazione che ha creato stampa su terminale il titolo della pagina del [sito del casinò](http://www.cashyland.tk) (<http://www.cashyland.tk>). Per farlo si è aiutato con questo [tutorial per creare i test](https://www.saucelabs.com) (<https://www.saucelabs.com>) e con quest'altro [tutorial per avviare i test senza interfaccia grafica](https://www.built.io/blog/run-selenium-tests-in-headless-browser) (<https://www.built.io/blog/run-selenium-tests-in-headless-browser>).

Come prima cosa ha dovuto scaricare le librerie di Selenium WebDriver seguendo questo [link](https://www.seleniumhq.org/download/) (<https://www.seleniumhq.org/download/>), poi ha creato un progetto di prova per cominciare a fare qualche test con Selenium che ha chiamato *SeleniumTest*, all'interno del quale ha importato tutte le librerie scaricate:



In seguito ha scritto il seguente codice:

```
// System.setProperty("webdriver.chrome.driver", "ChromeDriverPath");
// ChromeOptions options = new ChromeOptions();
// options.addArguments("headless");
// options.addArguments("window-size=1200x600");
WebDriver driver = new ChromeDriver(/* options */);
driver.get("http://www.cashyland.tk");

WebElement title = driver.findElement(By.tagName("title"));
String titleText = title.getText();
System.out.println(titleText);

GroovyTestCase.assertEquals(titleText, "Neuron HTML CSS Template");

driver.quit();
```

Dove la parte commentata serve per eseguire il test senza interfaccia grafica. Questo snippet di codice apre la pagina della gestione del casinò, trova l'elemento "title", prende il testo e certifica che il testo è uguale a "Neuron HTML CSS Template".

Matteo ha risolto, probabilmente, il problema che non lo lasciava collegare Jenkins alla repo di produzione semplicemente salvando le credenziali nel sistema con il comando:

```
git config --global credential.helper store
```

e modificando lo script eliminando le parti inutili del comando push.

```
cp -rf ./GestioneCasino/. ./GestioneCasinoProduction/
cd ./GestioneCasinoProduction
git pull origin master
git add .
git commit -m "Jenkins commit"
git push
```

Il funzionamento dello script con Jenkins non è potuto però essere testato dato che esso non funziona come dovrebbe.

Quando è presente un errore in un campo, quest'ultimo diventa rosso. Però quando si va per inserire il campo corretto, quest'ultimo deve ritornare al colore di partenza. Anche qua era presente un errore e questo è stato corretto così:

```
inputs[0].style.backgroundColor = "white";
```

Problemi riscontrati e soluzioni adottate

Mattia Toscanelli ha speso molto del suo tempo a mettere a posto un errore causato da un collega, ma purtroppo non lo ha trovato dunque è tornato sulla versione vecchia ed ha aggiunto le

Matteo ha riscontrato problemi con Jenkins, esso non esegue più le compilazioni correttamente dato che non riesce ad eseguire il fetch sulla cartella di GitHub. Questo problema c'è da mercoledì 13 ma non era stato notato. Non è stata ancora trovata una soluzione.

Matan ha avuto il problema dove non viene trovata la classe usata per fare gli *assert* all'interno della sua applicazione, ossia `GroovyTestCase`, anche se essa è stata importata. Infatti viene mostrato il seguente errore:

```
Error:(22, 23) java: cannot access junit.framework.TestCase  
class file for junit.framework.TestCase not found
```

Punto della situazione rispetto alla pianificazione

Rispetto alla pianificazione siamo in orario.

Programma di massima per la prossima giornata di lavoro

Matteo ha intenzione di rimettere in funzione Jenkins e testare se lo script funziona con esso.