

## Web-აპლიკაციათა ავტომატური ტესტირების SELENIUM ტექნოლოგია

მიხეილ გულიტაშვილი  
საქართველოს ტექნიკური უნივერსიტეტი  
რეზიუმე

განიხილება Web-აპლიკაციების პროგრამული უზრუნველყოფის ავტომატური ტესტირების საკითხები Selenium ტექნოლოგიით, მისი გამოყენების სფეროები და შედარება ხელით ტესტირებასთან, უპირატესობები და ნაკლოვანებები. შემოთავაზებულია Selenium ტექნოლოგიის ზოგადი აღწერა, მისი შექმნის ისტორია და განვითარება. Selenium - ის ძირითადი შემადგენელი მატესტირებელი კომპონენტები და მათი მოკლე მიმოხილვა.

### 1. შესავალი

Web - ზე ბაზირებული პროგრამული აპლიკაციების გამოყენება წარმოადგენს კომპიუტერის მომხმარებლების ცხოვრების ნაწილს. ყოველი ჩვენგანი იყენებს ინტერნეტ ბრაუზერს რათა შევამოწმოთ ელექტრონული ფოსტა, წავიკითხოთ ახალი ამბები, დავრეგისტრირდეთ სოციალურ ქსელში, მოვიძიოთ ჩვენთვის სასურველი ინფორმაცია და ა. შ. ჩვენ ისე თავისუფლად ვიყენებთ სიტყვას "ვიდეო" როგორც ნებისმიერ ზმნას. გამოუხატავთ ჩვენ დამოკიდებულებას ბრაუზერების მიმართ და ვმსჯელობთ მათ უპირატესობებზე [3].

დღესდღეობით არსებული პროგრამული უზრუნველყოფის უდიდესი ნაწილი არის Web-ზე ბაზირებული პროგრამული პროდუქტი, რომელთაც მოვიხმართ ინტერნეტ ბრაუზერების საშუალებით. ეპოქაში, სადაც არის მაღალი საპასუხისმგებლო პროგრამული უზრუნველყოფის პროცესები, სადაც მრავალი ორგანიზაციები იყენებენ Agile მეთოდოლოგიას, ტესტირების ავტომატიზაცია ხდება დღითიდღე პოპულარული. პროგრამული პროდუქტის ეფექტურ ტესტირებას დიდ მნიშვნელობას ანიჭებენ კომპანიები და ორგანიზაციები. პროგრამული უზრუნველყოფის ტესტირების ავტომატიზაცია ნიშნავს გამოიყენო სხვა პროგრამული ინსტრუმენტი რომლის საშუალებითაც მოახდენ სატესტირებელი პროგრამის მრავალჯერად ტესტირებას. ტესტირების ავტომატიზაციას გააჩნია სხვადასხვა უპირატესობები, რომელთაგანაც აღსანიშნავია მისი განმეორებადობა და სიჩქარე, რომელიც საჭიროა ტესტის გასაშვებად [1].

ტესტირების ავტომატიზაცია არის პროგრამული პროდუქტის შექმნის შემადგენელი ეტაპი, რომელიც მოიცავს: განმეორებად ტესტირებას; სწრაფ უკუკავშირს პროგრამისტებთან; ტესტ-ქეისების შესრულებაზე გაშვებას შეუზღუდავად; Agile და XP მეთოდოლოგიების მხარდაჭერას; კარგად ორგანიზებულ ტესტ-ქეისების დოკუმენტაციას; შეცდომების აღწერას; ხელით ტესტირებისას გამორჩენილი შეცდომების პოვნას.

ცხადია თუ მოვახდენთ ჩვენი პროგრამული პროდუქტის ავტომატურ ტესტირებას, ეს იქნება გარანტი შეცდომების თავიდან აცილების, პროგრამის საიმედოობის, ტესტირების დროის დაზოგვის, პროცესის დაჩქარების. მაგრამ არის პროგრამების ავტომატური ტესტირება ყოველთვის ხელსაყრელი? ის არ არის ყოველთვის მიზანშეწონილი, მაგალითად წარმოვიდგინოთ პროგრამული პროდუქტი, რომლის სამომხმარებლო ინტერფეისი ხშირად იცვლება, ამ შემთხვევაში მატესტირებელი სკრიპტების ცვლილება გარდაუვალია. ასევე როცა დასატესტირებელი პროგრამა პატარა და მარტივია, ამ შემთხვევებში ხელით ტესტირება უფრო მისაღებია, ვიდრე მისი ავტომატიზაცია. თუ აპლიკაცია სრულდება მოკლე ვალებში, მაშინ ხელით ტესტირება საუკეთესო გადაწყვეტილებაა, რადგან ტესტირების ავტომატიზაციას სჭირდება მეტი დრო [1].

თანამედროვე ინფორმაციულ ტექნოლოგიებში გავრცელებულია უამრავი კომერციული და უფასო პროგრამული ინსტრუმენტები, რომელთა საშუალებითაც შესაძლებელია ავტომატური ტესტირების განხორციელება. Selenium შესაძლოა განვიხილოთ როგორც ყველაზე უფრო ფართოდ

გავრცელებული და გამოყენებადი უფასო ტესტირების ინსტრუმენტი. იგი შედგება სხვადასხვა პროგრამული პროდუქტისგან, თვითოეული მათგანი თავისი ფუნქციებით უზრუნველყოფს ტესტირების ავტომატიზაციას. მისი პროგრამული ინსტრუმენტების სიმრავლე საშუალებას გვაძლევს, რომ დავატესტიროთ Web-აპლიკაციები ჩვენი მოთხოვნების შესაბამისად [3].

## 2. ძირითადი ნაწილი

Selenium არის უფასო პროგრამული უზრუნველყოფა, განკუთვნილი Web აპლიკაციების ავტომატური ტესტირებისთვის. Selenium ტესტები შესაძლებელია შეიქმნას თანამედროვე პროგრამირების ენებზე როგორცაა: C#, Java, Groovy, Perl, PHP, Python და Ruby. იგი თავსებადია Windows, Linux, Macintosh ოპერაციულ სისტემებთან [1].

Selenium შეიქმნა 2004 წელს Jason Huggins - ის მიერ კომპანია ThoughtWorks - ში. მან გაანალიზა რომ უკეთესი იქნებოდა დაეხმოდა ხელით ტესტირებისთვის საჭირო დრო და განეხორციელებინა ერთიდაიგივე ტესტი ავტომატურად, ყოველი პროგრამული ცვლილებისას. მან შექმნა JavaScript ბიბლიოთეკა, რომლითაც შეეძლო ემართა Web - გვერდი. ამ ბიბლიოთეკას ეწოდა Selenium Core, რომელიც საფუძვლად დაედო Selenium Remote Control (RC) და Selenium IDE - ს. Selenium RC იყო ინოვაცია რადგან არცერთ სხვა პროდუქტს არ შეეძლო ეკონტროლებინა Web ბრაუზერი პროგრამული ენის საშუალებით [1].

სიტყვა Selenium შეიქმნა Huggins-ის ზემოხსენებულად, რომელიც მიწერა თავის კონკურენტ ვინმე Mercury-ს და უთხრა, რომ შენ შეგიძლია განიკურნო Selenium დანამატებით [2].

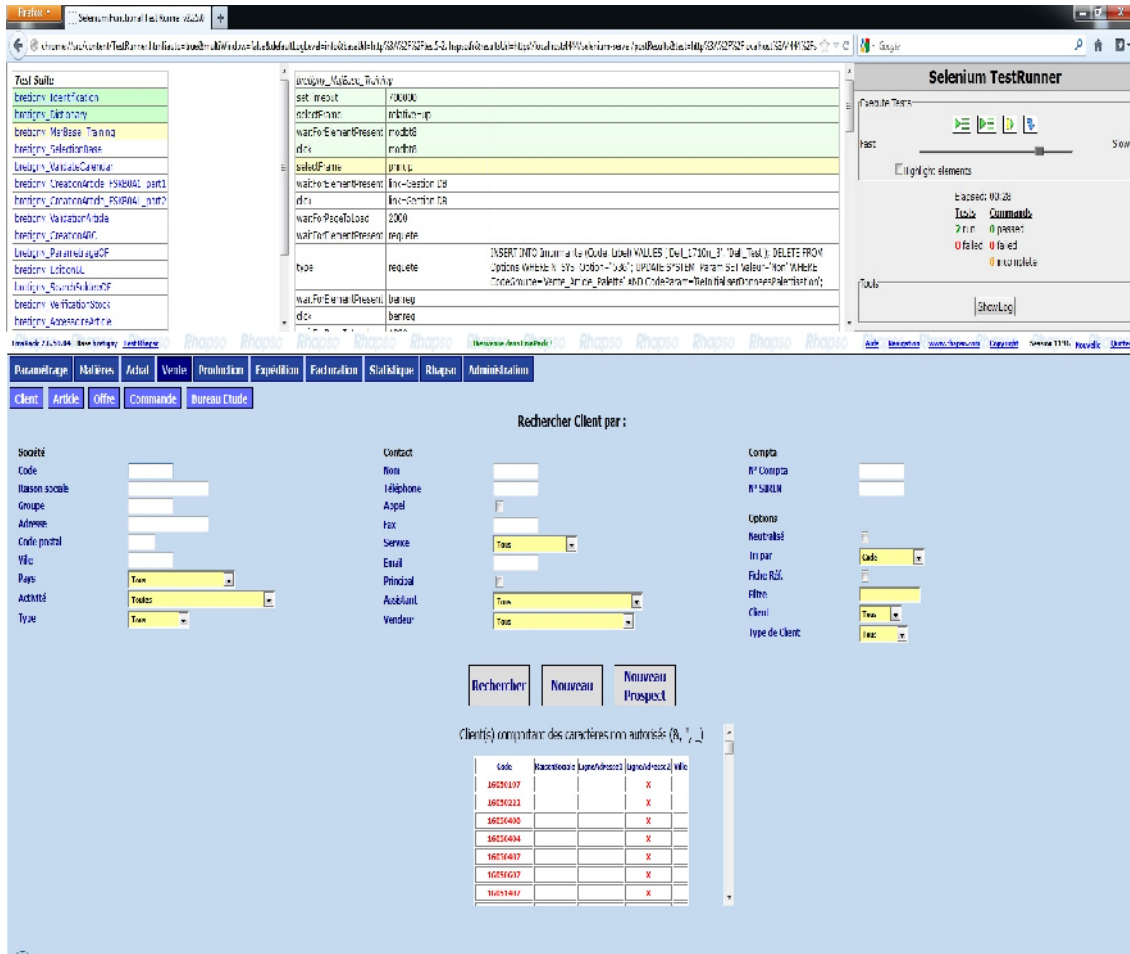
2006 წელს Google -ს ინჟინერმა, Simon Stewart დაიწყო მუშაობა პროექტზე რომელსაც მან უწოდა WebDriver. Google-ს ყავდა Selenium-ის გამოცდილი მომხმარებლები, მაგრამ ტესტირებს უწევდათ შეზღუდული ფუნქციებით მუშაობა, რაც განპირობებული იყო Javascript-ით. Simon-მა შექმნა მატესტირებელი ინსტრუმენტი, რომლითაც პირდაპირ ბრაუზერს მიმართავდა და იყენებდა ბრაუზერის მეთოდებს. ამით მან თავი აარიდა Javascript-ს. WebDriver პროექტის განხორციელების მიზანი იყო ის, რომ გადაეჭრა Selenium-ის „მტკიცეული“ საკითხები [1].

Selenium-ის ძირითადი კომპონენტებია: Selenium RC, Selenium IDE, Selenium Grid და Selenium WebDriver.

Selenium RC იყო მთავარი Selenium - ის პროექტი დიდი ხნის განმავლობაში, მანამ, სანამ შეიქმნებოდა ახალი და უფრო მძლავრი ინსტრუმენტი Selenium WebDriver. Selenium RC კვლავ არის აქტიურად განვითარების პროცესში, რომელიც თავსებადია მიმდინარე ვერსიის თანამედროვე ინტერნეტ ბრაუზერებთან და რომლსაც გააჩნია მხარდაჭერა შემდეგი ენების: Java, JavaScript, Ruby, PHP, Python, Perl და C#. Selenium RC საშუალებით ხდება Web აპლიკაციების სამომხმარებლო ინტერფეისების ავტომატური ტესტირება. ენას, რომელსაც იყენებს Selenium RC ავტომატური ტესტირებისთვის, ეწოდება Selense. 1-ულ ნახაზზე წარმოდგენილია Selenium RC. Selense ბრძანებები შეიძლება დაწერილ იქნას უფრო გაფართოებულ პროგრამირების ენაზე, ვიდრე ეს ნახაზზეა მოცემული - HTML ცხრილური ფორმატი.

Selenium IDE (Integrated Development Environment) არის Firefox პლაგინი, რომლითაც ხდება ტესტ-სკრიპტების, ტესტ-ქეისების შექმნა. მას გააჩნია მარტივი სამომხმარებლო ინტერფეისი, რომელიც უზრუნველყოფს სკრიპტების შექმნას. აქვს ჩაწერის ფუნქცია, რომელიც იწერს მომხმარებლის მოქმედებებს მოცემულ Web აპლიკაციაზე. ჩაწერილი სკრიპტების ექსპორტი შესაძლებელია დაპროგრამების სხვადასხვა ენაზე. ასევე მისი საშუალებით შესაძლებელია ჩაწერილი სკრიპტის შესრულებაზე გაშვება, მაგრამ მას არ გააჩნია პირობითი ოპერატორის ან ციკლის მხარდაჭერა [1].

მე-2 ნახაზზე გამოსახულია Selenium IDE. Selenium Grid აფართოებს Selenium RC - ის, რომელის საშუალებას იძლევა ტესტები გაუშვათ პარალელურად სხვადასხვა მანქანაზე.

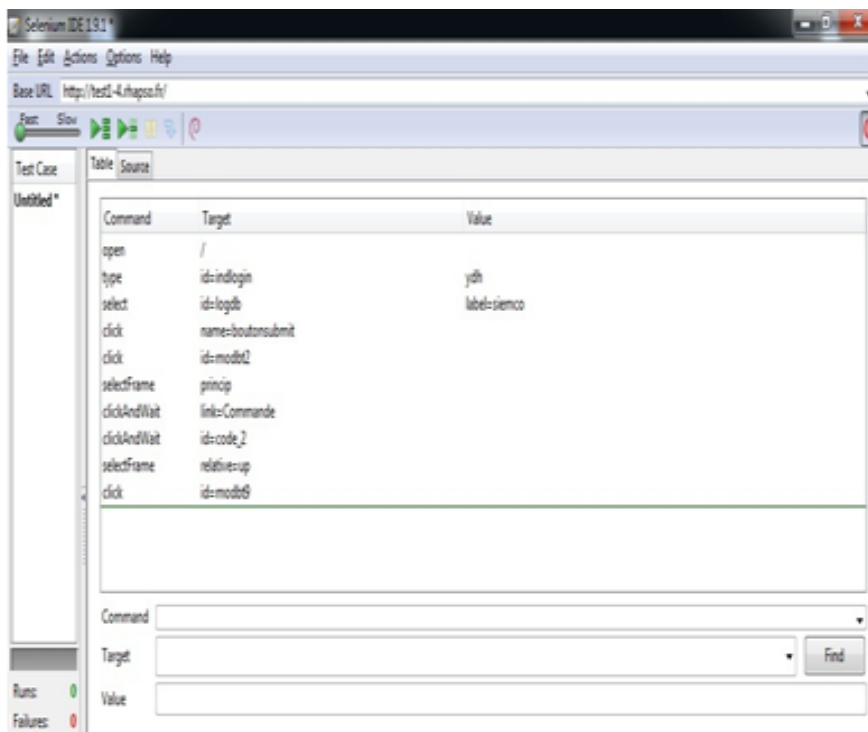


ნახ.1

მისი საშუალებით შესაძლებელია პარალელური ტესტირება, ასე რომ სხვადასხვა ტესტები შეიძლება გაკეთდეს ერთიდაიმავე დროს სხვადასხვა სერვერზე. მას გააჩნია ორი უპირატესობა. პირველი, თუ ჩვენ გვაქვს დიდი ტესტ-სუიტები, ან ნელა გასაშვები ტესტები, ჩვენ შეგვიძლია გავზარდოთ ტესტირების შესრულების სისწრაფე, სხვადასხვა მანქანების გამოყენებით. აგრეთვე თუ საჭიროა ერთიდაიმავე ტესტის გაშვება სხვადასხვა მოწყობილობაზე, ჩვენ შეგვიძლია გავუშვათ მოცემული ტესტი სხვადასხვა ოპერაციულ სისტემებზე პარალელურად. Selenium Grid აუმჯობესებს ტესტირებაზე დახარჯულ დროს პარალელური ტესტირების გამოყენებით [1].

### 3. დასკვნა

პროგრამის ყოველი ახალი ვერსიის შექმნისას შეგვიძლია გავუშვათ განმეორებადი ავტომატური ტესტები(ე. წ. Regression Testing) და დავადასტუროთ პროგრამის მუშაობის სისწორე, რადგან დავრწმუნდეთ რომ ვერსიის ცვლილებამ არ დააზიანა პროგრამის სხვა ფუნქციები. შეგნიშნოთ რომ ავტომატური ტესტირება იგივეა რაც ხელით ტესტირება, მათ შორის განსხვავებაა ის რომ ავტომატური ტესტების გაშვება ხორციელდება უფრო სწრაფად, შედეგები იწერება საშედეგო ფაილში, მისი შესრულებაზე გაშვება ხდება ადამიანის ჩარევის გარეშე, მისი გაშვება შეგვიძლია მრავალჯერადად. ყოველივე ეს იწვევს შეცდომების თავიდან აცილებას, ხარისხის ამაღლებას, დროის დაზოგვას და ხარჯების შემცირებას. განხილული ტექნოლოგია საშუალებას იძლევა სკრიპტების ავტომატურ რეჟიმში ჩაწერას, შესრულებაზე გაშვებას და პარალელური ტესტირების შესაძლებლობას. იგი არის პლატფორმისგან დამოუკიდებელი პროგრამული უზრუნველყოფა, რომლის ინსტალაცია შესაძლებელია Windows, Linux, Macintosh ოპერაციულ სისტემებზე.



ნახ.2

**ლიტერატურა:**

1. Hunt D., Newhook P., Kumar T., Selenium Documentation from Selenium Project, 2012
2. Wikipedia, [http://en.wikipedia.org/wiki/Selenium\\_\(software\)](http://en.wikipedia.org/wiki/Selenium_(software))
3. Sing Li., Article - Automate web application UI testing with Selenium, 2010. <http://www.developerfusion.com/article/84484/light-up-your-development-with-selenium-tests/>

**WEB APPLICATION AUTOMATIC TESTING SELENIUM TECHNOLOGY**

Gulitashvili Mikheil  
Georgian Technical University  
**Summary**

In the represented article considered software environment, Web applications automatic testing with Selenium technology. Automatic Testing usage aspects, its advantages by comparing with manual testing. General description of Selenium technology, history and development. Selenium testing components review and short description.

**ТЕХНОЛОГИЯ SELENIUM ДЛЯ АВТОМАТИЧЕСКОГО ТЕСТИРОВАНИЯ  
WEB-АПЛИКАЦИЙ**

Гулиташвили М.  
Грузинский Технический Университет

**Резюме**

Рассматриваются вопросы автоматического тестирования программного обеспечения Веб-аппликаций с использованием Selenium-технологии. Обсуждаются аспекты использования автоматического тестирования, ее преимущества и недостатки по сравнению с ручным тестированием. Дается общее описание Selenium-технологии, история ее создания и развития, функциональность компонентов Selenium-тестирования.