

Testowanie aplikacji internetowych narzędziami Selenium w chmurze – część 1

Autor: Łukasz Złocki

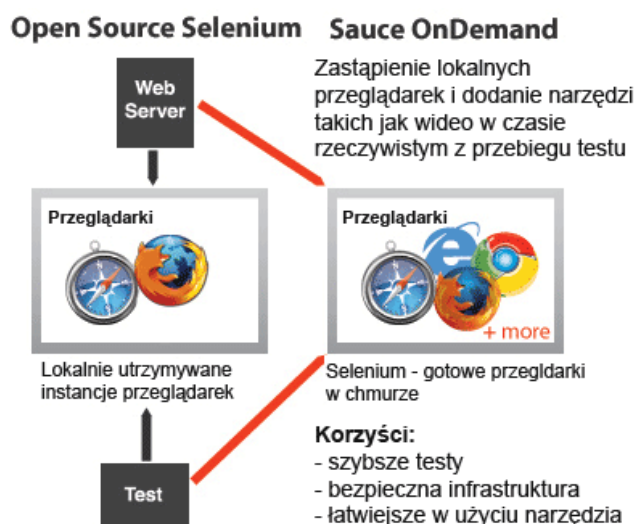
Selenium to popularny zestaw narzędzi open-source mających na celu ułatwienie procesu automatyzacji testowania aplikacji internetowych. Każde z tych narzędzi charakteryzuje się pewnymi właściwościami, dzięki którym badanie funkcjonalności aplikacji staje się łatwiejsze, szybsze i prostsze. Poza niezaprzeczalnymi zaletami, każde z nich posiada także pewne ograniczenia. Selenium IDE jest z tej rodziny narzędzi najbardziej ograniczone pod względem obsługi środowisk i przeglądarek, ale z drugiej strony budowanie skryptów testowych za jego pomocą jest niezmiernie proste i szybkie. Wykorzystując tę zaletę postanowiłem oprzeć dalszą część niniejszego artykułu na tym właśnie narzędziu, tym bardziej, iż jest to narzędzie najczęściej wybierane przez początkujących testerów aplikacji internetowych, automatyzujących skrypty testowe.

Podczas testowania aplikacji opartych na sieci web bardzo istotną rolę odgrywa środowisko na jakim prowadzone są testy. Zarówno deweloperzy, jak i testerzy doskonale zdają sobie sprawę, że dana aplikacja nie tylko może wyglądać inaczej w innej przeglądarce, ale również może działać inaczej lub nie działać wcale. Odpowiednio przygotowane środowisko testowe dla aplikacji internetowych powinno zapewniać obsługę wszystkich najpopularniejszych przeglądarek wraz z ich odpowiednikami dla różnych systemów operacyjnych.

W praktyce realizacja takich środowisk jest bardzo trudna i w dużym stopniu ograniczona nie tylko funduszami, ale także sprzętem, nie wspominając już o małych firmach, gdzie aplikacje często rozwijane są na mało nowoczesnych maszynach wykorzystując oprogramowanie open-source. Zorganizowanie tam takiego środowiska testowego graniczy z cudem, dlatego testy kompatybilności ograniczają się do przeglądarek zależnych od jednego systemu operacyjnego. A przecież Selenium wspiera wszystkie najpopularniejsze (te mniej popularne również, jednak nie bezpośrednio) przeglądarki dla różnych systemów operacyjnych! Jak tę cechę wykorzystać nie mając możliwości zbudowania odpowiedniego zaplecza dla naszych testów?

Testowanie w chmurze – zalety

Z pomocą przychodzi nam oferta serwisu [SauceLabs](https://saucelabs.com). SauceLabs to oprogramowanie i usługi oparte na projektach Selenium, które przyspieszają i ułatwiają proces testowania. SauceLabs to również lider w światowej społeczności Selenium. Oprogramowanie Selenium rozszerzone o usługi SauceLabs staje się bardziej produktywnie, ułatwiając prowadzenie testów funkcjonalnych we wszystkich popularnych przeglądarkach internetowych – Internet Explorer, Firefox, Safari, Chrome i Opera w systemach operacyjnych Windows i Linux!



Flagowym produktem SauceLabs jest usługa, która pozwala użytkownikom Selenium wykonać ich skrypty testowe równolegle w 10 najpopularniejszych obecnie przeglądarkach. Dzięki temu eliminowane zostają koszty związane z tworzeniem i utrzymywaniem środowisk testowych.

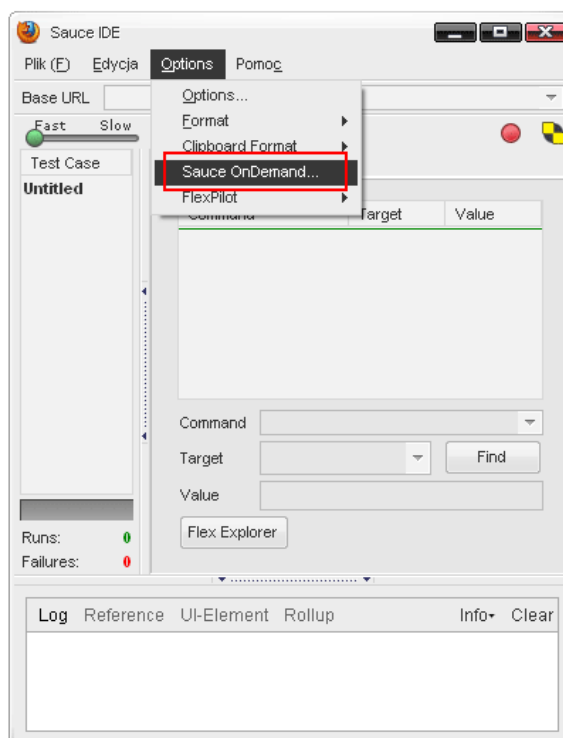
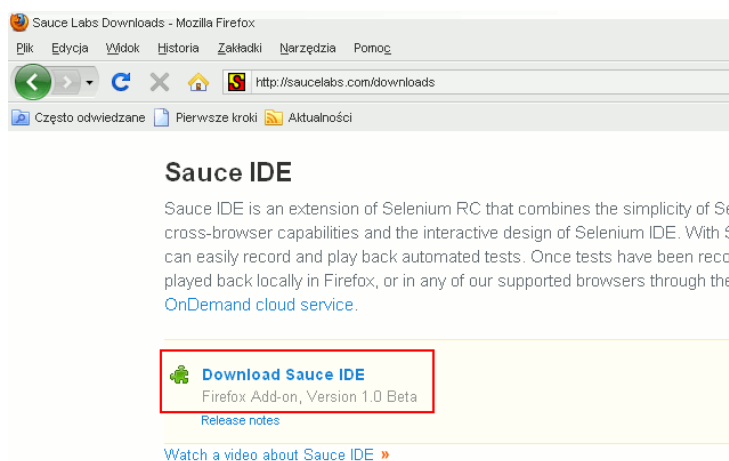
Testy mogą być uruchamiane w chmurze z takich narzędzi jak Sauce IDE (rozszerzenie Selenium IDE), Sauce RC (aplikacja kliencka do zarządzania serwerem Selenium i uruchamiania suit testowych) lub bezpośrednio z Selenium RC. Interfejs API integruje się z takimi frameworkami testowymi jak TestNG, JUnit czy PHPUnit. Można również podłączyć chmurę do istniejącej infrastruktury Selenium Grid lub wykorzystać narzędzie do zarządzania testami - Bromine.

Nieocenioną zaletą tej usługi jest możliwość nagrania filmu wideo z prowadzonego testu i wygenerowanie czytelny raportu w postaci HTML wraz ze zrzutami ekranów. Dodatkowo raz w tygodniu otrzymamy czytelny raport na skrzynkę email z przeprowadzonych przez nas testów.

Sauce IDE – Selenium IDE i obsługa innych przeglądarek

Przejdźmy zatem do praktycznego zastosowania dostępnych środowisk testowych w ofercie SauceLabs. Na początek musimy stworzyć sobie, darmowe oczywiście, konto na stronie projektu <http://saucelabs.com/pricing>. Jedyne ograniczenie darmowego konta to czasowe ograniczenie dla wykonywanych testów. W ciągu miesiąca łączny ich czas nie może przekroczyć 200 minut. Po rejestracji otrzymamy unikalny klucz (**API Key**), który będziemy wykorzystywać podczas połączeń do zdalnej infrastruktury.

Teraz pobieramy narzędzie [Sauce IDE](#) i instalujemy jako dodatek do przeglądarki Firefox. Wcześniej zainstalowane Selenium IDE zostanie rozszerzone o nowe możliwości, przy czym wcześniejsze rozszerzenia będą wciąż lokalnie aktywne. Jeżeli będziemy chcieli korzystać z tych rozszerzeń podczas uruchamiania w chmurze, musimy je upublicznić i mieć do nich dostęp poprzez adres URL. Nowa funkcjonalność to przede wszystkim obsługa zdalnych przeglądarek pracujących w chmurze oraz możliwość uruchamiania skryptu testowego dla wybranej przeglądarki przy wykorzystaniu zdefiniowanego serwera Selenium RC.

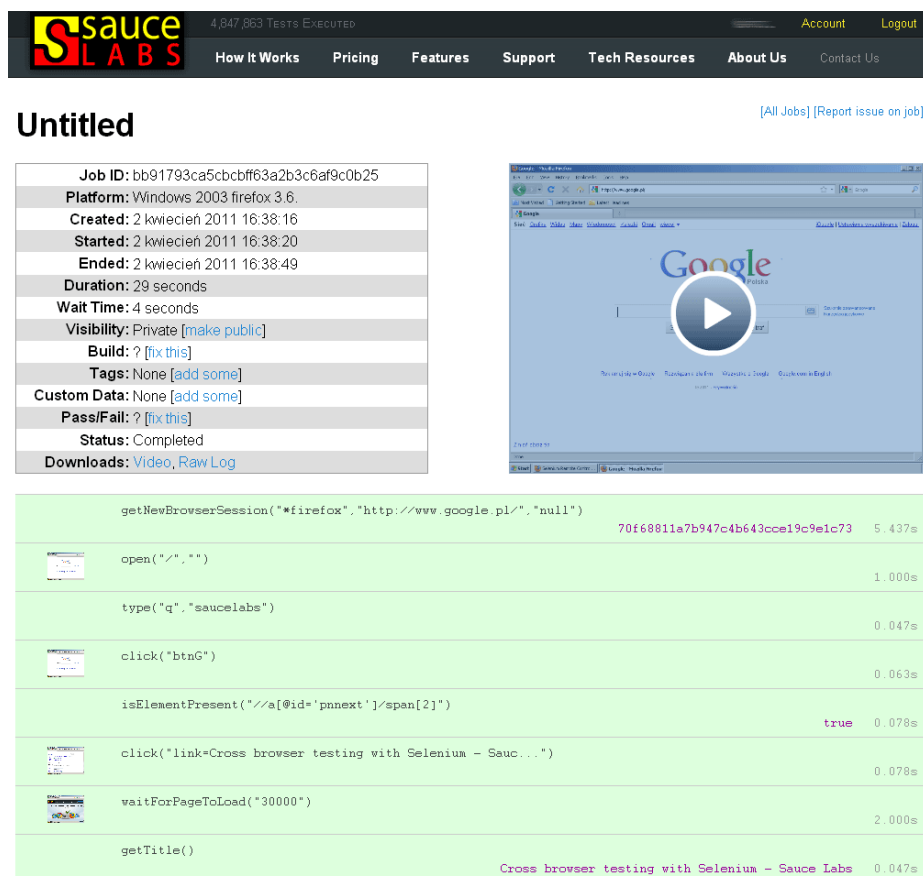


Nagrywanie skryptu testowego nie różni się niczym od sposobu w jaki robiliśmy to do tej pory, czyli wciskamy czerwony przycisk do nagrywania i symulujemy działania użytkownika w testowanej aplikacji. Po zakończeniu rejestrowania naszych czynności mamy możliwość ponownego ich odtworzenia.

Do tej pory mogliśmy uruchamiać wcześniej nagrane za pomocą Selenium IDE skrypty testowe jedynie w przeglądarce Firefox i to w wersji, na której dodatek jest obecnie zainstalowany. Teraz możemy wybrać spośród dwóch opcji: uruchamiania na zdefiniowanym serwerze Selenium RC lub uruchamiania w chmurze na dowolnej oferowanej przeglądarce. Wyboru dokonujemy w menu „Options/Sauce OnDemand...”.

Jeśli wybierzemy uruchomienie na serwerze Selenium RC, musimy określić host, na jakim serwer jest uruchomiony, port, na którym nasłuchuje oraz przeglądarkę na jakiej chcemy uruchomić nasz skrypt testowy. Natomiast wybór uruchomienia testu w chmurze wymaga od nas podania nazwy użytkownika (podawaliśmy ją podczas rejestracji), klucza dostępu służącego do identyfikacji (API Key) oraz oczywiście wybrania przeglądarki. Lista obsługiwanych przeglądarek jest spora i na bieżąco aktualizowana. Dodatkowo możemy jeszcze określić, czy chcemy nagrać film z przebiegu testu oraz określić maksymalny czas potrzebny na wykonanie testu – jeśli test będzie się wykonywał dłużej niż podany limit, wówczas zostanie przerwany. Na koniec możemy określić z jakich rozszerzeń użytkownika będziemy korzystać i podać do nich adres URL. Jeżeli korzystamy z wielu rozszerzeń, wówczas podajemy je w tablicy JSON np. ["url/to/ext1.js","url/to/ext2.js"]. Uruchamianie nagranych skryptów w chmurze odbywa się poprzez kliknięcie nowej ikonki znajdującej się obok przycisku do nagrywania.

Jeżeli jesteśmy zalogowani na stronie serwisu SauceLabs, wówczas możemy na bieżąco obserwować postęp uruchomionego testu (po rozpoczęciu odtwarzania skryptu testowego zostaniemy automatycznie przekierowani na stronę SauceLabs). W przypadku zakończenia skryptu testowego lub wystąpienia błędu otrzymujemy końcowy raport w postaci poleceń Selenium wraz ze zrzutami ekranów oraz (o ile zaznaczyliśmy odpowiedni checkbox) możemy obejrzeć film z przebiegu testu.



The screenshot displays the Sauce Labs web interface. At the top, there is a navigation bar with the Sauce Labs logo and links for 'How It Works', 'Pricing', 'Features', 'Support', 'Tech Resources', 'About Us', and 'Contact Us'. The main content area is titled 'Untitled' and includes a job summary table and a Selenium command log.

Property	Value
Job ID	bb91793ca5cbbff63a2b3c6af9c0b25
Platform	Windows 2003 firefox 3.6
Created	2 kwiecień 2011 16:38:16
Started	2 kwiecień 2011 16:38:20
Ended	2 kwiecień 2011 16:38:49
Duration	29 seconds
Wait Time	4 seconds
Visibility	Private [make public]
Build	? [fix this]
Tags	None [add some]
Custom Data	None [add some]
Pass/Fail	? [fix this]
Status	Completed
Downloads	Video, Raw Log

Command	Result	Duration
getNewBrowserSession("firefox", "http://www.google.pl/", "null")	70f68811a7b947c4b643cce19c9e1c73	5.437s
open("/", "")		1.000s
type("q", "sauce labs")		0.047s
click("btnG")		0.063s
isElementPresent("//a[@id='pnnext']/span[2]")	true	0.078s
click("link=Cross browser testing with Selenium - Sauc...")		0.078s
waitForPageToLoad("30000")		2.000s
getTitle()	Cross browser testing with Selenium - Sauce Labs	0.047s

Aplikacje w prywatnej sieci a chmura

Wszystko działa ładnie i pięknie, jest zewnętrzne środowisko na którym uruchamiane są testy aplikacji internetowych dostępnych poprzez podanie URLa z adresem, ale co z aplikacjami uruchamianymi lokalnie lub w wewnętrznej, prywatnej sieci?

Odpowiedzią jest aplikacja [Sauce Connect](#). Tworzy ona bezpośrednie połączenie SSH z chmury do określonego serwera w sieci prywatnej obsługującego naszą aplikację. Po ustanowieniu połączenia nasza aplikacja jest dostępna z internetu dla infrastruktury chmury poprzez sfalszowaną domenę widoczną w sieci z zewnątrz. Nie trzeba otwierać żadnych dodatkowych portów, a po zamknięciu połączenia prywatny serwer staje się niedostępny dla chmury.

Po pobraniu paczki z aplikacją Sauce Connect i po jej rozpakowaniu jedyne co musimy zrobić, aby uruchomić tunelowanie do naszej testowanej aplikacji, to podać wymagane parametry:

- ⤴ „wewnętrzny” serwer wraz z portem (port nie jest wymagany), na którym pracuje aplikacja, którą chcemy przetestować;
- ⤴ domena lub domeny, za pomocą których chcemy uzyskać dostęp do testowanej aplikacji z chmury;
- ⤴ nazwa użytkownika SauceLabs i klucz dostępu (API Key) – czyli dane wymagane do zalogowania się do serwisu SauceLabs.

Sauce Connect

Sauce Connect creates a direct line from our cloud to a specific test server on your private network. Test your applications securely behind a firewall with Sauce Connect. This feature connects a machine in your internal infrastructure to Sauce OnDemand via SSH. Only the Sauce Labs machines actively running your tests will have access and when a test is finished, access is revoked. You don't have to open any ports on your firewall, and once you shut Sauce Connect down your private server becomes invisible to our cloud again.

Note: we are currently conducting private beta testing of Sauce Connect 2, which for many use cases will dramatically increase test speed. If that's of interest to you, please [contact us](#).



Znając już wymagane parametry przechodzimy do katalogu, do którego wypakowaliśmy dane z pobranej paczki i uruchamiamy tunelowanie poleceniem:

```
sauce_connect -u <user> -k <api key> -s <webserver> -d <domain> [optionals]
```

- ⤴ „user” to nazwa użytkownika, którą podajemy w trakcie logowania do serwisu SauceLabs;
 - ⤴ „api key” to klucz dostępu, który jest wyświetlany na stronie z naszymi danymi;
 - ⤴ „webserver” to nazwa serwera, na którym pracuje testowana aplikacja (np. localhost, myserver.test, 192.168.7.45 itp.);
 - ⤴ „domain” jest nazwą domeny, pod którą dostępna jest testowana aplikacja i dla której są przygotowane skrypty testowe. Pod tą domeną testowana aplikacja będzie widoczna z zewnątrz. Można określić wiele domen, które chcemy obsłużyć (np. example.com, www.example.com i static.example.com)
- ```
sauce_connect -u <user> -k <api key> -s <webserver> -d example.com -d '*.example.com'
```

Więcej informacji o parametrach można otrzymać poprzez uruchomienie aplikacji z parametrem -h

```
sauce_connect -h
```

```
C:\WINDOWS\system32\cmd.exe - sauce_connect.exe -u Lukasz -k 899bd9f5-8d7e-4b95-939d-66b65f25dd8e -s devsoft.test -d *.devsoft.test
Microsoft Windows XP [Wersja 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\SauceConnect\windows>saucе_connect.exe -u _____ -k _____ -s _____ -d _____
* Debug messages will be sent to sauce_connect.log
:
: Have questions or need help with Sauce Connect?
: Contact us: http://saucelabs.com/forums
:
2011-04-01 14:45:16,463 - / Starting \
2011-04-01 14:45:16,463 - Please wait for "You may start your tests" to start your tests.
2011-04-01 14:45:16,463 - sauce_connect release 25
2011-04-01 14:45:20,259 - Forwarding: ['*'] -> ['80']
2011-04-01 14:45:20,275 - Successfully connected to _____:80 in 15ms
2011-04-01 14:45:22,790 - Tunnel host is provisioned (3e46195003709a9fd77a0c641e806536)
2011-04-01 14:45:23,884 - Tunnel host is new ..
2011-04-01 14:45:47,931 - Tunnel host is deploying ..
2011-04-01 14:45:55,713 - Tunnel host is booting ..
2011-04-01 14:46:21,838 - Tunnel host is running at maki8007.miso.saucelabs.com
2011-04-01 14:46:21,838 - Starting SSH process ..
2011-04-01 14:46:25,947 - Successfully connected to _____:80 in 15ms
2011-04-01 14:46:26,165 - Successfully connected to maki8007.miso.saucelabs.com:22 in 218ms
2011-04-01 14:46:26,165 - SSH is running. You may start your tests.
```

Kilka przykładów jak uruchomić tunelowanie do aplikacji w prywatnej sieci:

- testowanie wszystkich aplikacji w domenie example.com i aplikacji example.pl na lokalnym serwerze (127.0.0.1) obsługując protokół HTTPS

```
saucе_connect -u <user> -k <api key> -s 127.0.0.1 -p 80 -p 443 -d example.pl -d *.example.com
```

- testowanie aplikacji na serwerze hostname.com.pl dostępnej pod domeną example.com.pl

```
saucе_connect -u <user> -k <api key> -s hostname.com.pl -d example.com.pl
```

- testowanie wszystkich aplikacji w domenie myapp.test dostępnych na serwerze server.test

```
saucе_connect -u <user> -k <api key> -s server.test -d *.myapp.test
```

Będąc zalogowanym do swojego konta w serwisie SauceLabs mamy dostęp do widoku aktywnych tuneli, gdzie możemy je wyłączyć. Wyłączyć je możemy również poprzez zwykłe przerwanie procesu np. Ctrl+C w systemie Windows.

### Podsumowanie

Zdalne środowiska testowe jakie zapewnia infrastruktura SauceLabs to niewątpliwie ogromna zaleta, z której powinny skorzystać wszystkie osoby chcące sprawdzić działanie swoich aplikacji internetowych na różnych środowiskach i przeglądarkach. Nie tylko testerzy czy osoby odpowiedzialne za kontrolę jakości oprogramowania, ale również deweloperzy. Zapewnienie tak dużej liczby środowisk i przeglądarek w większości przypadków jest bardzo trudne do zrealizowania, szczególnie w małych firmach, i przy tym kosztowne. Dzięki chmurze SauceLabs testowanie aplikacji internetowych stało się prostsze, szybsze i przyjemniejsze. To co do tej pory spędzało sen z powiek osobom odpowiedzialnym za jakość oprogramowania internetowego, czyli między innymi stworzenie odpowiednich środowisk testowych i testowanie funkcjonalności dla różnych przeglądarek, ograniczone zostało jedynie do założenia konta i zainstalowania kilku dodatków do i tak już używanych narzędzi.

W kolejnych artykułach postaram się Państwu opisać jak, korzystając z innych narzędzi Selenium, wykorzystać środowiska testowe dostępne w chmurze oraz przedstawię inne, ciekawe aplikacje SauceLabs usprawniające pracę z tymi narzędziami.

**testerzy.pl**  
Sokolska 33/173  
40-086 Katowice  
+48 32 733 09 58  
mailto:kontakt@testerzy.pl