# LearningTestAutomation: Class1

## Setup

Stvara se instanca google chrome drivera *driver* defaultnim konstruktorom *ChromeDriver()* čime se pokreće Internet browser. Nakon toga se na konzoli ispisuje opisna poruka o završetku setupa – "Setup done".

## Test

Instanci *driver* atribut *url* postavlja se na youtubeovu web adresu i otvoreni browser ju učitava. Stvoren je *WebDriverWait* objekt *wait* s vremenom čekanja od pet sekundi kako bi izvođenje programa explicitno sačekalo pokušaj pronalaska button s tekstom "Slažem se" kojim će prihvatiti kolačiće i uvjete pružanja usluge. Na konzolu se ispisuje opisna poruka o pritisku buttona.

Element search bara pronađen je pomoću imena (*search\_query*) i spremljen u *IWebElement* objekt *searchQuery*. Na konzolu se ispisuje opisna poruka o završetku traženja elementa.

Element search button pronađen je pomoću id-a i spremljen je u objekt *searchButton*. Na konzolu se ispisuje opisna poruka o završetku traženja elementa.

U nedostatku znanja umjesto implicitnog čekanja iskorištena je sistemska metoda *Sleep()* da program sačeka jednu sekundu kako bi mogao uspješno unijeti tekst u search bar. (Bez navedenog čekanja program vraća "element not interactable" exception pozivom *SendKeys()* metode nad *searchQueary* objektom). Program se nanovo zaustavlja tri sekunde kako bi se tekst mogao jasno iščitati radi pokušaja postizanja komičnog efekta.

Uneseni pojam uklanja se pritiskom na "iks" element unutar search bara *searchClearButton pronađen pomoću id-a* search-clear-button. U search bar unosi se novi pojam stišće se search button i čeka jednu sekundu (bez tog čekanja program se iz nekog razloga odvije krivim redoslijedom: odradi idući korak i onda ostane na ovom koraku kao zadnjem).

Uneseni pojam se opet uklanja i unosi novi, nakon čega se prozor browsera maksimizira i završi s testiranjem.

## Teardown

Nakon izvršenih testova na konzolu se ispišu dvije poruke. Prozor namjerno nije zatvoren.

# Osobni osvrt

Selenium mi se čini kao easy to pickup – hard to master. Bilo je zabavno automatizirano čačkati po webu na ovaj način, utrošio sam nešto više vremena na ovaj zadatak zbog toga što me je uvukao u eksperimentiranje.

Kod realizacije ideje naviše sam zapeo kod rukovanja elementom do kojeg ne znam kako doći pošto nema id, nema određeni tekst, niti xpath koji ga precizno gađa, a radi se o izlistanim rezultatima pretraživanja. Nisam pronašao previše pomoći na netu pa pretpostavljam da je rješenje banalno, ali ja ga nisam uspio dokučiti.

Velik misterij bio mi je "element not interactable" exception u liniji 45 kod poziva *searchQuery.SendKeys()*. Nije mi najjasnije kako se element uspio pronaći i spremiti u objekt, no kada mu se pokuša nešto unijeti onda se s njim ne može interactati. Tražio sam postoji li kakva ugrađena *IsInteractable()* metoda pa da pokušam napraviti kombinaciju s *wait.until()*, no bez uspjeha (Ako se dobro sjećam, bio je neki paket koji nešto slično omogućuje, no još nisam htio toliko duboko zalaziti).

Čekanja (*waits*). U dokumentaciji sam pročitao kako nije dobro miješati eksplicitna i implicitna čekanja, ja sam svejedno pokušao (ne bih li riješio onaj prethodno spomenuti element "not interactable" exception), no implicitni uopće nije radio. Malo sam pogrebao po površini i mislio sam da mi je jasno, no očito ne dovoljno da ih uspješno iskoristim.

Sve u svemu, nisam imao nekih većih poteškoća oko svladavanja obaveznog dijela zadatka i uspješno sam pratio sve iz danih materijala.