



Univerzitet u Sarajevu
Elektrotehnički fakultet Sarajevo
Odsjek za računarstvo i informatiku



Drugi dio projekta - Izvještaj

Verifikacija i validacija softvera

Ime i prezime: Amina Bajrić

Broj indeksa: 18892

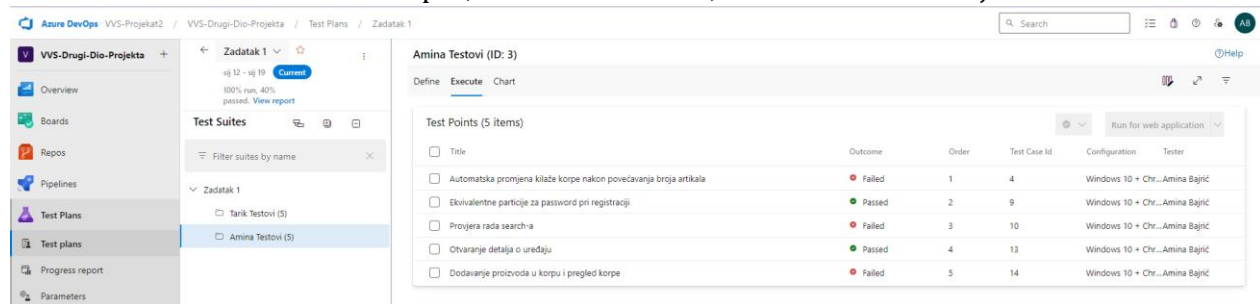
Svi zadaci urađeni u sklopu ovog projekta dokumentirani su na Azure DevOps platformi, a nalaze se u sklopu projekta na sljedećem linku: <https://dev.azure.com/VVS-Projekat2/VVS-Drugi-Dio-Projekta>

Zadatak 1

Stranica koja je korištena za testiranje je <https://demo.opencart.com/>

Ova stranica služi za pregledanje detalja o digitalnim uređajima, dodavanje istih u korpu, wish listu, upoređivanje specifikacija i slično.

Slijede testni slučajevi zajedno sa svim bitnim stavkama koje su vezane uz njih. Ovi testovi dodani su U Test Plans na Azure DevOpsu, kao Test Case-s, što se vidi na sljedećem screenshotu:



- **Automatska promjena kilaže korpe nakon povećanja broja artikala**

Detaljan opis testnog slučaja:

1. **Identifikator:** TEST CASE 4 (generisano automatski pri unosu testa u Azure Devops)
2. **Naziv testnog slučaja:** Automatska promjena kilaže korpe nakon povećavanja broja artikala
3. **Opis testnog slučaja:** Potrebno je dodati artikal u korpu. Nakon ovoga, otvara se korpa. Na stranici će biti prikazana labela koja govori kolika je ukupna težina korpe. Potom se poveća kvantitet za novododani proizvod (zbog preduslova koji će se dolje navesti u korpi će biti samo jedan artikal). Trebalo bi da se podatak o težini korpe ažurira bez da se uradi refresh stranice. Na kraju će se artikal obrisati iz korpe kako ne bi dolazilo do neočekivanih rješenja u narednim testiranjima.
4. **Potrebni preduslovi:** Prije izvršavanja testa obrisati sve artikle iz korpe, kako ne bi dolazilo do nedefinisanih ponašanja jer se kvantitet proizvoda postavlja na fiksnu veličinu.
5. **Testni podaci - Ulazne vrijednosti:**
 - u polje Quantity za proizvod unosi se "14"
6. **Koraci izvršavanja:**

Steps	Action	Expected result	Attachments
1.	Otvori https://demo.opencart.com/		
2.	Dodaj artikal u korpu klikom na dugme sa slikom korpe koja se nalazi ispod osnovnih informacija o artiklu. Dodaje se prvi artikal iz liste	Dodan 1 komad artikla	
3.	Otvori korpu klikom na dugme iz menija	Redirect na stranicu https://demo.opencart.com/index.php?route=checkout/cart&language=en-gb	
4.	Spasi trenutnu informaciju o kilaži korpe	Tekst labela spašen u varijablu	
5.	Povećaj broj komada dodanog artikla iz korpe na 14 unosom u polje Quantity za novododani artikal	Težina korpe povećana	
6.	Spasi promjene klikom na dugme za osvježavanje		
7.	Spasi novu informaciju o kilaži korpe	Tekst labela spašen u varijablu	
8.	Obriši novododani artikal iz korpe	Artikal obrisao	
9.	Uporedi staru i novu vrijednost kilaže korpe	Različite	

7. Izlazne vrijednosti:

- prikaz artikala u korpi i informacija o težini korpe

8. Post-uslovi-stanja:

- korpa na kraju prazna

Ovaj opis dodan je i kao summary u Azure DevOpsu za kreirani **Test Case**.

TEST CASE 4

4 Automatska promjena kilaže korpe nakon povećavanja broja artikala

Amina Bajric

0 comments

Add tag

Save & Close

Follow

Updated by Amina Bajric Yesterday

Steps

Summary

Associated Automation

Share

More

2

Description

1. **Identifikator:** TEST CASE 4 (generirano automatski pri unosu testa u Azure Devops)

2. **Naziv testnog slučaja:** Automatska promjena kilaže korpe nakon povećavanja broja artikala

3. **Opis testnog slučaja:** Potrebno je dodati artikal u korpu. Nakon ovoga, otvara se korpa. Na stranici će biti prikazana labela koja govori kolika je ukupna težina korpe. Potom se poveća kvantitet za novododani proizvod (zbog preduktova koji će se doći navesti u korpi će biti samo jedan artikal). Trebalo bi da se podatak o težini korpe ažurira bez da se usudi refresh stranice. Na kraju će se artikal obrisati iz korpe kako ne bi dolazilo do neočekivanih jeđenja u narednim testiranjima. Tehnika korištena za testiranje je testiranje na osnovu **slučajeva upotrebe**.

4. **Potrebni preduktovi:** Prije izvršavanja testa obrisati sve artikle iz korpe, kako ne bi dolazilo do nedefinisanih ponašanja jer se kvantitet proizvoda postavlja na fiksnu veličinu.

5. **Testni podaci - Ulazne vrijednosti:**

- u polje Quantity za proizvod unosi se "14"

6. **Koraci izvršavanja:** Data u dijelu **Steps**

7. **Izlazne vrijednosti:**

- prikaz artikala u korpi i informacija o težini korpe

8. **Post-uslovi-stanje:**

- korpa na kraju prazna

Ovaj test je automatiziran koristeći Selenium IDE. U Attachments dijelu data je side datoteka koja sadrži sve testove automatizirane kroz selenium IDE, kao i exportovani cs file konkretnog testa.

Tehnika korištena za testiranje:

Testiranje na osnovu **slučajeva upotrebe**.

Automatizacija testa:

U attachemnts dijelu testa na Azure DevOpsu nalazi se .side datoteka u kojoj su svi testovi kao i exportovani .cs fajl.

Attachments				
+ Add attachment				
Name ↑	Size	Date Attached	Comments	
AutomatskaPromjenaKilažeKorpenakonpo...	2K	15.1.2024. 1:11		
VVS-SeleniumIDE-Amina.side	17K	15.1.2024. 1:08		

Automatizacija ovog testa urađena je kroz **Selenium IDE**. Screensot testa i rezultata izvršavanja dat je ispod:

https://demo.opencart.com/

	Command	Target	Value
1	✓ open	/	
2	✓ click	css= .col.nth-child(2) .img-fluid	
3	✓ click	id=button-cart	
4	✓ click	css= .list-inline-item:nth-child(4) .d-none	
5	✓ click	css=html	
6	✓ store text	css=h1	Weight
7	✓ click	name=quantity	
8	✓ type	name=quantity	14
9	✓ click	css= .btn:nth-child(3)	
10	✓ click	css=main	
11	✓ store text	css=h1	WeightNew
12	✓ execute script	return (\${Weight})!=\${(WeightNew)})	jednake
13	✓ click	css= .btn:nth-child(4) > .fas	
14	✗ assert	jednake	true

Log	Reference
Running 'Automatska promjena kilaza korpe nakon povecanja broja artikala'	
1. open on / OK	01:13:43
2. click on css= .col.nth-child(2) .img-fluid OK	01:13:43
3. click on id=button-cart OK	01:13:44
4. click on css= .list-inline-item:nth-child(4) .d-none OK	01:13:44
5. click on css=html OK	01:13:45
6. storeText on css=h1 with value Weight OK	01:13:45
7. click on name=quantity OK	01:13:45
8. type on name=quantity with value 14 OK	01:13:46
9. click on css= .btn:nth-child(3) OK	01:13:46
10. click on css=main OK	01:13:46
11. storeText on css=h1 with value WeightNew OK	01:13:46
12. executeScript on return (\${Weight})!=\${(WeightNew)}) with value jednake OK	01:13:46
13. click on css= .btn:nth-child(4) > .fas OK	01:13:47
14. assert on jednake with value true Failed: Actual value 'false' did not match 'true'	01:13:47
'Automatska promjena kilaza korpe nakon povecanja broja artikala' ended with 1 error(s)	
	01:13:47

Ovaj test pada jer funkcionalnost koja se provjerava ne radi ispravno.

Detaljan opis bug-a:

Pri pregledu korpe, korisnici imaju mogućnost da povećaju ili smanje količinu svakog od proizvoda iz korpe. Nakon što unesu odgovarajuću vrijednost u polje Quantity, korisnici klikom na odgovarajuće dugme za osvježavanje sačuvaju promjene. Postoji labela koja informiše korisnika o ukupnoj težini korpe. Za očekivati je da se ova labela update-uje automatski kada se promjena sačuva, bez da korisnik mora uraditi reload stranice. Greška je upravo u tome što se ova promjena ne desi automatski.

U sklopu Azure DevOpsa dodan je **bug**, koji je prikazan ispod sa svim bitnim stavkama za opis bug-a.

BUG 5

Automatska promjena informacije o kilazi korpe

Amina Bajrić0 commentsAdd tag

StatusNew

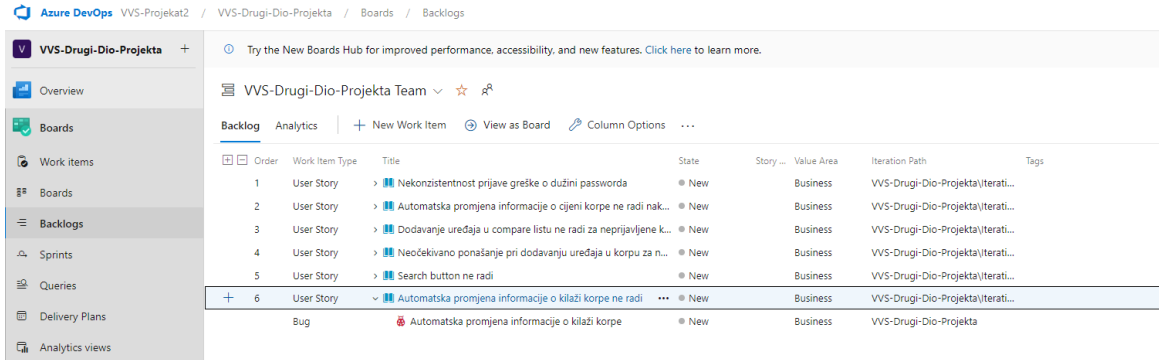
ReasonNew

AreaVVS-Drugi-Dio-Projekta

IntegrationVVS-Drugi-Dio-Projekta

Planning		
Resolved Reason		
Story Points		
Priority		
2		
Severity		
3 - Medium		
Activity		
Repro Steps		
12.1.2024. 14:19 Bug filed on "Automatska promjena kilaže korpe nakon povećavanja broja artikala"		
Step no.	Result	Title
1.	Passed	Otvori https://demo.opencart.com/
2.	Passed	Dodaj artikal u korpu
		Expected Result
		Dodan 1 komad artikla
3.	Passed	Otvori korpu
4.	Passed	Spasi trenutnu informaciju o kilaži korpe
		Expected Result
		Tekst labela spašen u varijablu
5.	Passed	Povećaj broj komada dodanog artikla iz korpe na 14
6.	Passed	Spasi promjene
7.	Passed	Spasi novu informaciju o kilaži korpe
		Expected Result
		Tekst labela spašen u varijablu
8.	Passed	Obriši novododani artikal iz korpe
		Expected Result
		Artikal obrisao
9.	Failed	Uporedi staru i novu vrijednost kilaže korpe
		Expected Result
		Različite
Comments: Kilaža korpe se trebala automatski uvećati, a to se nije desilo		
Test Configuration: Windows 10 + Chrome		

Bug je dodan i u backlog, kao dio UserStory-a sa imenom: **Automatska promjena informacije o kilaži korpe ne radi**



- *Ekvivalentne particije za password pri registraciji*

Detaljan opis testnog slučaja:

1. **Identifikator:** TEST CASE 9 (generisano automatski pri unosu testa u Azure Devops)
2. **Naziv testnog slučaja:** Ekvivalentne particije za password pri registraciji
3. **Opis testnog slučaja:** Pokušava se kreirati korisnički račun. Unosi se samo password, jer ostali podaci nisu bitni za ovaj test. Provjerava se da li se poruka o potrebnoj dužini passwprda ispisuje pod ispravnim uslovima, koristeći tehniku **ekvivalentnih particija**. Password može biti dužine između 4 i 20 karaktera, pa su ekvivalentne particije:




- password dužine manje od 4 karaktera
- password dužine veće od 20 karaktera
- password dužine između 4 i 20 karaktera

Za prve 2 particije, poruka upozorenja treba biti prikazana, a za treću element koji prikazuje poruku upozorenja ne treba biti vidljiv na stranici.

4. **Potrebni preduslovi:** Korisnik treba biti odjavljen iz aplikacije kako bi imao opciju Register
5. **Testni podaci - Ulazne vrijednosti:**

- [illegible]

- ## 6. Koraci izvršavanja:

Attachments				
+ Add attachment				
Name ↑		Size	Date Attached	Comments
 Ekivalentneparticipieczapasswordprirejist... 		3K	15.1.2024. 1:12	
 VVS-SeleniumIDE-Amina.side		17K	15.1.2024. 1:12	

[illegible]

Detaljan opis testnog slučaja:

1. **Identifikator:** TEST CASE 10 (generisano automatski pri unosu testa u Azure Devops)
2. **Naziv testnog slučaja:** Provjera rada search-a
3. **Opis testnog slučaja:** U search polje unosi se ime artikla koji sigurno postoji. Cilj je provjeriti da li search dugme ispravno radi, i da li će se prikazati uređaji sa unesenim imenom. Tehnika testiranja je testiranje na osnovu ***slučajeva upotrebe***.
4. **Potrebni preduslovi:** Potrebno je da postoji uređaj sa unesenim imenom
5. **Testni podaci - Ulazne vrijednosti:**
 - u polje search unosi se "MacBook"
6. **Koraci izvršavanja:**

Steps	Action	Expected result	Attachments
1.	Otvoriti stranicu https://demo.opencart.com/		
2.	U search polje unijeti ime uređaja koji sigurno postoji ("MacBook")		
3.	Kliknuti na dugme search	Prikaz uređaja sa unesenim imenom	
4.	Provjeriti lokaciju otvorenog prozora		
5.	Ako se otvori prozor "https://demo.opencart.com/error.html" test pada, u suprotnom prolazi		

7. Izlazne vrijednosti:

- prikaz uređaja sa unesenim imenom

8. Post-uslovi-stanja:

- nema post uslova

Ovaj opis dodan je i kao summary u Azure DevOpsu za kreirani **Test Case**.

Tehnika korištena za testiranje:

Testiranje na osnovu *slučajeva upotrebe*.

Automatizacija testa:

U attachemnts dijelu testa na Azure DevOpsu nalazi se .side datoteka u kojoj su svi testovi kao i exportovani .cs fajl.

Attachments			
+ Add attachment			
Name ↑	Size	Date Attached	Comments
ProvjeraradasearchaTest.cs	2K	15.1.2024. 1:13	
VVS-SeleniumIDE-Amina.side	17K	15.1.2024. 1:13	

Automatizacija ovog testa urađena je kroz **Selenium IDE**. Screensot testa i rezultata izvršavanja dat je ispod:

https://demo.opencart.com/			
	Command	Target	Value
1	✓ open	https://demo.opencart.com/	
2	✓ click	name=search	
3	✓ type	name=search	MacBook
4	✓ click	css=.btn-light	
5	✓ execute script	return window.location.href	currentURL
6	✓ execute script	return (\$currentURL)!="https://demo.opencart.com/error.html")	ispravanSearch
7	✗ assert	ispravanSearch	true

Log	Reference	
Running "Provjera rada search-a"		
1. open on https://demo.opencart.com/ OK		02:05:59
2. click on name=search OK		02:05:59
3. type on name=search with value MacBook OK		02:06:00
4. click on css=.btn-light OK		02:06:00
5. executeScript on return window.location.href with value currentURL OK		02:06:00
6. executeScript on return (\$currentURL)!="https://demo.opencart.com/error.html") with value ispravanSearch OK		02:06:00
7. assert on ispravanSearch with value true Failed: Actual value 'false' did not match 'true'		02:06:01
"Provjera rada search-a" ended with 1 error(s)		02:06:01

Ovaj test pada jer search ne radi.

Detaljan opis bug-a:

Nakon što korisnici unesu ime nekog uređaja koji postoji u Search polje i kliknu dugme za pretraživanje, desit će se internal server error i redirect na stranicu <https://demo.opencart.com/error.html> što naravno ne bi smjelo da se dešava, dakle search ne radi.

U sklopu Azure DevOpsa dodan je **bug**, koji je prikazan ispod sa svim bitnim stavkama za opis bug-a.

BUG 11

11 Serach button ne radi

Amina Bajrić

0 comments

Add tag

State

New

Area

VVS-Drugi-Dio-Projekta

Reason

New

Iteration

VVS-Drugi-Dio-Projekta

Planning

Resolved Reason



Story Points

Priority

3

Severity

2 - High

Activity

Repro Steps

12.1.2024. 23:12 Bug filed on "Provjera rada search-a"		
Step no.	Result	Title
1.	Passed	Otvoriti stranicu https://demo.opencart.com/
2.	Passed	U search polje unijeti ime uređaja koji sigurno postoji ("MacBook")
3.	Passed	Kliknuti na dugme search Expected Result Prikaz uređaja sa unesenim imenom
4.	Passed	Provjeriti lokaciju otvorenog prozora
5.	Failed	Ako se otvori prozor " https://demo.opencart.com/error.html " test pada, u suprotnom prolazi Comments: Otvorit će se link https://demo.opencart.com/error.html jer search button ne radi
Test Configuration:		Windows 10 + Chrome

Bug je dodan i u backlog, kao dio UserStory-a sa imenom: **Search button ne radi**

Azure DevOps VVS-Projekt2 / VVS-Drugi-Dio-Projekta / Boards / Backlogs

VVS-Drugi-Dio-Projekta +

Overview

Boards

Work items

Boards

Backlogs

Sprints

Queries

Delivery Plans

Analytics views

Try the New Boards Hub for improved performance, accessibility, and new features. [Click here to learn more.](#)

VVS-Drugi-Dio-Projekta Team

Backlog Analytics + New Work Item View as Board Column Options ...

Order	Work Item Type	Title	State	Story ...	Value Area	Iteration Path	Tags
1	User Story	> Nekonizistentnost prijave greške o dužini passworda	New		Business	VVS-Drugi-Dio-Projekta\iterati...	
2	User Story	> Automatska promjena informacije o cijeni korpe ne radi nak...	New		Business	VVS-Drugi-Dio-Projekta\iterati...	
3	User Story	> Dodavanje uređaja u compare listu ne radi za neprijavljene k...	New		Business	VVS-Drugi-Dio-Projekta\iterati...	
4	User Story	> Neočekivano ponašanje pri dodavanju uređaja u korpu za n...	New		Business	VVS-Drugi-Dio-Projekta\iterati...	
+ 5	User Story	> Search button ne radi	New		Business	VVS-Drugi-Dio-Projekta\iterati...	
	Bug	Serach button ne radi	New		Business	VVS-Drugi-Dio-Projekta	
6	User Story	> Automatska promjena informacije o kilaži korpe ne radi	New		Business	VVS-Drugi-Dio-Projekta\iterati...	

- **Otvaranje detalja o uređaju**

Detaljan opis testnog slučaja:

1. **Identifikator:** TEST CASE 13 (generisano automatski pri unosu testa u Azure Devops)
2. **Naziv testnog slučaja:** Otvaranje detalja o uređaju
3. **Opis testnog slučaja:** Nakon što se stranica otvori i prikažu dostupni uređaju, klikne se na prvi uređaj među prikazanim (u ovom slučaju "MacBook"). Nakon ovoga, trebalo bi da se desi automatski redirect na stranicu koja prikazuje detalje o traženom uređaju. U testu se provjerava da li se ovo zaista desilo, upoređujući očekivani link stranice i link koji je otvoren nakon klika. Korištena tehnika testiranja je testiranje na osnovu **slučajeva upotrebe**.
4. **Potrebni preduslovi:** Nema preduslova
5. **Testni podaci - Ulazne vrijednosti:**
 - nije potrebno ništa unositi
6. **Koraci izvršavanja:**

Steps	Action	Expected result	Attachments
1.	Otvoriti stranicu https://demo.opencart.com/		
2.	Kliknuti na link sa imenom prvog prikazanog uređaja	Otvaranje detalja o uređaju. Za slučaj MacBook-a otvara se stranica: https://demo.opencart.com/index.php?route=product/product&language=en-gb&product_id=43	
3.	Provjeriti da li link otvorene stranice odgovara linku na kojem se trebaju prikazati detalji traženog uređaja	Podudaranje	

7. Izlazne vrijednosti:

- prikaz detalja o uređaju koji je kliknut

8. Post-uslovi-stanja:

- nema post uslova

Isti ovaj opis kao dio summary-a:

ig "TEST CASE 13"

13 Otvaranje detalja o uređaju

Amina Bajric 0 comments Add tag

Save & Close Follow

State Design Area VVS-Drugi-Dio-Projekta

Reason New Iteration VVS-Drugi-Dio-Projekta

Updated by Amina Bajric Yesterday

Steps Summary Associated Automation

Description

- **Identifikator:** TEST CASE 13 (generisano automatski pri unosu testa u Azure Devops)
- **Naziv testnog slučaja:** Otvaranje detalja o uređaju
- **Opis testnog slučaja:** Nakon što se stranica otvori i prikažu dostupni uređaji, klikne se na prvi uređaj među prikazanim (u ovom slučaju "MacBook"). Nakon ovoga, trebalo bi da se desi automatski **redirect** na stranicu koja prikazuje detalje o traženom uređaju. U testu se provjerava da li se ovo zaista desilo. **upoređujući** očekivani link stranice i link koji je otvoren nakon klika. Korištena tehnika testiranja je testiranje na osnovu **slučajeva upotrebe**.
- **Potrebni preduслови:** Nema **preduslova**
- **Testni podaci - Ulazne vrijednosti:**
 - nije potrebno ništa unositi
- **Koraci izvršavanja:** Dato u dijelu **Steps**
- **Izlazne vrijednosti:**
 - prikaz detalja o uređaju koji je kliknut
- **Post-uslovi-stanja:**
 - nema post **uslova**

Test je automatiziran koristeći **Nunit Selenium automation framework**. Nalazi se na repozitoriju: https://dev.azure.com/VVS-Projekat2/VVS-Drugi-Dio-Projekta/_git/VVS-Nunit-testovi?path=%2f&version=GBmaster&a=contents

Tehnika korištena za testiranje:

Testiranje na osnovu **slučajeva upotrebe**

Automatizacija testa:

U summary dijelu koji je prikazan iznad dat je i link repozitorija na kojem se testovi nalaze, a isti link zalijepit ću i ovdje: https://dev.azure.com/VVS-Projekat2/VVS-Drugi-Dio-Projekta/_git/VVS-Nunit-testovi?path=%2f&version=GBmaster&a=contents

Testovi se nalaze na branchu **matser**.

Automatizacija ovog testa urađena je koristeći automation framework Nunit Selenium. Screensot testa i rezultata izvršavanja dat je ispod:

```
[Test]
public void OtvaranjeDetaljaOUredjaju_UpsjesanPrikaz()
{
    // Otvaranje početne stranice aplikacije
    driver.Navigate().GoToUrl("https://demo.opencart.com/");

    // Pronalazak linka sa imenom uređaja za prvi uređaj u listi, scroll do elementa i čekanje da bude vidljiv prije nego što se klikne
    IWebElement linkElement = driver.FindElement(By.XPath("//*[@id='content']/div[2]/div[1]/form/div/div[2]/div[1]/h4/a"));
    ScrollToElement(linkElement);
    WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(15));
    wait.Until(ExpectedConditions.ElementToBeClickable(linkElement));

    // Realizacija klika na link kako bi se otvorile informacije o uređaju
    IJavaScriptExecutor jsExecutor = (IJavaScriptExecutor)driver;
    jsExecutor.ExecuteScript("arguments[0].click();", linkElement);

    // Da bi se osiguralo da se klik obradi prije sljedeće naredbe, sačekamo
    Thread.Sleep(5000);

    // Da li su se otvorile informacije o kliknutom uređaju
    string ocekivaniURL = "https://demo.opencart.com/index.php?route=product/product&language=en-gb&product_id=43";
    Assert.AreEqual(ocekivaniURL, driver.Url);
}
```

```
private void ScrollToElement(IWebElement element)
{
    IJavaScriptExecutor jsExecutor = (IJavaScriptExecutor)driver;
    jsExecutor.ExecuteScript("arguments[0].scrollIntoView({block: 'center'});", element);
}
```

Test Detail Summary

✓ OtvaranjeDetaljaOUredjaju_UpsjesanPrikaz

📄 Source: [AminaTestovi.cs](#) line 14

🕒 Duration: 8,4 sec

- **Dodavanje proizvoda u korpu i pregled korpe**

Detaljan opis testnog slučaja:

1. **Identifikator:** TEST CASE 14 (generisano automatski pri unosu testa u Azure Devops)
2. **Naziv testnog slučaja:** Dodavanje proizvoda u korpu i pregled korpe
3. **Opis testnog slučaja:** Ovaj test provjerava funkcionisanje dodavanja u korpu za neprijavljenog korisnika. Korisnik ima opciju da doda uređaj u korpu bez obzira da li je prijavljen u aplikaciju. Nakon što korisnik klikne na dugme za dodavanje uređaja u korpu, prikazuje mu se iskočni prozor koji ga obavještava da je proizvod uspješno dodan u korpu. Potom se provjerava da li se uređaj zaista nalazi u korpi. Korištena tehnika testiranja je testiranje na osnovu **slučajeva upotrebe**.
4. **Potrebni preduslovi:** Korpa je prazna i korisnik nije prijavljen u aplikaciju.
5. **Testni podaci - Ulazne vrijednosti:**
- nije potrebno ništa unositi
6. **Koraci izvršavanja:**

Steps	Action	Expected result	Attachments
1.	Otvoriti stranicu https://demo.opencart.com/		
2.	Kliknuti na dugme za dodavanje u korpu u sklopu kartice koja prikazuje prvi uređaj iz liste	Pop-up window koji govori korisniku da je proizvod uspješno dodan u korpu	
3.	Klik na dugme Shopping Cart iz menija	Otvaranje korpe	
4.	Provjera da li je uređaj dodan u korpu	Uređaj je u korpi	

7. **Izlazne vrijednosti:**
 - prikaz sadržaja korpe
8. **Post-uslovi-stanja:**
 - nema post uslova-stanja

Isti ovaj opis kao dio summary-a:

TEST CASE 14

14 Dodavanje proizvoda u korpu i pregled korpe

Amina Bajrić

0 comments

Save & Close

Follow

Updated by Amina Bajrić Yesterday

Steps Summary Associated Automation

Description

- **Identifikator:** TEST CASE 14 (generirano automatski pri unosu testa u Azure DevOps)
- **Naziv testnog slučaja:** Dodavanje proizvoda u korpu i pregled korpe
- **Opis testnog slučaja:** Ovaj test provjerava funkcionalnost dodavanja u korpu za neprijavljenog korisnika. Korisnik ima opciju da doda uređaj u korpu bez obzira da li je prijavljen u aplikaciju. Nakon što korisnik klikne na dugme za dodavanje uređaja u korpu, prikazuje mu se modalni prozor koji ga obavještava da je proizvod uspješno dodan u korpu. Potom se provjerava da li se uređaj zaista nalazi u korpi. Korištena tehnika testiranja je testiranje na osnovu **slučajeva upotrebe**.
- **Potrebni preduslovi:** Korpa je prazna i korisnik nije prijavljen u aplikaciju.
- **Testni podaci - Ulazne vrijednosti:**
 - nije potrebno ništa unositi
 - potrebno ništa unositi
 - potrebno ništa unositi
- **Koraci izvršavanja:** Definisano u dijelu **Steps**
- **Izlazne vrijednosti:**
 - prikaz sadržaja korpe
 - nema post uslova-stanja

Test je automatiziran koristeći **Nunit Selenium automation framework**. Nalazi se na repozitoriju: https://dev.azure.com/VVS-Projekat2/VVS-Drugi-Dio-Projekta/_git/VVS-Nunit-testovi?path=%2f&version=GBmaster&a=contents

Tehnika korištena za testiranje:

Testiranje na osnovu *slučajeva upotrebe*

Automatizacija testa:

U summary dijelu koji je prikazan iznad dat je i link repozitorija na kojem se testovi nalaze, a isti link zalijepit ću i ovdje: https://dev.azure.com/VVS-Projekat2/VVS-Drugi-Dio-Projekta/_git/VVS-Nunit-testovi?path=%2f&version=GBmaster&a=contents

Testovi se nalaze na branchu **matser**.

Automatizacija ovog testa urađena je koristeći automation framework Nunit Selenium. Screensot testa i rezultata izvršavanja dat je ispod:

```
[Test]
0 references | Please sign-in to New Relic CodeStream to see Code Level Metrics
public void DodavanjeProizvodaUKorpuIPregledKorpe()
{
    // Otvaranje početne stranice aplikacije
    driver.Navigate().GoToUrl("https://demo.opencart.com/");

    // Pronalazak dugmeta koje dodaje element u korpu za prvi element u listi, scroll do elementa i čekanje da bude vidljiv prije nego što se klikne
    IWebElement addToCartButton = driver.FindElement(By.XPath("//*[id=\"content\"]/div[2]/div[1]/form/div/div[2]/div[2]/button[1]"));
    ScrollToElement(addToCartButton);
    WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(15));
    wait.Until(ExpectedConditions.ElementToBeClickable(addToCartButton));

    // Klik na dugme za dodavanje u korpu
    IJavaScriptExecutor jsExecutor = (IJavaScriptExecutor)driver;
    jsExecutor.ExecuteScript("arguments[0].click();", addToCartButton);

    // Da bi se osiguralo da se klik obradi prije sljedeće naredbe, sačekamo
    Thread.Sleep(5000);

    // Tražimo link iz menija koji otvara korpu kako bismo provjerili da li se element dodao u korpu
    IWebElement openCart = driver.FindElement(By.XPath("//*[id=\"top\"]/div[2]/div[2]/ul/li[4]/a"));
    ScrollToElement(openCart);
    wait.Until(ExpectedConditions.ElementToBeClickable(openCart));

    // Klik na link i čekanje
    jsExecutor.ExecuteScript("arguments[0].click();", openCart);
    Thread.Sleep(5000);

    // Provjera da li ima elemenata u korpi (preduslov je da prije testa bude prazna)
    IWebElement korpaPraznaInfo = driver.FindElement(By.XPath("//*[id=\"content\"]/p"));
    Assert.IsFalse(korpaPraznaInfo.Displayed);
}
3 references | Please sign-in to New Relic CodeStream to see Code Level Metrics | 1/2 passing
private void ScrollToElement(IWebElement element)
{
    IJavaScriptExecutor jsExecutor = (IJavaScriptExecutor)driver;
    jsExecutor.ExecuteScript("arguments[0].scrollIntoView({block: 'center'});", element);
}
```

```
Test Detail Summary
❌ DodavanjeProizvodaUKorpuIPregledKorpe
📄 Source: AminaTestovi.cs line 37
🕒 Duration: 13,5 sec

Message:
Expected: False
But was: True

Stack Trace:
AminaTestovi.DodavanjeProizvodaUKorpuIPregledKorpe() line 66
1) at VVS_Drugi_dio_projekta.Tests.AminaTestovi.DodavanjeProizvodaUKorpuIPregledKorpe()
```

Ovaj test pada jer dodavanje proizvoda u korpu za neulogovane korisnike ne radi konzistentno.

Detaljan opis bug-a:

Ako korisnik koji nije ulogovan pokuša da doda proizvod u korpu, on će dobiti poruku da je proizvod uspješno dodan u korpu. Međutim, kada korisnik pokuša pregledati svoje uređaje koje je dodao u korpu, desit će se da će se ispisati informacija da je korpa prazna. Dakle, korisnici koji nisu prijavljeni zapravo nemaju mogućnost da dodaju proizvode u korpu. Međutim, korisnici ne samo da nisu obaviješteni o tome da nemaju ovu mogućnost, već im je čak rečeno da su ovo uspješno odradili što dovodi do nekonzistentnosti. Potrebno je obavijestiti korisnike da se najprije moraju prijaviti da bi mogli dodavati proizvode u korpu.

U sklopu Azure DevOpsa dodan je **bug**, koji je prikazan ispod sa svim bitnim stavkama za opis bug-a.

BUG 16

16 Neočekivano ponašanje pri dodavanja proizvoda u korpu za neprijavljenog korisnika

Amina Bajrić

0 comments

Add tag

State

New

Area

VVS-Drugi-Dio-Projekta

Reason

New

Iteration

VVS-Drugi-Dio-Projekta

Planning

Resolved Reason

Story Points

Priority

2

Severity

3 - Medium

Activity

Repro Steps

13.1.2024. 18:02

Bug filed on "Dodavanje proizvoda u korpu i pregled korpe"

Step no.	Result	Title
1.	Passed	Otvoriti stranicu https://demo.opencart.com/
2.	Passed	Kliknuti na dugme za dodavanje u korpu u sklopu kartice koja prikazuje prvi uređaj iz liste Expected Result Pop-up window koji govori korisniku da je proizvod uspješno dodan u korpu
3.	Passed	Klik na dugme Shopping Cart iz menija Expected Result Otvaranje korpe
4.	Failed	Provjera da li je uređaj dodan u korpu Expected Result Uređaj je u korpi Comments: Uređaj neće biti dodan u korpu zato što korisnik nije prijavljen. Međutim, ovo nije uredu, jer je korisniku rečeno da je uređaj uspješno dodan u korpu. Potrebno je potpuno onemogućiti dodavanje u korpu ako korisnik nije prijavljen.

Test Configuration:Windows 10 + Chrome

Bug je dodan i u backlog, kao dio UserStory-a sa imenom: **Neočekivano ponašanje pri dodavanja uređaja u korpu za neprijavljenog korisnika**

Azure DevOpsVVS-Projekat2 / VVS-Drugi-Dio-Projekta / Boards / Backlogs

VVS-Drugi-Dio-Projekta

Try the New Boards Hub for improved performance, accessibility, and new features. [Click here](#) to learn more.

VVS-Drugi-Dio-Projekta Team

BacklogAnalyticsNew Work ItemView as BoardColumn Options

Order	Work Item Type	Title	State	Story ...	Value Area	Iteration Path	Tags
1	User Story	> Nekonstistentnost prijave greške o dužini passworda	New		Business	VVS-Drugi-Dio-Projekta\iterati...	
2	User Story	> Automatska promjena informacije o cijeni korpe ne radi nak...	New		Business	VVS-Drugi-Dio-Projekta\iterati...	
3	User Story	> Dodavanje uređaja u compare listu ne radi za neprijavljene k...	New		Business	VVS-Drugi-Dio-Projekta\iterati...	
+	User Story	> Neočekivano ponašanje pri dodavanju uređaja u korpu ...	New		Business	VVS-Drugi-Dio-Projekta\iterati...	
	Bug	Neočekivano ponašanje pri dodavanja proizvoda u korpu...	New		Business	VVS-Drugi-Dio-Projekta	
5	User Story	> Search button ne radi	New		Business	VVS-Drugi-Dio-Projekta\iterati...	
6	User Story	> Automatska promjena informacije o kilaži korpe ne radi	New		Business	VVS-Drugi-Dio-Projekta\iterati...	

Zadatak 2

S obzirom da aplikacija korištena za prvi i treći zadatak nije bila pogodna za API testiranje, za ovaj zadatak koristila sam jednostavni API koji se nalazi na sljedećem linku <https://restful-booker.herokuapp.com/apidoc/index.html>

Testovi iz ovog zadatka nalaze se na sljedećem repozitoriju: https://dev.azure.com/VVS-Projekat2/VVS-Drugi-Dio-Projekta/_git/VVS-Drugi-Dio-Projekta

Također je napravljen Pipeline koji je vezan uz ovaj repozitorij. Pipeline se nalazi na sljedećem linku: https://dev.azure.com/VVS-Projekat2/VVS-Drugi-Dio-Projekta/_build?definitionId=1

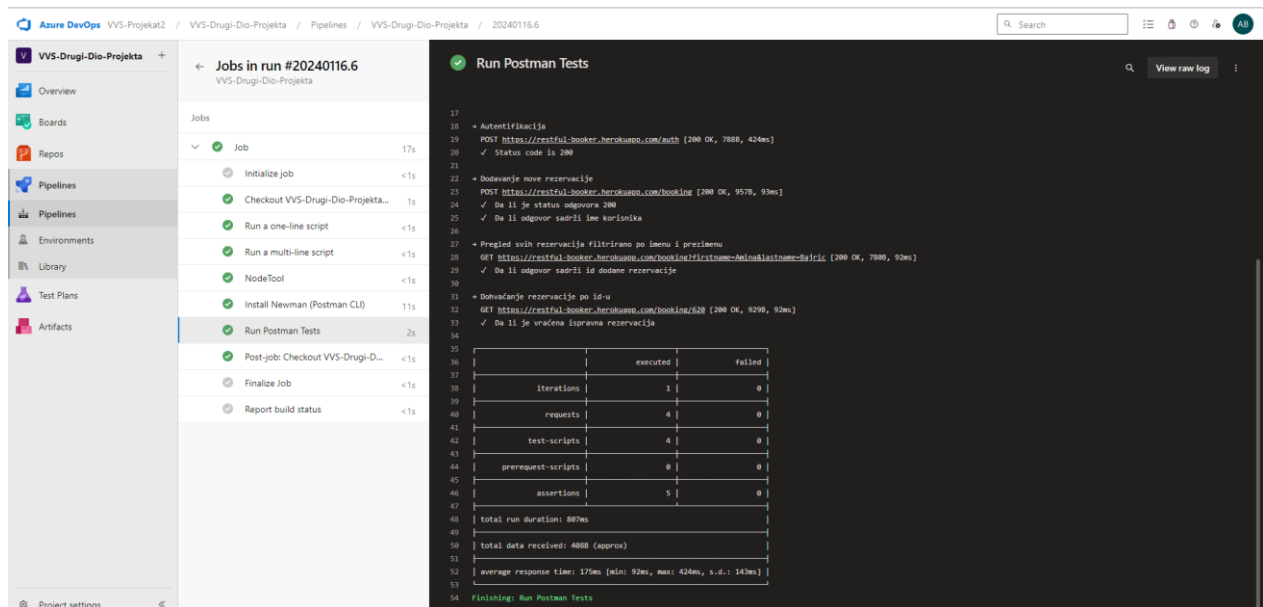
Rezultati pokretanja API testova koji će ispod biti navedeni preko pipeline-a su na sljedećim slikama:

Azure DevOps interface showing the 'Jobs in run #20240116.6' for the 'VVS-Drugi-Dio-Projekta' pipeline. The jobs list includes:

- Initialize job
- Checkout VVS-Drugi-Dio-Projekta...
- Run a one-line script
- Run a multi-line script
- NodeTool
- Install Newman (Postman CLI)
- Run Postman Tests
- Post-job: Checkout VVS-Drugi-Dio-Projekta...
- Finalize Job
- Report build status

Azure DevOps interface showing the 'Run Postman Tests' job details. The output shows the Newman command line script and the results of the API tests:

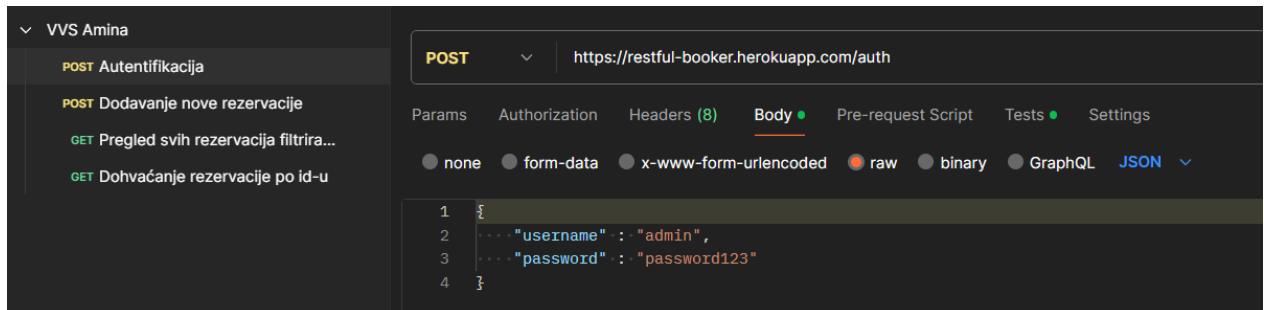
```
1 Starting: Run Postman Tests
2 =====
3 Task : Command line
4 Description : Run a command line script using Bash on Linux and macOS and cmd.exe on Windows
5 Version : 2.231.0
6 Author : Microsoft Corporation
7 Help : https://docs.microsoft.com/azure/devops/pipelines/tasks/yml/vs/command_line
8 =====
9 Generating script.
10 Script contents:
11 newman run https://api.postman.com/collections/27247259-8738a7c-f2c8-4883-927a-822e1d5a1d5b?access_key=TW6T-8JmP8Y9H8J3499XVPS2ABQ2H9
12 =====
13 /usr/bin/bash --noprofile --norc /home/vsts/work/_temp/87f34521-c42b-422b-b467-b29acc867f9.sh
14 newman
15
16 VVS Amina
17
18 + Autentifikacija
19 POST https://restful-booker.herokuapp.com/auth [200 OK, 788B, 42ms]
20 ✓ Status code is 200
21
22 + Dodavanje nove rezervacije
23 POST https://restful-booker.herokuapp.com/bookings [200 OK, 957B, 93ms]
24 ✓ Da li je status odgovora 200
25 ✓ Da li odgovor sadrži ime korisnika
26
27 + Pregled svih rezervacija filtrirano po imenu i prezimenu
28 GET https://restful-booker.herokuapp.com/bookings?firstname=Helina&lastname=Byjric [200 OK, 788B, 92ms]
29 ✓ Da li odgovor sadrži id dodane rezervacije
30
31 + Dohvaćanje rezervacije po id-u
32 GET https://restful-booker.herokuapp.com/bookings/638 [200 OK, 929B, 92ms]
33 ✓ Da li je vraćena ispravna rezervacija
34
```



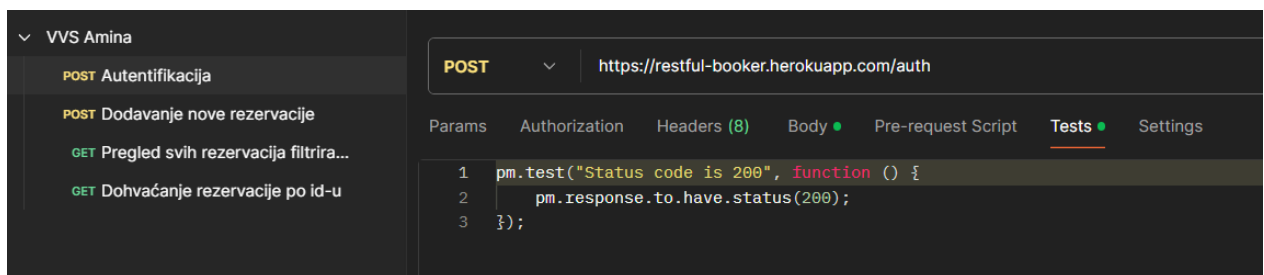
• Autentifikacija

Ovo je jednostavni zahtjev koji se izvršava samo sa ciljem da se dobije pristupni token, pa je u sklopu njega samo dodan test koji provjerava da li je status odgovora 200.

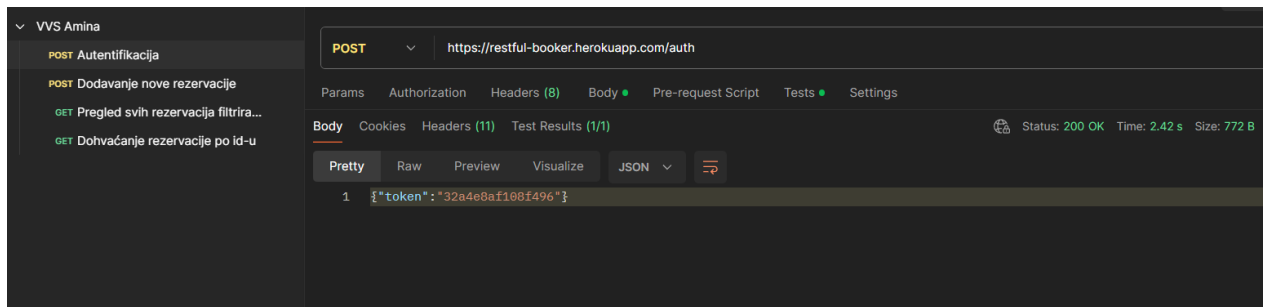
Najprije se uputi POST zahtjev i navede tijelo na sljedeću rutu, a sve bitno vidljivo je na slici



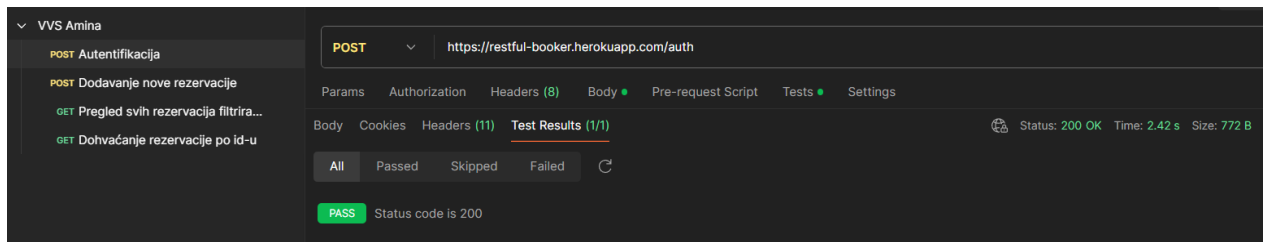
Test:



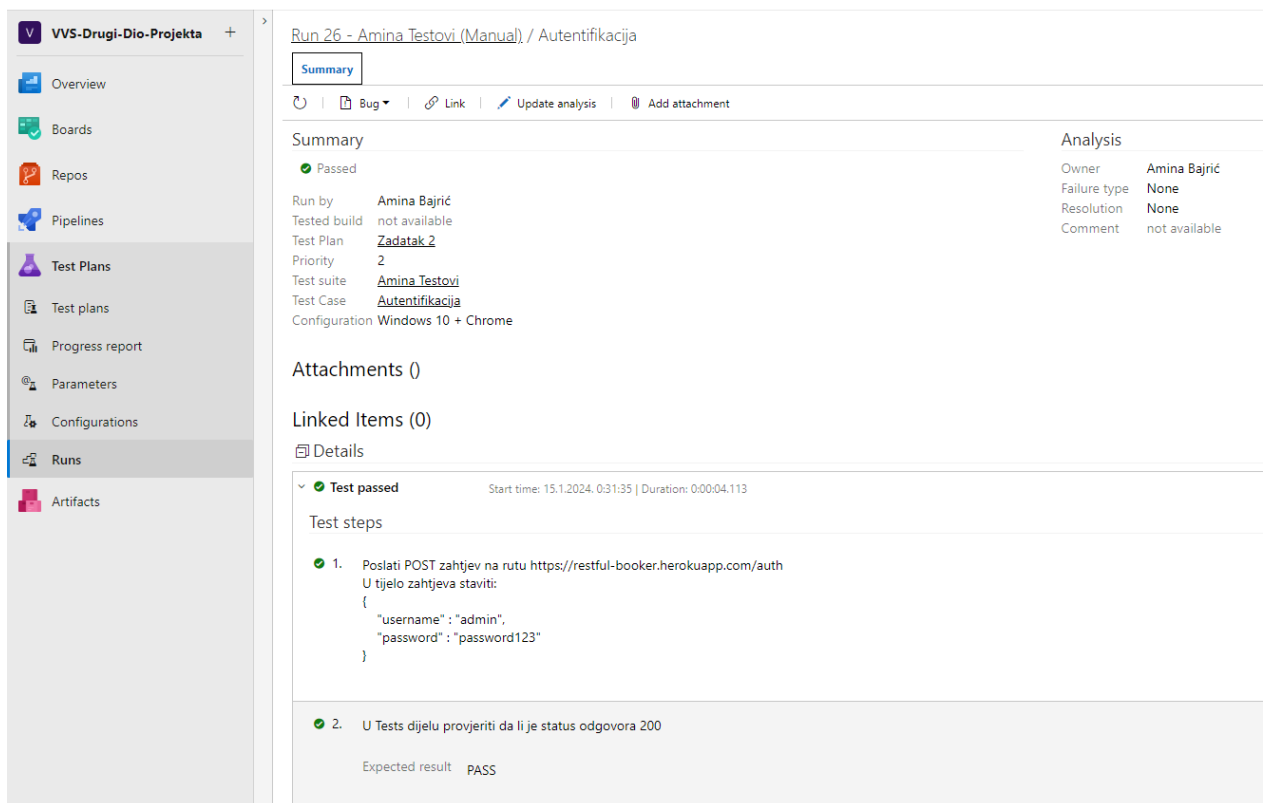
Odgovor:



Rezultati testa:

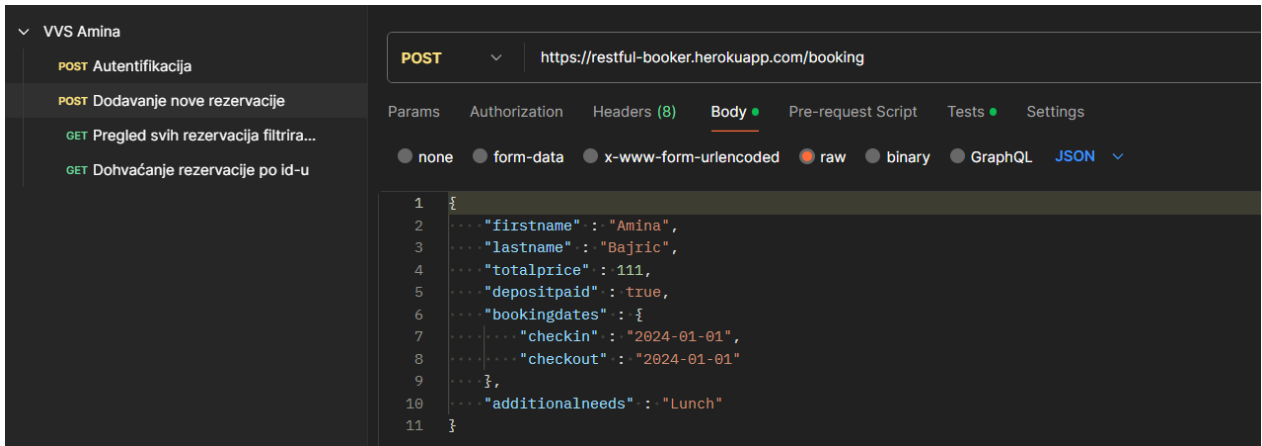


Test je dokumentiran na Azure DevOps platformi:



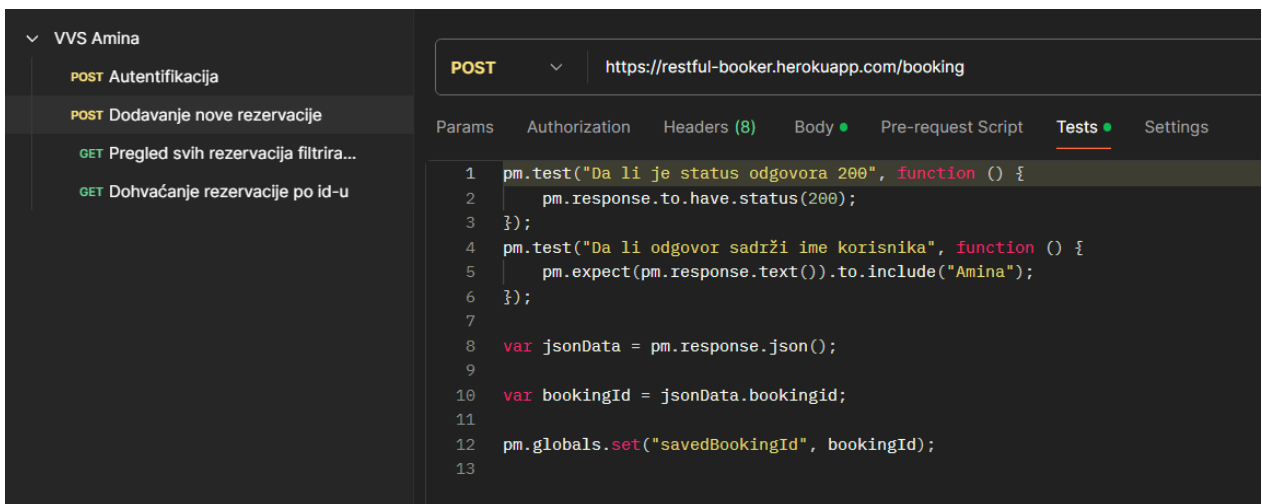
- ***Dodavanje nove rezervacije***

Slanjem POST zahtjeva kreira se nova rezervacija, a ruta i tijelo zahtjeva se mogu vidjeti na slici.

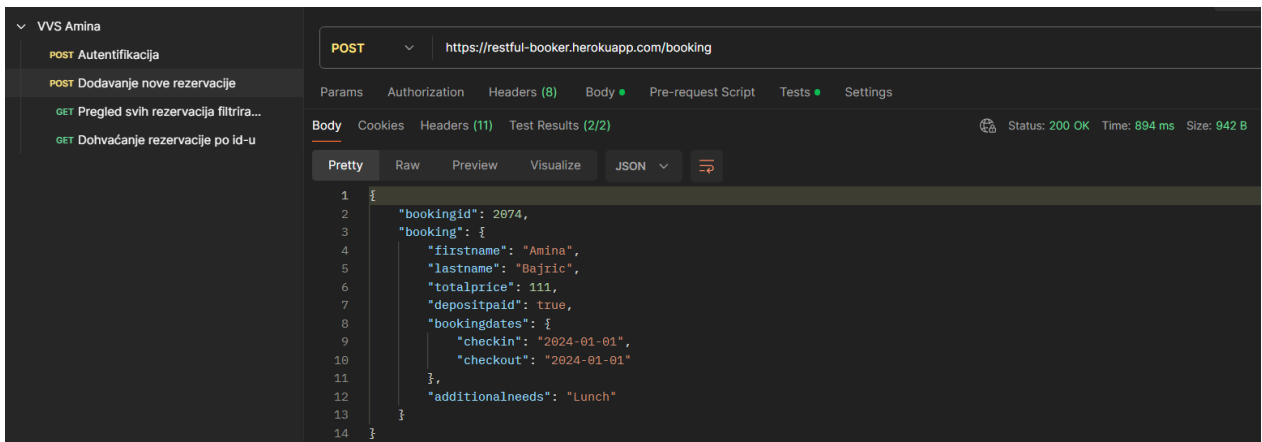


Test:

Najprije se provjerava da li je status odgovora 200, a zatim da li odgovor sadrži string „Amina“ koji je bio sadržan u tijelu zahtjeva. Id kreirane rezervacije se spašava u globalnu varijablu u svrhu provjera koje će biti obavljene u narednim testovima.



Odgovor:



Rezultati testa:

VVS Amina

POST Autentifikacija

POST Dodavanje nove rezervacije

GET Pregled svih rezervacija filtrira...

GET Dohvaćanje rezervacije po id-u

POST

https://restful-booker.herokuapp.com/booking

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

BodyCookiesHeaders (11)Test Results (2/2)

Status: 200 OKTime: 894 msSize: 942 B

AllPassedSkippedFailed

PASS

Da li je status odgovora 200

PASS

Da li odgovor sadrži ime korisnika

Test je dokumentiran na Azure DevOps platformi:

VVS-Drugi-Dio-Projekta

Overview

Boards

Repos

Pipelines

Test Plans

Test plans

Progress report

Parameters

Configurations

Runs

Artifacts

Run 26 - Amina Testovi (Manual) / Dodavanje nove rezervacije

Summary

Summary

Analysis

Attachments ()

Linked Items (0)

Details

Test passed

Start time: 15.1.2024. 0:31:44 | Duration: 0:00:01.938

Test steps

1. Poslati POST zahtjev na rutu: https://restful-booker.herokuapp.com/booking
U tijelu zahtjeva navesti sljedeće:
{
 "firstname": "Amina",
 "lastname": "Bajric",
 "totalprice": 111,
 "depositpaid": true,
 "bookingdates": {
 "checkin": "2024-01-01",
 "checkout": "2024-01-01"
 },
 "additionalneeds": "Lunch"
}

2. Provjeriti da li je status odgovora 200

Expected result PASS

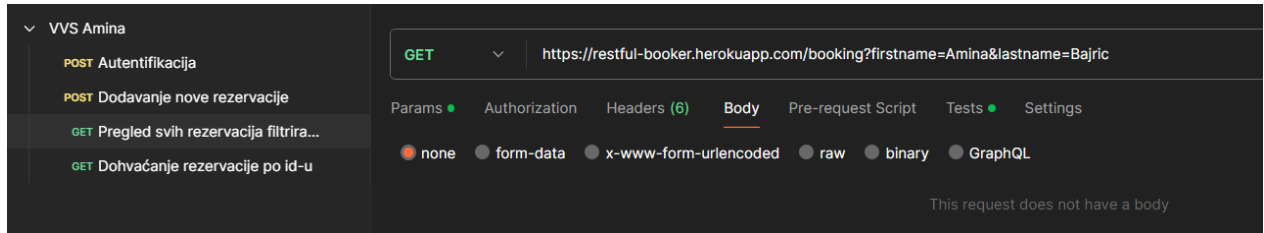
3. Provjeriti da li tijelo odgovora sadrži string "Amina"

Expected result PASS

4. Id kreirane rezervacije spasiti u globalnu varijablu

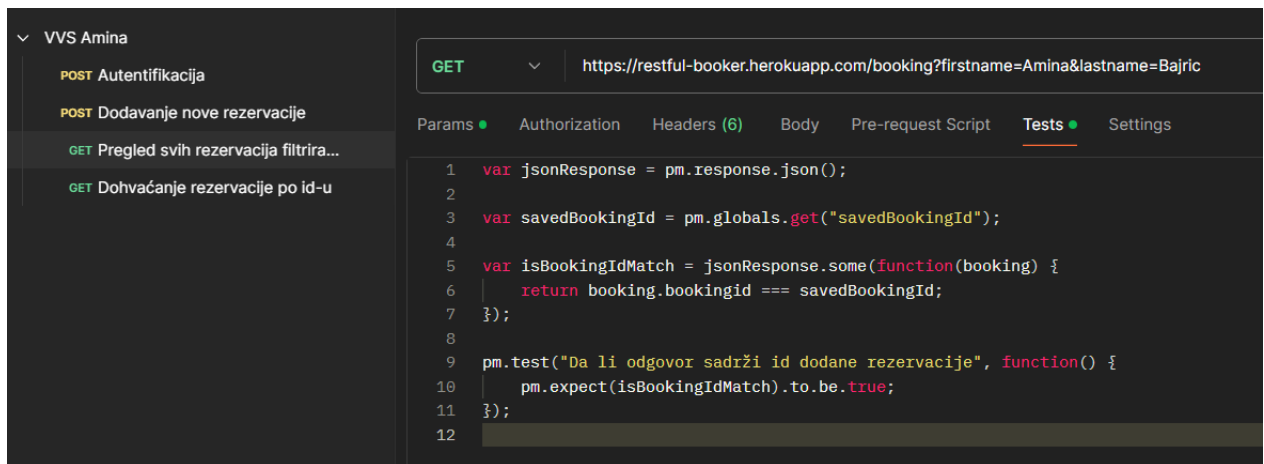
- **Pregled svih rezervacija filtrirano po imenu i prezimenu**

Šalje se get zahtjev gdje se kao parametri u ruti šalju ime i prezime korisnika za kojeg je rezervacija ranije kreirana. Cilj je vidjeti da li ispravno radi GET zahtjev za rezervacije sa filtriranjem.

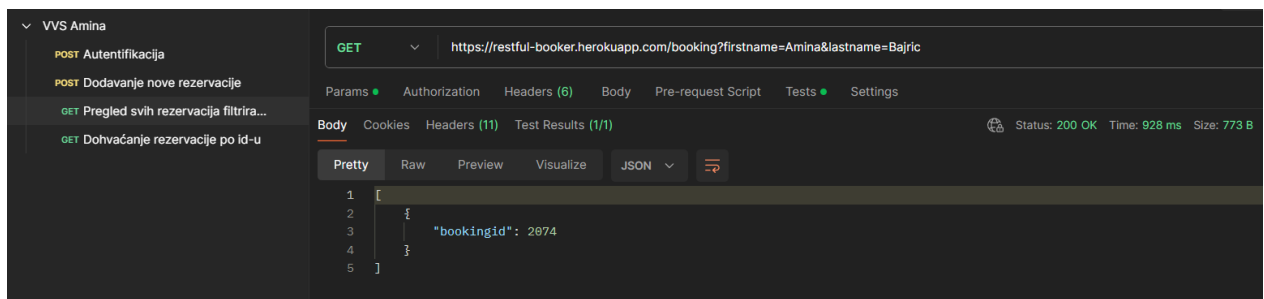


Test:

U tijelu odgovora za ovakav GET zahtjev vraćaju se samo id-evi rezervacija. Stoga se, koristeći globalnu varijablu koja je ranije spašena, provjerava da li odgovor sadrži taj id.



Odgovor:



Rezultati testa:

The screenshot shows a REST client interface with a sidebar on the left containing a list of endpoints: **POST** Autentifikacija, **POST** Dodavanje nove rezervacije, **GET** Pregled svih rezervacija filtrirano..., and **GET** Dohvaćanje rezervacije po id-u. The main panel displays a **GET** request to `https://restful-booker.herokuapp.com/booking?firstname=Amina&lastname=Bajric`. The **Test Results (1/1)** tab is active, showing a **PASS** status with the message "Da li odgovor sadrži id dodane rezervacije". The status bar indicates **Status: 200 OK**, **Time: 928 ms**, and **Size: 773 B**.

Test je dokumentiran na Azure DevOps platformi:

The screenshot shows the Azure DevOps interface for a test run. The left sidebar lists navigation options: Overview, Boards, Repos, Pipelines, Test Plans, Test plans, Progress report, Parameters, Configurations, **Runs**, and Artifacts. The main content area displays the test run details for "Run 26 - Amina Testovi (Manual) / Pregled svih rezervacija filtrirano po imenu i prezimenu". The **Summary** tab is selected, showing a **Passed** status. The test was run by **Amina Bajrić** on build **not available**, using test plan **Zadatak 2** (Priority 2) and test suite **Amina Testovi**. The test case is **Pregled svih rezervacija filtrirano po imenu i prezimenu**, configured for **Windows 10 + Chrome**. The **Analysis** section shows the owner as **Amina Bajrić** with no failure type, resolution, or comment. The **Attachments (0)** and **Linked Items (0)** sections are empty. The **Details** tab shows the test steps: 1. "Poslati GET zahtjev na rutu: https://restful-booker.herokuapp.com/booking?firstname=Amina&lastname=Bajric. Zadani firstname i lastname postoje jer su dodani u tesnom slučaju koji se izvršava prije ovog" and 2. "Provjeriti da li odgovor sadrži id rezervacije koja je ranije dodana i čiji id je sačuvan u globalnoj varijabli". The expected result is **PASS**.

• **Dohvaćanje rezervacije po id-u**

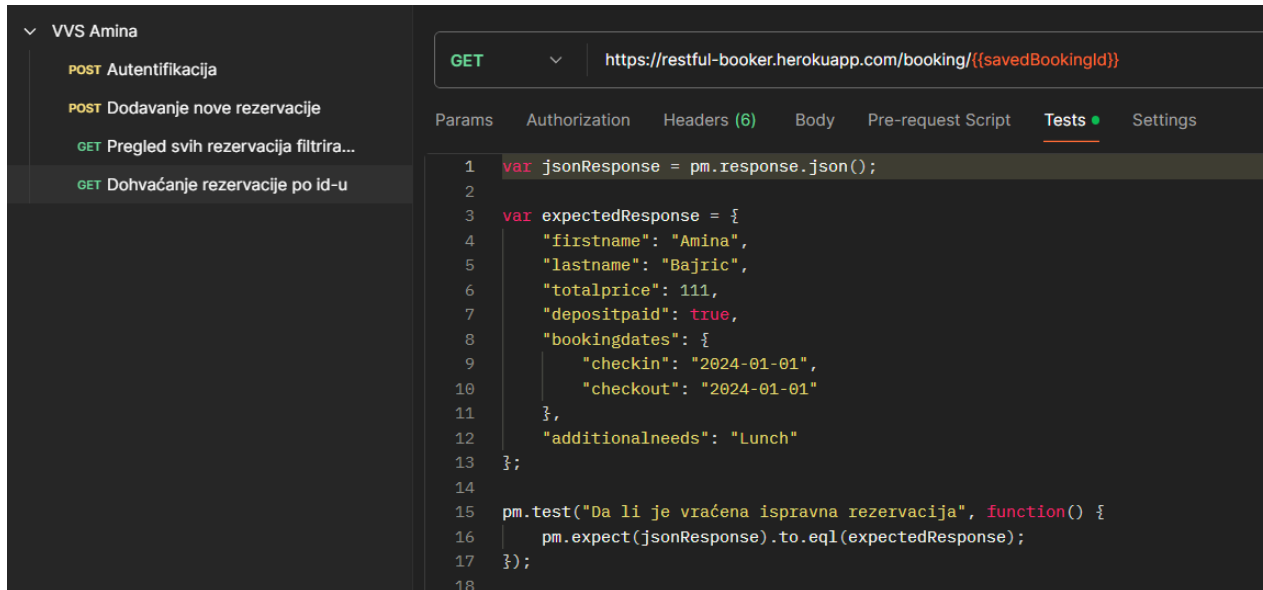
Šalje se get zahtjev gdje se kao dio rute šalje id rezervacije koja je kreirana u jednom od ranijih testova, a koji je spašen u globalnu varijablu.

The screenshot shows a REST client interface with a sidebar on the left containing a list of endpoints: **POST** Autentifikacija, **POST** Dodavanje nove rezervacije, **GET** Pregled svih rezervacija filtrirano..., and **GET** Dohvaćanje rezervacije po id-u. The main panel displays a **GET** request to `https://restful-booker.herokuapp.com/booking/{{savedBookingId}}`. The **Params** tab is active, showing the **Query Params** section with a table:

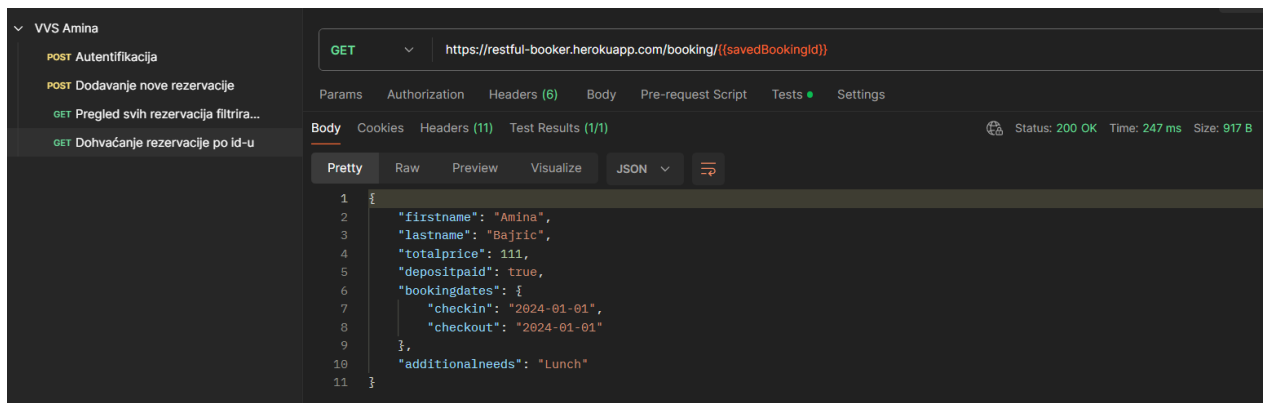
Key	Value
Key	Value

Test:

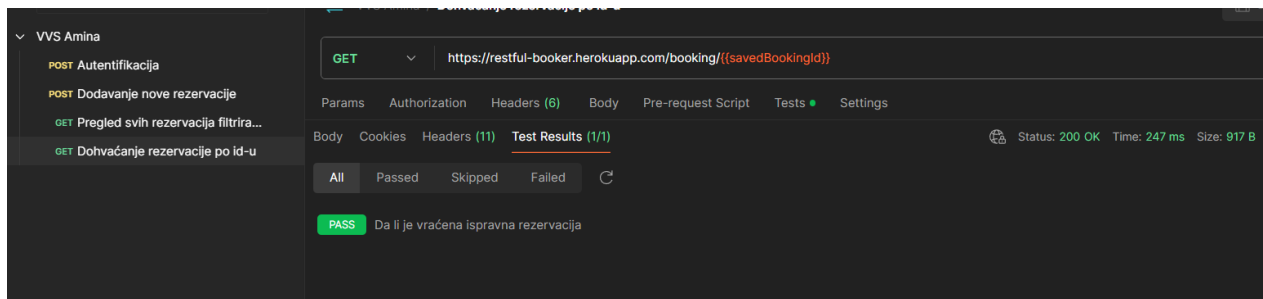
Znajući kakav oblik imaju podaci koji se vraćaju pri ovakvom GET zahtjevu, te kakvi podaci su zadani pri kreiranju rezervacije, provjerava se da li odgovor koji je vraćen se potpuno poklapa sa očekivanim.



Odgovor:



Rezultati testa:



Test je dokumentiran na Azure DevOps platformi:

VVVS-Drugi-Dio-Projekta

Overview

Boards

Repos

Pipelines

Test Plans

Test plans

Progress report

Parameters

Configurations

Runs

Artifacts

Run 26 - Amina Testovi (Manual) / Dohvaćanje rezervacije po id-u

Summary

Bug | Link | Update analysis | Add attachment

Summary

Analysis

Attachments ()

Linked Items (0)

Details

Test passed

Start time: 15.1.2024, 0:32:01 | Duration: 0:00:02.395

Test steps

1. Poslati GET zahtjev na rutu: https://restful-booker.herokuapp.com/booking/{savedBookingId} savedBookingId je globalna varijabla koja je sačuvana u jednom od testova koji se izvršavaju prije ovog

2. Provjeriti da li se tijelo odgovora podudara sa rezervacijom koja je dobijena kao odgovor pri kreiranju u jednom od ranijih testova

Expected result: PASS

Zadatak 3

Za određivanje performansi aplikacije koristili smo alat Loadster. Stranica nad kojom se vrši testiranje je ista kao i u prvom zadatku: <https://demo.opencart.com/>

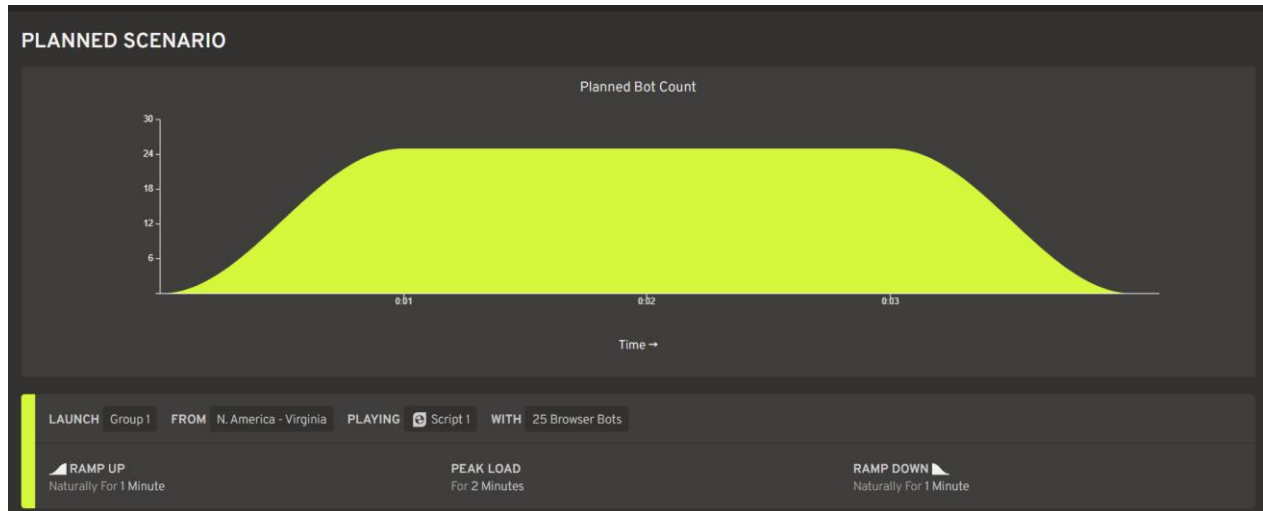
Load testiranje

Za izvršenje load testiranja stavili smo se u poziciju korisnika koji je posjetio stranicu s ciljem da pregleda dostupne uređaje, uporedi im specifikacije, doda ih u korpu, wish listu i slično. Cilj je vidjeti ponašanje stranice pri uobičajenom opterećenju kakvo se može očekivati. Aplikaciju smo targetirali sa 25 botova. Dodatne informacije o vremenu provedenom na stranici i rezultatima testiranja će biti prikazane ispod. Detaljan testni scenario koji smo koristili za ovaj tip testiranja je sljedeći:

- Otvaranje stranice <https://demo.opencart.com/>
- Login u aplikaciju
- Klikom na logo stranice navigira se na početnu stranicu
- Otvaraju se detalji trećeg uređaj iz liste dostupnih klikom na link u sklopu kartice u kojoj je prikazan. Ovo korisnika vodi na stranicu: https://demo.opencart.com/index.php?route=product/product&language=en-gb&product_id=42
- Pregleda se description i specifications za uređaj
- Pregledaju se sve slike dostupne za uređaj
- Uređaj se doda u wish list klikom na odgovarajuće dugme
- Vraćanje na početnu stranicu klikom na logo. Ovo vodi na stranicu <https://demo.opencart.com/index.php?route=common/home&language=en-gb>
- Otvaranje detalja o drugom uređaju iz liste dostupnih. Ovo vodi na stranicu https://demo.opencart.com/index.php?route=product/product&language=en-gb&product_id=40
- Dodavanje ovog uređaja u compare listu klikom na odgovarajuće dugme
- Povratak na početnu stranicu klikom na logo
- Otvaranje detalja o prvom uređaju iz liste dostupnih. Otvara se stranica https://demo.opencart.com/index.php?route=product/product&language=en-gb&product_id=43
- Uređaj se dodaje u compare listu klikom na odgovarajuće dugme
- Klikom na link koji se pojavi u pop-up prozoru ide se u compare list. Ovim se otvara stranica <https://demo.opencart.com/index.php?route=product/compare&language=en-gb>
- Oba proizvoda koja su upravo dodana u compare listu dodaju se u korpu klikom na odgovarajuće dugme
- Klikom na odgovarajuće dugme iz menija otvara se korpa. Ovo vodi na stranicu <https://demo.opencart.com/index.php?route=checkout/cart&language=en-gb>
- Poveća se količina prvom uređaju u korpi unosom odgovarajuće vrijednosti u polje Quantity
- Klikne se dugme za osvježavanje promjena koje se nalazi pored uređaja
- Ukloni se drugi uređaj iz korpe klikom na odgovarajuće dugme

Rezultati testiranja:

Planirani scenario:



Osnovne statističke informacije nakon izvršenog testiranja prikazane su na sljedećem screenshotu.

DURATION 0:05:24	BOTS 25	ITERATIONS 4	HITS 8.555	ERRORS 221
AVG RESPONSE TIME 0.200s	P50 RESPONSE TIME 0.095s	P90 RESPONSE TIME 0.538s	DOWNLOADED 106.7 MB	UPLOADED 4.6 MB

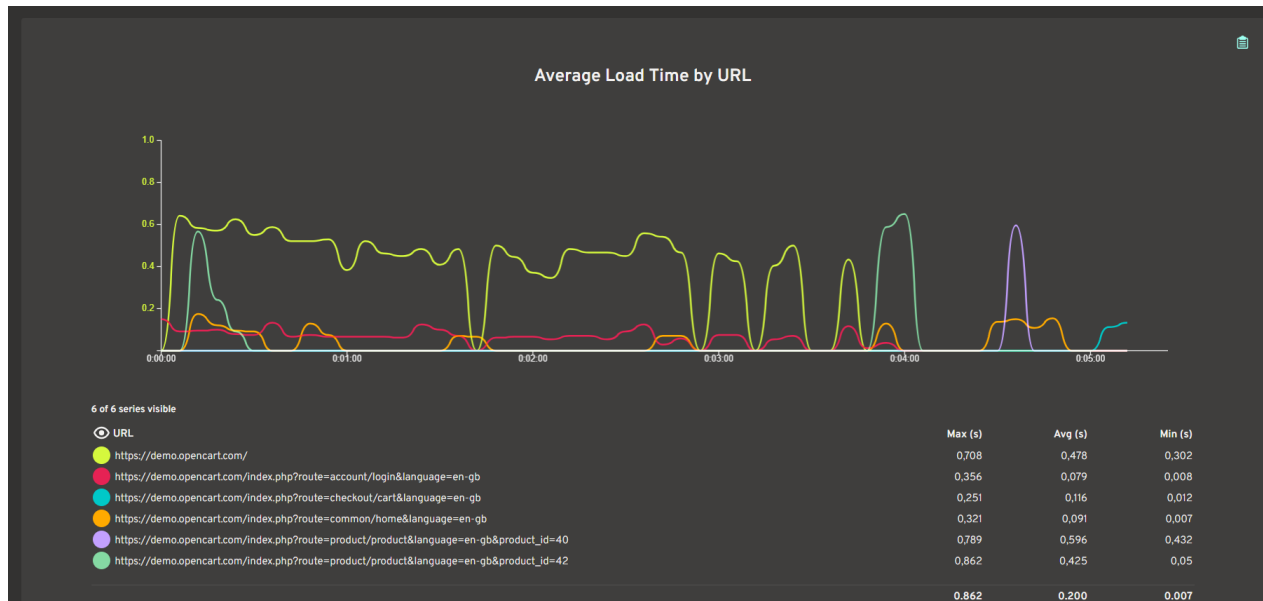
Vidimo da se test izvršavao ukupno 5 minuta.

Prosječno vrijeme odgovora je 0.2s

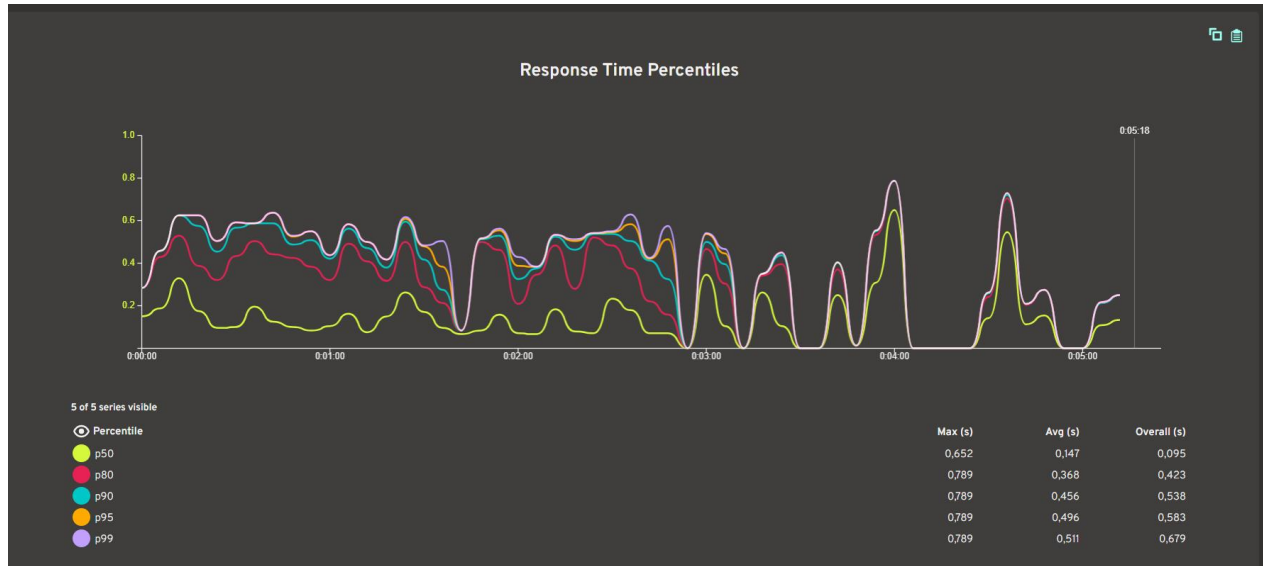
P90 response time je 0.538s što znači da 10% najsporijih zahtjeva ima vrijeme odgovora duže od ovoga.

P50 govori da 50% najsporijih zahtjeva ima vrijeme odgovora duže od navedenog broja.

RESPONSE TIMES



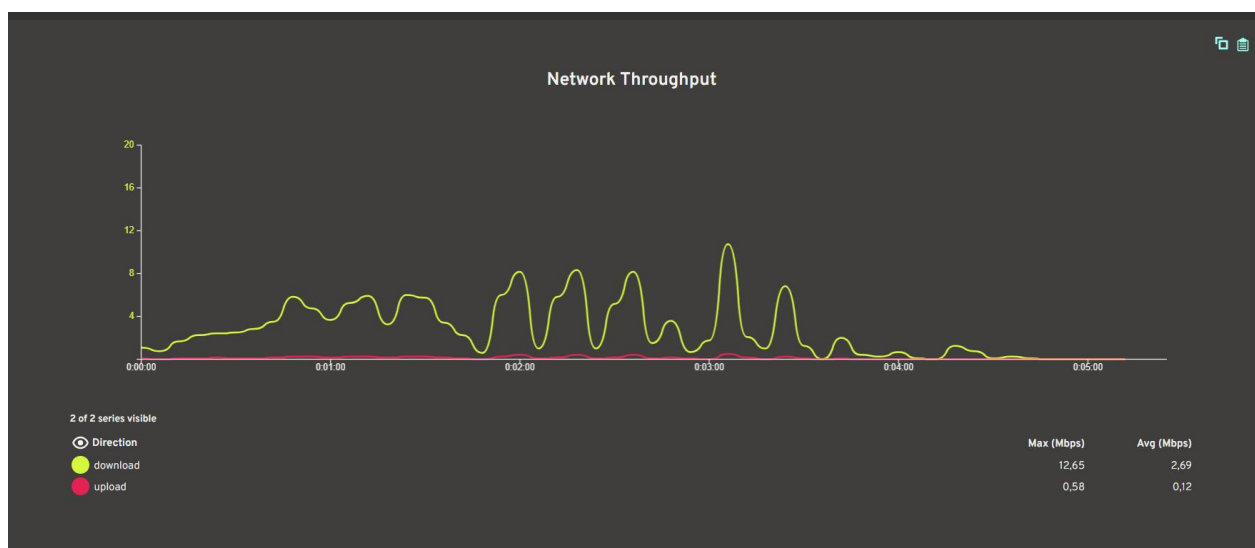
Na grafu iznad prikazano je vrijeme učitavanja po linkovima stranica. Vidimo da je maksimalno vrijeme izvršavanja 0.862s, minimalno 0.007s. Prosjek koji imamo od 0.200s je zadovoljavajući. Niti jedna stranica nema vrijeme učitavanja duže od 1s.



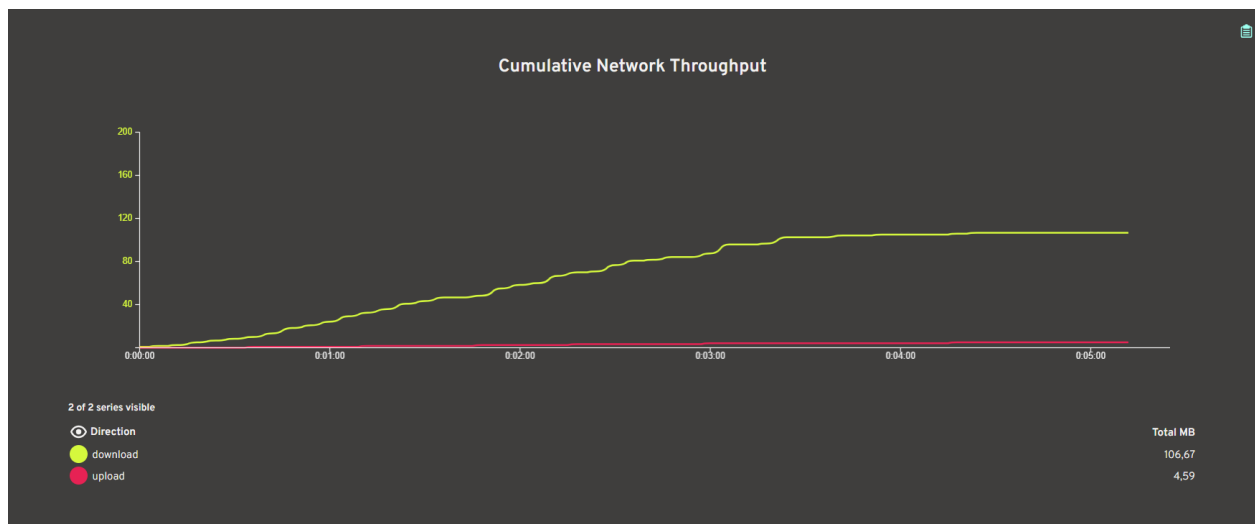
Osvrnut ćemo se na P90 response time, dok se analogno može zaključiti i za ostale procenete. Imamo situaciju da je prosječan P90 0.456s, što znači da 10% najsporijih zahtjeva ima vrijeme odgovora duže od 0.456s, ali ipak 90% zahtjeva će odgovoriti za $\leq 0.456s$ što je ponovno zadovoljavajuća situacija.

Total Time Spent			
<div> <div>Q URL</div> <div>Filter by URL</div> </div>			
URL ^	Total (s)	Avg (s)	Request Count
https://demo.opencart.com/	107.11	0.478	224
https://demo.opencart.com/index.php?route=account/login&language=en-gb	36.97	0.079	465
https://demo.opencart.com/index.php?route=checkout/cart&language=en-gb	0.928	0.116	8
https://demo.opencart.com/index.php?route=common/home&language=en-gb	10.23	0.091	112
https://demo.opencart.com/index.php?route=product/product&language=en-gb&product_id=40	4.77	0.596	8
https://demo.opencart.com/index.php?route=product/product&language=en-gb&product_id=42	6.81	0.425	16
	166.82	0.200	833

NETWORK

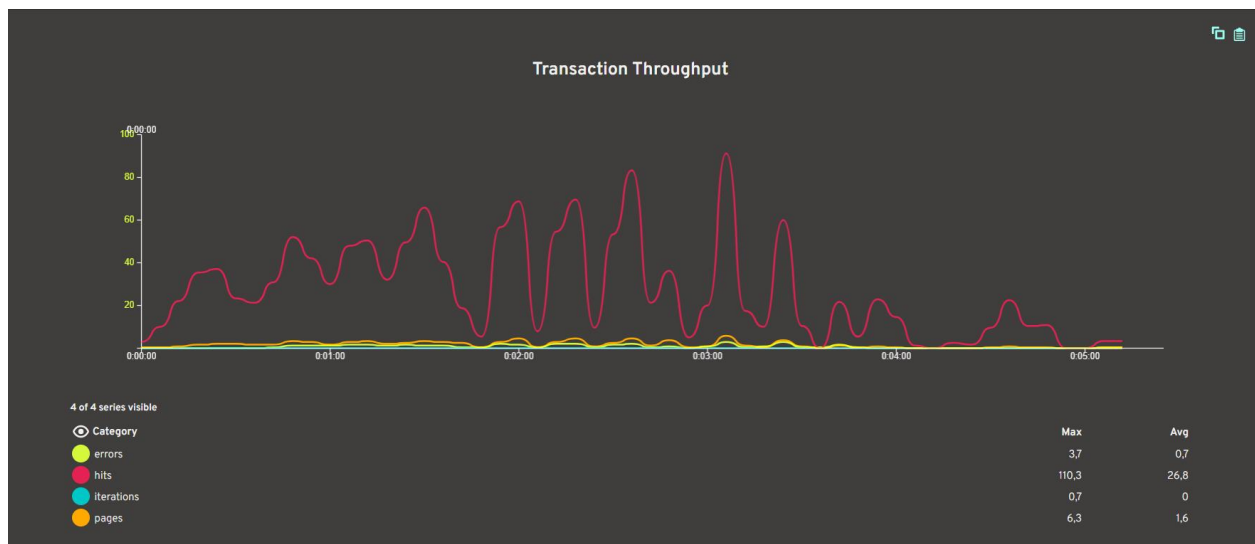


Za download podataka, najveća brzina koja je postignuta za vrijeme izvršavanja je bila 12,65Mbps, a minimalna 2.69Mbps. Vidimo ne tako malu varijaciju između minimalne i maksimalne vrijednosti što ukazuje na varijacije u kvalitetu mrežnog prenosa tokom testiranja.

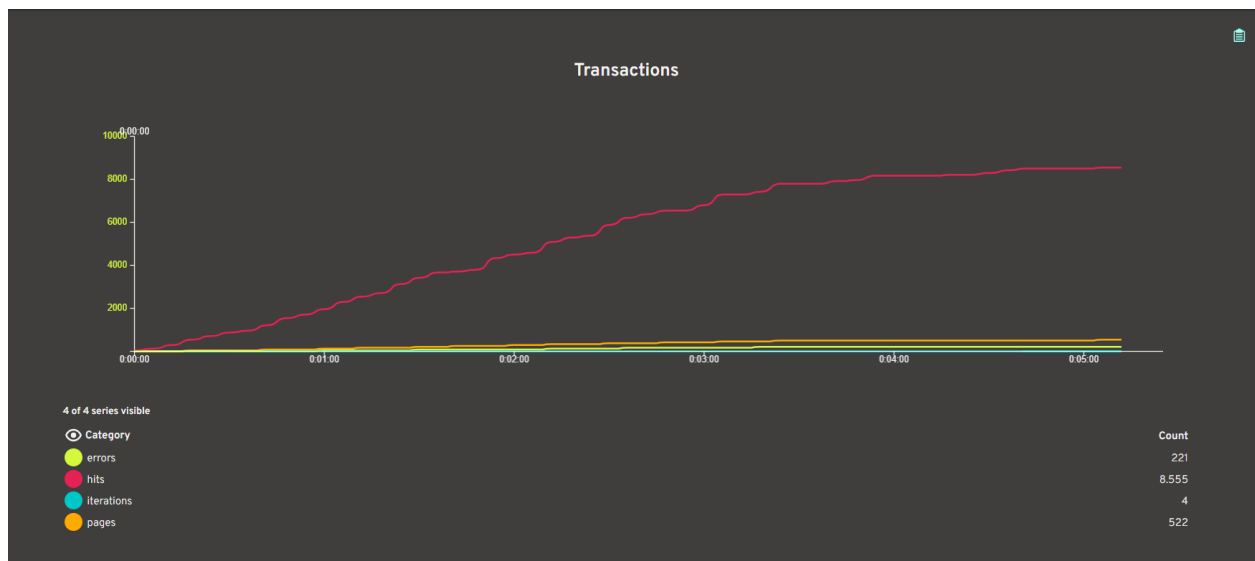


Kumulativna propusnost mreže linearno raste do otprilike treće minute testiranja, a upravo toliko dugo i traje proces slanja zahtjeva, a nakon 3 minute botovi postepeno napuštaju aplikaciju. Ovo ukazuje na dobru skalabilnost sistema u pogledu mrežne propusnosti pod povećanim opterećenjem.

TRANSACTIONS



Na ovom grafu vidimo kako se kroz vrijeme mijenja broj transakciju u sistemu. Najviše transakcija koje su se dogodile odnose se na hits, odnosno na zahtjeve koje web poslužitelj prima od korisnika. Vidimo da je maksimalan broj zahtjeva u jednom momentu 110 otprilike.



Broj transakcija linearno raste u pogledu korisničkih zahtjeva, što ponovo ukazuje na linearnu skalabilnost u odnosu na povećanje opterećenja, što je pozitivna stvar.

Zaključak je da pri targetiranju sistema sa 25 botova u trajanju od otprilike 3 minute, gdje se šalju jednostavni zahtjevi, sistem ima zadovoljavajuće performanse.

Stress testiranje

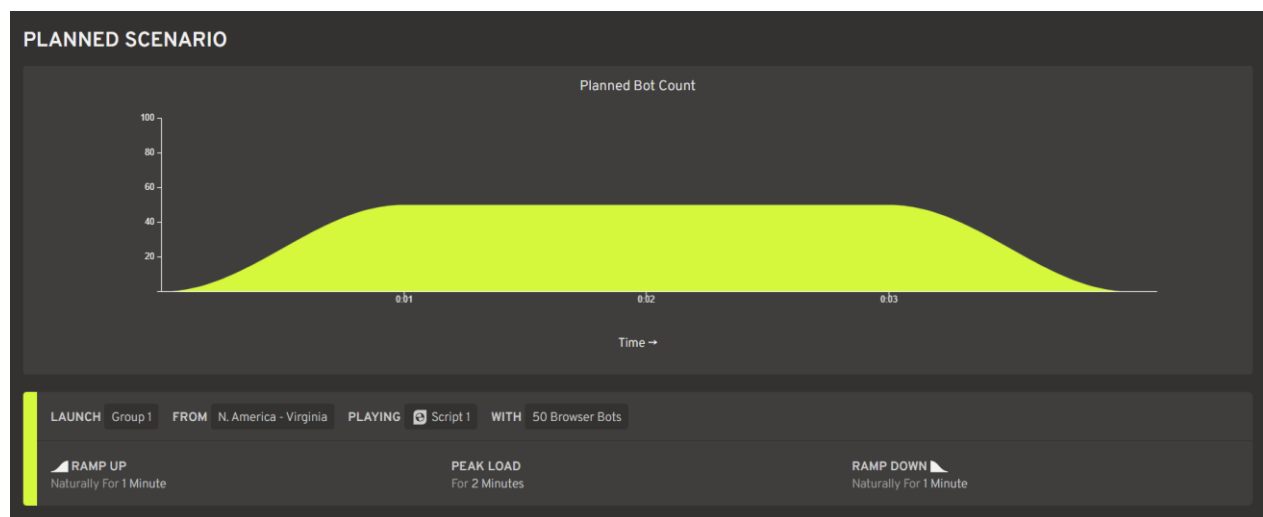
Za izvršenje stress testiranja iskoristili smo testni scenarij u kojem korisnik obavlja više zahtjeva nego što je uobičajeno da korisnik šalje pri jednom kraćem korištenju aplikacije. Cilj je analizirati ponašanje stranice pri opterećenju većem od onog koje se može očekivati u normalnim uslovima. Aplikaciju smo targetirali sa 50 botova. Dodatne informacije o vremenu provedenom na stranici i rezultatima testiranja će biti prikazane ispod. Detaljan testni scenario koji smo koristili za ovaj tip testiranja je sljedeći:

- Otvaranje stranice <https://demo.opencart.com/>
- Login u aplikaciju
- Klikom na logo stranice navigira se na početnu stranicu
- U kratkom vremenskom rasponu između pojedinih akcija koje će se nabrojati vrši se sljedeće:
otvaranje detalja o prvom uređaju (otvara se stranica https://demo.opencart.com/index.php?route=product/product&language=en-gb&product_id=43) -> vraćanje na početnu stranicu klikom na logo (otvara se stranica <https://demo.opencart.com/index.php?route=common/home&language=en-gb>) -> otvaranje detalja o drugom uređaju (https://demo.opencart.com/index.php?route=product/product&language=en-gb&product_id=40) -> klik na logo za vraćanje na početnu -> otvaranje detalja o trećem uređaju (https://demo.opencart.com/index.php?route=product/product&language=en-gb&product_id=42) -> klik na logo za povratak na početnu -> otvaranje detalja o četvrtom uređaju (https://demo.opencart.com/index.php?route=product/product&language=en-gb&product_id=30) -> klik na logo za povratak na početnu
- Prva 2 proizvoda dodati u korpu

- Sve prikazane proizvode (4) dodati u wish listu
- Prva 2 proizvoda još po 2 puta dodati u korpu
- Sve proizvode koji su prikazani dodati u compare listu
- Otvoriti korpu klikom na dugme iz menija (otvara se <https://demo.opencart.com/index.php?route=checkout/cart&language=en-gb>)
- Povećati količinu prvog proizvoda unosom odgovarajuće vrijednosti u polje Quantity i klikom na dugme za osvježavanje promjena
- Povećati količinu drugog proizvoda
- Povećati količinu prvog proizvoda
- Ukloniti drugi proizvod iz korpe klikom na odgovarajuće dugme
- Otvoriti wish listu klikom na dugme iz menija (otvara se https://demo.opencart.com/index.php?route=account/wishlist&language=en-gb&customer_token=20d967d2cab78998735a5b0b0)
- Ukloniti sve proizvode iz wish liste klikom na odgovarajuće dugme.

Rezultati testiranja:

Planirani scenario:



Osnovne statističke informacije nakon izvršenog testiranja prikazane su na sljedećem screenshotu.

DURATION 0:04:14	BOTS 50	ITERATIONS 0	HITS 70.035	ERRORS 2.415
AVG RESPONSE TIME 0.723s	P50 RESPONSE TIME 0.695s	P90 RESPONSE TIME 1.03s	DOWNLOADED 1.0 GB	UPLOADED 44.6 MB

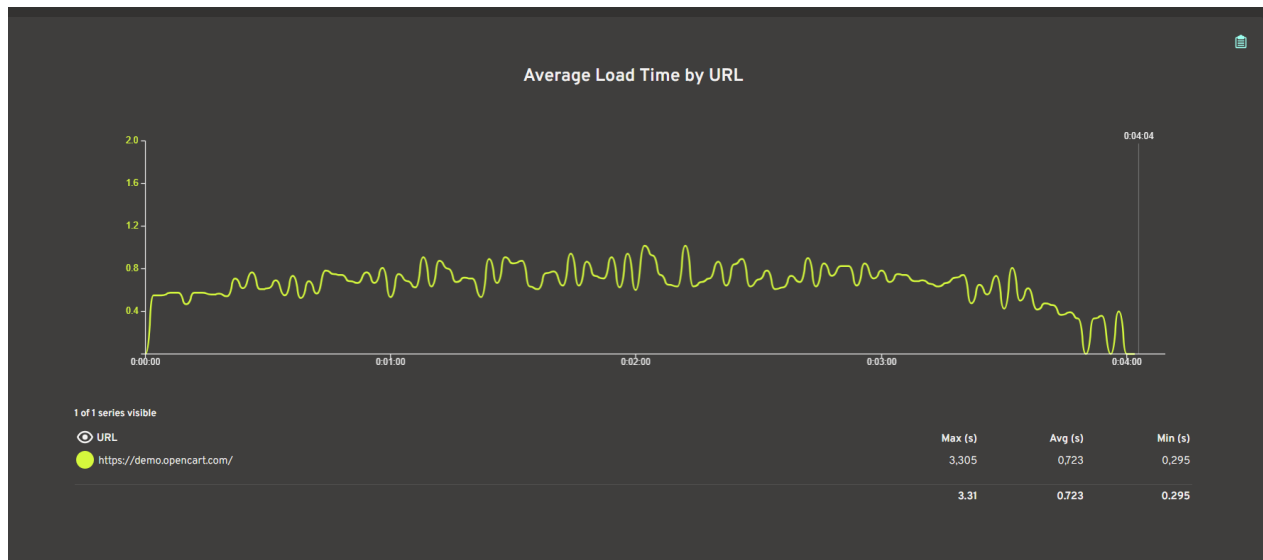
Vidimo da se test izvršavao ukupno 4 minute.

Prosječno vrijeme odgovora je 0.723s.

P90 response time je 1.03s što znači da 10% najsporijih zahtjeva ima vrijeme odgovora duže od ovoga.

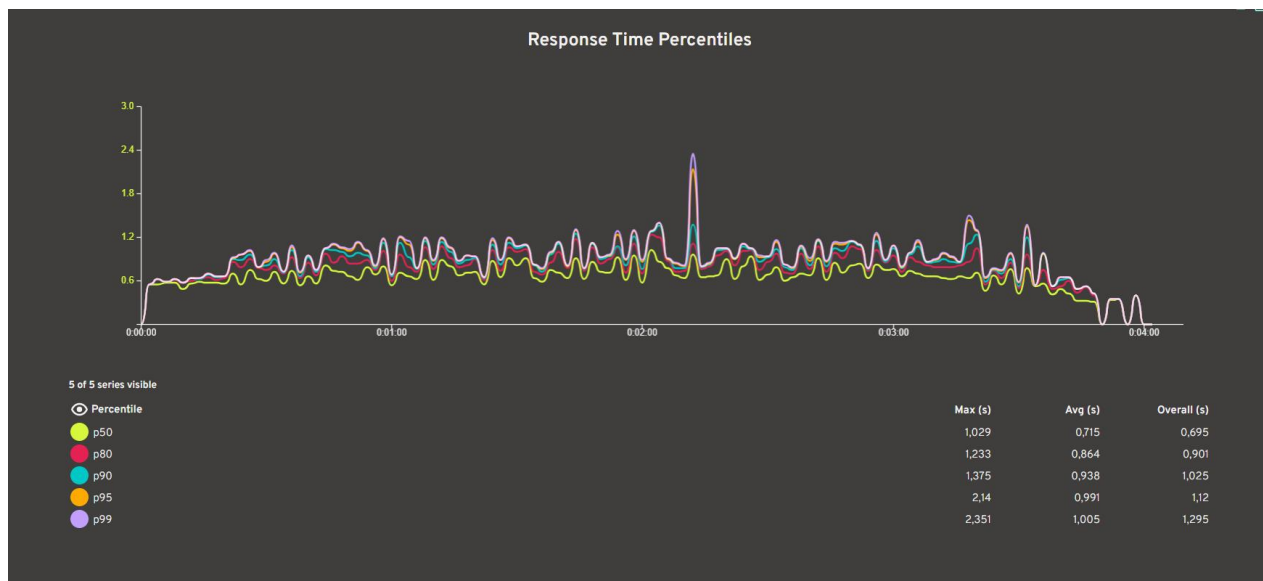
P50 govori da 50% najsporijih zahtjeva ima vrijeme odgovora duže od navedenog broja.

RESPONSE TIMES



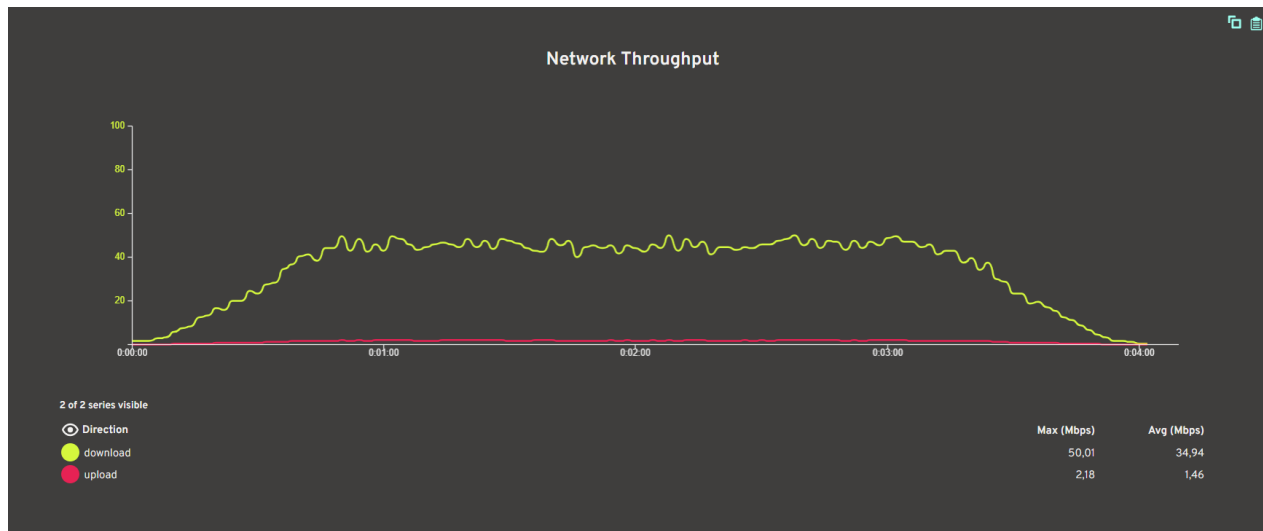
Na ovom grafu vidimo prikaz prosječnog vremena učitavanja po stranicama. Međutim, informacije su prikazane samo za početnu stranicu, iako je više stranica trebalo da bude otvarano ovim testom. Dakle, stranica nije izdržala ovakav test, iako se ne radi baš o prevelikom opterećenju.

Ipak su se zahtjevi nastavili slati prema stranici, iako nisu mogli biti uspješno obrađeni, pa ćemo svakako razmotriti ostale grafove.

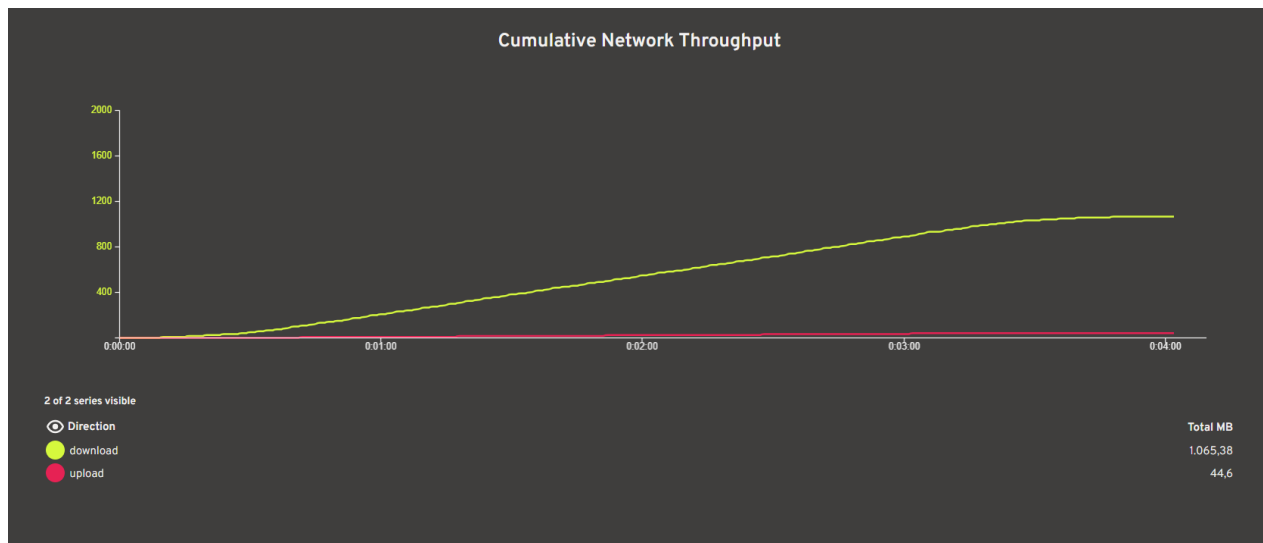


Što se tiče prosječnih vremena odgovora, vidimo da u najgorem slučaju, 1% najsporijih zahtjeva može imati vrijeme odgovora od čak 2,351s, a u prosjeku 1.005s. Što se tiče neke prosječne situacije, 20% najsporijih zahtjeva ima prosječno vrijeme odgovora duže od 0.864s.

NETWORK

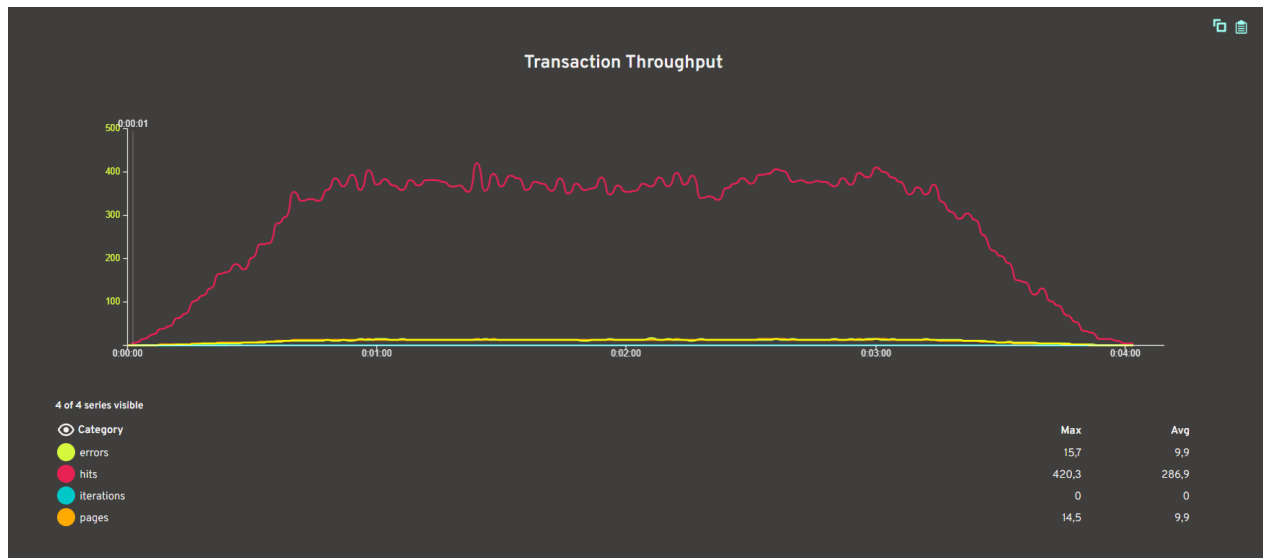


Maksimalna propusnost mreže koja je bila ostvarena pri downloadu je 50Mbps, a minimalna 34.96Mbps.

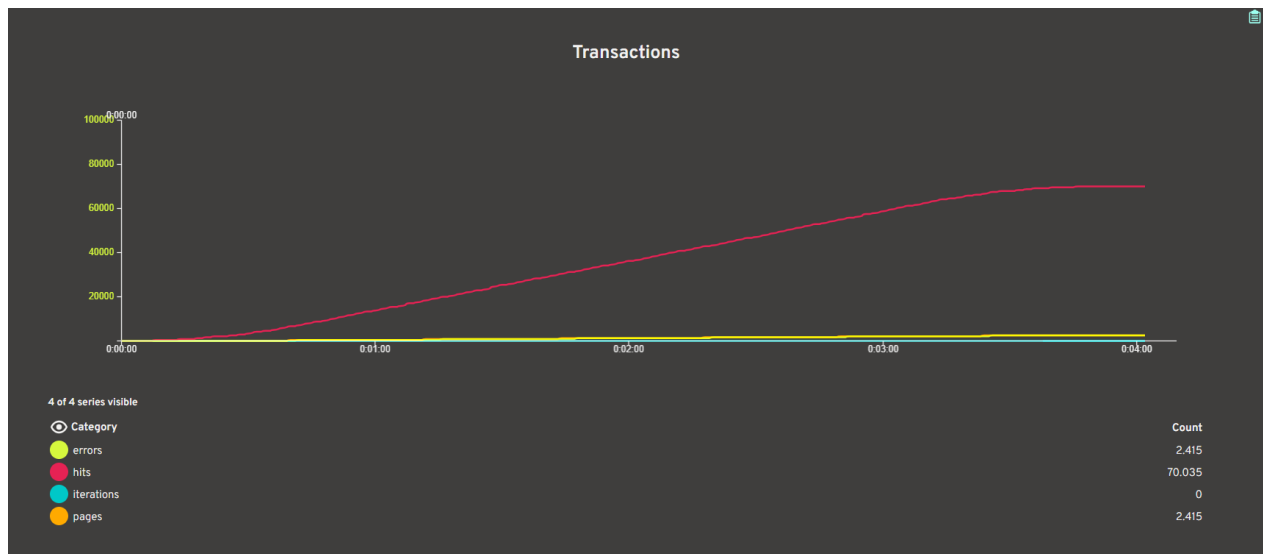


Što se tiče kumulativne mrežne propusnosti, opet imamo linearan rast i ukupno otprilike 1GB podataka prenesenih preko mreže za download.

TRANSACTIONS



Što se tiče propusnosti transakcija, vidimo da je najveći broj transakcija u jednom momentu bio 420 i radi se o kruisničkim zahtjevima.



Vidimo linearan rast transakcija.

Generalni zaključak je da pri ovakvom stress testiranju aplikacija ne može da podnese opterećenje kojem je izložena. Ostale statističke vrijednosti analizirane su s ciljem da se vidi kako se aplikacija nosi sa pristizanjem velike količine zahtjeva, iako nije previše mjerodavno jer ne uspijeva da ih obradi, tako da uglavnom vraća odgovore o neuspješno obrađenim zahtjevima, ali svejedno analizirana je komunikacija sa stranicom pri velikom opterećenju.