



# Napredne Veb Tehnologije

## Testiranje – Student 2

Erdelji Marko

SV49/2020

## Contents

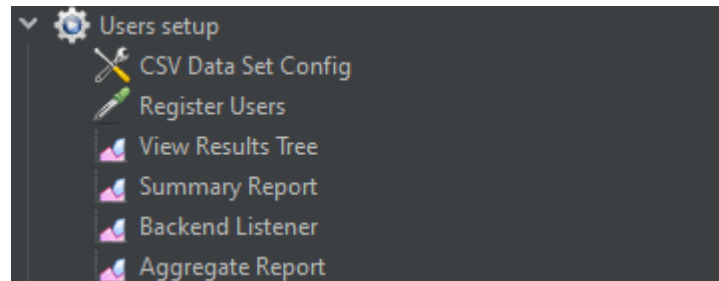
1. Incijalni setup .....	3
2. Property registration .....	4
3. Admin accept property .....	5
4. Admin reject property.....	6
5. Smartdevice setup .....	7
.....	8
6. Smartdevice availability info .....	9
7. Lamp statistics info.....	10
8. Turning on lamp .....	<b>Error! Bookmark not defined.</b>
9. Car gate history .....	12
10. Opening car gate .....	13
11. Sprinkler history .....	14
12. Turning sprinkler on .....	15
13. Obzervacije.....	16

# 1. Incijalni setup

Da bi se testirala naša veb aplikacija potrebno je prvobitno setapovati test bazu u kojoj ce se cuvati podaci potrebni za pozivanje testnih slučajeve za naš load test. Postoje dva setapa koja se prvobitno trebaju odraditi da bismo odradili prva 3 scenarija:

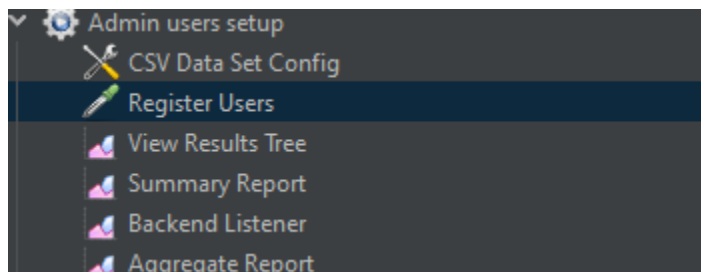
## 1. Postavljanje registracije korisnika:

- Kroz 100 niti, prolazi se kroz generisani CSV fajl (kreiran pomoću Faker biblioteke i Python skripte).
- Za svaku nit, uzima se po red kako biste kreirali običnog korisnika.
- Poziva se endpoint za registraciju i registruje se 100 testnih korisnika.
- Postavljeni su listeneri za proveru uspešnosti inicijalizacije.



## 2. Postavljanje registracije administratora:

- Kroz 10 niti, prolazi se kroz generisani CSV fajl (koristeći prethodno pomenute alate).
- Za svaku nit, uzima se po red kako biste kreirali administratorskog korisnika.
- Poziva se endpoint za registraciju i registruje se 10 testnih admin korisnika.
- Postavljeni su listeneri za proveru uspešnosti inicijalizacije.

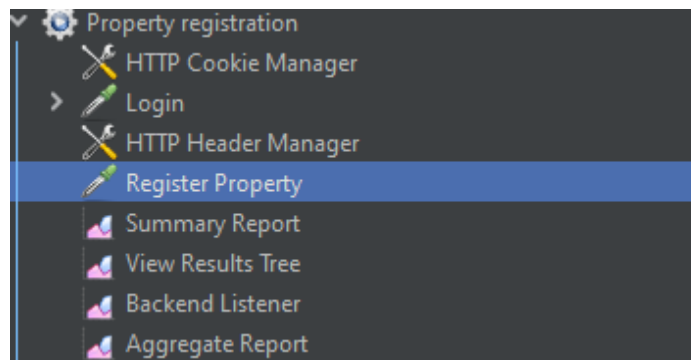


## 2. Property registration

Koristeći prethodno pomenuti csv fajl za korisnike, posle setupovanja korisnika u bazu koriste se već registrovani korisnici da bi se testiralo opterećenje na registraciji propertija.

**Prolazi se kroz sledeći pipeline za testiranje propertija:**

- Kroz N niti se prolazi kroz csv i zatim pogadja login endpoint
- Korišćenjem HTTP cookie managera se preuzima cookie za odredjeni thread u zavisnosti od logina
- Za svakog od usera se poziva endpoint za registraciju propertija.
- Korišćenjem raznih listenera kojim JMeter raspolaže zatim prikupljamo rezultate poziva endpointa



**Rezultati:**

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Register Property	643	3527	5	60013	6519.89	0.93%	4.6/sec	1.04	28.56	231.6
Login	612	5251	401	60028	5610.23	0.33%	4.3/sec	4.84	1.07	1139.7
TOTAL	1255	4367	5	60028	6153.93	0.64%	8.9/sec	5.87	29.39	674.4

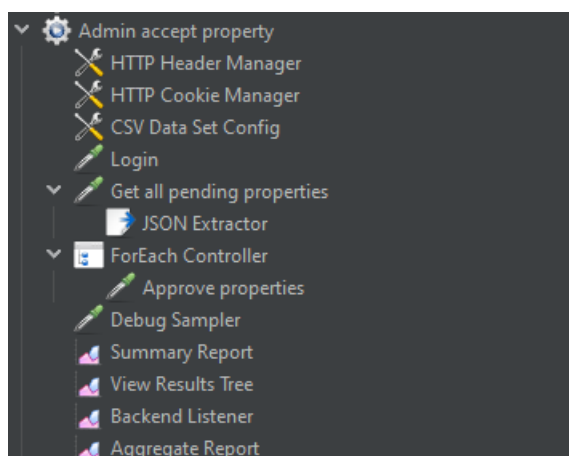
Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/...	Sent KB/sec
Login	1800	5018	4397	6061	6707	33268	340	60038	0.67%	6.0/sec	6.71	1.49
Register Pro...	1800	3346	2611	4499	5395	32393	5	60043	1.28%	6.1/sec	1.37	37.55
TOTAL	3600	4182	3584	5692	6322	32787	5	60043	0.97%	12.1/sec	8.07	38.92

### 3. Admin accept property

Koristeći prethodno pomenuti csv fajl za admin korisnike, posle setupovanja admin korisnika u bazu i posle pozivanja testnog slučaja za registraciju propertyja, koriste se već registrovani admin korisnici i pending propertyja da bi se testiralo opterećenje na prihvatanju propertyja od strane administratora.

**Prolazi se kroz sledeći pipeline za testiranje prihvatanja propertyja:**

- Kroz N niti se prolazi kroz csv i zatim pogadja login endpoint
- Korišćenjem HTTP cookie managera se preuzima cookie za određeni thread u zavisnosti od logina.
- Svaki admin user zatim dobavlja sve pending propertyje perthodno kreiranje korišćenjem testnog slučaja za registraciju propertyja.
- Korišćenjem ForEach controller-a se redom za svaki property poziva endpoint za prihvatanje propertyja
- Korišćenjem raznih listenera kojim JMeter raspolaže zatim prikupljamo rezultate poziva endpointa



Rezultati:

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/...	Sent KB/sec
Login	100	4186	4817	5213	5259	6525	1741	6745	0.00%	13.0/sec	14.50	3.22
Get all pendi...	100	1709	1106	3723	3786	3870	507	3871	0.00%	15.7/sec	57.60	13.22
Approve pro...	473	1580	1554	1989	2097	2200	729	2658	0.42%	47.6/sec	12.57	42.70
TOTAL	673	1986	1606	4450	4953	5234	507	6745	0.30%	43.3/sec	38.77	34.27

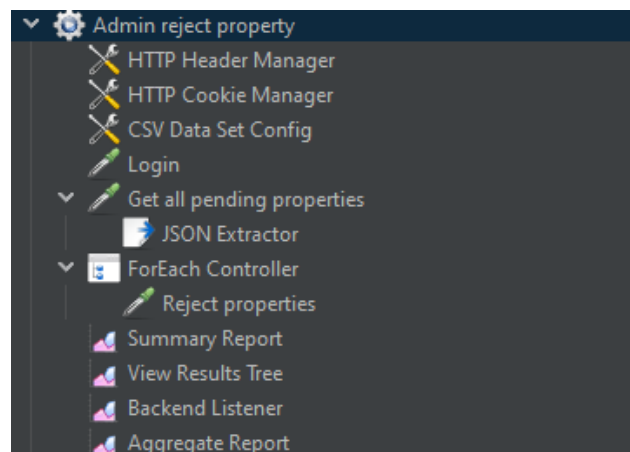
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Login	100	4186	1741	6745	1335.36	0.00%	13.0/sec	14.50	3.22	1140.1
Get all pending ...	100	1709	507	3871	1170.26	0.00%	15.7/sec	57.60	13.22	3757.0
Approve proper...	473	1580	729	2658	304.36	0.42%	47.6/sec	12.57	42.70	270.2
TOTAL	673	1986	507	6745	1174.75	0.30%	43.3/sec	38.77	34.27	917.6

## 4. Admin reject property

Koristeći prethodno pomenuti csv fajl za admin korisnike, posle setupovanja admin korisnika u bazu i posle pozivanja testnog slučaja za registraciju propertyja, koriste se već registrovani admin korisnici i pending propertyja da bi se testiralo opterećenje na odbijanju propertyja od strane administratora.

### Prolazi se kroz sledeći pipeline za testiranje odbijanja propertyja:

- Kroz N niti se prolazi kroz csv i zatim pogadja login endpoint
- Korišćenjem HTTP cookie managera se preuzima cookie za određeni thread u zavisnosti od logina.
- Svaki admin user zatim dobavlja sve pending propertyje prethodno kreiranje korišćenjem testnog slučaja za registraciju propertyja.
- Korišćenjem ForEach controller-a se redom za svaki property poziva endpoint za odbijanje propertyja
- Korišćenjem raznih listenera kojim JMeter raspolaže zatim prikupljamo rezultate poziva endpointa



### Rezultati:

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
Login	100	3506	3579	5051	5193	6281	1991	6285	0.00%	13.7/sec	15.31	3.40
Get all pending p...	100	2416	2284	3670	3691	3709	800	3714	0.00%	16.6/sec	60.74	13.95
Reject properties	1000	1426	1472	1508	1518	1535	97	1598	0.00%	63.1/sec	15.35	58.59
TOTAL	1200	1682	1478	2261	3662	5010	97	6285	0.00%	55.7/sec	33.49	48.15

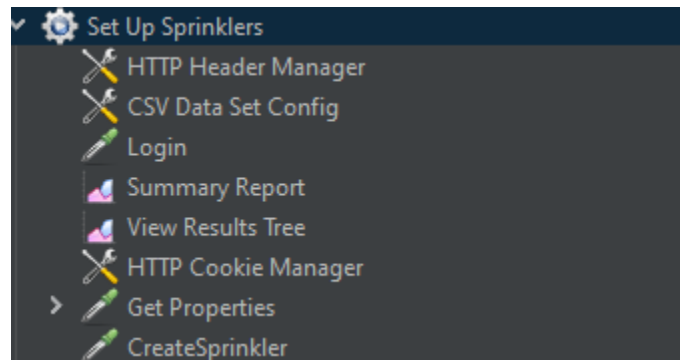
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Login	100	3506	1991	6285	1187.45	0.00%	13.7/sec	15.31	3.40	1140.1
Get all pending prop...	100	2416	800	3714	1042.92	0.00%	16.6/sec	60.74	13.95	3755.0
Reject properties	1000	1426	97	1598	138.85	0.00%	63.1/sec	15.35	58.59	249.0
TOTAL	1200	1682	97	6285	775.37	0.00%	55.7/sec	33.49	48.15	615.4

## 5. Smartdevice setup

Da bi se testirali ostali test scenariji potrebno je setapovati test bazu sa dodatnim sprinklerima, cagate-ovima i lampama. Postoje tri setapa za svaki od uređaja, koja se trebaju odraditi da bismo odradili ostale scenarije:

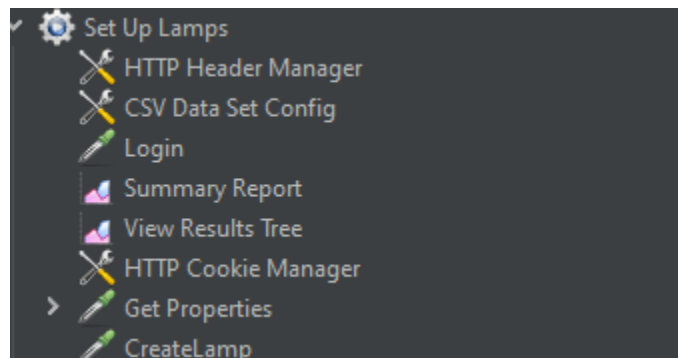
### 1. Inicijalizacija sprinklera:

- Kroz N niti se prolazi kroz csv I zatim pogađa login endpoint
- Korišćenjem HTTP cookie managera se preuzima cookie za određeni thread u zavisnosti od logina.
- Poziva se endpoint za dobavljanje userovih proprijetija.
- Ako user poseduje property dodaje se sprinkler u njega.



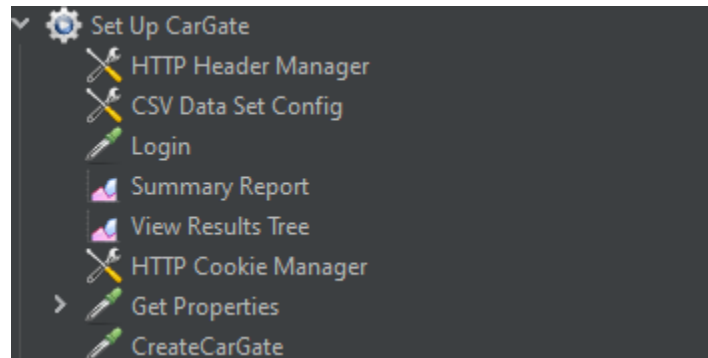
### 2. Inicijalizacija lampi:

- Kroz N niti se prolazi kroz csv I zatim pogađa login endpoint
- Korišćenjem HTTP cookie managera se preuzima cookie za određeni thread u zavisnosti od logina.
- Poziva se endpoint za dobavljanje userovih proprijetija.
- Ako user poseduje property dodaje se lampa u njega.



### 3. Inicijalizacija car gate-ova:

- Kroz N niti se prolazi kroz csv I zatim pogadja login endpoint
- Korišćenjem HTTP cookie managera se preuzima cookie za odredjeni thread u zavisnosti od logina.
- Poziva se endpoint za dobavljanje userovih propertija.
- Ako user poseduje property dodaje se car gate u njega.



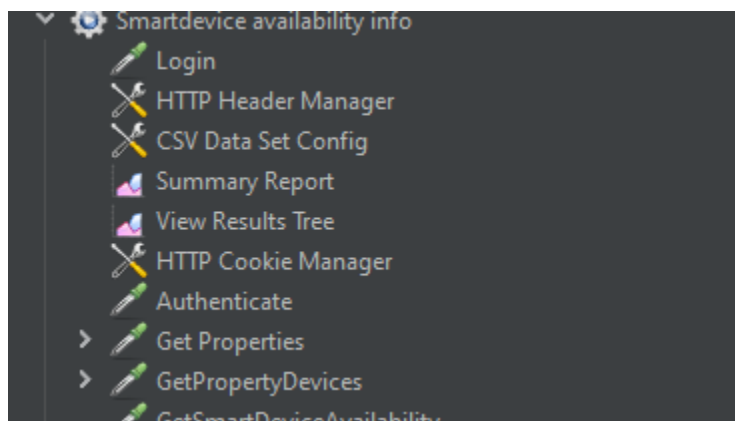


## 6. Smartdevice availability info

Koristeći prethodno pomenuti csv fajl za korisnike, posle setupovanja korisnika u bazu I posle pozivanja testnog slučaja za registraciju propertiija I approvovanje I posle setapovanja smartdevice-ova, koriste se već registrovani korisnici, propertiiji I device-ovi da bi se testiralo opterećenje na dobavljanju dostupnosti korisnikovog uređaja.

### Prolazi se kroz sledeći pipeline za testiranje dostupnosti uređaja:

- Kroz N niti se prolazi kroz csv I zatim pogadja login endpoint
- Korišćenjem HTTP cookie managera se preuzima cookie za odredjeni thread u zavisnosti od logina.
- Svaki user zatim dobavlja sve svoje propertije prethodno kreirane korišćenjem testnog slučaja za registraciju propertiija.
- Svaki user zatim dobavlja sve svoje uređaje prethodno kreirane korišćenjem setup-a za kreiranje uređaja.
- Za prvi nadjeni uređaj poziva se endpoint za dobavljanje dostupnosti.



### Rezultati:

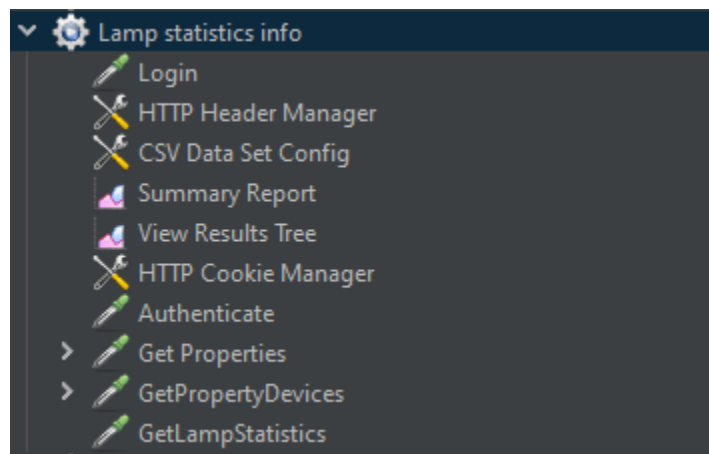
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Login	1000	5216	407	60066	10770.50	2.90%	4.2/sec	4.58	3.55	1127.7
Authenticate	1000	4064	5	60064	13381.46	4.70%	4.3/sec	1.39	3.61	330.1
Get Properties	1000	3733	17	60063	12225.94	3.70%	4.3/sec	2.51	3.61	595.1
GetPropertyDev...	1000	3510	17	60117	11864.97	6.90%	4.3/sec	15.68	3.83	3719.8
GetSmartDevic...	1000	2224	10	60062	9298.42	8.70%	4.3/sec	0.94	4.01	223.5
TOTAL	5000	3750	5	60117	11631.11	5.38%	20.8/sec	24.32	18.04	1199.2

## 7. Lamp statistics info

Koristeći prethodno pomenuti csv fajl za korisnike, posle setupovanja korisnika u bazu i posle pozivanja testnog slučaja za registraciju propertiija i approvovanje i posle setapovanja lampi, koriste se već registrovani korisnici, propertiiji i device-ovi da bi se testiralo opterećenje na dobavljanju statistike korisnikove lampe.

### Prolazi se kroz sledeći pipeline za testiranje statistike lampe:

- Kroz N niti se prolazi kroz csv i zatim pogadja login endpoint
- Korišćenjem HTTP cookie managera se preuzima cookie za odredjeni thread u zavisnosti od logina.
- Svaki user zatim dobavlja sve svoje propertiije prethodno kreirane korišćenjem testnog slučaja za registraciju propertiija.
- Svaki user zatim dobavlja sve svoje uređaje prethodno kreirane korišćenjem setup-a za kreiranje uređaja.
- Za prvu nadjenu lampu poziva se endpoint za dobavljanje statistike.



### Rezultati:

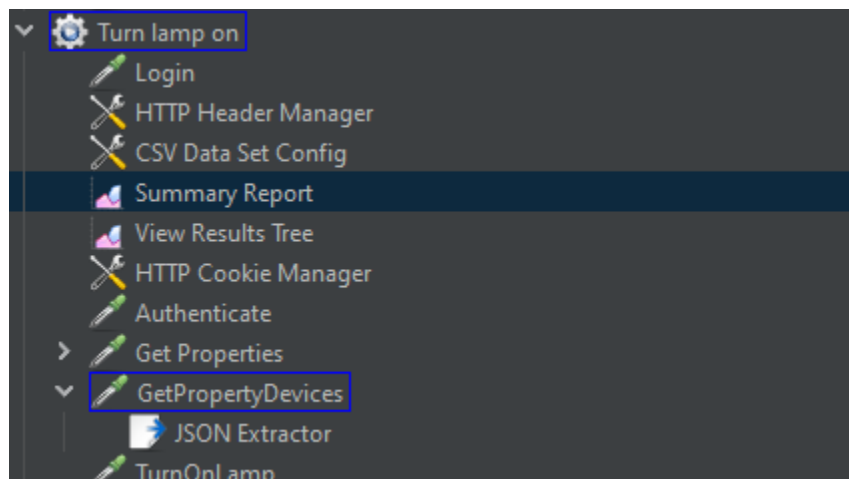
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Login	1000	4629	338	60058	9335.70	2.60%	5.2/sec	5.74	4.46	1129.1
Authenticate	1000	2182	4	60046	9842.01	2.70%	5.3/sec	1.68	4.47	323.6
Get Properties	1000	2522	15	60056	10083.42	2.90%	5.4/sec	3.14	4.53	596.5
Debug Sampler	2000	0	0	2	0.37	0.00%	10.8/sec	4.72	0.00	447.3
GetPropertyDevices	1000	2476	6	60060	10103.40	5.60%	5.4/sec	19.85	4.81	3763.2
GetLampStatistics	1000	2743	4	60057	10791.11	8.60%	5.4/sec	1.19	5.00	225.0
TOTAL	7000	2079	0	60060	8620.40	3.20%	36.4/sec	35.20	22.62	990.3

## 8. Turn lamp on

Koristeći prethodno pomenuti csv fajl za korisnike, posle setupovanja korisnika u bazu i posle pozivanja testnog slučaja za registraciju svojih uređaja i aprobovanje i posle setupovanja lampi, koriste se već registrovani korisnici, svojih uređaja i lampi da bi se testiralo opterećenje na paljenju korisničkih lampi.

### Prolazi se kroz sledeći pipeline za testiranje paljenja lampi:

- Kroz N niti se prolazi kroz csv i zatim pogadja login endpoint
- Korišćenjem HTTP cookie managera se preuzima cookie za određeni thread u zavisnosti od logina.
- Svaki user zatim dobavlja sve svoje svojstva prethodno kreirane korišćenjem testnog slučaja za registraciju svojih uređaja.
- Svaki user zatim dobavlja sve svoje uređaje prethodno kreirane korišćenjem setup-a za kreiranje uređaja.
- Za prvu nadjenu lampu poziva se endpoint za paljenje lampi.



Rezultati:

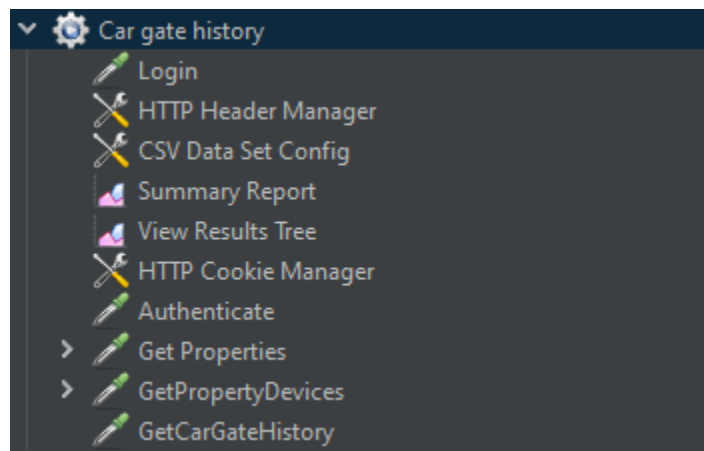
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Login	1000	5338	299	60054	12388.32	4.80%	4.4/sec	4.76	3.73	1119.1
Authenticate	1000	3238	4	60069	12616.14	4.60%	4.5/sec	1.45	3.78	330.7
Get Properties	1000	2690	13	60029	10990.96	3.50%	4.5/sec	2.62	3.77	597.0
GetPropertyDevices	1000	2587	7	60059	10811.61	6.70%	4.5/sec	16.43	4.02	3728.9
TurnOnLamp	1000	2721	6	60056	11175.76	10.20%	4.5/sec	0.76	4.11	171.7
TOTAL	5000	3315	4	60069	11667.05	5.96%	21.7/sec	25.26	18.86	1189.5

## 9. Car gate history

Koristeći prethodno pomenuti csv fajl za korisnike, posle setupovanja korisnika u bazu i posle pozivanja testnog slučaja za registraciju properti i aprobovanje i posle setapovanja cagate-ova, koriste se već registrovani korisnici, properti i cagate-ovi da bi se testiralo opterećenje na dobavljanju istorije korisnikovog car gate-a.

**Prolazi se kroz sledeći pipeline za testiranje dobavljanja istorije cagate-a:**

- Kroz N niti se prolazi kroz csv i zatim pogadja login endpoint
- Korišćenjem HTTP cookie managera se preuzima cookie za određeni thread u zavisnosti od logina.
- Svaki user zatim dobavlja sve svoje propertije prethodno kreirane korišćenjem testnog slučaja za registraciju properti.
- Svaki user zatim dobavlja sve svoje uređaje prethodno kreirane korišćenjem setup-a za kreiranje uređaja.
- Za prvi nadjeni car gate poziva se endpoint za dobavljanje istorije car gate-a.



Rezultati:

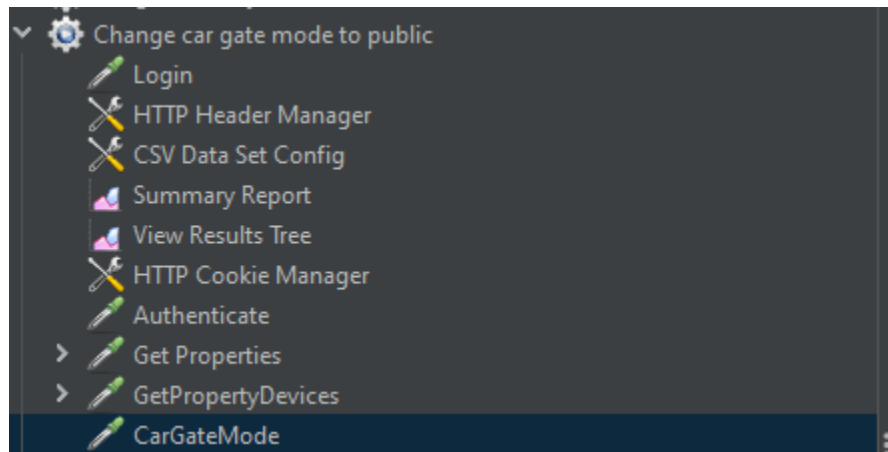
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Login	1000	3434	263	60056	9115.45	2.50%	6.0/sec	6.66	5.17	1129.6
Authenticate	1000	2825	3	60067	11830.96	4.10%	6.1/sec	1.96	5.14	328.9
Get Properties	1000	2933	11	60039	11657.61	4.00%	6.2/sec	3.62	5.21	597.5
GetPropertyDevices	1000	2280	10	60057	9986.99	6.90%	6.2/sec	44.07	5.54	7256.1
GetCarGateHistory	1000	2365	5	60058	10320.97	10.00%	6.2/sec	1.36	5.44	222.7
TOTAL	5000	2767	3	60067	10640.49	5.50%	30.2/sec	56.19	25.96	1907.0

## 10. Change car gate mode (Public)

Koristeći prethodno pomenuti csv fajl za korisnike, posle setupovanja korisnika u bazu I posle pozivanja testnog slučaja za registraciju propertiija I approvovanje I posle setapovanja cargate-ova, koriste se već registrovani korisnici, propertiiji I cargate-ovi da bi se testiralo opterećenje na menjaju moda korisnikovog car gate-a na public.

**Prolazi se kroz sledeći pipeline za testiranje menjanja moda cargate-a na public:**

- Kroz N niti se prolazi kroz csv I zatim pogadja login endpoint
- Korišćenjem HTTP cookie managera se preuzima cookie za odredjeni thread u zavisnosti od logina.
- Svaki user zatim dobavlja sve svoje propertiije prethodno kreirane korišćenjem testnog slučaja za registraciju propertiija.
- Svaki user zatim dobavlja sve svoje uredjaje prethodno kreirane korišćenjem setup-a za kreiranje uredjaja.
- Za prvi nadjeni car gate poziva se endpoint za menjanje moda car gate-a na public.



Rezultati:

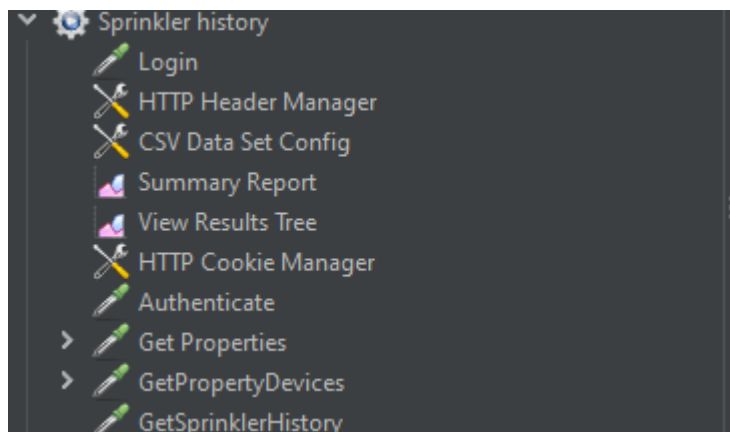
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Login	1000	3722	294	60032	9443.95	2.40%	5.6/sec	6.20	4.81	1130.1
Authenticate	1000	2334	4	60076	9536.88	2.10%	5.7/sec	1.78	4.78	321.5
Get Properties	1000	2031	14	60028	8642.30	1.70%	5.6/sec	3.25	4.69	595.5
GetPropertyDevices	1000	2323	4	60041	9921.44	4.30%	5.6/sec	40.67	4.99	7446.7
CarGateMode	1000	1871	3	60052	8072.62	5.80%	5.6/sec	0.72	5.15	130.6
TOTAL	5000	2456	3	60076	9171.59	3.26%	27.1/sec	51.02	23.59	1924.9

# 11. Sprinkler history

Koristeći prethodno pomenuti csv fajl za korisnike, posle setupovanja korisnika u bazu I posle pozivanja testnog slučaja za registraciju propertiija I approvovanje I posle setapovanja sprinklera, koriste se već registrovani korisnici, propertiiji sprinkleri da bi se testiralo opterećenje na dobavljanju istorije korisnikovog sprinklera.

## Prolazi se kroz sledeći pipeline za testiranje dobavljanja istorije sprinklera:

- Kroz N niti se prolazi kroz csv I zatim pogadja login endpoint
- Korišćenjem HTTP cookie managera se preuzima cookie za odredjeni thread u zavisnosti od logina.
- Svaki user zatim dobavlja sve svoje propertiije prethodno kreirane korišćenjem testnog slučaja za registraciju propertiija.
- Svaki user zatim dobavlja sve svoje uredjaje prethodno kreirane korišćenjem setup-a za kreiranje uredjaja.
- Za prvi nadjeni sprinkler poziva se endpoint za dobavljanje istorije sprinklera.



Rezultati:

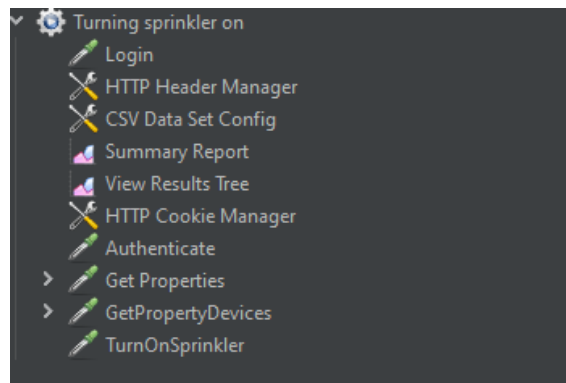
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Login	1000	3530	344	60068	9653.25	2.80%	4.6/sec	5.02	3.90	1128.2
Authenticate	1000	2401	4	60057	11139.09	3.60%	4.6/sec	1.47	3.87	327.0
Get Properties	1000	2084	13	60053	9838.81	2.80%	4.7/sec	2.71	3.91	596.4
GetPropertyDevices	1000	3239	7	60058	12614.42	7.50%	4.7/sec	32.87	4.16	7214.2
GetSprinklerHistory	1000	2628	4	60061	11566.19	11.40%	4.7/sec	1.04	4.09	228.4
TOTAL	5000	2776	4	60068	11030.79	5.62%	22.8/sec	42.20	19.59	1898.8

## 12. Turning sprinkler on

Koristeći prethodno pomenuti csv fajl za korisnike, posle setupovanja korisnika u bazu I posle pozivanja testnog slučaja za registraciju propertiija I approvovanje I posle setapovanja sprinklera, koriste se već registrovani korisnici, propertiiji sprinkleri da bi se testiralo opterećenje na paljenju korisnikovog sprinklera.

### Prolazi se kroz sledeći pipeline za testiranje paljenja sprinklera:

- Kroz N niti se prolazi kroz csv I zatim pogadja login endpoint
- Korišćenjem HTTP cookie managera se preuzima cookie za odredjeni thread u zavisnosti od logina.
- Svaki user zatim dobavlja sve svoje propertiije prethodno kreirane korišćenjem testnog slučaja za registraciju propertiija.
- Svaki user zatim dobavlja sve svoje uređaje prethodno kreirane korišćenjem setup-a za kreiranje uređaja.
- Za prvi nadjeni sprinkler poziva se endpoint za paljenje sprinklera.
- 



### Rezultati:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Login	1000	4023	306	60046	10860.39	3.60%	6.2/sec	6.82	5.32	1124.5
Authenticate	1000	2925	4	60057	12111.15	4.30%	4.6/sec	1.47	3.86	329.5
Get Properties	1000	2275	18	60075	10333.25	3.10%	4.6/sec	2.70	3.89	596.7
GetPropertyDevices	1000	3199	6	60062	12620.68	7.80%	4.6/sec	32.58	4.13	7194.8
TurnOnSprinkler	1000	2957	4	60057	11955.27	12.00%	4.7/sec	0.67	4.16	146.7
TOTAL	5000	3076	4	60075	11620.71	6.16%	22.6/sec	41.50	19.56	1878.4

## 13. Obzervacije

Kroz intenzivno testiranje je uočeno nekoliko bootleneckova vezanih za docker containere koje koristimo.

- Baza (container) sama po sebi po default konfiguraciji ima maksimalno 100 konekcija i jako malo shared memorije sa nasim uredjajem.
- NGINX sam po sebi ima samo 1024 workera po defaultu, broj smo povećali na više naravno uz gubitak resursa

Povećanjem resursa kojima raspolažu nasi docker containeru dobija se fino na performansama, medjutim sam računar počinje da “zabada”.

Sa povećanjem broja simulatora uočen je veći load na mqtt container, takodje je povećan broj resursa kojim mqtt container raspolaže.

Velikim povećanjem broja uredjaja na mašini se i povećava RAM koji simulacija rezerviše, samim tim rad svih ostalih uredjaja je lošiji.