

# Testowanie aplikacji

Nr przypadku testowego	Liczba użytkowników	Spawn rate	Host
1	5	1	http://127.0.0.1:8000/
2	50	1	http://127.0.0.1:8000/
3	500	1	http://127.0.0.1:8000/
4	1000	1	http://127.0.0.1:8000/
5	5000	1	http://127.0.0.1:8000/

Testowana była aplikacja internetowa main.py, która wyświetla napis „Hello, world!” w oknie przeglądarki po połączeniu się z hostem. W ramach testów skupiono się na sprawdzeniu wydajności serwera aplikacji pod obciążeniem. Testy zostały przeprowadzone na lokalnym serwerze odpalonym z poziomu komputera z procesorem AMD Athlon 200GE 3.20Ghz, 8GB pamięci RAM oraz systemem operacyjnym Windows 10 Pro. Do testów wykorzystano Locust w wersji 2.15.1. Pod uwagę brane były: ilość użytkowników, czas odpowiedzi serwera i liczba żądań na sekundę.

## 1. Pierwszy przypadek testowy:

Testy trwały przez czas odpowiedni do osiągnięcia połączenia z wymaganą ilością użytkowników + 1 minuta, a maksymalna liczba użytkowników równocześnie korzystających z aplikacji wynosiła 5. Wykresy dostępne na stronie Locust przedstawiają między innymi liczbę żądań na sekundę, czas odpowiedzi serwera, liczbę błędów oraz liczbę użytkowników. W trakcie testów nie wystąpiły żadne błędy, a czas odpowiedzi serwera po osiągnięciu maksymalnej ilości użytkowników wahał się między 30ms a 40ms. Liczba żądań na sekundę wahała się między 200ms a 250ms.

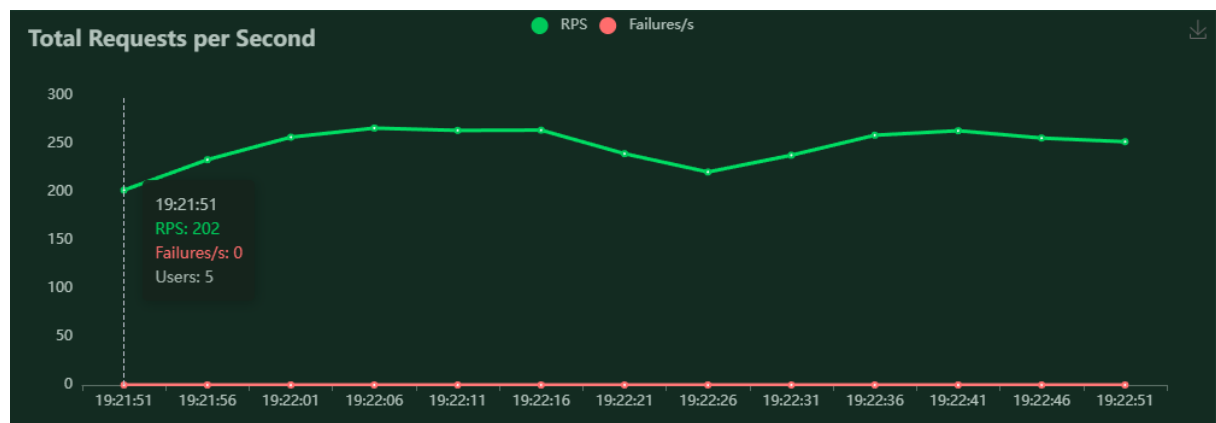
### Request Statistics

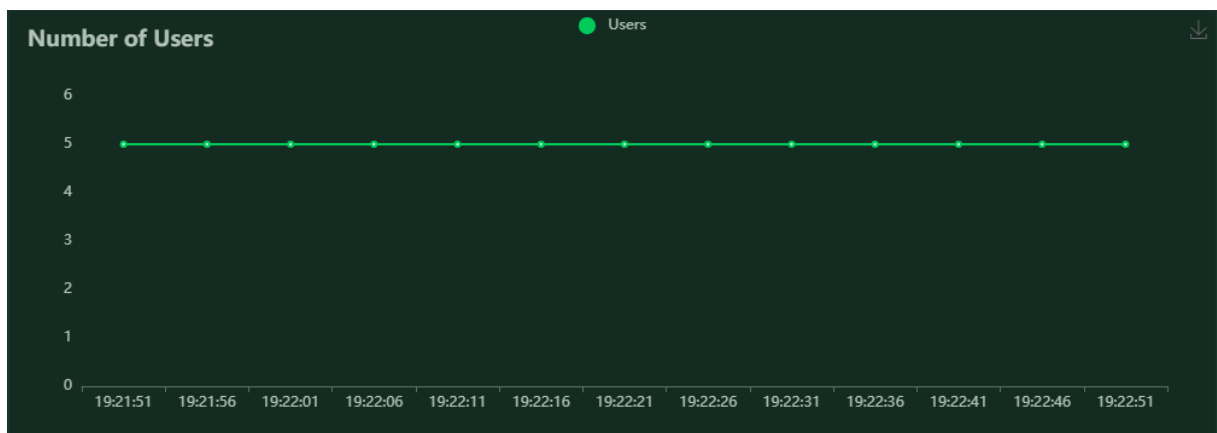
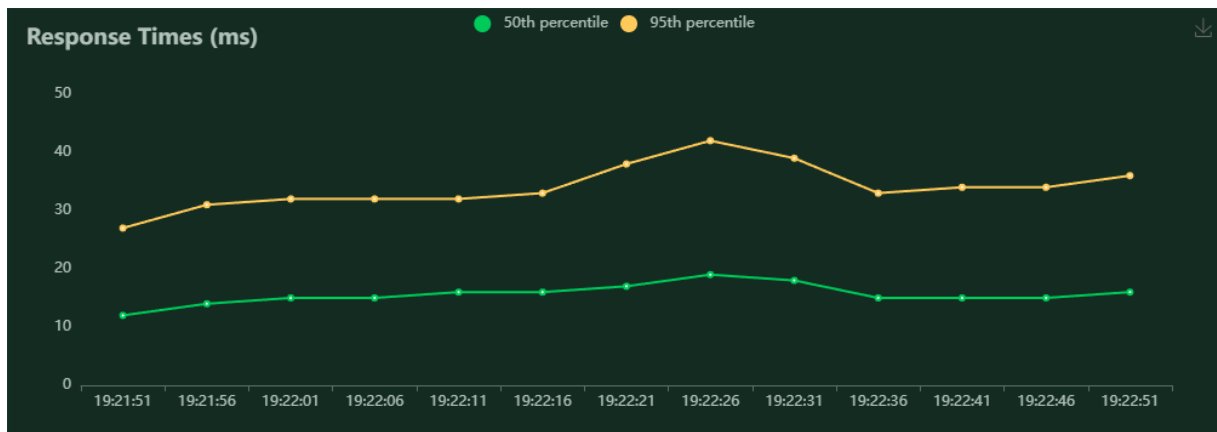
Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	/	5464	0	27	6	109	13	81.7	0.0
GET	/hello	5686	0	14	3	82	269	85.1	0.0
POST	/hello	5685	0	14	3	61	14	85.0	0.0
Aggregated		16835	0	18	3	109	99	251.8	0.0

### Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	/	26	27	29	32	37	41	51	110
GET	/hello	14	14	15	17	19	21	28	83
POST	/hello	14	14	15	17	19	21	28	62
Aggregated		16	18	23	26	29	34	44	110

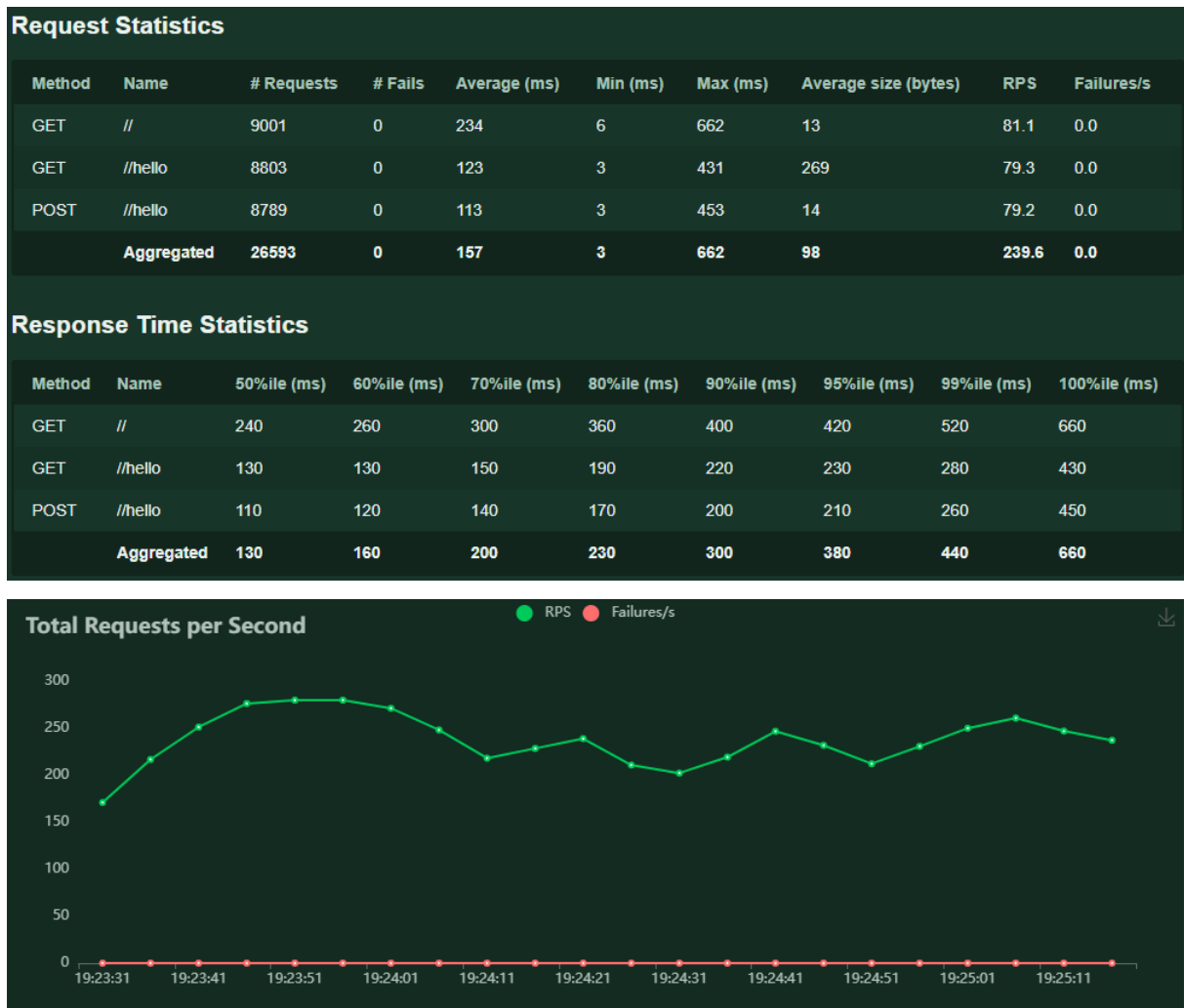
### Total Requests per Second

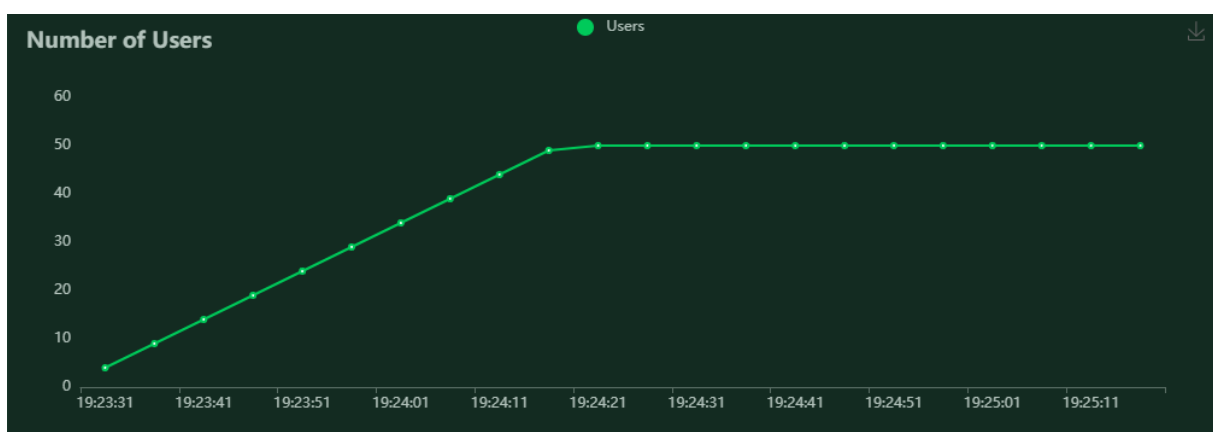
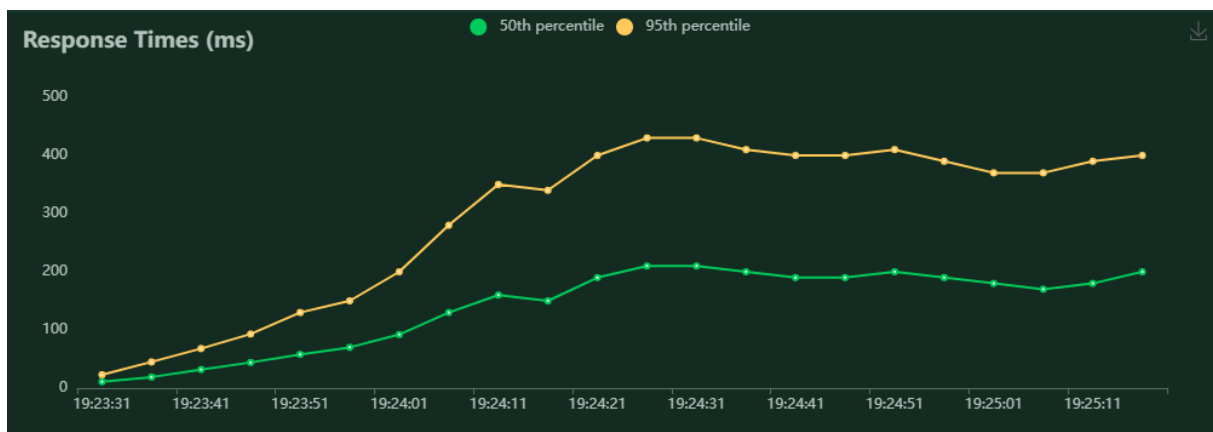




## 2. Drugi przypadek testowy:

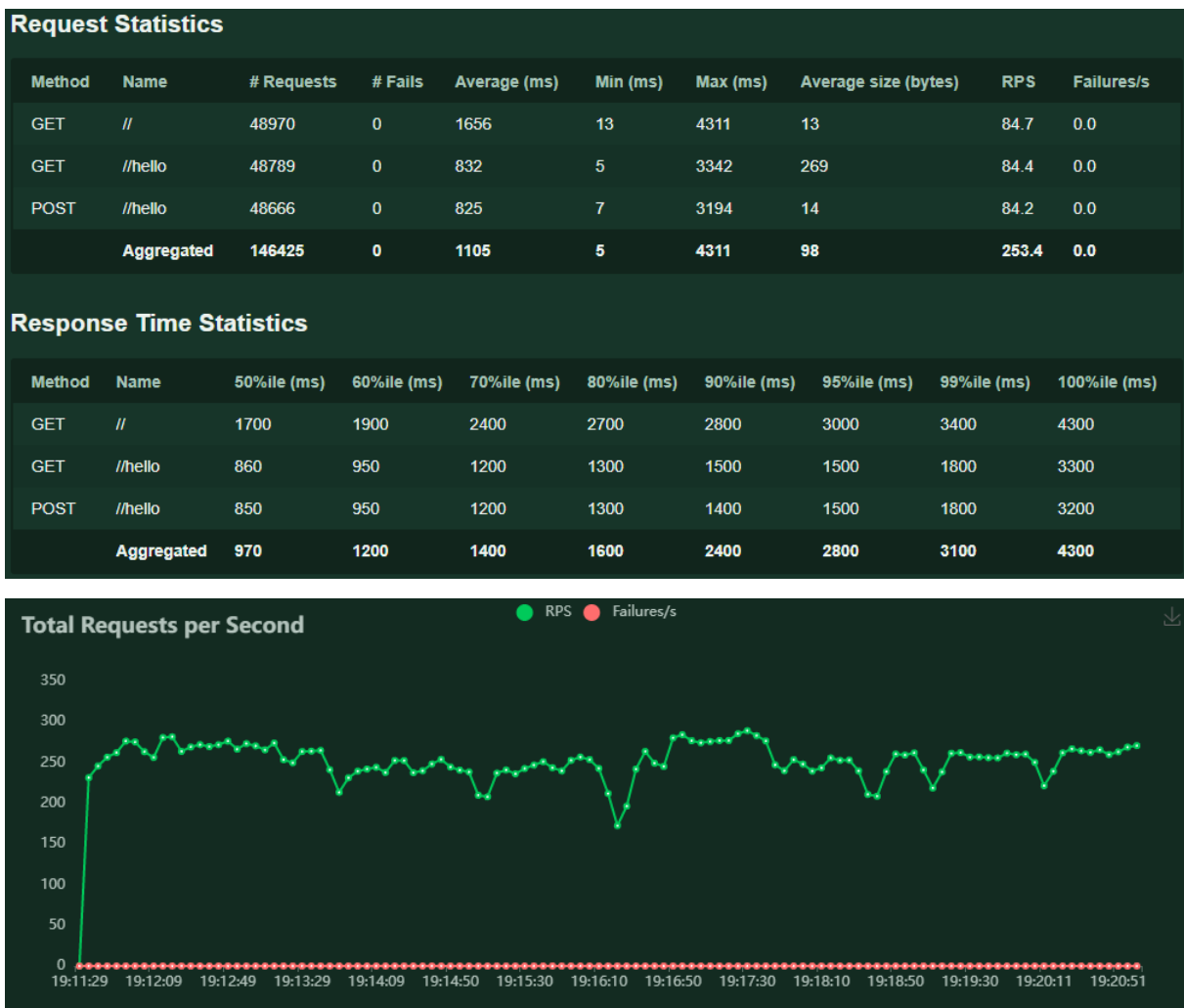
Testy trwały przez czas odpowiedni do osiągnięcia połączenia z wymaganą ilością użytkowników + 1 minuta, a maksymalna liczba użytkowników równocześnie korzystających z aplikacji wynosiła 50. Wykresy dostępne na stronie Locust przedstawiają między innymi liczbę żądań na sekundę, czas odpowiedzi serwera, liczbę błędów oraz liczbę użytkowników. W trakcie testów nie wystąpiły żadne błędy, a czas odpowiedzi serwera po osiągnięciu maksymalnej ilości użytkowników wahał się między 340ms a 430ms. Liczba żądań na sekundę wahała się między 171ms a 280ms.

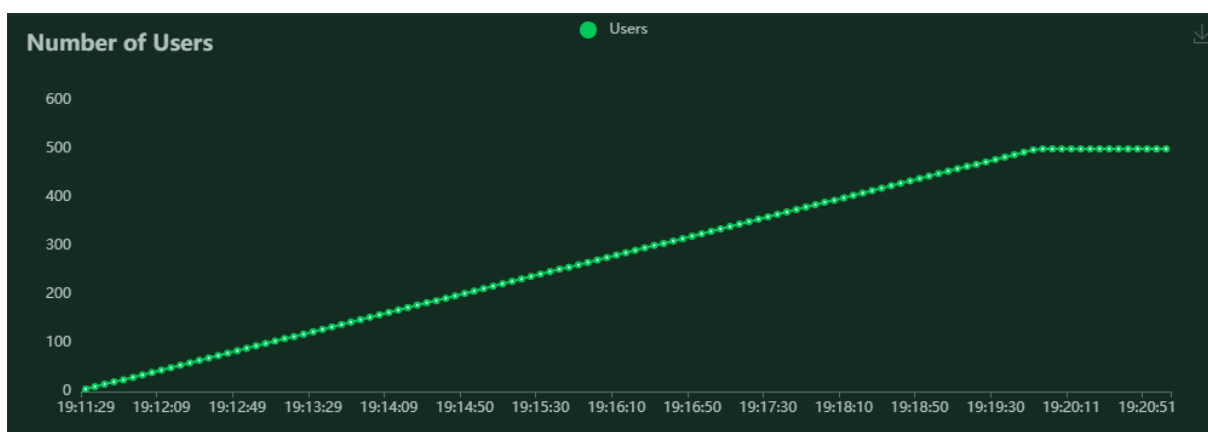
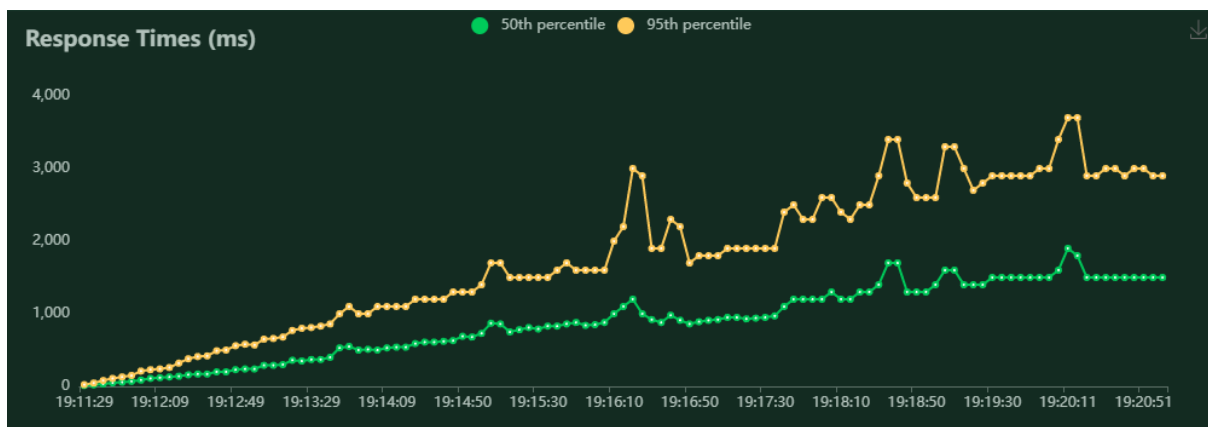




### 3. Trzeci przypadek testowy:

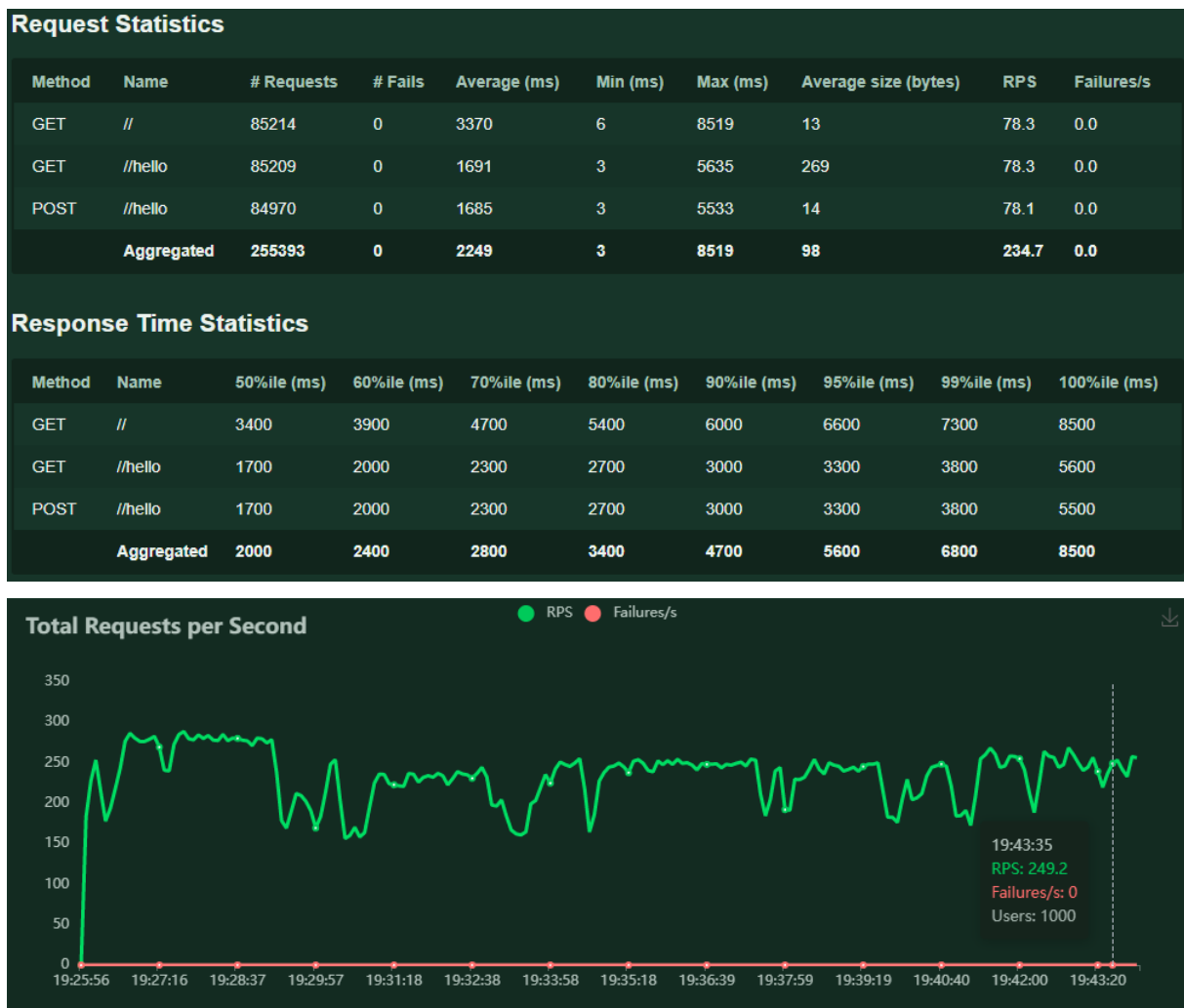
Testy trwały przez czas odpowiedni do osiągnięcia połączenia z wymaganą ilością użytkowników + 1 minuta, a maksymalna liczba użytkowników równocześnie korzystających z aplikacji wynosiła 500. Wykresy dostępne na stronie Locust przedstawiają między innymi liczbę żądań na sekundę, czas odpowiedzi serwera, liczbę błędów oraz liczbę użytkowników. W trakcie testów nie wystąpiły żadne błędy, a czas odpowiedzi serwera po osiągnięciu maksymalnej ilości użytkowników wahał się między 3000ms a 3700ms. Liczba żądań na sekundę wahała się między 172ms a 289ms.



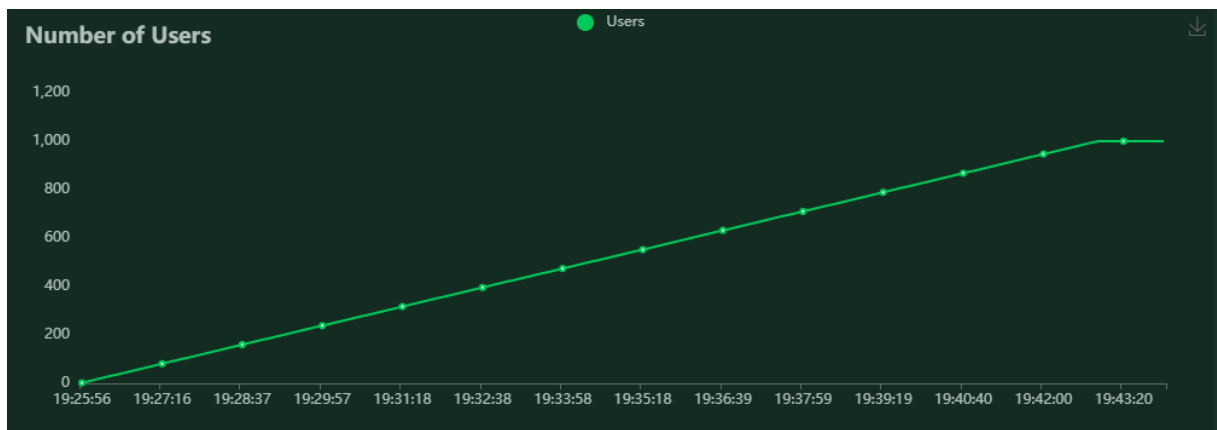
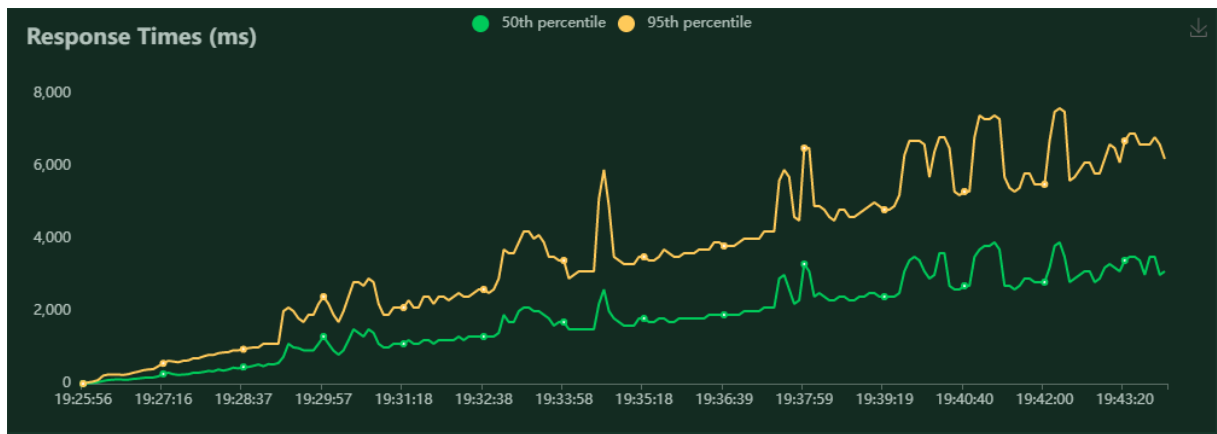


#### 4. Czwarty przypadek testowy:

Testy trwały przez czas odpowiedni do osiągnięcia połączenia z wymaganą ilością użytkowników + 1 minuta, a maksymalna liczba użytkowników równocześnie korzystających z aplikacji wynosiła 1000. Wykresy dostępne na stronie Locust przedstawiają między innymi liczbę żądań na sekundę, czas odpowiedzi serwera, liczbę błędów oraz liczbę użytkowników. W trakcie testów nie wystąpiły żadne błędy, a czas odpowiedzi serwera po osiągnięciu maksymalnej ilości użytkowników wahał się między 5800ms a 6900ms. Liczba żądań na sekundę wahała się między 156ms a 286ms.





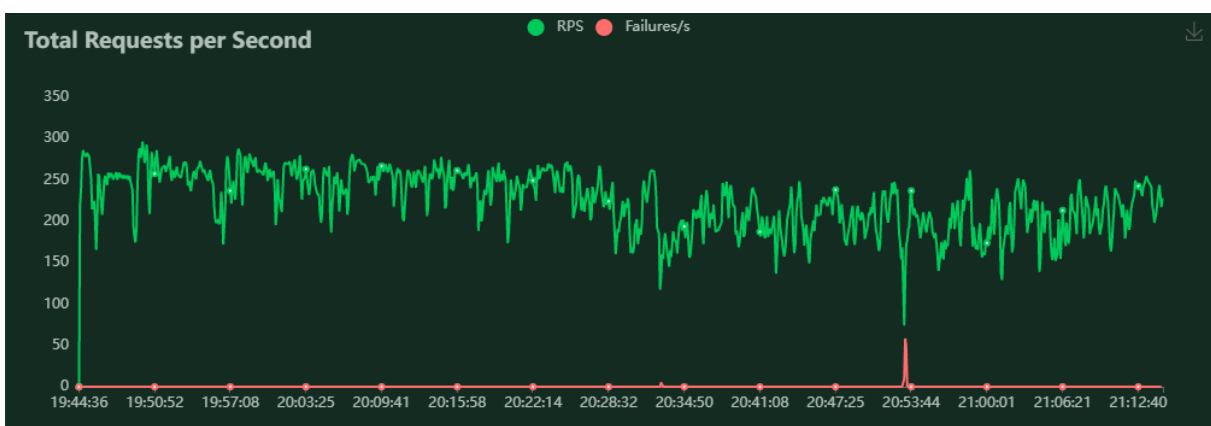


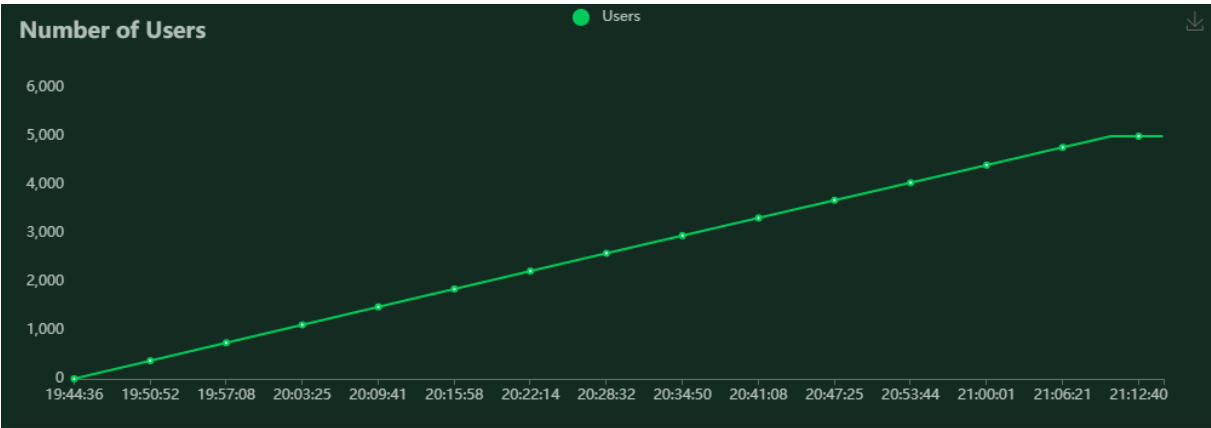
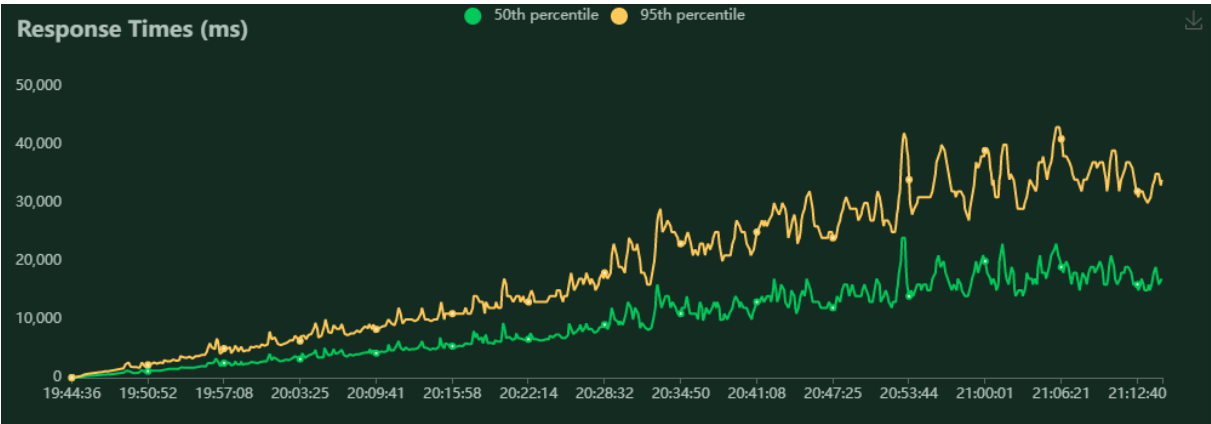
## 5. Piąty przypadek:

Testy trwały przez czas odpowiedni do osiągnięcia połączenia z wymaganą ilością użytkowników + 1 minuta, a maksymalna liczba użytkowników równocześnie korzystających z aplikacji wynosiła 5000. Wykresy dostępne na stronie Locust przedstawiają między innymi liczbę żądań na sekundę, czas odpowiedzi serwera, liczbę błędów oraz liczbę użytkowników. W trakcie testów wystąpiły trzy błędy związane najprawdopodobniej z „lagiem” komputera, a czas odpowiedzi serwera po osiągnięciu maksymalnej ilości użytkowników wahał się między 30000ms a 39000ms. Liczba żądań na sekundę wahała się między 160ms (z pominięciem błędu, podczas którego najniższa wartość wyniosła 74ms) a 296ms.

Request Statistics									
Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	/	406043	318	17411	6	43838	12	75.1	0.1
GET	/hello	406618	156	8729	3	26111	268	75.2	0.0
POST	/hello	405384	163	8718	3	26102	13	75.0	0.0
Aggregated		1218045	637	11620	3	43838	98	225.4	0.1

Response Time Statistics									
Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	/	14000	22000	26000	30000	34000	37000	40000	44000
GET	/hello	7300	11000	13000	15000	17000	18000	21000	26000
POST	/hello	7300	11000	13000	15000	17000	18000	21000	26000
Aggregated		9900	13000	15000	18000	26000	32000	38000	44000





### Failures Statistics

Method	Name	Error	Occurrences
GET	//	[Ermo 10061] [WinError 10061] Nie można nawiązać połączenia, ponieważ komputer docelowy aktywnie go odmawia.	318
GET	//hello	[Ermo 10061] [WinError 10061] Nie można nawiązać połączenia, ponieważ komputer docelowy aktywnie go odmawia.	156
POST	//hello	[Ermo 10061] [WinError 10061] Nie można nawiązać połączenia, ponieważ komputer docelowy aktywnie go odmawia.	163

# Wnioski

Poszczególne testy różniły się ilością połączonych użytkowników, liczbą żądań na sekundę oraz czasem odpowiedzi serwera. Liczba żądań i czas odpowiedzi rósł proporcjonalnie do ilości użytkowników. Żaden test nie zgłosił błędu (wyjątkiem był piąty przypadek testowania, wydaje mi się, że błędy wyskoczyły przez chwilowy „lag” komputera). Aplikacja jest wydajna.