

Jmeter Test Sonuçları

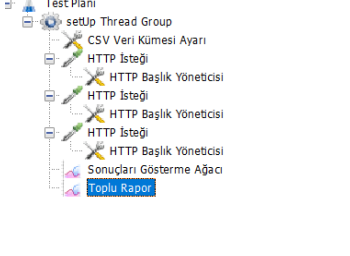
Test senaryosu - 1

Amaç: Bucket'ı aşacak şekilde requestler atarak kullanıcının rate limit'e takıldığını görme.

Yöntem:

- Rate limiter dakikada 10 tokena sahiptir.
- Bir get requesti 3, bir post requesti 5 token tüketir.
- Sırasıyla bir get ve bir post request atılır, sonrasında atılan get requestinin rate limitera takılması beklenir.
- Test metodu her kullanıcı random bir ip'ye sahip olacak şekilde 100 eşzamanlı kullanıcı 1 kere çalıştırılır.
- Toplamda 300 istek atılır ve %33.33 hata oranı olması beklenir.

Sonuç:



Toplu Rapor												
İsim: Toplu Rapor												
Yorumlar:												
Sonuçları dosyaya yaz / Dosyadan oku												
Dosya ismi: Gözet... Sadece Log/Görüntüleme: <input type="checkbox"/> Hatalar <input type="checkbox"/> Başarılar <input type="button" value="Ayarla"/>												
Etiket	# Örnek	Ortalama	Ortalama	90% Line	95% Line	99% Line	En Az	En Çok	Hata %	Transfer ...	KB/sn	Sent KB/sec
HTTP İsteği	300	254	153	475	548	1557	30	6893	33,33%	37,1/sec	11,66	6,8
TOPLAM	300	254	153	475	548	1557	30	6893	33,33%	37,1/sec	11,66	6,8

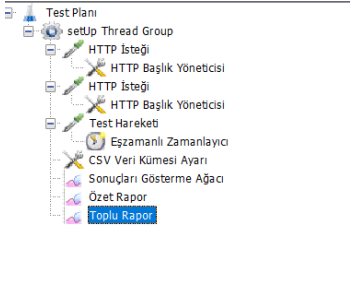
Test senaryosu - 2

Amaç: Rate Limiter enabled/disabled olacak şekilde backend servise request atma.

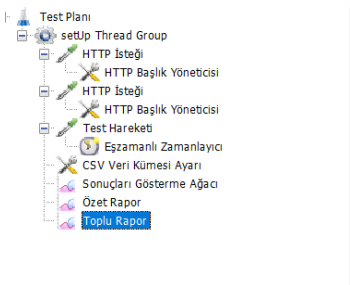
Yöntem:

- 100 eşzamanlı kullanıcı tarafından bir get ve bir post request atılır, 60 saniye beklenilir. Bu işlem sürekli tekrar edilir.
- Test kodu rate limiter varken ve yokken çalıştırılır.
- Response timelar incelenir.

Sonuç:



Toplu Rapor												
İsim: Toplu Rapor												
Yorumlar:												
Sonuçları dosyaya yaz / Dosyadan oku												
Dosya ismi: Gözet... Sadece Log/Görüntüleme: <input type="checkbox"/> Hatalar <input type="checkbox"/> Başarılar <input type="button" value="Ayarla"/>												
Etiket	# Örnek	Ortalama	Ortalama	90% Line	95% Line	99% Line	En Az	En Çok	Hata %	Transfer ...	KB/sn	Sent KB/sec
HTTP İsteği	1600	388	357	697	826	2069	29	3357	0,00%	3,7/sec	1,31	0,7
TOPLAM	1600	388	357	697	826	2069	29	3357	0,00%	3,7/sec	1,31	0,7



Toplu Rapor												
İsim: Toplu Rapor												
Yorumlar:												
Sonuçları dosyaya yaz / Dosyadan oku												
Dosya ismi: Gözet... Sadece Log/Görüntüleme: <input type="checkbox"/> Hatalar <input type="checkbox"/> Başarılar <input type="button" value="Ayarla"/>												
Etiket	# Örnek	Ortalama	Ortalama	90% Line	95% Line	99% Line	En Az	En Çok	Hata %	Transfer ...	KB/sn	Sent KB/sec
HTTP İsteği	1600	230	228	407	503	612	29	1078	0,00%	3,8/sec	1,28	0,7
TOPLAM	1600	230	228	407	503	612	29	1078	0,00%	3,8/sec	1,28	0,7

Test senaryosu - 3

Amaç: Response'a delay vererek etkisini gözlemleme.

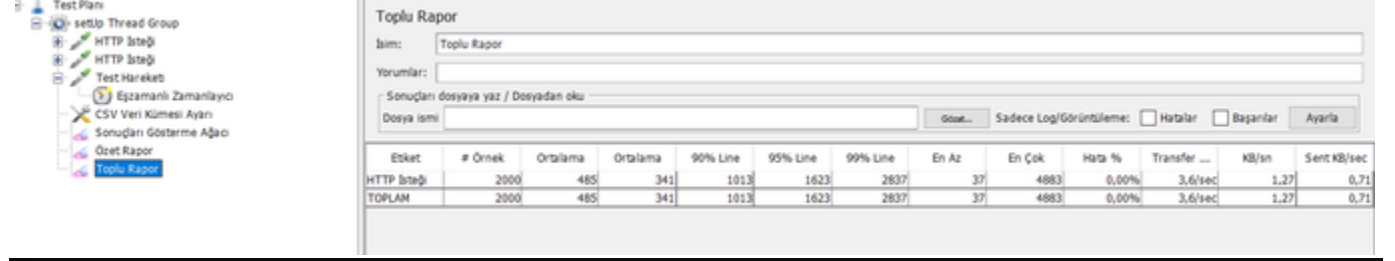
Yöntem:

- Wiremock ile servis oluşturulur.
- Gateway wiremock'a yönlendirilir.

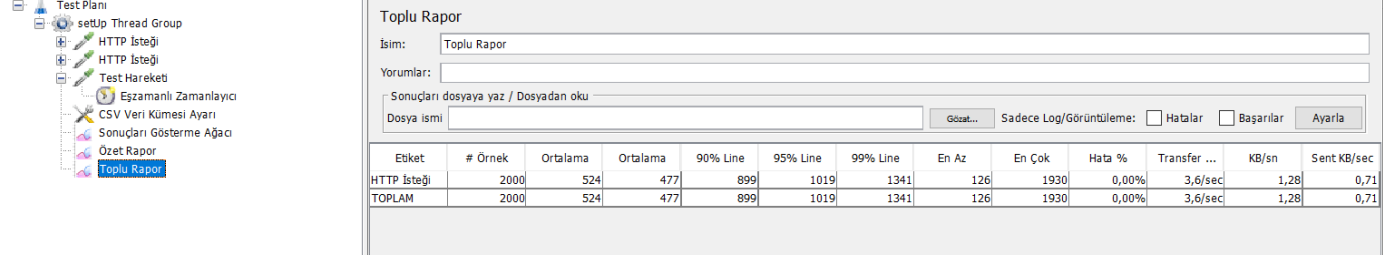
- Wiremock 10, 100 ve 1000 ms geciktirilerek test senaryosu denenir.
- Test senaryosu: 100 eşzamanlı kullanıcı tarafından bir get ve bir post request atılır, 60 saniye beklenilir. Bu işlem sürekli tekrar edilir.

Sonuç:

10 ms delay:



100 ms delay:





1000 ms delay:

Test Planı

- setUp Thread Group
- HTTP İsteği
- HTTP İsteği
- Test Hareketi
- Ezamanlı Zamanlayıcı
- CSV Veri Kümesi Ayarı
- Sonuçları Gösterme Ağacı
- Ozet Rapor
- Toplu Rapor**

Toplu Rapor

İsim: Toplu Rapor

Yorumlar:

Sonuçları dosyaya yaz / Dosyadan oku

Dosya ismi:

Gözet... Sadece Log/Görüntüleme: ☐ Hatalar ☐ Başarılar ☐ Ayarla

Etket	# Örnek	Ortalama	Ortalama	90% Line	95% Line	99% Line	En Az	En Çok	Hata %	Transfer ...	KB/sn	Sent KB/sec
HTTP İsteği	2000	3978	4370	6022	6304	7416	1026	9214	0,00%	3,1/sec	1,09	0,6
TOPLAM	2000	3978	4370	6022	6304	7416	1026	9214	0,00%	3,1/sec	1,09	0,6

