

בדיקות קיבול ועומס

תמהיל השימוש הצפוי במערכת

- כ-10,000 משתמשים רשומים (בסה"כ)
- עד כ-500 משתמשים המחוברים למערכת בו זמנית (מתוכם כ-50 רשומים ו-450 אורחים)
- כ-50 חנויות במערכת, כאשר לרובן המוחלט יהיה כבעל חנות אחד וכ-2 מנהלי חנות
- פעולות הרכישה יהוו כ-5% מכלל הפעולות במערכת
- רוב הפעולות במערכת אמורות להיות פעולות "קלות" מבחינת המשאבים שהן דורשות, למשל – צפייה בעגלה, חיפוש חנות, צפייה במוצרים בחנויות וכו'.
- קצב ההצטרפות הצפוי: כ-30 משתמשים רשומים חדשים מדי יום.
- רוב הרשומות שיאוחסנו במערכת – היסטוריית רכישות. כמות החנויות צפויה כאמור להיות כמה עשרות, אלפים בודדים של משתמשים רשומים, אלפים בודדים של מוצרים בחנויות. כמות הרכישות, לעומת שאר הדברים שצוינו קודם, צפויה לגדול באופן מהר משמעותית. הצפי הוא סדר גודל של כמה עשרות מיליונים של רשומות של היסטוריית רכישות.

את הטסטים האוטומטיים הגדרנו בעזרת פלטפורמת JMeter. נפרט במסמך זה את תרחישי הטסטים שהגדרנו, ניתוח התוצאות שלהם (בעזרת גרפים ותרשימים מתאימים) ונסיק מסקנות לגבי טיב הביצועים של המערכת על פי המדדים שנקבעו עבור גרסה 4.

התרחישים בטסטים תואמים את דרישות רמת השירות

בעזרת הטסטים שלנו נראה עמידה במדדים הבאים:

- i. התמודדות עם 100 בקשות בו זמנית (אירועים כגון התחברות, רכישה וכו') תוך עמידה בזמן תגובה של לכל היותר שנייה לכל בקשה.
- ii. תמיכה בכמה מיליוני רשומות (חנויות, משתמשים, היסטוריית רכישות). (מטעמי זמן ריצה ומשאבים מוגבלים שלא מאפשרים להריץ כמות ת'רדים התואמת תרחיש כזה בזמן סביר – שכן זה תרחיש שאמור להתפרש לאורך תקופה ארוכה ולא מספר שעות/ימים, אין לנו תוצאות שתומכות בכך שהמערכת שלנו עומדת בטסט, אך נסביר כיצד הבדיקה תתבצע). בנוסף, נגדיר טסטים שימחישו את יכולת המערכת לייצר כמות רשומות סבירה בזמן קצר
- iii. תמיכה ב-1000 מבקרים במערכת בכל רגע נתון.

טסט ראשון: בדיקת קיבול – המערכת מסוגלת להתמודד עם 500 משתמשים פעילים המבצעים פעולות "נפוצות"/"קלות", במקרה הזה - צפייה בעגלה, במשך זמן קצר

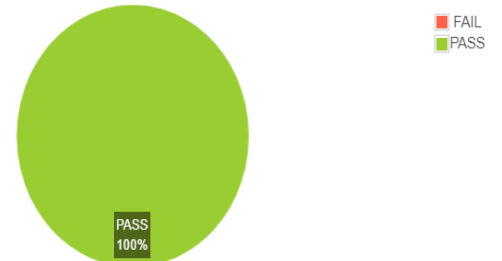
תיאור התרחיש: במשך זמן קצר (כ-10 שניות), מתבצעים 500 חיבורים של משתמשים למערכת (העומס הוא "אינקרמנטלי", כלומר מדי שנייה מתווספים 50 חיבורים חדשים). לאחר מכן, על פני 10 שניות (גם כאן, באופן אינקרמנטלי), כל אחד מהמשתמשים הללו מבקש לצפות בעגלה שלו (פעולה "נפוצה" שהגיוני שתקרה הרבה פעמים בטווח זמן יחסית קצר, והיא גם לא יקרה מבחינת זמן הריצה שלה). מתבצע וידוא שהresponse של השרת שלנו הוא success לכל הבקשות שהתקבלו בתרחיש, ושכולן נענו תוך שנייה לכל היותר.

ניתוח תוצאות:

APDEX (Application Performance Index)

Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
1.000	500 ms	1 sec 500 ms	Total
1.000	500 ms	1 sec 500 ms	Enter Request
1.000	500 ms	1 sec 500 ms	GetCart Request

Requests Summary



Statistics

Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	1000	0	0.00%	41.97	19	346	30.00	48.00	80.90	291.85	50.29	816.75	9.38
Enter Request	500	0	0.00%	46.32	19	346	28.00	69.20	226.20	316.97	50.72	566.44	8.32
GetCart Request	500	0	0.00%	37.62	24	91	36.00	48.00	52.00	77.99	49.97	1064.99	10.44

כפי שניתן לראות, המערכת עומדת בעומס הצפוי למערכת של 500 משתמשים המחוברים בו זמנית, ומצליחה לספק זמן תגובה של לכל היותר שנייה אל הבקשות שהתקבלו (לפי דיאגרמת העוגה – כל הבקשות נענו בהצלחה, לפי הסטטיסטיקות – כל הבקשות נענו תוך פחות משנייה).

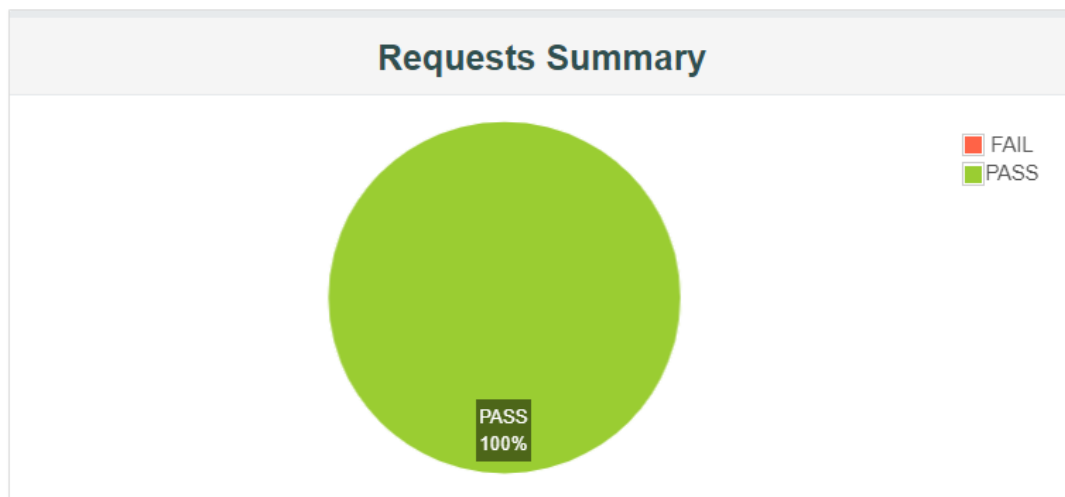
טסט שני: בדיקת עומס- התמודדות של המערכת עם 1000 פעולות בו זמנית (תרחיש קיצוני שהמערכת שלנו צפויה לא לעמוד בו, לפחות לא ברמת זמני התגובה הדרושים)

תיאור התרחיש: במשך זמן מאוד קצר (5 שניות) מתבצעים 1000 חיבורים של משתמשים למערכת. לאחר מכן, על פני שנייה כל אחד מהמשתמשים הללו מבקש לצפות בעגלה שלו (פעולה "נפוצה" שהגיוני שתקרה הרבה פעמים בטווח זמן יחסית קצר, והיא גם לא יקרה מבחינת זמן הריצה שלה, אבל כאן הדגמנו תרחיש קיצוני עם כמות חיבורים בה המערכת לא צפויה לעמוד בזמני תגובה הרצויים). מתבצע וידוא שהresponse של השרת שלנו הוא success לכל הבקשות שהתקבלו בתרחיש, ושכולן נענו תוך שנייה לכל היותר.

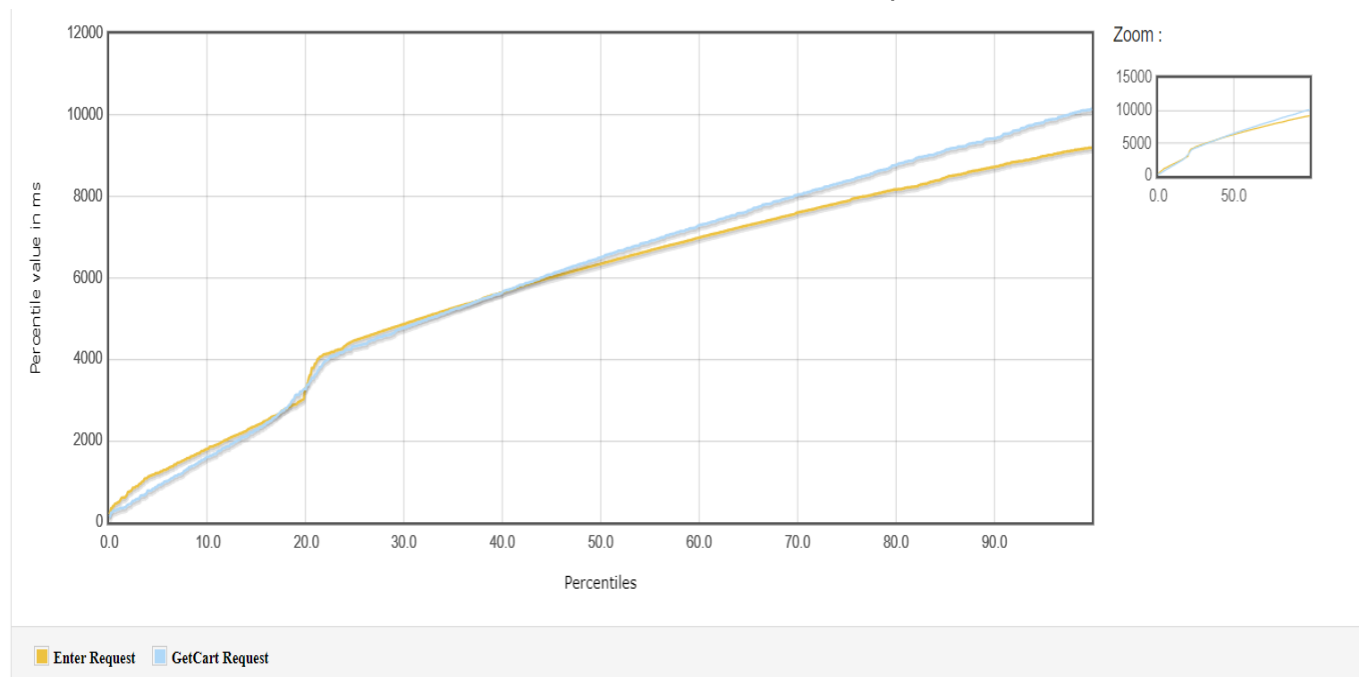
זה תרחיש קיצוני שמטרתו לבחון את קצה גבול היכולת של המערכת.

ניתוח תוצאות:

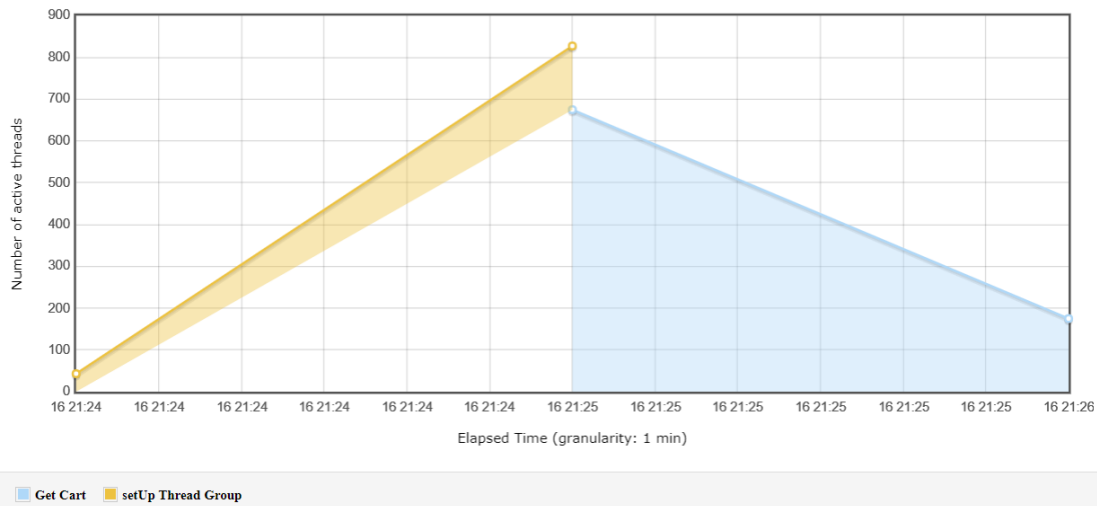
אמנם כל הבקשות נענו בהצלחה



אבל ניתן לראות כי זמני התגובה מאוד גדולים, רק עבור כ-5% מהבקשות זמן התגובה היה פחות משנייה ועבור שאר הבקשות הוא טיפס לעד כ-10 שניות.



כמות המשתמשים הפעילים לאורך הבדיקה:



כלומר, המערכת מסוגלת להתמודד עם 1000 משתמשים מחוברים בו זמנית ולטפל בסופו של דבר בכל בקשותיהם, אך זמני התגובה עלולים להיפגע.

טסט שלישי: בדיקת קיבול – המערכת מסוגלת להתמודד עם 100 בקשות "כבדות" (ופחות תדירות בהשוואה לאחרות, כגון רכישה) בזמן קצר

לפי תמהיל השימוש שהגדרנו, יש לתמוך ב-500 משתמשים המחוברים בו זמנית. כמו כן, הגדרנו כי כ-5% מסך הפעולות במערכת יהיו פעולות רכישה. לכן תרחיש סביר בהתאם לתמהיל השימוש הוא כ-25 רכישות בטווח זמן קצר. בטסט הזה נוודא כי המערכת מסוגלת לתמוך ב-100 בקשות הללו בהצלחה בזמן תגובה של פחות משנייה.

תחילה יצרנו 100 חיבורים של משתמשים (על פני 5 שניות). לאחר מכן, כל אחד מוסיף מוצר קיים במערכת לעגלה שלו (גם כאן, מפוזר על פני 5 שניות).

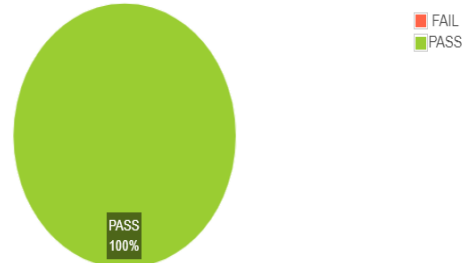
לבסוף, על פני 5 שניות – כל אחד מבצע רכישה של המוצר הזה.

הוספנו Assert על זמן התגובה (תוך שנייה) והסטטוס שלה (200-success), ניתן לראות על פי התוצאות שהתקבלו שהמערכת עמדה בטסט בהצלחה 😊

APDEX (Application Performance Index)

Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
0.900	500 ms	1 sec 500 ms	Total
0.845	500 ms	1 sec 500 ms	AddProductToCart Request
0.855	500 ms	1 sec 500 ms	Enter Request
1.000	500 ms	1 sec 500 ms	PurchaseCart Request

Requests Summary



Statistics

Requests	Executions			Response Times (ms)								Throughput	Network (KB/sec)	
	Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total		300	0	0.00%	294.81	26	554	354.00	521.00	535.95	546.99	39.13	530.35	11.51
AddProductToCart Request		100	0	0.00%	430.98	224	543	463.00	529.00	538.00	543.00	39.76	325.16	10.45
Enter Request		100	0	0.00%	414.13	138	554	451.50	533.40	538.90	553.93	41.48	463.26	6.80
PurchaseCart Request		100	0	0.00%	39.32	26	75	37.50	54.00	57.00	74.85	20.02	426.81	9.13

טסט רביעי: בדיקת עומס - המערכת לא קורסת כאשר השירותים החיצוניים כבויים בעת רכישה

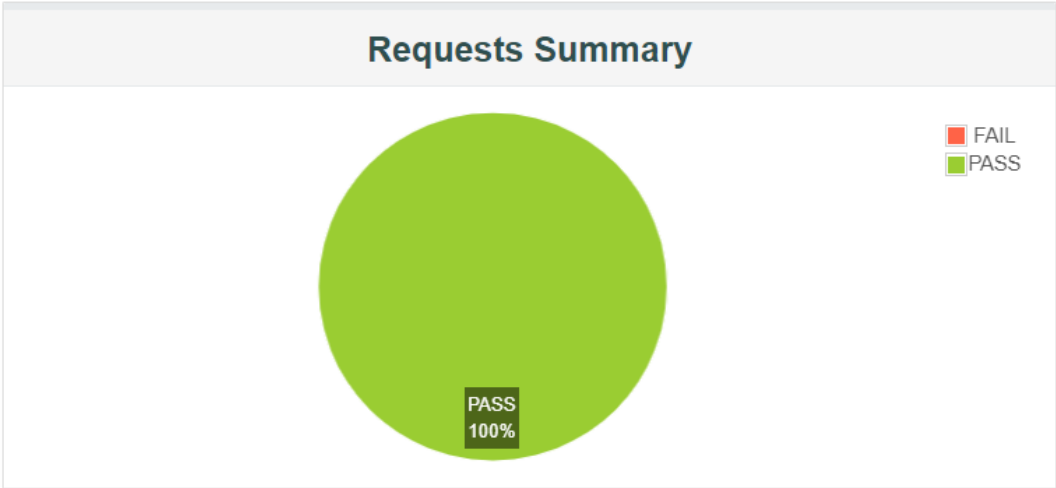
לפי דרישת העומס (SLO), על המערכת להיות זמינה ונגישה באופן תמידי, למעט פעולות סגירה יזומות.

את טסט זה יש להריץ כשלפחות אחת מהמערכות החיצוניות כבויות (ניתן לבצע זאת באופן יזום בעזרת הקובץ env. בפרויקט שלנו).

המטרה שלו היא לבדוק שהמערכת לא קורסת בזמן שהשירותים החיצוניים כבויים, אלא פועלת כרגיל ומגיבה בהודעת שגיאה מתאימה.

לשם כך ביצענו את אותו תרחיש מהטסט הקודם, וביצענו עבור בקשות הרכישה assert על סטטוס התגובה (400- bad request), על מנת לוודא שהמערכת מודיעה למשתמש שביקש לרכוש כי השירות הזה אינו זמין כרגע, ושלא מתרחשת במקום שגיאה לא צפויה אחרת כלשהי.

אכן כך היה:



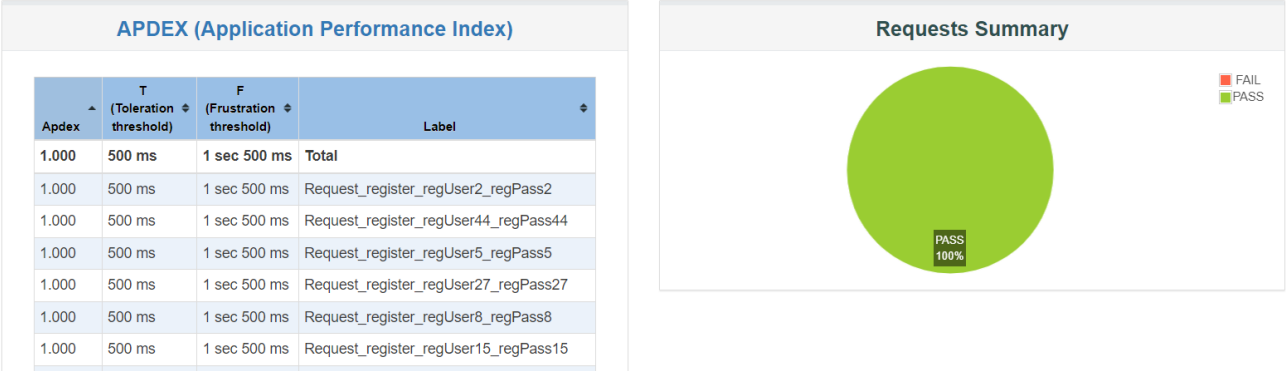
טסט חמישי: בדיקת עומס – התמודדות עם תרחיש שיוצר כמות סבירה של רשומות (100-200) בזמן קצר ולאחר מכן מוחק רשומות (פעולות בתרחיש – רישום, התחברות, מחיקת משתמש על ידי האדמין)

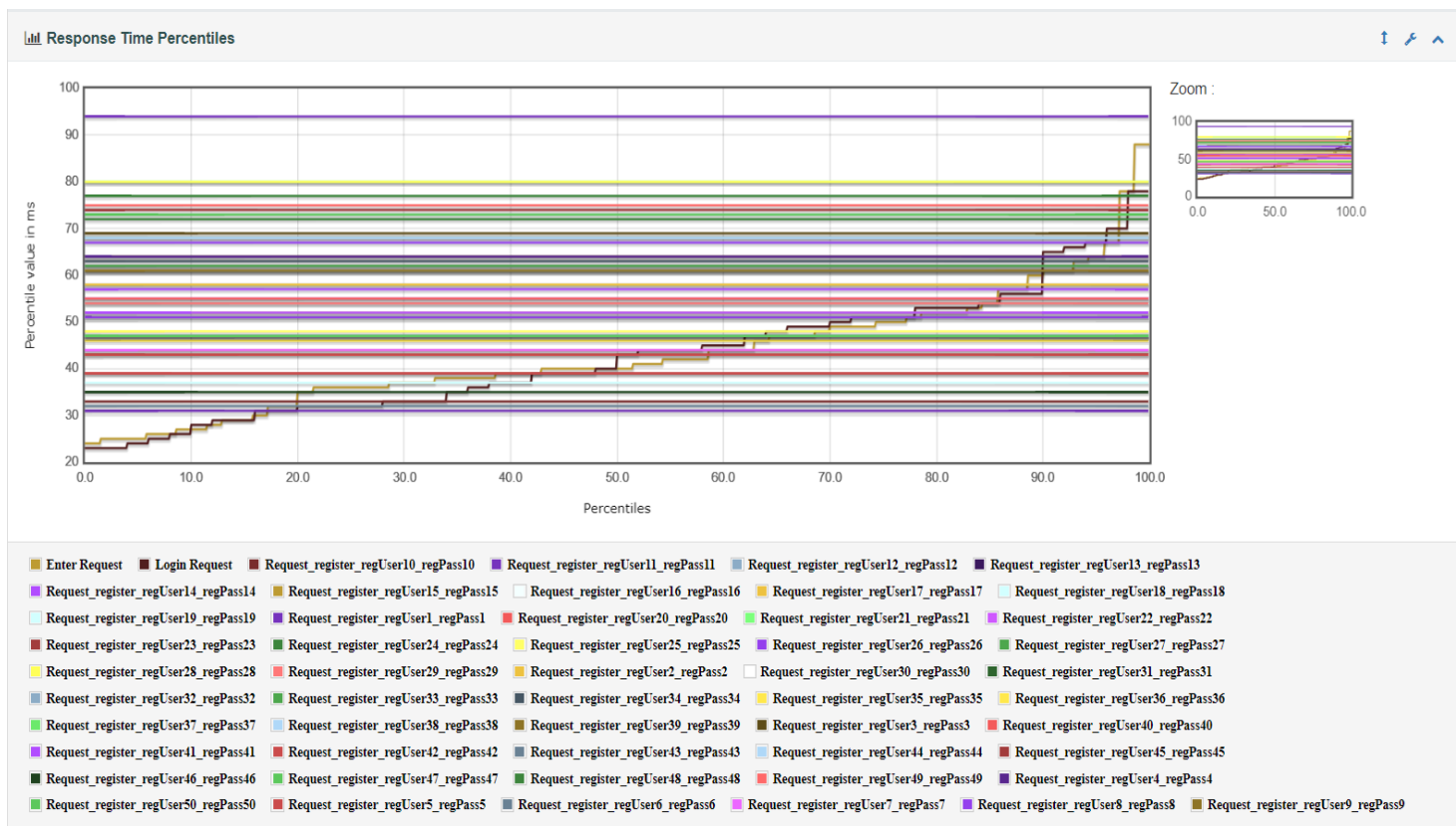
לפי דרישת SLO מהעבודה, על המערכת לתמוך ברישום והתחברות של כמות משתמשים חדשים התואמת את צרכי המערכת בזמן קצר.

בטסט הזה נדגים תרחיש בו נרשמים 50 משתמשים חדשים למערכת (על פני 20 שניות), מתחברים 70 אורחים (על פני 20 שניות), המשתמשים שנרשמו מבצעים login (על פני 20 שניות) ולבסוף האדמין מוחק את 50 המשתמשים שנרשמו (על פני 20 שניות).

מתבצע Assert על זמן התגובה לכל בקשה (לכל היותר שנייה) ועל כך שהסטטוס הוא 200-success.

ואכן המערכת עומדת בדרישות, כפי שניתן לראות מתוצאות הטסט:





כאן ניתן לראות כי כל הבקשות באמת טופלו בזמן קצר כנדרש.

טסט שישי: בדיקת עומס – התמודדות עם תרחיש שיוצר כמות סבירה של רשומות (200-100) בזמן קצר ולאחר מכן מוחק רשומות (פעולות בתרחיש – רישום, התחברות, יצירת חנות)

לפי דרישת SLO מהעבודה, על המערכת לתמוך ברישום והתחברות של כמות משתמשים חדשים התואמת את צרכי המערכת בזמן קצר, כמו גם יצירת כמות חנויות סבירה בזמן קצר.

בתרחיש זה נירשם עם 50 משתמשים למערכת (על פני 20 שניות), נתחבר איתם למערכת (על פני 20 שניות), ולבסוף נפתח עם כל אחד מהם חנות עם שם ייחודי (זהה לשם המשתמש).

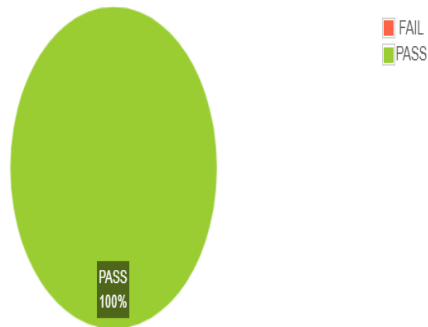
נבצע assert על כך שסטטוס התגובות הוא ok-200 ושכל בקשה נענתה תוך לכל היותר שנייה.

תוצאות:

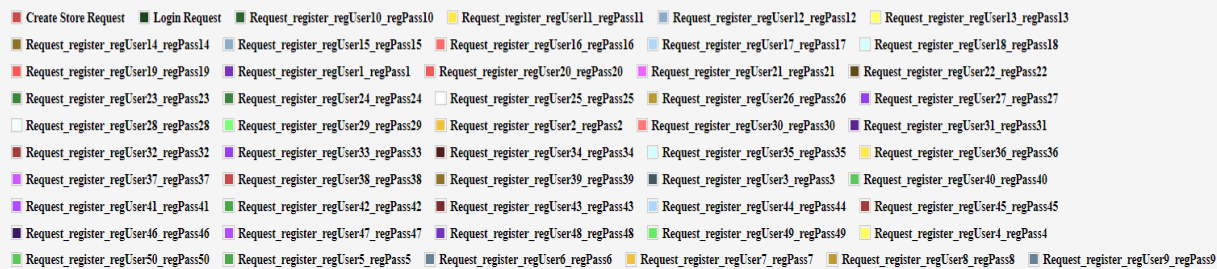
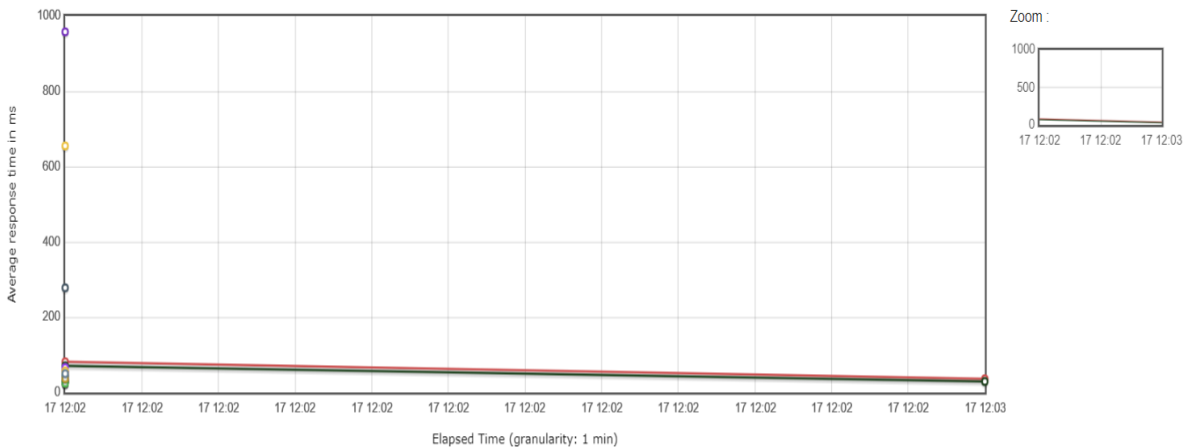
APDEX (Application Performance Index)

Apdex ▲	T (Toleration threshold) ▼	F (Frustration threshold) ▼	Label ▼
0.987	500 ms	1 sec 500 ms	Total
0.500	500 ms	1 sec 500 ms	Request_register_regUser2_regPass2
0.500	500 ms	1 sec 500 ms	Request_register_regUser1_regPass1
0.990	500 ms	1 sec 500 ms	Create Store Request
0.990	500 ms	1 sec 500 ms	Login Request
1.000	500 ms	1 sec 500 ms	Request_register_regUser44_regPass44
1.000	500 ms	1 sec 500 ms	Request_register_regUser5_regPass5

Requests Summary

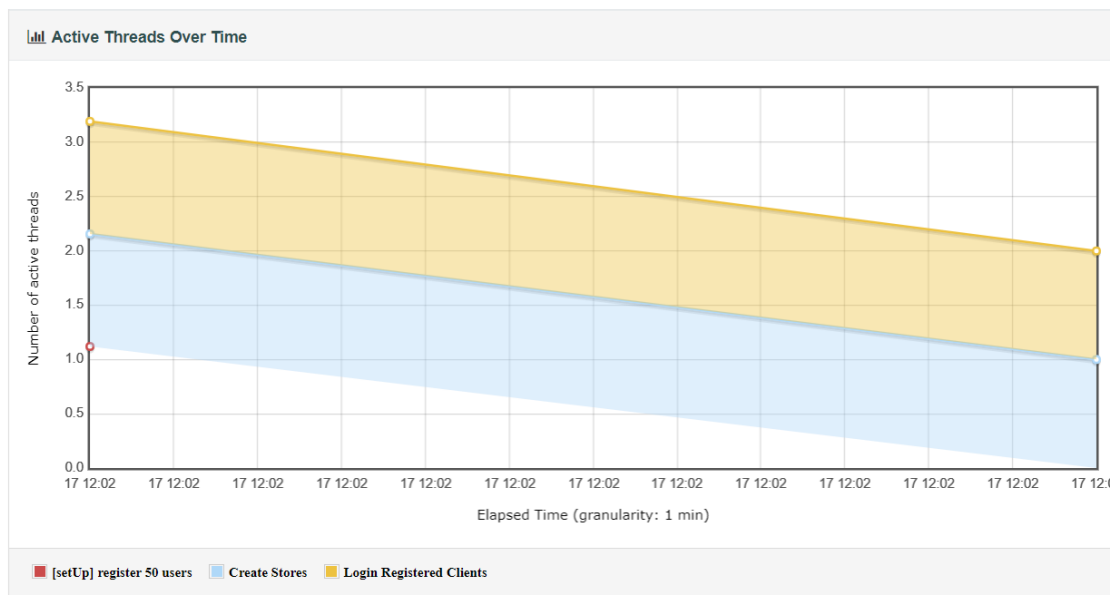


Response Times Over Time



כפי שניתן לראות, כל בקשה טופלה (בהצלחה) תוך פחות מ200ms.

כמות משתמשים פעילים לפי זמן:



טסט שביעי: בדיקת עומס – התמודדות עם תרחיש שיוצר כמה מיליוני רשומות

לפי דרישת העומס במערכת, עלינו לתמוך בכמות רשומות "בלתי מוגבלת" (בהתאם לצרכי המערכת). המערכת שלנו צפויה להחזיק לאורך פעולתה כמה מיליוני רשומות. זו כמות עצומה שאמורה להצטבר לאורך זמן רב של פעילות של המערכת, ולא בצורה נקודתית/במשך זמן קצר כפי שטסט מסוגל לבדוק. המשאבים החישוביים שאמורים להיות במחשב ממנו מתבצעת הבדיקה צריכים להיות עצומים על מנת להריץ טסט כזה תוך כמות זמן לגיטימית, כיוון שיש צורך בתמיכה בכמות ת'רדים אדירה לשם כך והרבה כוח חישוב. לאור האמור לעיל הרצת הבדיקה הזו אינה פיזיבילית ולא נוכל לספק תוצאות עבורה, אלא רק להגדיר תרחיש:

נגדיר כעת תרחיש בדיקה שמטרתו לבדוק האם המערכת מסוגלת להתמודד עם כמות הרשומות הצפויה לה.

נפתח 1000 חנויות חדשות ממשתמש רשום במערכת, אותו משתמש יוסיף לאחר מכן 1000 מוצרים חדשים לכל חנות – עם מיליון פריטים זמינים במלאי עבור כל אחד מהם.

כעת, נירשם עם 10,000 משתמשים חדשים למערכת, כל אחד יתחבר, יוסיף מוצר כלשהו לעגלה מאחת החנויות ויקנה אחד ממנו עבור כל אחד מ-1000 המוצרים השונים בחנות, ללא חפיפה), ויתנתק מהמערכת.

לבסוף, מייסד החנות יבקש את היסטוריית הרכישה של החנות ונוודא שהגיעו 10 מיליון רשומות עבור response המתאים.

סה"כ נוצרו מתרחיש זה 1000 רשומות, 1000 חנויות, 10,000 משתמשים, מיליון מוצרים ו-10 מיליון רכישות.