**שרת PROXY מאובטח**

מסמך עיצוב

יובל ערד

גרסה 2

17.2.16

**היסטוריית גרסאות המסמך**

|  |  |  |
| --- | --- | --- |
| **תאריך** | גרסה | **תקציר השינויים** |
| 16.2.16 | 1 |  |
| 17.2.16 | 2 | נוספו הגדרות, מבט על ודיון בארכיטקטורה, מחלקות Request ו compression, עיצוב נתונים |
|  |  |  |
|  |  |  |



**1. הקדמה**

1.1 מטרה

*מסמך זה נועד לתאר את המוצר ואת השינויים בו לאורך שלבי הפיתוח*, מסמך זה מיועד למפתח המוצר על מנת לעקוב אחרי השינויים והעדכונים בתהליך הפיתוח.

1.2 המוצר

* *המוצר הוא שרת PROXY מאובטח, כל תקשורת האינטרנט שמועברת בין הלקוח לשרת מועברת באמצעות SSL כך שהיא מאובטחת.*

1.3 קישור למסמכים קודמים

*קישור למסמכים / קבצים חיצוניים הרלוונטיים למסמך. לדוגמא: מסמך אפיון, עמוד web עם עיצוב הממשק הכללי שאתם מתכננים וכו'.*

1.4 הגדרות

*SSL – פרוטוקול המאפשר חיבור מאובטח ומוצפן בין שרת ללקוח*

*http –* פרוטוקול לתקשרות עם אתרי אינטרנט

דחיסה – תהליך של הקטנת הנפח שמידע תופס, גוף התשובה מהאתר מגיע לפעמים דחוס, ויש להרחיב אותו חזרה עלמנת לשנות פרמטרים ולדחוס חזרה.

1. ארכיטקטורת המערכת

חלק זה כולל את תיאור מבנה המערכת ופירוט המודולים השונים בה

2.1 מבט על

המערכת מחולקת למחלקות כשכל מחלקה אחראית על צד אחד של התקשורת – מחלקה אחת על השרת עצמו, מחלקה אחת על התקשורת מול הלקוח ומחלקה נוספת על התקשורת מול האתר.

2.2 פירוט רכיבי המערכת

Server – מחלקה המייצגת את השרת עצמו, אחראית על יצירת החיבור עם הלקוחות.

Client - מחלקה המייצגת את החיבור בין השרת ללקוח, מכילה socket של החיבור בין השרת ללקוח ואחראית על הטיפול בבקשה של הלקוח, יורשת מהמחלקה Thread על מנת לאפשר עבודה עם מספק לקוחות במקביל.

Request - מחלקה המייצגת את בקשת הHTTP שתישלח לאתר ואת התגובה שתישלח ללקוח, תפקידה לעבד את הבקשה, ולשנות את הפרמטרים שיש לשנות בבקשה ובתגובה של של האתר על מנת שהלקוח יקבל תשובה תקינה

Compression – מחלקה המכילה פונקציות סטטיות שתפקידן לטפל בדחיסה של גוף התשובה מהאתר על מנת שניתן יהיה לשנות פרמטרים.

2.3 דיון בנושא העיצוב הנבחר

בחרתי בחלוקה למחלקות לפי הרכיבים השונים של המערכת – השרת עצמו, התקשורת עם הלקוח והתקשורת עם האתר, בנוסף, יש חלוקה לפונקציות שלכל אחד תפקיד מוגדר, כך נשמר עקרון הכימוס (Encapsulaiton)מה שמאפשר אבחון של בעיות ביתר קלות ודיוק.

בנוסף הפרויקט כתוב בשפת Python שהיא שפה נוחה שמאפשר שינויים קלים בקוד ולכן טובה מאוד לאפליקציות צד שרת שדורשות עדכונים מהירים, ולא בהכרח דורשות ממשק גרפי, אך אפשרות לגשת ולשנות את הקוד במהירות ובנוחות ללא עיכובים.

1. עיצוב נתונים ופרוטוקולים

כאן יופיע תיעוד של מבני נתונים שונים / פרוטוקולים המשמשים אותנו במערכת

השרת מתחבר ללוקח בתקשורת מאובטחת באמצעות SSL, ושולח לו בקשת HTTP, שהיא בקשה טקסטואלית, הבקשה עצמה עוברת שינוי כל, שכל הכתובות של השרת הופכות לכתובות של האתר שאליו תועבר הבקשה, הבקשה לאחר מכן נשלחת לאתר ומתקבלת תשובה טקסטואלית גם היא, בה כל כתובות האינטרנט משונות לכתובות שיעברו דרך השרת ואז היא נשלחת ללקוח.

4. ממשק משתמש

התקשורת עם השרת תיעשה באמצעות בקשות http:

כך על מנת לשלוח בקשת POST לאתר מסוים שכתובתו website\_url תישלח הבקשה הבאה לשרת:

POST /site?url=website\_url

Header1:

Header2:

Post\_params

כשpost\_params הם כל הפרמטרים של בקשת הpost שתישלח לאתר וכל הheaderים הם אלו שישלחו

לאתר.

התשובה שתחוזר תהיה אותה התשובה מהאתר רק שכל כתובות הurl יהיו אלו של העברת הבקשה דרך השרת

5. נספחים

**כל דבר שהייתם רוצים להוסיף בנוגע לעיצוב המערכת**