**שרת PROXY מאובטח**

מסמך עיצוב

יובל ערד

גרסה 1

16.2.16

**היסטוריית גרסאות המסמך**

|  |  |  |
| --- | --- | --- |
| **תאריך** | גרסה | **תקציר השינויים** |
| 16.2.16 | 1 |  |
|  |  |  |
|  |  |  |
|  |  |  |



**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

*בחלק זה יש לציין את כל ההגדרות, המושגים וראשי התיבות בהם תשתמשו בהמשך המסמך.*

1. ארכיטקטורת המערכת

חלק זה כולל את תיאור מבנה המערכת ופירוט המודולים השונים בה

2.1 מבט על

בסעיף זה יש להסביר את החלוקה העיקרית למודולים השונים במערכת, והיחסים/קשרים ביניהם. יש להגדיר באופן ברור את תחום האחריות של כל מודול או רכיב, ולאיזו פונקציונליות (כפי שהוגדרה במסמך האפיון שנכתב לפני כן) הוא מתקשר (בין אם מודול אחד בלבד או מספר מודולים יחד).  
חשוב להסביר לא רק מה/איך נעשה אלא גם מדוע דווקא כך.

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 במקבילץ

2.3 דיון בנושא העיצוב הנבחר

כאן יש לפרט ובעיקר להסביר מדוע בחרתם דווקא בחלוקת הרכיבים/תפקידים הזו ולא אחרת. מה היתרונות שלכם מבחינתה, וגם מה החסרונות שאתם מודעים אליהם. במידה וחשבתם על חלופות אחרות, יש לציין אותן ולהסביר בקצרה כיצד הן שונות מהפתרון שנבחר - ומדוע החלטתם בסופו של דבר שלא לבחור בחלופות האלו.  
כאן יש להתייחס גם לגבי שפת התכנות שבה בחרתם לכתוב כל רכיב, ומדוע דווקא בשפה זו ולא אחרת.

1. עיצוב נתונים ופרוטוקולים

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

בסעיף זה יש לפרט את כל סוגי המידע אשר מועבר או נשמר במערכת - בין אם בין רכיבים, בין מודולים או בכל צורה אחרת (כלומר גם אם זה משמש רק לתקשורת פנימית ולא ע"ג הרשת, או למשל אם זה משהו ששומרים לקובץ/בסיס נתונים).

* לכל סוג מידע כזה יש לפרט על השדות אותן הוא מכיל ומאיזה סוג כל שדה, מה טווח הערכים הרלוונטי לגביו וכל הגבלה או מידע נוסף אחר שאתם מוצאים לנכון (למשל: האם מותר שהשדה יהיה ריק ומה המשמעות של זה, אורך או ערך מינימלי/מקסימלי למספר או למחרוזת)
* יש לציין לאיזו מטרה משמש כל מבנה נתונים.
* עבור פרוטוקול תקשורת-נתונים בין רכיבים שונים:
* מה המצבים השונים לכל סוג של תקשורת (למשל אותנטיקציה, הרשמה, שליחת קובץ וכו')
* באילו ייצוגי מידע נעשה שימוש + תרשים זרימה של המצבים השונים שיכולים להיות וסדר השלבים שלהם (למשל כיצד נעביר מידע על מקרים של סיסמה שגויה, משתמש חסום, התחברות מוצלחת וכו')
* מומלץ להגדיר מראש קודים של בקשות/פעולות, תוצאות/תגובות ושגיאות אשר ישותפו בין כך הרכיבים הרלוונטיים
* יש להתייחס לאופי המידע הנשמר (האם הוא טקסטואלי או בינארי) ולייצגו בצורה נוחה והולמת בהתאם
* יש לפרט אודות מבנה בסיס הנתונים (חלוקה לטבלאות, השם והטיפוס של כל עמודה/שדה, קשרים בין טבלאות ואילוצים כלשהם על עמודות)

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**