2. רקע תיאורטי

2.1 רשתות מחשבים

נסקור כאן בקצרה מצד עולם רשתות המחשבים את הפעולות הנדרשות ליצירת קשר בין ישויות בעולם האינטרנט.

תחילה נגדיר מושגים בהם נשתמש:

1. ישות – היא כל רכיב המחובר לרשת בעל כתובת IP
2. Session – שיחה בין ישויות ברשת האינטרנט, לצורך הקמת השיחה אנחנו חייבים לפתוח socket בכל צד, כלומר בצד הserver ובצד הclient
3. Socket – בהגדרתו מורכב מIP וPORT, למשל כמו צינור בין שני צדדים הרוצים לדבר ביניהם. אמון על העברת התכנים בין האפליקציות בצורה תקינה, כלומר שאם למשל עובדים עם skype ובמקביל הbrowser מבקש דף אינטרנט, חשוב לנו שהתוצאות יתקבלו למקום הנכון ולא יהיה בלבול בין התוכנות.
4. כתובת IP – היא כתובת בשכבה השלישית network layer, עבור כל ישות ברשת האינטרנט, המאפשרת לזהות ולתקשר איתה ברשת.
5. PORT – מוגדר בשכבה הרביעית transport layer, אמון על העברת המידע לאפליקציה הנכונה.

בעולם האינטרנט של ימנו רוב סוגי התקשורת עבור משתמשים היא מסוג client-server (על מודל זה נרחיב בהמשך), כלומר ישנו לקוח אחד או יותר המעוניינים לגשת לserver מסוים ולקחת ממנו נתונים או לחילופין לבקש ממנו לבצע כל מיני פעולות (כמו למשל לבצע חישובים המצריכים עיבוד טוב יותר מאשר הרכיבים במחשב הביתי).

כאשר אנחנו ניגשים להסביר את תהליך הקמת השיחה וקיום השיחה בפועל בין שני הישויות, ישנם הרבה פרטים שעליהם לא נרחיב כאן, אבל באופן יותר כללי אנחנו נדרשים לכך שהserver "יאזין" על PORT וכך הclient יוכל לגשת ולקבל את הנתונים הדרושים.



**4230**

**80**



2.2 מודל client – server

client-server modelהיא ארכיטקטורת תוכנה לחישוב מבוזר, אשר מגדירה את היחס בין תוכנות משתפות פעולה.

המודל מחלק את המשימות או עומס העבודה בין הserver שהוא למעשה ספק השירות של דרישות הלקוח, לבין הclient שהוא מבקש השירות, הלקוח.

כלומר ישנה הפרדה ביניהם באופן כזה שהclient לא צריך לדעת מה קורה עם הניהול מאוחר, למשל אם הוחלט לעבור לDatabase אחר, הserver חייב להיות מעודכן בכך אבל מצד הclient זה "שקוף".

מודל client-server הוא אחד מתצורות ההתקשרות הנפוצות ברשתות מחשבים שהפך לאחד מהרעיונות המרכזיים והרבה applications משתמשים בו.

הserver הוא מחשב בדר"כ חזק יותר מהמחשב הביתי המריץ תוכנה פסיבית, המאזינה על port ידוע (כמו למשל 80 עבור HTTP וכו') או port אחר שהוגדר מראש[[1]](#footnote-1) כלפי הclient ומחכה לקבל בקשות.

הclient לעומתו בדרך כלל מהווה את ממשק המשתמש הוא מופעל על ידי המשתמש ופונה -

לשרת כאשר הוא זקוק למידע או שירותים ממנו.

בדרך כלל, תוכנות השרת והלקוח רצות על גבי מחשבים שונים והתקשורת ביניהן מתבצעת על גבי

רשת מחשבים. עם זאת, תוכנות השרת והלקוח יכולות לפעול גם על גבי אותו מחשב. מכונת שרת

הוא מחשב המריץ תוכנת שרת אחת או יותר, אשר חולקת את המשאבים שלה עם הלקוחות.

הלקוח אינו חולק את המשאבים שלו, אלא מבקש תכנים ושירותים מהשרת.

לפיכך, הלקוח הוא זה שיוזם את ההתקשרות עם השרת, אשר ממתין לבקשות נכנסות.



צד לקוח

יישום הקיים בצד המשתמש (דפדפן, אפליקציות שונות, בין אם מחשב ובין אם טלפון חכם) והוא בדר"כ יוזם החיבור למול השרת לצורך ביצוע פעולות שהוא לא מסוגל לבצע, בין אם מדובר ביכולת חישוב או בגישה למאגרי נתונים (DB) שאין לו יכולת/הרשאה להגיע אליהם.

כל עמוד המוצג למשתמש מורכב למעשה מכמה חלקים:

1. קובץ css האחראי על עיצוב הדף
2. Script הכתוב בjavascript האחראי על הפונקציונאליות של הדף
3. קובץ HTML מגדיר את הדף והפקדים השונים ומכיל script וקישור לקובץ css

צד שרת

כאן מתבצעות הפעולות שמאחורי הבקשות של הלקוח, למשל אם ניקח את פרוטוקול HTTP (אחראי על העברת קבצי HTML) ובו ישנם פקודות GET ו-POST, כאשר הלקוח מקליד כתובת URL הוא למעשה יוצר פקודת GET וניגש לשרת שבמקרה שלנו מאזין על port 80 ומבקש ממנו דף אינטרנט מסוים, השרת מחפש את הדף המבוקש ומעביר אותו ללקוח.

ובמקרה בו הלקוח יבצע פקודת POST הוא מעלה נתונים לשרת והשרת כבר צריך לטפל בהם בין אם זה לבצע פעולות ולאחסן בבסיס נתונים וכו'.

https://www.ibm.com/support/knowledgecenter/en/SSAL2T\_9.1.0/com.ibm.cics.tx.doc/concepts/c\_clnt\_sevr\_model.html

https://en.wikipedia.org/wiki/Client%E2%80%93server\_model

http://webmaster.org.il/articles/how-does-server-side-works

<https://en.wikipedia.org/wiki/Client-side>

<https://skillcrush.com/2012/07/30/client-side-vs-server-side/>

2.3 ניתוח כלי פיתוח

2.3.1 סביבת עבודה

2.3.1.1 pyCharm

סביבה המותאמת לשפת python (מאת Jet Brain) המאפשרת לכתוב תוכנית בצורה מסודרת ו/או ע"פ OOP.

קיים debugger המאפשר לנו להריץ את הקוד step by step ולצפות במשתנים.

מתריע על שגיאות syntax וקונבנציות 8PEP .

מקל על איתחול התוכנית לאחר קריסה או טעות.

<https://confluence.jetbrains.com/display/PYH/Getting+Started+with+PyCharm>

2.3.1.2 notepad++

2.3.1.3 Tableau

2.3.1.4 Node.js command prompt

2.3.1.5 Firebase

2.3.1.6 **A**mazon **W**eb **S**ervices

2.3.2 שפות פיתוח

Javascript2.3.2.1

היא שפת תוכנה (ללא כל קשר לשפת java) ליצירת אתרים דינאמיים, בהם השינויים באתר ובדף האינטרנט לא מצריכים טעינה של הדף מחדש אלא מורצים בצד הלקוח.

בניגוד לשפות כמו Java שמתורגמות לשפת מכונה (compiled), שפת JavaScript מפורשת על ידי הדפדפן (או כל כלי אחר) בזמן אמת. זה הופך את השפה ליותר איטית בהשוואה לשפות מתורגמות אך זהו עניין זניח כשמדובר באתרי אינטרנט שרובם ככולם לא דורשים עיבודים מורכבים.

בתחילת דרכה, עיקר השימוש של השפה היה הרצת קוד בתוך הדפדפן בצד הלקוח. אולם מספר שנים לאחר מכן השפה פרצה את תחום הדפדפנים ועברה להיות שפה עצמאית שניתן ליצור איתה הכל, ואפילו להשתמש בה לתכנות צד שרת. הדבר נעשה אפשרי אודות לפיתוח סביבת ההרצה הנקראת [Node.js](https://nodejs.org/). בעזרת Node ניתן להריץ JavaScript ללא צורך בדפדפן. Node משתמש במנוע להרצת JavaScript שפותח על ידי Google ונקרא V8. על ידי שימוש בNode ניתן לפתח אפליקציות מבוססות JavaScript למכשירים ניידים ונייחים ולמגוון מערכות הפעלה.

<https://he.wikipedia.org/wiki/%D7%90%D7%AA%D7%A8_%D7%90%D7%99%D7%A0%D7%98%D7%A8%D7%A0%D7%98>

<https://skillcrush.com/2012/04/05/javascript/>

HTML2.3.2.2

CSS2.3.2.3

Python2.3.2.4

postgreSQL2.3.2.5

1. כיוון שכדי שהclient יוכל ליצור קשר למול הserver שלנו הוא נדרש לדעת לאיזה port לגשת [↑](#footnote-ref-1)