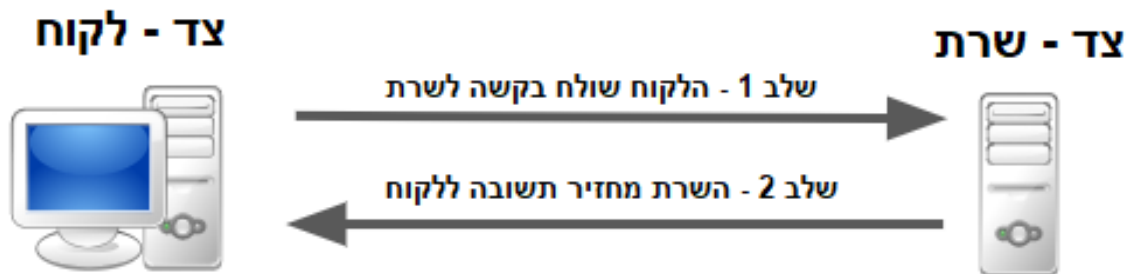


בקשות http

- כאשר בונים מערכות ממוחשבות, עולה הצורך של קבלת מידע מהשרת. מידע שאין אפשרות להפיק מקומית בצד לקוח מבלי לפנות לשירות מרוחק.
לדוגמא: קבלת שערי המרת מטבע, תחזית מזג אויר, מידע השמור במסד נתונים מרכזי ועוד.
- העבודה במודל שרת לקוח מתבססת על פרוטוקול http.
בכל פניה של הלקוח לשרת, מתבצע עיבוד של הבקשה וחזרה של תשובה המתאימה ללקוח.
את פרוטוקול http ניתן לחלק לשני חלקים מרכזיים:
 1. **http request message** - בקשה שנישלחת מהלקוח אל השרת
 2. **http response message** - התשובה שמוחזרת מהשרת.



סוגי verb

Verb - מגדיר את צורת העברת הנתונים הרצויה, או במילים אחרות ב method המתאים:

- Get – העברת הפרמטרים ב querystring.
- Post – העברת הפרמטרים בגוף הבקשה.

נוסף על צורת העברת הנתונים, קיימת מוסכמה על משמעות השימוש בכל אחת מהדרכים. להלן רשימה של המוסכמות העיקריות:

- Get – שליפת מידע.

- Post – הכנסת רשומה חדשה.
- Put – עדכון רשומה.
- Delete – מחיקת רשומה.

מבנה בקשה

בכל בקשה נשלח http request message מהלקוח לשרת. ההודעה כוללת את הנתונים הבאים:

- **head:**
 - Url
 - (get/post/put/delete) Verb
- **Body**

מבנה תשובה

התשובה המוחזרת מהשרות ללקוח כוללת את הנתונים הבאים:

- **head:**
 - Url
 - status code
- **Body**

בתשובת השרת ללקוח (response) נשלח גם http message אשר מכיל את אותם המאפיינים שנשלחו ב request עם שינוי קטן. במקום verb מתקבל status code.

status code המתאר את התשובה שהתקבלה מהשרת. דוגמה ל status-ים ידועים:

- 200 - OK
- 401 - Unauthorized
- 403 – Forbidden
- 404 - Not Found
- 500 - Internal Server Error

דוגמא לשימוש של status code:

- ננסה לשלוח פרטי מוצר לפי ID.
- במקרה שה ID אינו קיים במערכת, נחזיר ללקוח תשובה כי המוצר לא קיים במערכת.
- במקרה שלא ארעה שגיאה ולא הוגדר במפורש קוד חזרה מסויים - מנגנון ה http מחזיר תשובה עם קוד 200 – ok.

כאשר מנסים לגשת למשאב שאינו קיים במערכת, הקוד שפרוטוקול ה http שולח הוא 404. כלומר, גם במקרה שננסה לשלוח מוצר שאינו קיים, נרצה להחזיר את אותו קוד. כך לפי המיספר שחוזר ניתן לדעת אם הפניה הצליחה או ניכשלה ואת הסיבה לכישלון כגון הגדרת אבטחה או כתובת שגויה של השירות ועוד

advanced rest client

פעמים רבות - במהלך בניית השירות, נרצה לבדוק ולשלוח בקשות מסוגים שונים אל השרת עוד לפני יצירת קוד הAJAX.

למטרה זו נשתמש בכלים שונים המספקים לנו אפשרות לשלוח בקשות http לשרת (כגון: fiddler postman ועוד).

כלי חינוכי, נח ויעיל מאד לשימוש, הוא advanced rest client:



ARC

הניתן להורדה בקישור: <https://chrome.google.com/webstore/category/extensions>



כאשר נפתח את הכלי, נקבל את התצוגה הבאה:

Request

Use XHR

> Request URL

כאן נרשום את ה-url המלא אליו נרצה לפנות

☐ GET

☒ POST

☐ PUT

☐ DELETE

☐ PATCH

כאן נבחר את סוג הבקשה

Other methods

Custom content type

Raw headers

Headers form

Headers sets

Variables

כאן נוכל להוסיף ערכים לheader של request

A✓

0 bytes

Raw payload

Data form

Files

כאן נוכל להוסיף ערכים לbody של request

שליחת הבקשה לשירות

SEND