

## תרגיל 4

הרפו ריק, יש להוסיף קובץ README.md ואת הקבצים לפי המבנה של תרגיל 3. אין פיתוח צד שרת (יהיה בתרגיל 5).  
סרטון הסבר:

<https://hac.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=eff63c24-04ad-474c-990d-b10900c97dbc>

### • נביאים:

- <https://classroom.github.com/a/VF90otRh>

תאריך הגשה: 12/02/2024 23:59

### • שטראוס נשים:

- [https://classroom.github.com/a/6il\\_OGg\\_](https://classroom.github.com/a/6il_OGg_)

תאריך הגשה: 13/02/2024 23:59

### • שטראוס גברים:

- <https://classroom.github.com/a/J8M34pKP>

תאריך הגשה: 15/02/2024 23:59

## מטרת התרגיל: בניית אתר גישה למאגר תמונות מאדים של NASA בעזרת rest api

יש לבנות אתר כדף אחד (SPA) המאפשר:

1. חיפוש תמונות
2. שמירת תמונות לרשימת מועדפים
3. הצגת הרשימה בקרסלה

עליכם לקרוא ולהשיג את המידע באתר ה-NASA: <https://api.nasa.gov>  
תחת הלשונית Mars Rover Photos (זהו חלק המשימה שלכם: להתמודד עם REST API חיצוני). שימו לב  
יש רשימה ארוכה של מקורות באתר NASA. אנחנו מעוניינים רק ב-  
**MARS Rover Photos** (Image Data Gathered by NASA's Curiosity)

## הכנה לפיתוח: קבלת token

בתרגיל זה אתם תשתמשו ב- REST API פתוח המסופק על ידי אתר [api.nasa.com](https://api.nasa.com).  
חייבים להשתמש במפתח (API key) על מנת לשלוח בקשות. קיים מפתח "DEMO\_KEY" זמין בכל רגע אבל  
מפתח זה מוגבל ל-50 בקשות ביום מתוך אותה כתובת IP.  
לכן יש צורך לבקש מפתח פרטי.

חפשו את הטופס Generate API KEY:

1. יש להירשם
2. תקבלו במייל API KEY – זהו מפתח אישי שתצרכו כפרמטר לכל בקשה http
3. דאגו להגדיר אותו כ- **const API\_KEY = "xyx"** בתחילת הקוד javascript שלכם כך שהבודק יוכל להחליף אותו בקלות ע"י סקריפט.

שימוש במפתח:

יש לצרף את המפתח לכל בקשה http שתכתבו (fetch()) לדוגמא:

```
https://api.nasa.gov/mars-photos/api/v1/rovers?api_key=abcdefghijklmno
```

שימו לב, גם למפתח אישי יש הגבלה (כ-1000 בקשות בשעה). אם תגיעו ל-rate limit תצטרכו להמתין שעה או להשתמש במפתח אחר (כדאי שיהיו לכם 2).

## מאדים ותאריכים

קודם קצת מידע על מאדים:

Mars is a planet with a very similar daily cycle to the Earth.

The time it takes for a planet to spin on its axis so that the Sun appears in the same position in the daytime sky is known as solar day.

Mars solar is about day 24 hours, 39 minutes and 35 seconds.

A Martian day (referred to as "**sol**") is approximately 40 minutes longer than a day on Earth.

<https://proulpx.github.io/CS290-How-To-Guide/rover.html>

לכן כאשר נתייחס לתאריך נוכל להזין תאריך כדור הארץ או לפי מספר (יום) SOL.

## Rovers ומצלמות

את כל המידע על ה-api תוכלו למצוא באתרים

- <https://api.nasa.gov>
- <https://github.com/corinceraami/mars-photo-api>

תפקידכם קודם לחקור את ה-api ולמצוא את ה-endpoints הרלוונטיים לצרכים שלכם על מנת להשיג מידע עדכני. לדוגמה לכל Rover יש את המצלמות שלה. את זה חייבים להשיג דרך ה-api. אין לכתוב בצורה סטטית מידע על rovers and camera.

רמז: את רשימת ה-Rovers ניתן להשיג ע"י <https://api.nasa.gov/mars-photos/api/v1/rovers>

## דרישות התרגיל

### 1 - לצורך וולידציה, ראשית נרצה להשיג מהשרת (פעם אחת) עבור כל rover את

- a. תאריך ההתחלה: `landing_date`
- b. תאריך כדור הארץ עדכני ביותר: `max_date`
- c. תאריך SOL עדכני ביותר: `max_sol`
- d. רשימת המצלמות: `cameras`

כך נבצע וולידציה בהמשך ללא גישה לשרת.

### 2 - ממשק משתמש

**טופס חיפוש:**

1. יש לבחור rover ותאריך (או ההפך), ואחר כך מצלמה (חייבים לבדוק בוולידציה שהמצלמה הזמינה בהתאם ל-rover הנבחר).
2. יש לאפשר הזנת תאריך לפי תאריך כדור הארץ או לפי מספר סול. יש לבדוק בוולידציה שהתאריך נמצא בטווח שהוגדר עבור ה-rover הנבחר. באופן כללי ניתן או לבדוק אחרי הזנה, או להגביל מראש את הקלט מיד אחרי בחירת ה-rover כלומר לעדכן ב-select את התוכן תוך כדי בחירה.
3. מומלץ לחייב בחירת מצלמה או להגביל את מספר התמונות אם זה אפשרי (אחרת תקבלו אלפי תמונות).
4. כפתור חיפוש יבצע את החיפוש בהתאם לשלושת הפרמטרים  
דוגמה לחיפוש (חקרו את התשובה ב-inspector):

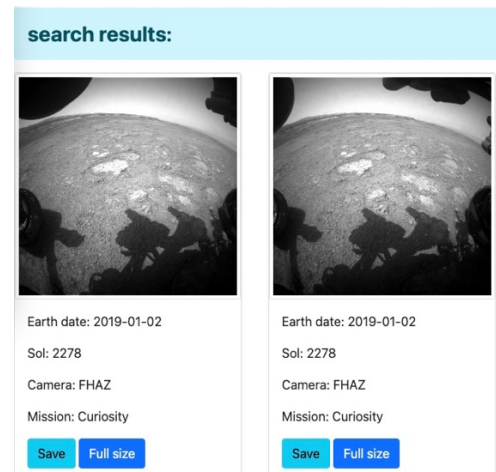
[https://api.nasa.gov/mars-photos/api/v1/rovers/curiosity/photos?sol=1000&camera=fhaz&api\\_key=DEMO\\_KEY](https://api.nasa.gov/mars-photos/api/v1/rovers/curiosity/photos?sol=1000&camera=fhaz&api_key=DEMO_KEY)

5. אם אין תמונות בתוצאת חיפוש יש להציג הודעה למשתמש בהתאם
  6. כפתור איפוס (מוחק כל קלט -אם יש dropdown/select יש להציג מצב ללא בחירה – מוחק גם את כל תוצאות החיפוש והודעות שגיאה למיניהם)
  7. כפתור למעבר לרשימת התמונות המועדפות (אין להציג את הטופס במצב זה)  
אין להזין ערכים דפולטיבים בטופס.
- יש להציג gif המתנה עם אנימציה כלשהי, בזמן ביצוע כל פעולה מול ה-api של צד שרת.

### תוצאות חיפוש:

- הצגת כל תמונה בגודל מצומצם עם
1. מידע : תאריך, מצלמה, rover
  2. כפתור שמירה (ברשימת המועדפים, ללא כפילויות). מודיעה על הצלחת השמירה ועל התמונה כבר שמורה ע"י modal.
  3. כפתור תצוגה (בחלון נפרד ברזולוציה מלאה).

דוגמא:



### רשימת התמונות המועדפות

רשימת התמונות תוצג ברשימה שתאפשר מחיקה של תמונה מהרשימה (אפשר להציג בטבלה, אפשר להציג רק את התאריך ומצלמה).  
יהיה כפתור "חזרה" לחזרה לטופס החיפוש.  
יהיה כפתור הפעלה/הפסקה של קרוסלה מוצג תחת הרשימה, ראו  
(<https://getbootstrap.com/docs/5.3/components/carousel/>)

### וולידציה

מותר להשתמש בוולידציה מובנת של html אך ברור שצריך לבדוק את הקלט בהתאם לנתונים ב-(1). במידה ומאפשרים קלט שגוי יש לדאוג להציג הודעה ספציפית עבור הקלט: קלט חסר, תאריך לא תואם לטווח של המשימה שנבחרה – יש לציין את הטווח (אתם משיגים מידע זה בתחילת התכנית). **ברמת התקשורת, אין לשלוח בקשה http ל-nasa אם הערכים לא עומדים בדרישות : תאריך בטווח המותר, rover ומצלמה השייכת לה.**

### שגיאות רשת

במקרה של error בגישה לשרת או כל תשובה עם סטטוס שונה מ-2XX, יש להציג הודעת שגיאה בעלת משמעות כגון  
"Search did not return a valid response, try again later"

"NASA servers are not available right now, please try again later".

### 3 - קוד Javascript

נקפיד על כתיבת קוד גנרי, מסודר, בסגנון תכנות פונקציונלית, סינטקס עדכני.  
יש לדאוג ככל האפשר לכתיבת פונקציות ומחלקות עם תפקיד ממוקד כגון וולידציה, יצירת html, תקשורת עם השרת.

### 4 - עיצוב

נקפיד על עיצוב, כרגיל רספונסיבי מבווס bootstrap. שימוש ב-css אך ורק אם אין פתרון ב-bootstrap (כגון הוספת תמונת ריקע).

### טיפים

קראו היטב את אתר NASA.  
ה-inspector של chrome הוא החבר הכי טוב שלכם: לשונית הרשת על מנת לחקור את ה-http. debugger.  
המאפשר לשים breakpoint ב-sources.

בהצלחה!

### הנחיות נוספות

1. עיצוב רספונסיבי לבחירתכם, כתבו את האתר באנגלית בלבד, השתמשו ב-bootstrap 5.
2. שם הדף יהיה index.html.
3. שמות קבצים אחרים לבחירתכם. דאגו לארגון קבצים בתיקיות JS/images. שימו לב לא לכתוב נתיבים שלמים (אין לכתוב http://localhost... בקוד שלכם כל עוד מדובר בקבצים שלכם).
4. יש לכתוב את הקוד Ajax רק באמצעות Promises/fetch/async. אין להשתמש בספריה חיצונית או לכתוב קוד על בסיס XMLHttpRequest.
5. כל דבר שלא מוגדר חד משמעית ואושר ע"י המרצה לבחירתכם חייב תיעוד ב-readme.
6. דאגו תמיד להציג תוכן עקבי, עדכני וברור, לא ליצור מצבים בהם המשתמש נופל "בין הכיסאות" (לדוגמא להימנע מהודעות לא רלוונטיות, וולידציה חלקית/שגויה, תצוגה לא ברורה).
7. מומלץ לעבור על רשימת טעויות נפוצות לפני הגשה.