



Calendar Skill

בדקו והוסיפו אירועים ליומן גוגל באמצעות Apps Script

זמן התקנה: 10-15 דקות

מה הסקיל הזה עושה?

הסקיל מאפשר לקלוד לקרוא ולנהל את יומן גוגל שלכם. תוכלו לשאול "מה יש לי היום?" או "הוסף פגישה מחר ב-10" וקלוד יטפל בזה!



אירועים קרובים

מה יש בשעות הקרובות



תצוגה שבועית

כל האירועים של השבוע



אירועי היום

צפייה בכל האירועים של היום



הוספת אירועים

יצירת אירועים חדשים

1 מה צריך לפני שמתחילים?

חשבון גוגל

צריך חשבון גוגל עם יומן פעיל. זה בחינם!

Node.js מותקן

בדקו שיש לכם Node.js:

```
node --version
```

טיפ: הסקיל משתמש ב- **clasp** - הכלי הרשמי של גוגל לניהול Apps Script מהטרמינל.



2 התקנת Clasp

Clasp הוא CLI של גוגל לניהול פרויקטי Apps Script:

```
# התקנה גלובלית  
npm install -g @google/clasp
```

התחברות לגוגל

```
clasp login
```

יפתח דפדפן להתחברות עם חשבון גוגל שלכם.

הפעלת API: אם מקבלים שגיאה, היכנסו ל-[הגדרות Apps Script](#) והפעילו את "Google Apps Script API".



יצירת פרויקט Apps Script

3

צרו תיקייה חדשה

1

```
mkdir calendar-api && cd calendar-api
```

צרו פרויקט Apps Script

2

```
clasp create --title "Calendar API" --type webapp
```

זה יצור קבצי `appsscript.json` ו-`.clasp.json`

העתיקו את קוד ה-API

3

צרו קובץ `Code.gs` עם התוכן הבא:

```
/**
 * Google Calendar API
 * Deploy as Web App
 */

const AUTH_TOKEN = 'YOUR_SECRET_TOKEN_HERE';

function doGet(e) {
  const token = e.parameter.token;
  if (token !== AUTH_TOKEN) {
    return jsonResponse({ error: 'Unauthorized' });
  }

  const action = e.parameter.action || 'today';
  let events;

  switch (action) {
    case 'today': events = getTodayEvents(); break;
    case 'week': events = getWeekEvents(); break;
    case 'upcoming':
      const hours = parseInt(e.parameter.hours) || 4;

```

```
        events = getUpcomingEvents(hours);
        break;
    }

    return jsonResponse({ count: events.length, events });
}

// ... (ראו קוד מלא ב-GitHub)
```

הקוד המלא נמצא בקובץ `references/calendar-api.gs` בסקיל.

הגדירו טוקן סודי

4

החליפו את `YOUR_SECRET_TOKEN_HERE` בסיסמה שתבחרו.

העלאה ופריסה

4

העלו את הקוד לגוגל

1

```
clasp push
```

פרסו כ-Web App

2

```
clasp deploy --description "Calendar API v1"
```

קבלו את ה-URL

3

```
clasp deployments
```

העתיקו את ה-URL שמסתיים ב- `/exec`

Deploy → לפתיחה בדפדפן ופרסו משם דרך

clasp open

אפשרות נוספת: הריצו



.New deployment

5 הגדרת הסקיל

אחרי שיש לכם URL וטוקן, בקשו מקלוד:

עם calendar עדכן את הסקיל

URL: `https://script.google.com/macros/s/YOUR_DEPLOYMENT_ID/exec`

TOKEN: הטוקן שבחרתם

או עדכנו ידנית את קובץ ה-SKILL.md בתיקיית הסקיל.

דוגמאות לשימוש

אחרי ההגדרה, פשוט תשאלו את קלוד:

בדיקת אירועים

"?מה יש לי היום ביומן"

"?מה יש לי השבוע"

"?יש לי פגישות ב-4 שעות הקרובות"

הוספת אירועים

"הוסף פגישה עם דני מחר ב-10:00 עד 11:00"

"קבע לי תזכורת לשיחה ביום רביעי ב-14:00"

פעולות זמינות

פרמטרים

תיאור

פעולה

-	אירועי היום	today
-	אירועי השבוע	week
(ברירת מחדל: 4) hours	שעות קרובות	upcoming
end , start	טווח תאריכים	range
endTime , startTime , title	הוספת אירוע	add

פקודות Clasp שימושיות

התחברות והתנתקות

```
clasp login
clasp logout
```

ניהול פרויקט

```
clasp push      # העלאה לגוגל
clasp pull      # הורדה מגוגל
clasp open      # פתיחה בדפדפן
```

פריסה

```
clasp deploy      # פריסה חדשה
clasp deployments # רשימת פריסות
clasp undeploy ID # הסרת פריסה
```

דיבאג

```
clasp logs      # צפייה בלוגים
```

פתרון בעיות

"Script API not enabled"

היכנסו ל-[הגדרות Apps Script](#) והפעילו את ה-API.

"Unauthorized"

- וודאו שהטוקן בקוד תואם לטוקן שאתם שולחים
- בדקו שאין רווחים מיותרים

"Not found"

- וודאו שאתם משתמשים ב-URL שמסתיים ב- `/exec`
- בדקו שהפריסה פעילה

אין הרשאות ליומן

וודאו שבפריסה בחרתם:

- Execute as: **Me**
- Who has access: **Anyone**

קוד מלא ב-GitHub 

← חזרה לדף הראשי

יש שאלות? פתחו Issue ב-GitHub