

מדריך להגדרת שליחת מייל באמצעות SMTP של גוגל:

*תחילה מומלץ להשתמש ב TLS ובפורט 587.

ניצור משתמש ב Gmail – (כדאי לאמת עם מס' טלפון – יכול לעזור)

*חשוב לשים סיסמה חזקה (אפשר להיכנס לכל אתר Generate password – אנחנו שמנו באורך 8)

שלב ראשון – להגדיר את כל הנתונים בקוד (בד"כ ה Username-וה Sender יהיו זהים)

```
26 app.config.update(dict(  
27     DEBUG = True,  
28     MAIL_SERVER = 'smtp.gmail.com',  
29     MAIL_PORT = 587,  
30     MAIL_USE_TLS = True,  
31     MAIL_USE_SSL = False,  
32     MAIL_DEFAULT_SENDER = _____ ,  
33     MAIL_USERNAME = _____ ,  
34     MAIL_PASSWORD = _____ ,  
35 ))
```

נבצע מתחת את הפקודה הבאה:

```
mail = Mail(app)
```

כעת כדי לשלוח אימייל נשתמש בפונ':

```
def sendEmail(subject, body, participants):  
    if len(participants) > 0:  
        msg = Message(subject, recipients=participants)  
        msg.body = body  
        mail.send(msg)
```

לאחר מכן צריך להיכנס לקישור הבא ולאפשר את ההגדרה: (בד"כ השגיאה תהיה עם המס' 535)

<https://myaccount.google.com/lesssecureapps>

כעת נסה לשלוח מייל, במידה ותהיה שגיאה נוספת (בד"כ השגיאה תהיה עם המס' 534)

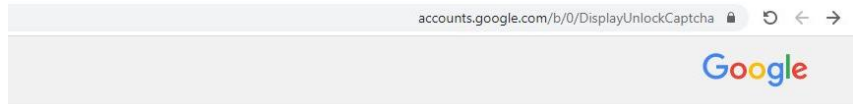
*חשוב מאוד – לסגור את כל הדפדפנים, להשאיר דפדפן אחד פתוח(לא גלישה פרטית) והוא מחובר למשתמש ב Google.

ניכנס לקישור:

<https://accounts.google.com/DisplayUnlockCaptcha>

ונעשה אישור.

*חשוב להשאיר את ה Tab פתוח ובמקביל לבצע ניסיון של שליחת אימייל. (אנחנו השארנו יומיים)



הגישה לחשבון הופעלה

נסה להיכנס שוב לחשבון Google מהמכשיר החדש שלך או מהאפליקציה.

הגדרה נוספת שיכולה לעזור היא לאפשר Pop ו Imap.

ניכנס לקישור:

<https://mail.google.com/mail/#settings/fwdandpop>

אנחנו הגדרנו את ההגדרות כך:

קישורים בהם נעזרנו:

<https://stackoverflow.com/questions/16512592/login-credentials-not-working-with-gmail-smtp>

<https://www.youtube.com/watch?v=BfYsdNaHrps>

Kaffeine heroku

- מכיוון שבחרנו להעלות לשרת חינמי של Heroku, אם במשך חצי שעה אין תנועה לאתר, הוא נכנס למצב של שינה וכך למעשה המתזמן (scheduler) שאנו משתמשים בו כדי לעדכן את המשתמשים באמצעות שליחת מייל, הופך ללא רלוונטי.

לכן על מנת להשאיר את האתר "קשוב" למתזמן נאלצנו למצוא דרך מקורית כדי שתנועה לאתר לא תפסיק במקרה והמשתמשים לא נכנסו אליו.

בחרנו להשתמש באתר שנקרא: [/http://kaffeine.herokuapp.com](http://kaffeine.herokuapp.com)
אתר זה שולח ping כל חצי שעה לאתר ובכך משאיר את האתר ער וקשוב למתזמן.

פתרונות נוספים ניתן למצוא בקישור:
<https://stackoverflow.com/questions/5480337/easy-way-to-prevent-heroku-idling>

Auth דרך Gmail

הגדרות התחברות דרך Gmail:

```
CONF_URL = 'https://accounts.google.com/.well-known/openid-configuration'
oauth = OAuth(app)
google = oauth.register(
    name='google',
    client_id='1028413226216-2ivlgf73ho77h09de0js4mur4kos4gr6.apps.googleusercontent.com',
    client_secret='TZD2-5wPZI_UBmXAtmuzxI5d',
    access_token_url='https://accounts.google.com/o/oauth2/token',
    access_token_params=None,
    authorize_url='https://accounts.google.com/o/oauth2/auth',
    authorize_params=None,
    api_base_url='https://www.google.com/oauth2/v1/',
    server_metadata_url=CONF_URL,
    client_kwargs={'scope': 'openid email profile'})
```

נתיב ופונל התחברות:

```
@app.route('/gmailLogin/')
def gmailLogin():

    google = oauth.create_client('google')
    redirect_url = url_for('auth', _external=True)
    return google.authorize_redirect(redirect_url)

@app.route('/auth/')
def auth():

    google = oauth.create_client('google')
    token = google.authorize_access_token()
    resp = google.get("https://www.googleapis.com/oauth2/v2/userinfo")
    user_info = resp.json()
    email = user_info["email"].lower()
    firstName = user_info["given_name"]
    lastName = user_info["family_name"]

    existing_user = users.find_one({'email': email})

    if existing_user is None:
        users.insert({'email': email, 'password': None, 'firstName': firstName, 'lastName': lastName, 'state': "",
                      'phoneNum': "", 'favoritesTenders': [], 'confirmed': None})
        session["email"] = email
    else: # user exists
        if existing_user["confirmed"] is None: # already registered via Gmail at least once
            session["email"] = email
        else: # the user already registered in the regular way so cannot login with gmail
            session["need_login_with_pwd"] = True
    return redirect(url_for("home"))
```