

פונקציות Lambda – אתר Shortly

change_link_password

הסבר קצר:

פונקציה זו מעדכנת את הסיסמה של לינק ספציפי. היא בודקת בעלות על הלינק, מוודאת שהלינק מוגן בסיסמה, מאמתת את הסיסמה הנוכחית ולאחר מכן מעדכנת את הסיסמה החדשה ב-DynamoDB.

קלט לדוגמה:

```
}
body": "{ \"LinkId\": \"link123\", \"userId\": \"userA\", \"currentPassword\": \"oldPass\", \"newPassword\": \"newPass123\" }
```

delete_image

הסבר קצר:

פונקציה זו מאפסת את תמונת הפרופיל של משתמש לתמונת ברירת מחדל או לתמונת placeholder. ספציפית ב-DynamoDB. היא אינה מוחקת בפועל קבצים מ-S3.

קלט לדוגמה:

```
}
body": "{ \"userId\": \"user123\", \"pictureUrl\": \"https://example.com/new_default_avatar.jpg\" }
```

create_mailing_list

הסבר קצר:

פונקציה זו יוצרת רשימת תפוצה חדשה. היא מקבלת `initiatorId` (מזהה יוצר הרשימה), שם לרשימה (name), רשימת חברים (`recipientsIds`), ורשימת כתובות אימייל (`recipientsEmails`). הפונקציה מחזירה את כתובות האימייל של החברים שצוינו, מאחדת אותן עם האימיילים (תוך טיפול בכפילויות), ושומרת את הרשימה בטבלת `Mailing_Lists`.

קלט לדוגמה:

```
}
body": {"initiatorId": "userABC", "name": "My New Group", "recipientsIds":
  ["friend1", "friend2"], "recipientsEmails": ["email1@example.com",
    "email2@example.com"]
}
```

get_all_links

הסבר קצר:

פונקציה זו מחזירה את כל הלינקים מטבלת ה-`Links`. היא מבצעת סריקה מלאה של הטבלה ומחזירה את כל הפריטים שנמצאו.

קלט לדוגמה (אין צורך ב-body):

```
{}
```

check_unread_notifications

הסבר קצר:

פונקציה זו בודקת אם למשתמש מסוים יש התראות שעדיין לא נקראו. היא מאמתת את קיום המשתמש ולאחר מכן סורקת את טבלת ההתראות כדי למצוא התראות השייכות למזהה המשתמש וסטטוס `IsRead` השווה ל-0 (לא נקרא).

קלט לדוגמה:

```
}
body": {"UserId": "some-user-id-123"}
{}
```

ban_user

הסבר קצר:

פונקציה זו משנה את סטטוס הפעילות (**IsActive**) של משתמש ושל כל הלינקים שבבעלותו. היא מקבלת **UserId** ודגל בוליאני **IsActive**. אם המשתמש הופך ללא פעיל (חסום), גם הלינקים שלו מושבתים.

קלט לדוגמה:

```
}
{"body": "{\"UserId\": \"userXYZ\", \"IsActive\": false"}
{
```

get_all_active_links

הסבר קצר:

פונקציה זו מחזירה את כל הלינקים שהם גם ציבוריים וגם פעילים. היא מבצעת סריקה של טבלת ה-**Links** ומסננת את התוצאות כדי לכלול רק לינקים שבהם **IsActive** הוא **true**.

קלט לדוגמה (אין צורך ב-body):

```
{}
```

delete_link

הסבר קצר:

פונקציה זו "מבטלת" לינק על ידי שינוי סטטוס **IsActive** שלו ל-**false** ב-**DynamoDB**. היא אינה מוחקת את הלינק בפועל מהטבלה, אלא מסמנת אותו כלא פעיל.

קלט לדוגמה:

```
}
{"body": "{\"LinkId\": \"link-to-deactivate-123"}
{
```

get_all_notifications

הסבר קצר:

פונקציה זו מחזירה את כל ההתראות עבור משתמש נתון, וממיינת אותן לבקשות חברות ממתיונות והתראות אחרות (שאינן נקראו ובנות 7 ימים אחרונים). עבור בקשות חברות, היא גם מחזירה את פרטי השולח (שם משתמש ותמונה) מטבלת המשתמשים.

קלט לדוגמה:

```
}
{"body": {"userId": "some-user-id-123"}
{
```

get_user_mailing_lists

הסבר קצר:

פונקציה זו מחזירה את כל רשימות התפוצה שנוצרו על ידי משתמש מסוים ([InitiatorId](#)). היא מבצעת פעולת [Scan](#) על טבלת [Mailing_Lists](#) ב-[DynamoDB](#), כלומר סורקת את כל הטבלה, ומסננת את התוצאות לפי מזהה המשתמש שסופק. הפונקציה מחזירה רק את מזהה הרשימה ([ListId](#)) ושם הרשימה ([ListName](#)).

קלט לדוגמה (מה שהפונקציה מקבלת):

```
{"body": {"InitiatorId": "user-123-abc"}}
```

get_link_details

הסבר קצר:

פונקציה זו מספקת מידע מפורט על קישור יחיד. היא שולפת את הגדרות הקישור (כמו האם הוא פרטי או מוגן בסיסמה), סופרת את סך הלחיצות עליו, ומסכמת את נתוני הלחיצות לפי מדינות. המידע נאסף מטבלת [Links](#).

קלט לדוגמה:

```
{ "body": {"linkId": "link-xyz-789"} }
```

new_notif

הסבר קצר:

פונקציה זו יוצרת התראה חדשה ושומרת אותה במסד הנתונים (טבלת **Notifications**). היא מקבלת את פרטי ההתראה (למי מיועדת, התוכן, וכו'), מייצרת עבודה מזהה ייחודי (**NotifId**), ומוסיפה אותה לטבלה.

קלט לדוגמה:

```
{
"body": "{\"ToUserId\": \"user-456\", \"FromUserId\": \"user-123\", \"Text\": \"You have a new friend request!\", \"LinkId\": \"link-abc\"}"
}
```

get links by user id

הסבר קצר:

פונקציה זו מחזירה את כל הקישורים שנוצרו על ידי משתמש ספציפי. היא מנסה לבצע שאילתה יעילה באמצעות אינדקס (**GSI**), ואם האינדקס לא קיים, היא חוזרת לחלופה פחות יעילה של סריקת כל הטבלה (**Scan**) כדי למצוא את הקישורים הרלוונטיים.

קלט לדוגמה:

```
{
"body": "{\"UserId\": \"user-xyz-123\"}"
}
```

get user links

הסבר קצר:

פונקציה זו מחזירה את כל הקישורים הציבוריים (**IsPrivate: false**) והפעילים (**IsActive: true**) של משתמש מסוים. היא מבצעת שאילתה כדי לקבל את כל הקישורים של המשתמש, ולאחר מכן מסננת אותם ברמת הקוד כדי להחזיר רק את אלו שעונים על שני התנאים.

קלט לדוגמה:

```
{
"body": "{\"UserId\": \"a1\"}"
}
```

get_user_by_id

הסבר קצר:

פונקציה זו מרכיבה פרופיל משתמש מלא על ידי איסוף נתונים ממספר טבלאות. היא שולפת מידע בסיסי על המשתמש, את רשימת ההישגים שלו (כולל פרטי ההישג), ואת הקישורים שיצר. הפונקציה גם מבצעת סינון פרטיות: אם המשתמש שצופה בפרופיל אינו בעל הפרופיל, קישורים פרטיים יוסתרו מהתשובה.

קלט לדוגמה:

```
{ "body": "{ \"ProfileOwnerId\": \"user-profile-456\", \"LoggedInUserId\": \"user-viewer-123\"}" }
```

is_user_banned

הסבר קצר:

פונקציה זו בודקת אם חשבון משתמש מסוים פעיל או חסום. היא ניגשת לטבלת המשתמשים, שולפת רק את השדה `IsActive` (כדי לחסוך במשאבים), ומחזירה תשובה בוליאנית (`true` אם פעיל, `false` אם לא).

קלט לדוגמה:

```
{  
  "body": "{ \"userId\": \"user-to-check-789\"}"  
}
```

mark_notifications_as_read

הסבר קצר:

פונקציה זו מסמנת את כל ההתראות של משתמש מסוים כ"נקראו". היא מאתרת את כל ההתראות שטרם נקראו עבור המשתמש באמצעות שאילתה, ולאחר מכן עוברת על כל אחת מהן ומעדכנת את הסטטוס שלה ל"נקרא" (`IsRead = 1`).

קלט לדוגמה:

```
{  
  "body": "{ \"userId\": \"user-abc-456\"}"  
}
```

get_public_links

הסבר קצר:

פונקציה זו מיועדת להצגת "פיד" ציבורי של קישורים. היא סורקת את כל טבלת הקישורים ומסננת אותם ברמת מסד הנתונים כדי לקבל רק קישורים שהם גם ציבוריים (`IsPrivate: false`) וגם פעילים (`IsActive: true`). היא גם מבקשת רק שדות מידע ספציפיים כדי לייעל את השליפה.

קלט לדוגמה (אין צורך ב-body):

```
{
```

get_user_friends

הסבר קצר:

פונקציה זו מחזירה את רשימת החברים של משתמש מסוים. היא קוראת את רשימת מזהי החברים מהפרופיל של המשתמש, ואז עבור כל מזהה, היא פונה שוב למסד הנתונים כדי לשלוף את הפרטים הבסיסיים של אותו חבר (שם, תמונה, ואימייל).

קלט לדוגמה:

```
{  
  "body": "{\"UserId\": \"main-user-123\"}"  
}
```

respond_to_friend_request

הסבר קצר:

פונקציה זו מנהלת את המענה לבקשת חברות. היא מקבלת את מזהה ההתראה של הבקשה (`NotifId`) ואת החלטת המשתמש (אישור/דחייה). היא מעדכנת את סטטוס הבקשה המקורית, שולחת התראה חזרה לשולח המקורי עם התשובה, ואם הבקשה אושרה, היא מעדכנת את רשימות החברים (`Friends`) של שני המשתמשים.

קלט לדוגמה:

```
{  
  "body": "{\"NotifId\": \"notif-abc-123\", \"accept\": true}"  
}
```

restore_link

הסבר קצר:

פונקציה זו משחזרת קישור שנמחק (מחיקה "רכה") על ידי הגדרת הדגל `IsActive` שלו בחזרה ל-`true`. היא מקבלת מזהה קישור, ומשתמשת בפעולת עדכון עם תנאי (`ConditionExpression`) כדי לוודא שהקישור קיים לפני ביצוע השחזור.

קלט לדוגמה:

```
{
  "body": "{\"LinkId\": \"link-to-restore-123\"}"
}
```

set_link_password

הסבר קצר:

פונקציה זו מגדירה סיסמה לקישור. היא דורשת מזהה קישור, מזהה משתמש וסיסמה חדשה. הפעולה תצליח רק אם המשתמש הוא הבעלים של הקישור והקישור אינו מוגן בסיסמה קיימת (`attribute_not_exists(IsPasswordProtected) OR IsPasswordProtected = :p_false`). זה מונע דריסה של סיסמאות קיימות ומבטיח שרק הבעלים יכול להגדיר סיסמה.

קלט לדוגמה:

```
{
  "body": "{\"LinkId\": \"link-unsecure-789\", \"userId\": \"owner-user-123\",
    \"newPassword\": \"MySecret123\"}"
}
```


post_create_user

הסבר קצר:

פונקציה זו מופעלת באופן אוטומטי על ידי שירות **AWS Cognito** מיד לאחר שמשתמש חדש מסיים הרשמה. היא לוקחת את פרטי המשתמש מהאירוע של **Cognito**, מכינה רשומת פרופיל חדשה עם ערכי ברירת מחדל (כמו תאריך הצטרפות, סטטוס פעיל וכו'), ושומרת אותה בטבלת המשתמשים (**Users**) ב-**DynamoDB**. זו הדרך לסנכרן משתמשים חדשים ממערכת האימות למסד הנתונים של האפליקציה.

קלט לדוגמה (אירוע מ-Cognito):

```
{
  "request": {
    "userAttributes": {
      "sub": "cognito-unique-id-123",
      "email": "test@example.com",
      "name": "Test User",
      "nickname": "tester"
    }
  }
}
```

new_short_url

הסבר קצר:

פונקציה זו יוצרת קישור מקוצר חדש. היא מקבלת כתובת **URL** ארוכה ומזהה משתמש, מייצרת קוד קצר וייחודי, ובודקת שאינו קיים כבר בטבלה. היא שומרת את הקישור החדש בטבלת **Links**, ובנוסף, מעדכנת את רשומת המשתמש בטבלת **Users** על ידי הוספת מזהה הקישור החדש לרשימת הקישורים המשויכת אליו.

קלט לדוגמה:

```
{
  "body": "{\"url\": \"https://www.wikipedia.org/wiki/Very_Long_Article_Name\",
    \"userId\": \"user-abc\"}"
}
```

remove_link_password

הסבר קצר:

פונקציה זו מסירה הגנת סיסמה מקישור. היא מקבלת מזהה קישור ומזהה משתמש. הפונקציה מבצעת עדכון בטבלת הקישורים, אך ורק בתנאי (ConditionExpression) שהמשתמש הוא הבעלים של הקישור ושהקישור אכן מוגן כרגע בסיסמה (IsPasswordProtected = :p_true). אם התנאים מתקיימים, ההגנה והסיסמה מוסרות.

קלט לדוגמה:

```
{
  "body": "{\"linkId\": \"link-secure-456\", \"userId\": \"owner-user-789\"}"
}
```

send_friend_request

הסבר קצר:

פונקציה זו שולחת בקשת חברות ממשתמש אחד לאחר. היא מבצעת מספר בדיקות חיוניות: שהמשתמשים קיימים, שהם לא אותו אדם, ושאינן ביניהם כבר בקשת חברות קיימת (מכל סוג). אם כל הבדיקות עוברות, היא יוצרת התראה חדשה מסוג "pending" המייצגת את בקשת החברות.

קלט לדוגמה:

```
{
  "body": "{\"FromUserId\": \"user123\", \"ToUserId\": \"user456\"}"
}
```

request_image_upload_url

הסבר קצר:

פונקציה זו מייצרת כתובת URL מאובטחת וזמנית (Pre-signed URL) המאפשרת לאפליקציית צד-לקוח להעלות קובץ ישירות לאחסון S3. היא מקבלת את סוג התוכן של הקובץ (contentType), יוצרת שם קובץ ייחודי, ומחזירה את כתובת ה-URL להעלאה (uploadUrl) ואת הכתובת הציבורית הסופית של הקובץ (finalUrl) לאחר שההעלאה תושלם.

קלט לדוגמה:

```
{
  "body": "{\"contentType\": \"image/png\"}"
}
```

send_mail

הסבר קצר:

פונקציה זו שולחת אימייל שיתוף עבור קישור לרשימת נמענים. היא אוספת כתובות אימייל ממספר מקורות (רשימות תפוצה, חברים, וכתובות שהוזנו ידנית), יוצרת רשימה ייחודית, ומכינה אימייל מותאם אישית. לבסוף, היא מתחברת לשרת דואר (SMTP) באמצעות פרטים השמורים במשתני סביבה ושולחת את האימייל לכל נמען.

קלט לדוגמה:

```
{
  "body": "{ \"senderId\": \"user-A\", \"linkId\": \"link-123\", \"groupId\": \"group-alpha\", \"friendIds\": [\"user-B\"], \"recipientsEmails\": [\"new@example.com\"], \"site\": \"https://shortly.com/link-123\"}"
}
```

get_all_users_with_stats

הסבר קצר:

פונקציה זו מחזירה את כל המשתמשים ממסד הנתונים ומספקת מידע סטטיסטי עליהם. אם מסופק `userId` בקלט, היא גם מחזירה את הלינקים המשייכים לאותו משתמש.

קלט לדוגמה:

```
{
  "body": "{ \"userId\": \"894980c8-e8a6-4921-9bb0-f917671caa65\"}"
}
```

toggle_link_privacy

הסבר קצר:

פונקציה זו משנה את סטטוס הפרטיות (`IsPrivate`) של לינק נתון. היא מקבלת `linkId` וקובעת את מצבו הנוכחי של הלינק. לאחר מכן, היא הופכת את הסטטוס (מפרטי לציבורי או מציבורי לפרטי) ומעדכנת את הלינק במסד הנתונים.

קלט לדוגמה:

```
}
{"body": "{ \"linkId\": \"someLinkId123\"
{
```

verify_link_password

הסבר קצר:

פונקציה זו בודקת סיסמה עבור לינק מוגן. היא מקבלת `linkId` וסיסמה. אם הלינק קיים והוא מוגן בסיסמה, הפונקציה משווה את הסיסמה שהוזנה עם הסיסמה המאוחסנת. אם הן תואמות או אם הלינק אינו מוגן בסיסמה כלל, היא מעניקה גישה.

קלט לדוגמה:

```
}
{"body": {"linkId": "someLinkId123", "password": "mysecretpassword"}
{
```

verify_admin_status

הסבר קצר:

פונקציה זו בודקת אם משתמש הוא חלק מקבוצת `Admins` במאגר משתמשים של `Cognito`. היא קובעת באופן דינמי את מזהה מאגר המשתמשים מתוך שם ערימת ה-`CloudFormation` ואת שם הקבוצה מסביבת הריצה.

קלט לדוגמה:

```
}
{"body": {"UserId": "username@example.com"}
{
```

update_user_profile

הסבר קצר:

פונקציה זו מעדכנת את פרטי הפרופיל של משתמש במסד הנתונים של `DynamoDB`. היא מקבלת `userId` ושדות נוספים כמו `FullName`, `Country`, `Picture`. היא בונה ביטוי עדכון דינמי ומעדכנת רק את השדות שסופקו בקלט.

קלט לדוגמה:

```
}
body": {"userId": "user123", "FullName": "Jane Doe", "Country": "Canada"}
{
```

track_click

הסבר קצר:

פונקציה זו מטפלת בקליקים על לינקים. היא מחזירה את פרטי הלינק, כולל אם הוא פעיל. אם הקליק אינו של בעל הלינק עצמו, היא מעלה את מונה הקליקים של הלינק. היא גם בודקת אם הקליק מגיע לאבני דרך המזכות בהישגים, ובהתאם לכך מעניקה הישגים, שולחת התראות ומעדכנת את רשומת המשתמש. בנוסף, הפונקציה בודקת אם הלינק מוגן בסיסמה; אם כן, היא מחזירה דגל המציין זאת במקום את כתובת ה-URL המקורית.

קלט לדוגמה:

```
}
{"body": {"code": "test-link-123", "userId": "test-user-456"}
{
```

get_all_active_users

הסבר קצר:

פונקציה זו מחזירה את כל המשתמשים הפעילים (שבהם **IsActive** מוגדר ל-true) מטבלת ה-**Users** ב-**DynamoDB**. היא מבצעת סריקה של הטבלה ומסננת את התוצאות.

קלט לדוגמה (אין צורך ב-body):

```
{}
```