

מדריך: סגירת הזמנה ב-Base44 באמצעות דף Notify

האינטגרציה ל-Base44 מחברת את האפליקציה או האתר של בית העסק לדף התשלום (iFrame) של המסוף שלו, ומאפשרת קבלת תשלום או תשלומים באמצעות כרטיסי אשראי וארנקים דיגיטליים.

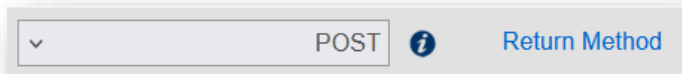
ה-Prompt שלהלן יוצר דף Notify עם Webhook (כתובת URL) עבור האתר או האפליקציה של בית העסק, ומאפשר לו לקבל תשובה חזרה עם כל פרטי העסקה כפי שבוצעו, בנוסף לפרמטרים נוספים, ותשובה האם העסקה עברה או לא.

דרישות לעבודה עם ה-Prompt

- אתר או אפליקציית בית העסק צריכים להיות מחוברים לדף התשלום iFrame של טרנזילה, באמצעות האינטגרציה שלנו.
במידה ובית העסק עדיין לא ביצע את החיבור, נדרש קודם להריץ את ה-Prompt של האינטגרציה, ע"פ המדריך "עבודה עם האינטגרציה של טרנזילה ל- Base44 עבור ביצוע חיובים רגילים או תשלומים" או ע"פ המדריך "עבודה עם האינטגרציה של טרנזילה ל- Base44 עבור ביצוע חיובים חוזרים".
- בית העסק צריכה להיות סביבת Back Order שנבנתה עבור האתר או האפליקציה שלו ב-Base44, לדוגמה – מערכת לניהול הזמנות, שם ניתן לראות את סטטוס הרכישות שבוצעו ("ממתין לתשלום" או "שולם").

יצירת דף ה-Notify

לפני שמריצים את ה-Prompt, יש להיכנס להגדרות דף הסליקה של המסוף, לבחור בהגדרות מתקדמות ולוודא ש-Return Method מוגדר על POST:



כעת נעתיק את ה-Prompt הארוך הבא במלואו ונדביק אותו בחלון השאילתה של האפליקציה:

Create a Notify Page – Back Order Closer

Instructions for Creating a Tranzila Notify URL on Base44

The Tranzila notify URL is a webhook endpoint that Tranzila calls after processing a payment to inform your system of the transaction result.

This allows you to automatically update order status based on actual payment outcomes.

1. Create the Function

Create a new function file in your Base44 project:

functions/tranzila-notify.js

Key Components:

- Form data parser – Tranzila sends data as application/x-www-form-urlencoded.
- Order matching logic – Match orders by amount + recent timestamp.
- Status update logic – Update order status to paid or failed.
- Response – Always return "OK" with status 200 to acknowledge receipt.

Example Implementation:

```
import { createClient } from "npm:@base44/sdk@0.1.0";
```

```
const base44 = createClient({  
  appld: Deno.env.get("BASE44_APP_ID"),  
});
```

```
Deno.serve(async (req) => {
```

```
if (req.method !== "POST") {
  return new Response("Method Not Allowed", { status: 405 });
}

try {
  // Parse Tranzila form-data
  const formData = await req.formData();
  const notify = {};
  for (const [key, value] of formData.entries()) {
    notify[key] = value;
  }

  // Extract fields from notification
  const responseCode = notify["Response"]; // "000" or "0" = success
  const amount = parseFloat(notify["sum"] || 0);

  let order = null;

  // Match by amount + pending status + recent timestamp
  order = await base44.db
    .from("orders")
    .select("*")
    .eq("status", "pending")
    .eq("total_amount_nis", Math.round(amount))
    .gte("created_at", new Date(Date.now() - 30 * 60 * 1000).toISOString())
    .order("created_at", { ascending: false })
    .limit(1)
    .single();

  // If found, update order status
  if (order) {
    await base44.db
      .from("orders")
      .update({
        status: responseCode === "000" || responseCode === "0" ? "paid" : "failed",
        tranzila_reference: notify["TranzilaTK"] || null,
        confirmation_code: notify["ConfirmationCode"] || null,
        raw_data: JSON.stringify(notify),
      })
      .eq("id", order.id);
  }

  return new Response("OK", { status: 200 });
} catch (error) {
  console.error("Tranzila Notify Error:", error);
  return new Response("Error", { status: 500 });
}
});
```

2. Public Access

Unlike Supabase, Base44 functions are public by default.
There is no JWT requirement — as long as the function is deployed, Tranzila can call it.

3. Order Creation Strategy

Before starting the payment:

- Create an order with status = pending.
- Do not put the order ID into myid.
- Store amount and timestamp in your DB.

Matching Logic:

1. Primary → by amount and pending status.
2. Secondary → by recent timestamp (e.g. within the last 30 minutes).

3. Response Codes

Tranzila will send:

- "000" or "0" → Successful payment.
- Any other value → Failed payment.

Your function must:

- Return "OK" with status 200 (otherwise Tranzila retries).
- Return errors (404, 500) only for debugging.

6. Security Considerations

Safe:

- Function only accepts POST requests.
- Data includes Tranzila transaction tokens.
- Matching strategy prevents random orders from being updated.

Caution:

- Don't expose sensitive data in responses.
- Log notifications for debugging.
- Validate amounts match expected order totals.
- Use a time window to avoid stale matches.

7. Testing & Debugging

- Check function logs in Base44 dashboard.
- Verify successful notifications update orders.
- Watch for errors:
 - o No orders found → Order not in pending or amount mismatch.
 - o CORS → Only relevant if accessed from browser (Tranzila itself won't need CORS).

8. Complete Flow

9. User initiates payment → order created as pending.
10. Payment form is loaded in Tranzila for the user to complete the payment.
11. User completes payment in Tranzila.
12. Tranzila calls your notify endpoint.
13. Your function receives notification → finds order by amount + timestamp.
14. Order updated to paid or failed.

נלחץ על **החץ השחור** להעלאת השאילתה לביצוע:

Create a Notify Page – Back Order Closer

Instructions for Creating a Tranzila Notify URL on Base44

The Tranzila notify URL is a webhook endpoint that Tranzila calls after processing a payment to inform your system of the transaction result. This allows you to automatically update order status based on actual payment outcomes.

1. Create the Function

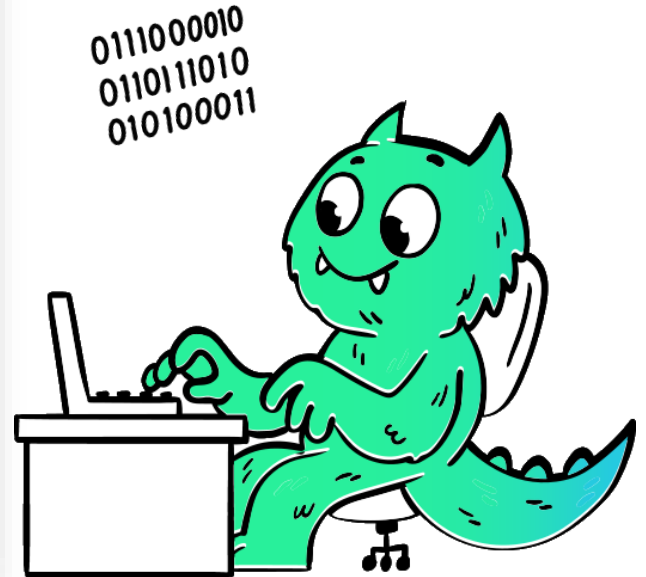
Create a new function file in your Base44 project:

```
functions/tranzila-notify.js
```

Key Components:

- Form data parser – Tranzila sends data as application/x-www-form-urlencoded.

⚙️ + Discuss

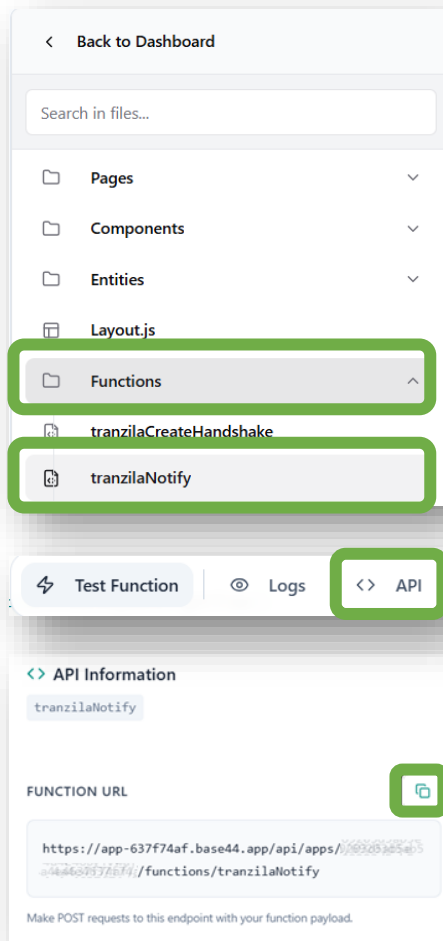


כעת ניגש ל-**Dashboard** של האתר/האפליקציה של בית העסק ונבחר ב-**Code**:

Dashboard Preview

- Overview
- Users
- Data
- Analytics
- Domains
- Integrations Beta
- Security
- Code**
- Agents Beta
- Logs
- API

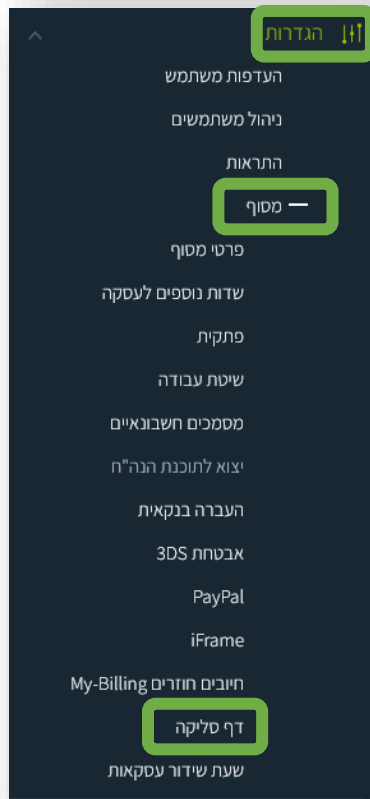
נבחר ב- **Functions** ואז ב- **tranzilaNotify**:



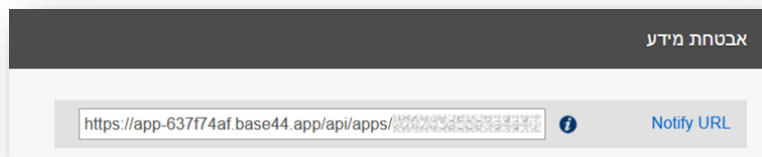
בצד ימין של המסך למטה נבחר ב- **API**:

נעתיק את כתובת ה- **URL (Webhook)** ש- **Base44** יצר:

ניכנס למערכת **my.tranzila** ונבחר בהגדרות < מסוף < דף סליקה:



בטאב הנוסף שנפתח נבחר באבטחת מידע, נדביק את כתובת ה-URL שהתקנו בשדה **Notify URL** ונשנה את השדה **פורמט Notify** ל-QSTR. לסיים, נלחץ על הכפתור **שמירה** המופיע בראש העמוד:

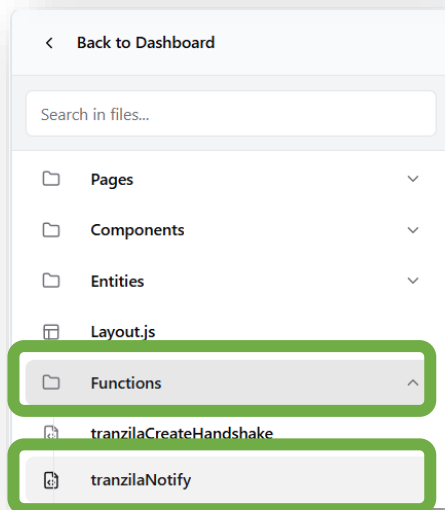
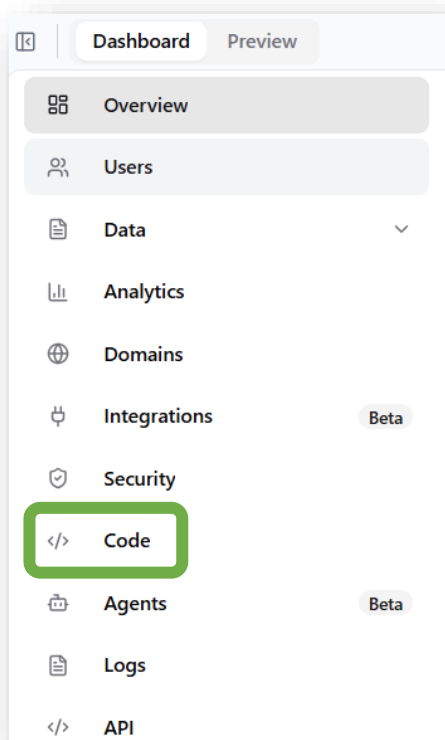


כעת נבצע עסקת בדיקה באתר או באפליקציה של בית העסק, ונוודא את שני התנאים הבאים:

1. שהתקבלה תשובה לדף **Notify**.

לצורך הבדיקה, נפעל על פי השלבים הבאים:

ניגש שוב ל-**Dashboard** של האתר/האפליקציה של בית העסק ונבחר שוב ב-**Code**:

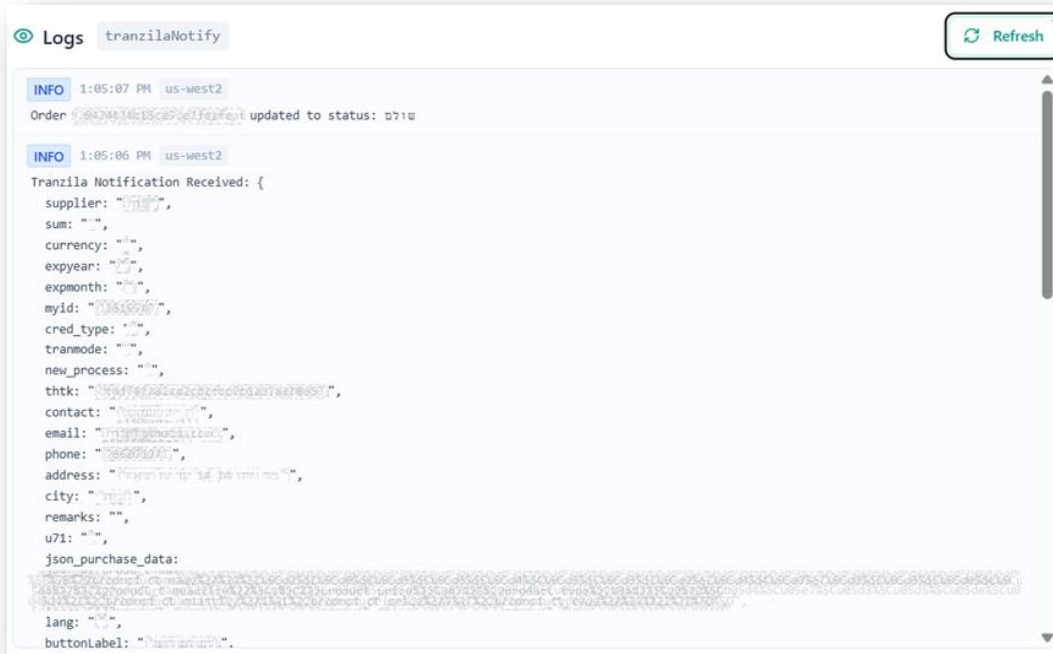


נבחר ב-**Functions** ואז ב-**tranzilaNotify**:

בצד ימין של המסך למטה נבחר ב-Logs:



כעת נוכל לראות אם התקבלה תשובה או לא.
דוגמה לתשובה שחזרה:



2. **שעסקת הבדיקה שביצענו מופיעה ב-Back Order (מערכת ניהול הזמנות) של בית העסק בסטטוס "שולם".**
דוגמה להזמנה שנסגרה:

