AI Automation / Zapier Workflows

Map data fields between apps with format transformations — prevents “field not found” errors and data loss.
Difficulty: Intermediate
Model: GPT-4 / Claude / Gemini
Use Case: Zapier Setup, Data Integration, App-to-App Mapping
Updated: May 2026
Why This Prompt Exists
The most common Zapier failure is “field not found” — because field names differ between apps, and data needs transformation.

You get:

- Zaps that fail because a field name changed or doesn’t exist
- data that lands in the wrong format (dates, phone numbers, currencies)
- missing required fields because no default was provided
- manual data cleanup after every automation run
- hours wasted testing field mappings

But field mapping can be systematic:

- source field: where data comes from (trigger or previous action)
- target field: where data goes (action field)
- transformation: format changes (date format, string concatenation, math)
- default value: what to use if source field is empty
- required vs. optional: which fields must be mapped

Without mapping guidance, Zaps break silently.

This prompt generates complete field mapping plans between apps.

The Prompt

```
Assume the role of a Zapier integration specialist who maps fields
between apps.

Your task is to create a complete field mapping plan from source to
target.

Generate:

1. SOURCE APP & TRIGGER
   - App: [name]
   - Trigger event: [e.g., "New Form Submission"]
   - Available fields: [list with sample data]

2. TARGET APP & ACTION
   - App: [name]
   - Action event: [e.g., "Create Contact"]
   - Required fields: [list]
   - Optional fields: [list]

3. FIELD MAPPING TABLE

| Target Field | Source Field | Transformation | Default Value |
Required? |
|--------------|--------------|----------------|---------------|------
-----|
```

| [field name] | [source field] | [e.g., "MM/DD/YYYY → YYYY-MM-DD"] | [if empty] | Yes/No |

4. TRANSFORMATION DETAILS (for complex mappings)
   - Date formatting: [source format] → [target format]
   - Phone numbers: [source format] → [E.164 format]
   - Name splitting: "Full Name" → "First Name + Last Name"
   - Address parsing: [how to split address fields]
   - Custom formulas: [Zapier formula code if needed]

5. REQUIRED FIELD RISK ASSESSMENT
   - Fields with no source: [list]
   - Fields that need default values: [list with proposed defaults]
   - Fields that may be missing in some records: [list]

6. COMPLETE MAPPING INSTRUCTIONS
   - Step-by-step field mapping for Zapier interface

INPUTS:

Source app and trigger:
[E.G., "Typeform - New Entry"]

Target app and action:
[E.G., "Salesforce - Create Lead"]

Sample source data (optional):
[PASTE JSON OR EXAMPLE FORM SUBMISSION]

```
Known field mismatches (optional):
[E.G., "Typeform has 'Full Name', Salesforce has 'FirstName' and
'LastName'"]

RULES:
- Map all required target fields (unmapped required fields cause
failures)
- Provide default values for source fields that may be empty
- Document transformations explicitly (date format, phone format,
etc.)
- Flag fields that have no source and no default (Zap will fail for
these)
- Test mapping with sample data before deploying
- Use Zapier's built-in formatter when possible (Date/Time, Number,
Text)
```

How To Use It

- Run this before building any Zap that moves data between apps.
- Pay attention to required fields — missing one will break the Zap.
- Set default values for fields that might be empty in the source.
- Use the built-in Zapier Formatter app for complex transformations.
- Test the mapping with real sample data before deploying.

Example Input

Source app and trigger:
“Google Forms – New Form Response”

Target app and action:
“HubSpot – Create Contact”

Sample source data:
“Timestamp: 2025-03-15 14:30:00, Full Name: John Smith, Email: john@example.com, Phone: (555) 123-4567, Company: Acme Inc”

Known field mismatches:
“Google Forms has ‘Full Name’, HubSpot has ‘First Name’ and ‘Last Name'”

Why It Works
Most Zapier users guess at field mappings — then debug failures when data doesn’t appear correctly.

This framework improves outcomes by forcing:

- source and target field inventories (what data is available vs. needed)
- explicit mapping table (clear one-to-one correspondence)
- transformation documentation (how to change formats)
- default value planning (what to use when source is empty)
- required field risk assessment (what will break if missing)

Great field mapping doesn’t guess — it ensures every required field gets data in the right format.

Build Better AI Systems

Subscribe for advanced prompt engineering, AI coding tools, debugging frameworks, and practical strategies for developers and engineers.

Carefully engineered prompts for people doing real work.

Share this:

- Share on Facebook (Opens in new window) Facebook
- Share on X (Opens in new window) X

See also Rate Limit Planner