Iterable Pipelines in Digioh

Last Modified on 06/25/2026 9:36 am EDT

This article covers what is specific to **Iterable** when building or troubleshooting a Digioh pipeline: how Iterable expects fields and data types, the top-level email requirement, and the errors you will see most often in pipeline logs - including how to tell an Iterable outage apart from a configuration problem. For pipeline fundamentals (task types, the item data payload, logs, conditions), see *Understanding Digioh Pipelines*.

Iterable field syntax

Iterable stores custom profile fields under dataFields. In the Map Data step, write the output field as:

dataFields.your_field_name

The top-level email must be mapped

Iterable uses a **top-level email or userId** to identify the contact. It must be explicitly mapped - separate from any email or userId you place inside dataFields. If a campaign uses a custom field for email (e.g. form.custom_1) instead of the native form.email, the top-level email arrives blank and Iterable rejects every record.

A common 100% failure: A pipeline sending email only inside dataFields with a blank top-level email identifier will fail on every single submission. Always confirm the top-level email is mapped.

Validate custom email fields: When using a custom field for email, enable email syntax validation on that field in the form builder. A trailing space like jane@email.com will be rejected, and the log error will not make the cause obvious.

Match the data type Iterable expects

Iterable is strict about field types - stricter than Klaviyo. If a field expects a date and receives a plain string, or expects a boolean and receives the string “true,” Iterable will reject the submission. Use the Transform option in Map Data to convert values to the type Iterable expects. Watch especially for:

- Date of birth and timestamps - must be a date/timestamp format, not a plain string.
- Opt-in / boolean fields - note that form.opt_in comes through as the string "true", not a real boolean, so it needs a transform to a true/false value.

Empty mappings: If no fields are mapped, the pipeline still calls Iterable but sends an empty payload - nothing is written. If submissions arrive but no data lands in Iterable, check that the Map Data step actually has the fields mapped and that Remove Unmapped Fields has not been unchecked.

Common Iterable errors

Iterable errors surface in the pipeline log's Response field. The most common ones and what to do:

Error in log	What it means and what to do
"successCount": 0, "invalidEmails": [""]	The top-level email is blank or invalid. For email-based Iterable projects every submission must include an email. Map the correct top-level email field and enable email syntax validation if it is a custom field.
"Peer from the ring-balancer" / pool exhausted	Iterable's load balancer could not reach a healthy backend - effectively a 503. Check statusgator.com/services/iterable and re-run the log once Iterable recovers.
'T' is an invalid start of a value	A JSON parse error: Iterable timed out and returned plain text instead of JSON. Iterable-side - same status check as above.
event disallowed from tracking	The event name being sent is not enabled in the Iterable account settings (Iterable blocks unconfigured events to prevent spam calls). Ask the client to enable the event in their Iterable project.

When to use Render Template for Iterable

If Iterable expects a payload shape that Map Data cannot produce - nested objects, conditional fields, or arrays of subscriptions - use a Render Template task to build the JSON exactly. See the Render Template section of Understanding Digioh Pipelines for syntax and the fragility warning.