✅What is Filter by Zapier?

Filter by Zapier is a step you add to your Zap that acts like a **checkpoint**.
It decides whether your Zap should **continue** or **stop** based on conditions you set.

Think of it as:

> “Only run the next steps if these rules are true.”

✅Why use Filters?

- Prevent unnecessary actions (and save tasks).
- Make sure your Zap runs **only when data meets your criteria**.
- Add **control** and **accuracy** to your automation.

✅How does it work?

1. **Trigger fires** (e.g., new order, new email).
2. Add **Filter by Zapier** as the next step.
3. Set **conditions** using fields from the trigger:
    - Example:
        - Amount > 500
        - Status = Paid
4. Zapier checks the conditions:
    - If **true**, Zap continues to the next steps.
    - If **false**, Zap stops (no further actions).

✅Example Use Case

You have a Zap that sends a Slack alert for new orders.
But you only want alerts for **high-value orders** (amount > $500).

Steps:

- Trigger: New order.
- Filter: Amount > 500.
- Action: Send Slack message.

Result: Only big orders trigger Slack alerts.

✓Visual Analogy

Imagine a **security gate**:

- Cars (your data) arrive.
- Gate checks: “Is this car allowed?”
- If yes → Pass through.
- If no → Stop.

✓Best Practices

- Use Filters **early** in the Zap to avoid wasting tasks.
- Combine multiple conditions with **AND/OR** logic.
- Normalize data (e.g., convert text to number) before comparing.
- Add clear descriptions so you remember why the filter exists.

✓Common Uses

- Only send emails if status = confirmed.
- Only create tasks if priority = high.
- Only post to Slack if amount > 1000.

✓Limitations

- If the filter fails, the Zap **stops completely** (no fallback like Paths).
- Each Zap can have multiple filters, but too many can make it hard to manage.
- Filters don’t run actions—they only allow or block the flow.
