

Approval Request Workflow — HTTP-Triggered Auto-Approval Logic App

Article · Azure Logic Apps · Workflow Definition Language (schema 2016-06-01) · Content version 1.0.0.0

Overview

This Logic App implements an **automated approval-routing workflow** for expense or purchase requests. It exposes an HTTP endpoint that accepts approval requests, evaluates the requested amount against a configurable approval threshold, and returns an immediate synchronous decision:

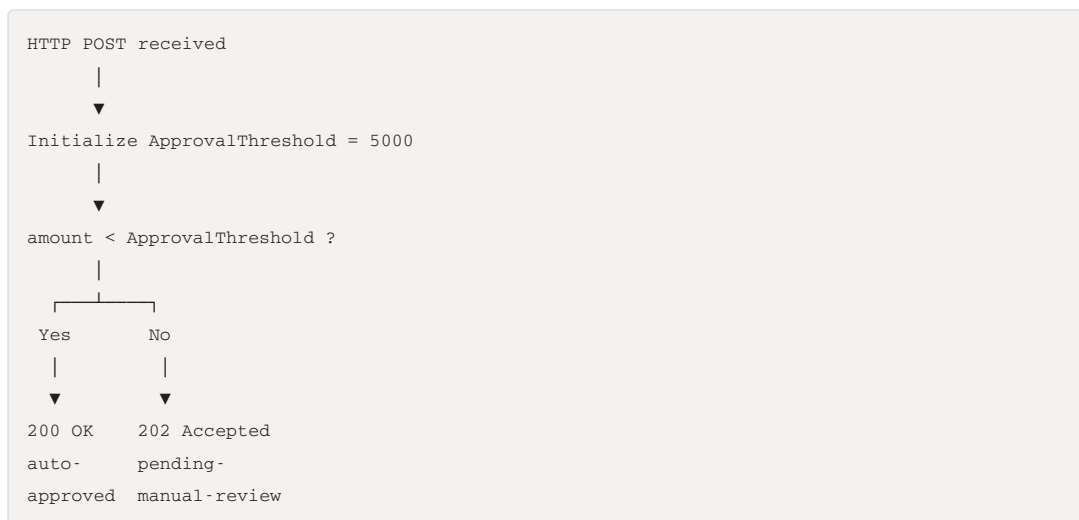
- Requests **below the threshold** are auto-approved and acknowledged with an HTTP `200 OK` response.
- Requests **at or above the threshold** are flagged for manual review and acknowledged with an HTTP `202 Accepted` response.

Because both outcomes respond synchronously to the caller, this workflow functions as a lightweight **approval decision API** that upstream systems (forms, ticketing tools, procurement apps) can call directly.

Workflow architecture

Component	Name	Type
Trigger	When_an_approval_request_is_received	Request (HTTP)
Action 1	Initialize_approval_threshold	InitializeVariable
Action 2	Check_if_auto_approval_applies	If (Condition)
Action 2a (true branch)	Respond_approved	Response (HTTP)
Action 2b (false branch)	Respond_pending_review	Response (HTTP)

Execution flow:



Trigger: When an approval request is received

The workflow starts with a **Request trigger** (`type: Request` , `kind: Http`), which provisions a callable HTTPS endpoint when the Logic App is saved. The trigger activates whenever an external system sends an HTTP `POST` to the generated endpoint URL.

The trigger enforces a **JSON request schema** to validate and type the incoming payload:

```
{
  "type": "object",
  "properties": {
    "requestId": { "type": "string" },
    "requesterEmail": { "type": "string" },
    "amount": { "type": "number" },
    "costCenter": { "type": "string" }
  },
  "required": ["requestId", "requesterEmail", "amount"]
}
```

Field	Type	Required	Description
<code>requestId</code>	string	Yes	Caller-supplied identifier used to correlate the request and response.
<code>requesterEmail</code>	string	Yes	Email address of the person submitting the request.
<code>amount</code>	number	Yes	Monetary amount being requested; drives the approval decision.
<code>costCenter</code>	string	No	Optional cost center for attribution or reporting.

Defining the schema on the trigger makes the payload properties available as strongly-typed dynamic content (for example, `triggerBody()?['amount']`) in downstream actions.

Example request:

```
POST https://<logic-app-endpoint> HTTP/1.1
Content-Type: application/json

{
  "requestId": "REQ-2026-0042",
  "requesterEmail": "jdoe@contoso.com",
  "amount": 3200,
  "costCenter": "CC-1180"
}
```

For more information, see [Receive and respond to inbound HTTPS calls in Azure Logic Apps](#).

Actions

1. Initialize_approval_threshold

Type: `InitializeVariable` · **Runs after:** trigger

This action declares an integer variable that defines the auto-approval cutoff:

```

{
  "variables": [
    {
      "name": "ApprovalThreshold",
      "type": "integer",
      "value": 5000
    }
  ]
}

```

Variable	Type	Value	Purpose
ApprovalThreshold	integer	5000	Maximum amount eligible for automatic approval.

Centralizing the threshold in a variable, rather than hard-coding it inside the condition expression, keeps the business rule in a single, easily maintained location. See [Store and manage values by using variables in Azure Logic Apps](#).

2. Check_if_auto_approval_applies

Type: If (Condition) · **Runs after:** Initialize_approval_threshold (Succeeded)

This control action evaluates whether the requested amount qualifies for automatic approval:

```

{
  "and": [
    {
      "less": [
        "@triggerBody()?['amount']",
        "@variables('ApprovalThreshold')"
      ]
    }
  ]
}

```

The expression compares the `amount` property from the trigger payload against the `ApprovalThreshold` variable using the `less` operator — the condition is **true** when `amount < 5000`. The `?` operator (`triggerBody()?['amount']`) performs a null-safe property access, preventing a runtime failure if the property is absent.

For expression reference, see [Reference guide to workflow expression functions in Azure Logic Apps](#) and [Add conditions to control workflow actions](#).

True branch: Respond_approved

When the amount is under the threshold, the workflow returns an immediate approval:

```

{
  "statusCode": 200,
  "headers": { "Content-Type": "application/json" },
  "body": {
    "requestId": "@triggerBody()?['requestId']",
    "status": "auto-approved",
    "reviewedBy": "system"
  }
}

```

```
}
```

The response echoes the caller's `requestId` for correlation, sets `status` to `auto-approved`, and attributes the decision to `system` — providing a clear audit signal that no human reviewer was involved.

False (else) branch: `Respond_pending_review`

When the amount meets or exceeds the threshold, the workflow defers the decision:

```
{
  "statusCode": 202,
  "headers": { "Content-Type": "application/json" },
  "body": {
    "requestId": "@triggerBody()?['requestId']",
    "status": "pending-manual-review",
    "reason": "Amount exceeds auto-approval threshold"
  }
}
```

The `202 Accepted` status code is semantically appropriate here: the request has been received and accepted for processing, but the final outcome (manual review) is not yet complete. The `reason` field tells the caller *why* automatic approval did not apply.

Both branches use the native **Response** action; see [Receive and respond to inbound HTTPS calls in Azure Logic Apps](#).

Parameters and connections

The definition declares a single standard parameter:

Parameter	Type	Default	Notes
<code>\$connections</code>	Object	<code>{}</code>	Placeholder for managed API connections. Empty in this workflow because only built-in native operations (Request, Variables, Condition, Response) are used — no external connectors are required.

The absence of managed connections means this Logic App has **no external service dependencies**, no connection authentication to maintain, and minimal failure surface.

Response summary

Scenario	Condition	HTTP status	status value	Reviewed by
Auto-approval	<code>amount < 5000</code>	200 OK	<code>auto-approved</code>	<code>system</code>
Manual review required	<code>amount >= 5000</code>	202 Accepted	<code>pending-manual-review</code>	Deferred to human reviewer

Purpose and functionality

This workflow encapsulates a common enterprise pattern: **threshold-based approval triage**. Its value lies in removing low-risk requests from human review queues while guaranteeing that high-value requests receive scrutiny. Typical integration scenarios include:

- A procurement or expense form that calls this endpoint on submission and displays the decision to the requester in real time.
- A ticketing system that routes tickets to an approvals queue only when the workflow returns `202`.
- A parent Logic App or Power Automate flow that uses this workflow as a reusable decision component.

Extension considerations: the boundary condition is worth noting — an amount of exactly `5000` routes to manual review because the comparison uses `less` rather than `lessOrEquals`. Natural next steps for a production implementation include sourcing `ApprovalThreshold` from Azure App Configuration or a workflow parameter instead of a hard-coded value, adding a notification action (for example, Office 365 Outlook or Microsoft Teams) in the else branch to alert reviewers, and persisting decisions to a data store for audit history.

Additional resources

- [What is Azure Logic Apps?](#)
- [Schema reference for Workflow Definition Language](#)
- [Receive and respond to inbound HTTPS calls](#)
- [Add conditions to control workflow actions](#)
- [Store and manage values by using variables](#)
- [Workflow expression functions reference](#)