ChatGPT

Google Workspace Flows: API Access and Integration Overview

Introduction to Google Workspace Flows

Google Workspace Flows is a new **no-code workflow automation tool** introduced by Google in 2025. It allows users to **orchestrate multi-step processes across Google Workspace apps** (Gmail, Drive, Calendar, Chat, Forms, etc.) with the help of Google's Gemini AI model [1] [2]. Flows is designed to handle context-rich tasks (e.g. analyzing content, drafting responses) that go beyond simple triggers, effectively **integrating AI ("Gems") into automation** [3]. For example, a flow might automatically summarize an incoming email with Gemini and route it to the right place, or update a spreadsheet and notify someone based on a form submission. Currently, Google Workspace Flows is in an **early preview (alpha)** phase – available in English only to customers enrolled in the Gemini for Workspace Alpha program [4] [5]. This means access is limited (generally to certain Workspace editions or beta participants) and features are still evolving.

Availability and Current Access Constraints

Workspace Flows is in private preview (alpha). It was unveiled around Google Cloud Next 2025 and is not yet generally available to all Workspace users [6]. Only domains **accepted into the Gemini (AI) trusted tester program** have access, and even then only in English for now [4]. Admins of eligible organizations may need to enable the Flows alpha or Gemini features for their users. Notably, Google Workspace Flows (and its third-party integrations) appear to be **limited to higher-tier Workspace plans** (e.g. Business Standard/Plus, Enterprise) that have the Gemini AI add-on entitlements [7]. Regular Workspace users outside the alpha cannot use Flows yet. Because it's pre-GA, the **feature set and APIs are subject to change**, and Google advises against publicly publishing any solutions built on Flows until it's officially launched [8]. In summary, as of late 2025 Workspace Flows is in a closed beta – primarily for testing within enterprises – and not broadly open for production use.

Is There an API for Google Workspace Flows?

There is *no public REST or RPC API for managing Workspace Flows (e.g. no documented endpoints to programmatically create, list, or execute flows). Google has not released any RESTful service like `flows.googleapis.com`, nor published OpenAPI/Discovery documents or protobuff definitions for a Flows API as of this writing. Instead, **Workspace Flows is built on the Google Workspace Add-ons platform** [9]. In other words, Flows doesn't expose a standalone external API – rather, developers can **extend Flows by building custom "steps" as Workspace add-ons** [10]. These custom steps allow integration of third-party or custom logic into the no-code Flows environment. Under the hood, flows and steps are defined through the add-on's manifest (appsscript.json) and executed via Apps Script or alternate runtimes.

Custom Steps via Add-ons (Extension Mechanism)

To integrate with Flows, a developer creates a **Google Workspace Add-on** and adds a Flows section to its manifest. For example, you would define one or more `workflowElements` in the manifest under `"addOns": { "flows": { ... } }` with each element representing a custom step (or action) that users can add to their flows [11]. Each step definition includes: an `id`, name/description, input and output parameters, and references to handler functions (for configuration and execution) [12]. Specifically:

- `onConfigFunction` – an Apps Script function that builds the UI for configuring the step (e.g. a card interface to collect any inputs from the user, such as an email address or parameters). This runs when a user adds/configures the step in their flow.
- `onExecuteFunction` – the function that executes the step's logic when the flow runs. This is where you put the code to perform the action (send an email, create a CRM record, call an API, etc.). It receives any inputs the user provided and must return any defined outputs [12]. Execution is synchronous: the flow waits for this function to finish before moving to the next step.

Google provides a **Flow Event object** to these functions, consistent with the standard Workspace Add-on event payload structure [13] [14]. For instance, when a flow runs your custom step, it triggers an `actionInvocation` event with any input values, and your `onExecuteFunction` can access those via `event.workflow.actionInvocation.inputs` [15]. You can also log activity or errors for the flow's audit log [16]. (Currently, no separate webhook/callback mechanism exists – the logic runs within the add-on execution context.)

Because this is essentially an extension of Google's add-ons framework, **there is no "direct" OAuth endpoint or GraphQL for Flows** itself – instead, developers interact by writing code in Apps Script (or an HTTPS endpoint for alternate runtimes) and deploying it as an add-on. Google's documentation explicitly frames this as *"extending Flows by updating your add-on's manifest to contain a flow-specific section."* [9] There is *no officially documented experimental API* either; all developer guidance revolves around the add-on model. In short, **Workspace Flows does not yet offer a public API** in the traditional sense – the only way to programmatically interface with it is to build a **custom step add-on** or use the provided UI.

Manifest and Protobuf Notes

Since no external API is published, there aren't public protobuff definitions for Flows accessible to developers at this time. The flow step definitions live in the add-on's JSON manifest rather than a standalone API schema. For reference, a manifest snippet for a simple custom step (e.g. a "Calculator" step) looks like this:

```
"addOns": {
  "common": { ... metadata ... },
  "flows": {
    "workflowElements": [
      {
        "id": "calculateStep",
        "state": "ACTIVE",
        "name": "Calculate",
        "description": "... performs a calculation ...",
        "workflowAction": {
          "inputs": [ { "id": "value1", "dataType": {"basicType":
"INTEGER"} }, ... ],
          "outputs": [ { "id": "result", "dataType": {"basicType":
"INTEGER"} } ],
          "onConfigFunction": "onConfigCalculate",
          "onExecuteFunction": "onExecuteCalculate"
        }
      }
    ]
  }
}
```

This manifest structure (particularly the `workflowAction` with inputs/outputs and function hooks) is how Flows are configured – it's part of the **Google Workspace Add-ons manifest specification**. Google has not (yet) open-sourced a proto or Discovery doc for this; however, it aligns conceptually with other Google Workspace add-on components. The lack of a direct API means there is **no official way to programmatically create or modify flows outside of the Flows builder UI**, and no public endpoint to trigger flows from external systems (aside from actually sending the trigger events like an email or form submission). All automation is configured within the Flows UI or via these manifest-defined steps.

Authentication Model and OAuth Scopes

Authentication in Workspace Flows leverages Google Workspace Add-on OAuth scopes. Because Flows runs within a Google Workspace domain and often performs actions on behalf of a signed-in user, it uses that user's credentials (with consented scopes) to execute steps. For Google-provided steps (like Gmail or Drive actions), Google presumably manages the necessary authorizations under the hood when you enable Flows (likely requiring you to grant access to your Gmail, Drive, etc. the first time). For **custom steps**, the add-on must declare whatever OAuth scopes are needed to do its work, and users will be prompted to authorize those when they install/enable the step. This is identical to how normal add-ons work [17] [18].

For example, if your flow's custom step needs to read Gmail messages or send email, you must include the corresponding Gmail API scopes (`https://www.googleapis.com/auth/gmail.readonly`, `.../gmail.send`, etc.) in your manifest. The user will be asked to approve those permissions when adding your integration [19]. Similarly, to call external APIs via UrlFetch or an OAuth2 library in Apps Script, you'd include `https://www.googleapis.com/auth/script.external_request` [18]. The **authorization flow** is the same as for other Workspace add-ons: the user sees a consent screen listing the permissions your step requires (e.g. "Send email as you," "Read your Google Sheets," etc.) and must approve to enable the step [20].

It's worth noting that Google treats **Flows' third-party integrations as Marketplace add-ons**. Administrators can control which integrations are available by using domain allowlists for Google Workspace Marketplace apps [21]. (API-based admin controls cannot block these; they must be managed like Marketplace apps [22].) In some cases, a third-party integration will require its own OAuth connection – for instance, an Asana or Salesforce integration might prompt the user to log in to their Asana/Salesforce account to authorize Google to act on their behalf. Google's admin guide mentions **"helper apps" for third-**

party integrations that may need to be installed from the Marketplace [23], which implies these connectors use the add-on infrastructure and OAuth2 flows for the non-Google service.

In summary, **Flows runs in the context of the user's Google Workspace session**, and it abides by the OAuth scope consent model. There are no unique OAuth scopes just for Flows; instead, it uses the existing scopes for whatever services your flow touches. Developers building custom steps should carefully include all necessary scopes in the manifest (and none more than necessary) just as they would for a regular add-on.

Gmail Integration and Email Automation

One of the most powerful aspects of Workspace Flows is its **deep integration with Gmail for email automation**. Flows can directly interact with Gmail both as a **trigger (starter)** and as an **action**:

- **Gmail-Based Triggers:** You can set a flow to start **"when I receive an email"** under certain conditions. This trigger can be filtered by criteria similar to Gmail search: sender, recipients, subject keywords, specific words in the body, message size, presence of attachments, etc. [24]. For example, a flow could kick off whenever an email from a VIP client arrives, or when an email with subject containing "URGENT" is received. According to documentation, *"When you receive an email with [a specific] sender, subject, or content"* is one of the available starters in the Flows interface [25]. This effectively modernizes the idea of Gmail filters by allowing the incoming email to *trigger a multi-step workflow*. In the context of email response automation, this means as soon as an incoming message meets your criteria, Flows can automatically respond or take other actions (with AI assistance if needed).

- **Gmail Actions (Send/Draft Email):** Within a flow's sequence of steps, you can include actions that interface with Gmail. Google has built-in steps like *"Send an email"* or *"Draft an email reply"* using your Gmail account. For instance, a flow could analyze an inbound email (via a Gemini AI step) and then **automatically draft a reply in Gmail** – ready for you or your team to review and send [26]. Another flow might take parsed data and send out a notification email to someone. The alpha documentation even gives an example of using Flows to draft customer support responses: *"Flows…drafts a helpful reply [to a customer] and flags it for the support team to review and send."* [27]. So, both fully automated sending and assisted drafting are possible. Additionally, flows can perform other Gmail operations like labeling messages or marking them (this isn't explicitly detailed in docs we have, but some user examples mention adding labels or moving emails to tasks).

- **AI-Powered Email Processing:** Because Flows is integrated with Gemini AI, you can incorporate AI steps to handle email content. For example, a flow trigger on new emails could pass the email text to a **"Gemini summarize" step**, then use the summary in a follow-up action. One user-described use case: *summarize each point of an email, prioritize them, create tasks, notify via Chat if high priority, and even monitor if you've responded to all points* [28] [29]. This highlights that beyond just sending emails, Flows can intelligently triage and respond to emails, making it very useful for helpdesk or inbox management automation.

Overall, Gmail integration is first-class in Workspace Flows – **you can trigger flows on incoming mail and automate outgoing mail**. This makes it feasible to set up scenarios like autoresponders with brainpower (e.g. AI-crafted replies), escalation of certain emails to chat or task systems, or daily digests of emails. All such interactions run within your Google account (using Gmail API behind scenes), so they respect Gmail's security and permission model.

Integration with Third-Party Services (Connectors & Webhooks)

Google Workspace Flows isn't limited to Google's own apps; it's designed to connect with external services as well, making it a potential replacement for tools like Zapier or Make.com (with the added AI capabilities). There are two main ways Flows can integrate with third parties:

1. Pre-Built Third-Party Integrations (Connectors): Google has been working with partners to provide out-of-the-box connectors for popular business applications [30]. In the Workspace Flows alpha, **integrations for services like Asana, Jira, Salesforce, QuickBooks, and Mailchimp have been mentioned** [31] [32]. These likely come as **Workspace Integration add-ons** available through the Google Workspace Marketplace [33]. For example, an official Salesforce connector might allow a flow step to create a new lead or update an account in Salesforce. The admin documentation explicitly lists *"connectors to third-party services, such as Asana, Mailchimp, and Salesforce, that let users take action on third-party content from … Workspace Flows."* [33]. Once the admin enables a given integration (and the user has a subscription to that third-party service if required [34]), it becomes available as a step in the Flow builder. Under the hood, these connectors are Google-vetted add-ons that handle the API calls to the third party. Users might need to sign into the third-party account to authorize the connection the first time (for example, authorize the Asana add-on to access their Asana data). After that, the flow can perform operations in the external app. Current limitations: the set of available connectors is limited (the ones above and possibly a few others in alpha), and these may be available only to certain Workspace tiers or the alpha program. Additionally, because these are essentially add-ons, they run with whatever **permissions** the third-party API and the user's account allow. Expect some variation in capabilities depending on what each connector supports.

2. Custom Integrations via Webhooks or HTTP: For any service that doesn't have an official connector, Workspace Flows can likely integrate through **webhooks or custom HTTP requests**. The Workspace Learning Center materials suggest you can *"connect [flows] to other services with webhooks"* [35] and *"take actions in third-party services"* [36]. In practice, there are a couple ways to do this:
- **Outgoing Webhook Step:** You could have a step in a flow that uses a generic "Webhook" connector – e.g., make an HTTP POST to a given URL with some payload. Some automation platforms have this built-in, and Flows seems to at least document it conceptually. This would allow sending data to any endpoint (for instance, a custom webhook your application provides, or a service like Slack's incoming webhook). While we don't have the exact UI screenshot, the Learning Center guide's title implies this is a feature. The user would input the URL and maybe map some data to send. This provides a lightweight way to integrate without coding an entire add-on. - **Apps Script HTTP Requests:** If more complex logic or two-way communication is needed, a developer's custom step code can call external REST APIs using `UrlFetch` in Apps Script. This is essentially building your own connector in code. With the `script.external_request` scope granted, your `onExecuteFunction` can perform `UrlFetchApp.fetch()` to any API endpoint (e.g., create a ticket in ServiceNow or send a message to Slack via API). The response can then be parsed and used in subsequent steps (via output variables). This method requires writing code, but it gives full flexibility to integrate with virtually any service's API. The downside is that it's not user-friendly for a non-developer – it would be packaged as an add-on step that someone has to install. But for internal use or for a vendor building a connector before Google offers an official one, this is a viable path.

It's important to highlight that **administrators have control** over these third-party connections. Since they behave like add-ons, an admin can decide which Marketplace integrations are allowed. The admin guide notes you manage access via the Marketplace allowlist and **you cannot simply block them via API permissions toggle** [37] – meaning if an integration is allowed, it can be used by users even if the third-party app isn't marked as "trusted" in the usual OAuth settings (this is a nuance in admin controls due to how Google implemented these connectors). So, companies will likely vet which external systems employees are permitted to tie into Flows.

Current limitations: Because Flows is in alpha, the third-party integration catalog is not fully fleshed out. Many common apps (Slack, Trello, etc.) might not have official connectors yet – though Google says it's *"working with partners"* to add more [30]. It's also possible that during alpha/beta, only Google-internal testers or select partners have access to build custom connectors. As Flows moves toward general availability, expect to see more integrations added and possibly an SDK for third parties to contribute their own flow steps (likely via the add-on mechanism we described). For now, developers can experiment with custom steps or webhooks for any missing integrations.

Key Takeaways for Developers

1. No Official REST API (Yet): Google Workspace Flows does not provide a public API interface in the traditional sense – you can't call a "Flows API" to create or trigger workflows from outside Google. All interactions are either through the **Flows web UI** (`flows.workspace.google.com`) or by using the **Workspace Add-ons framework** to extend functionality. If your goal was to programmatically control flows (e.g., create flows on the fly via code), that's not supported at this time. Developers should not find any mention of a `flows.googleapis.com` in Google's API library – it simply isn't exposed.

2. Extend Flows via Custom Steps: What Google *does* support is a way to **insert custom functionality into Flows**. This is done by creating a Workspace Add-on (Apps Script or alternate runtime) and defining one or more **Flow steps** in its manifest [9] [10]. Each step you build becomes available to users as a building block in their flows. This is how you integrate proprietary systems or specialized logic. For example, if you have an internal database or API that Google doesn't natively support, you could write a custom step that queries that API and returns data into the flow. The developer docs provide guidance (and a quickstart) for building a simple calculator step, which illustrates the pattern of collecting inputs and producing outputs synchronously [12]. Keep in mind that during the alpha, Google has advised *not* to publish these add-ons to the public Marketplace [8] – distribution should be limited (e.g., just within your domain or test users) until Flows is further along. In the future, we might see an official gallery of Flow steps or the ability for third parties to list their connectors for easy user installation.

3. Authentication & Scopes Mirror Add-on Model: When developing integrations, remember that **all actions run under the user's identity**. You must request OAuth scopes for any Google services your code touches (Mail, Drive, etc.) [19]. The user will need to authorize those. If integrating external APIs, you might use service-specific OAuth within your code (for instance, using an OAuth2 library in Apps Script to get a token for an external service – which then also requires the `external_request` scope and possibly `/script.oauth2` scope). The security context is quite granular: e.g., a custom Gmail-related step could use the Gmail API via Apps Script, but it will only be able to see the user's data after that user grants access. Also, if your step uses any sensitive data or APIs, expect the usual Google verification process for OAuth if you intend to deploy it broadly.

4. Gmail and Workflow Automation: If your use-case involves email, Workspace Flows is well-suited out of the box. You can automate email responses, triage, and notifications with minimal coding. The built-in Gmail trigger on new messages (with filters) is a key feature [25]. In the past, achieving "run a script when a Gmail arrives" required workarounds (like Gmail add-on triggers on open, or polling via Apps Script). Flows provides a more direct, event-driven model for email. Combined with Gemini AI, it opens possibilities like automated email summaries, smart routing (e.g., "if urgent and from VIP, forward to Slack and draft immediate reply"). Developers looking to extend this might write custom steps for specialized email processing, but many scenarios can be handled with the native steps (Gemini text analysis, Gmail send, etc.). It's a good idea to consult Google's provided templates and examples; for instance, Google showcased a flow that *"when a form submission arrives, uses a Gem to draft a summary and then drafts an email reply for support to send"* [27] – this demonstrates combining Gmail actions with AI and other triggers.

5. Third-Party Integration Considerations: For non-Google services, first check if Google or a partner has an official connector in the Flows alpha. If one exists (like Asana or Salesforce), using it will save you a lot of effort – those steps are pre-built and likely handle authentication via OAuth2. If an official integration is not available, you have the two options discussed (webhook vs. custom code). When building custom integrations, be mindful of API limits and latency. Flows steps run synchronously; if you call an external API that takes time to respond, it will slow the whole flow. There might be execution time limits similar to other Apps Script-based add-ons (often ~30 seconds per function run, though this isn't explicitly stated for Flows yet). Also consider error handling – if the external call fails, your step should handle it gracefully and perhaps output an error message (which could be logged or even used in a conditional step in the flow if you design for that).

6. Current Limitations and Future Outlook: As of now, **Workspace Flows is an alpha product**, so any integrations you build are for exploratory or internal testing purposes. There may be features not yet available (for example, conditionals/loops beyond basic "Decide/Check" steps, or the ability to trigger flows via an API call or button – currently triggers are predefined events). We also don't have info on whether flows can be exported/imported or managed as code (Infrastructure-as-Code style) – likely not yet. Google is actively working on this product, so keep an eye on updates. It's possible that as it matures, they might introduce a management API (to, say, programmatically deploy flow templates or fetch flow run logs) or expand the triggers (perhaps to Drive events, which are hinted at in some docs). In the developer preview program and release notes [38], Google recently introduced a Workspace **Events API** (for Drive, etc.), which could eventually tie into Flows – for instance, a Drive file change could trigger a flow. This isn't explicitly documented for Flows yet, but it aligns with the direction.

For now, developers aiming to leverage Google Workspace Flows should **focus on building robust custom steps** and **using Flows in tandem with existing Google APIs**. You might use Flows as the user-facing orchestrator and still use Apps Script or REST APIs for heavy lifting in steps. The key benefit of Flows is the easy UI and AI integration, so even without a direct API, it can significantly speed up automation tasks for end-users. Just approach it as an add-on development exercise, and refer to Google's guides on building flow steps (the quickstarts and samples) for best practices. Google's official documentation on extending Workspace Flows and the manifest reference will be your primary resources [10] [12], along with the support content for using Flows.

References and Further Reading

- Google Workspace Flows Developer Guide – *Extend Google Workspace Flows* (Google for Developers) [10] [12]
- Google Workspace Add-ons Manifest – *Flows section and workflowElements* (JSON example) [11]
- Google Workspace Blog – *New AI capabilities in Workspace (Flows announcement)* [1] [30]
- Google Workspace Learning Center – *Workspace Flows training and help* (end-user documentation) [4] [36]
- Google Workspace Admin Help – *Support access to Workspace Integrations* (admin controls for third-party connectors) [33]
- *About Google Workspace Flows* (Unofficial Google Sites wiki with early info) [2] [25]
- Reddit discussion by early users (anecdotal use cases of Flows in action) [28] [29]

---

[1] [3] [27] [30] Announcing the latest AI capabilities in Google Workspace with Gemini | Google Workspace Blog
https://workspace.google.com/blog/product-announcements/new-ai-drives-business-results

[2] [5] [6] [25] [26] Google Workspace Flows - About Google Workspace Flows
https://sites.google.com/view/workspace-flows/about

[4] [35] [36] Google Workspace Flows training and help - Google Workspace Learning Center
https://support.google.com/a/users/answer/16275487?hl=en

[7] [21] [22] [23] [33] [34] [37] Support access to Workspace Integrations - Google Workspace Admin Help
https://support.google.com/a/answer/15967955?hl=en

[8] [11] [12] Build a step | Google Workspace add-ons | Google for Developers
https://developers.google.com/workspace/add-ons/workflows/build-a-step

[9] [10] [16] Extend Google Workspace Flows | Google Workspace add-ons | Google for Developers
https://developers.google.com/workspace/add-ons/workflows

[13] [14] [15] The Workspace Flows event object | Google Workspace add-ons | Google for Developers
https://developers.google.com/workspace/add-ons/workflows/event-objects

[17] [18] [19] [20] Scopes | Google Workspace add-ons | Google for Developers
https://developers.google.com/workspace/add-ons/concepts/workspace-scopes

[24] Workspace Flows (Closed Alpha) Documentation
https://48970008.fs1.hubspotusercontent-eu1.net/hubfs/48970008/Workspace%20Flows%20(Closed%20Alpha)%20Documentation-1.pdf

[28] [29] [38] Google Workspace Flows : r/googleworkspace
https://www.reddit.com/r/googleworkspace/comments/1oz1n89/google_workspace_flows/

[31] [32] Google Gemini Workspace Flows: How I Automated My Entire Workflow : r/AISEOInsider
https://www.reddit.com/r/AISEOInsider/comments/1ox2kbz/google_gemini_workspace_flows_how_i_automated_my/