

Google Cloud - Co... · Following

# Next-Generation Google Apps Script Development: Leveraging Antigravity and Gemini 3.0

7 min read · 6 hours ago



Kanshi Tanaike

Following ▾

Listen

Share

More



## Abstract

This article demonstrates a cutting-edge workflow for Google Apps Script development using Google Antigravity and Gemini 3.0. By integrating `gas-fakes` via the Model Context Protocol (MCP), we establish an environment where autonomous agents can generate, unit-test, and execute cloud-based scripts locally, revolutionizing the standard GAS development lifecycle.

## Introduction

Google Antigravity has officially been released. [Ref](#) This is a revolutionary “Agent-first” IDE powered by Gemini 3, designed to empower autonomous AI agents to plan, code, and verify tasks across the Editor, Terminal, and Browser. It is anticipated that this platform will trigger a paradigm shift in how we develop

applications and auto-generate comprehensive documentation, moving the industry from simple code completion to fully agentic workflows.

I believe this technology brings a transformative evolution to the development of Google Apps Script (GAS) by finally bridging the gap between local AI agents and the GAS cloud runtime. I have previously published several articles on modernizing GAS development, specifically regarding secure local execution and sandboxing. [Ref](#) and

[Open in app ↗](#)

# Medium



combination allows the AI to autonomously generate, unit-test, and refactor GAS code in a local environment before deployment, solving the historical challenge of local GAS execution.

## Environment Setup

To implement this workflow, we must first establish a communication bridge between the Antigravity IDE and the Google Apps Script runtime.

### 1. Install Google Antigravity

Begin by setting up the IDE. Please check the official release and installation guide at <https://antigravity.google/>.

### 2. Install gas-fakes

We will use `gas-fakes` to emulate the GAS environment locally. Install the CLI tool via `npm`.

```
npm -g install @mcpher/gas-fakes
```

### 3. Authorize Access to Google Services

To allow `gas-fakes` to interact with your Google services (Drive, Sheets, etc.) for real-world testing, you must authorize the client.

First, create a `.env` file to store your project configuration. The tool will prompt you for the Project ID (which can be found in your GCP console).

```
gas-fakes init
```

Next, run the authorization command. This will guide you through the OAuth flow to log into your Google account and grant the necessary permissions.

```
gas-fakes auth
```

Finally, enable the required Google APIs for your project to ensure all necessary services are accessible.

```
gas-fakes enableAPIs
```

## Verification:

Once the setup is complete, run a simple test command to verify that `gas-fakes` is correctly configured. This command uses the tool's sandbox to execute a script that retrieves the name of the root folder in your Google Drive.

```
gas-fakes -s "const rootFolder = DriveApp.getRootFolder(); const rootFolderName = rootFolder.getName(); Logger.log(rootFolderName)"
```



If the command executes without errors and prints your root folder's name, your local environment is ready for development.

## 4. Install Clasp

Install [Clasp](#), the command-line tool for Google Apps Script:

```
npm install -g @google/clasp
```

Authorize Clasp by following the instructions in the [official GitHub repository](#).

## Integration: Model Context Protocol (MCP)

The core of this workflow is the **Model Context Protocol (MCP)**. The `gas-fakes` and `clasp` CLI functions as an MCP server, allowing the Gemini 3 agents within Antigravity to execute tools and scripts directly. The official documentation for installing MCP servers can be viewed [here](#).

For this integration, the MCP server must be manually configured. Open a file named `mcp_config.json` and paste the following configuration:

```
{  
  "mcpServers": {  
    "gas-fakes": {  
      "command": "gas-fakes",  
      "args": [  
        "mcp"  
      ],  
      "disabled": false,  
      "disabledTools": []  
    },  
    "clasp": {  
      "command": "clasp",  
      "args": [  
        "mcp"  
      ]  
    }  
  }  
}
```

After saving the file, refresh the MCP server list in Antigravity. When installed correctly, the `gas-fakes` and `clasp` tools will become visible to the AI agent, as shown below:

Manage MCP servers

6 / 100 tools View raw config Refresh

clasp	5 / 5	gas-fakes	Configure	Enabled
gas-fakes	1 / 1	1. run-gas-by-gas-fakes		<input checked="" type="checkbox"/>
Use this to safely run Google Apps Script in a sandbox using gas-fakes. # Important - Use the extension of the Google Apps Script files as `js`. Don't use `gs` - When you provide the generated Google Apps Script to the tool "gas-fakes" of the MCP server "gas-development-kit-extension", please be careful of the following rule. For example, when you generated a Google Apps Script like `function sample() { script }`, please add `sample();` to execute the function. Or, you can also create a Google Apps Script without enclosing the script with `function sample() { script }`.				

## Practical Demonstration

With the environment configured, we can now task the Gemini 3 agent with creating and testing Google Apps Scripts. The agent plans the execution, writes the code, and uses the `gas-fakes` tool to verify the output.

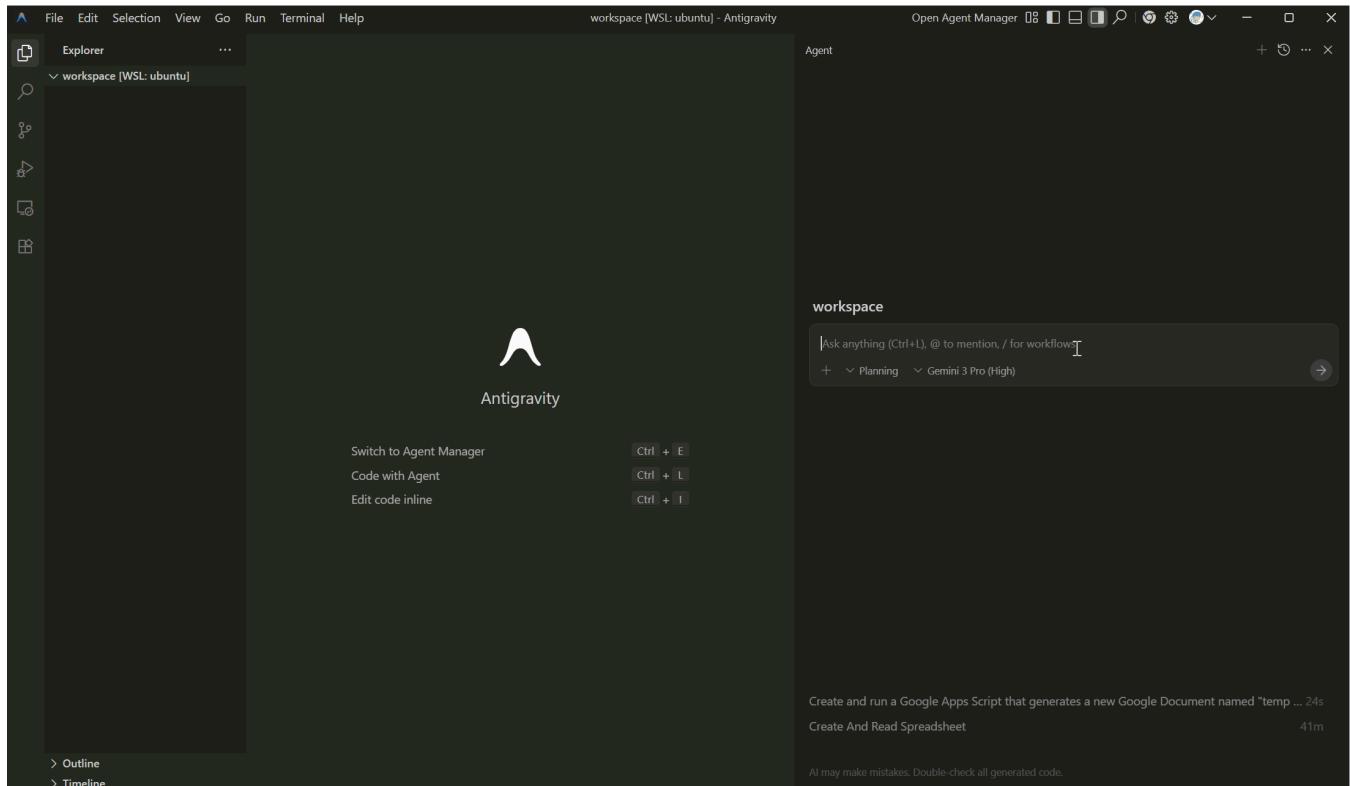
### Sample 1: Document Creation

Prompt:

Create and run a Google Apps Script that generates a new Google Document named



**Result:** The agent successfully calls `DocumentApp` through the local bridge.



The generated script is as follows:

```
function createDoc() {
  var doc = DocumentApp.create("temp document");
  var body = doc.getBody();
  body.appendParagraph("sample");
  console.log("Document created: " + doc.getId());
}

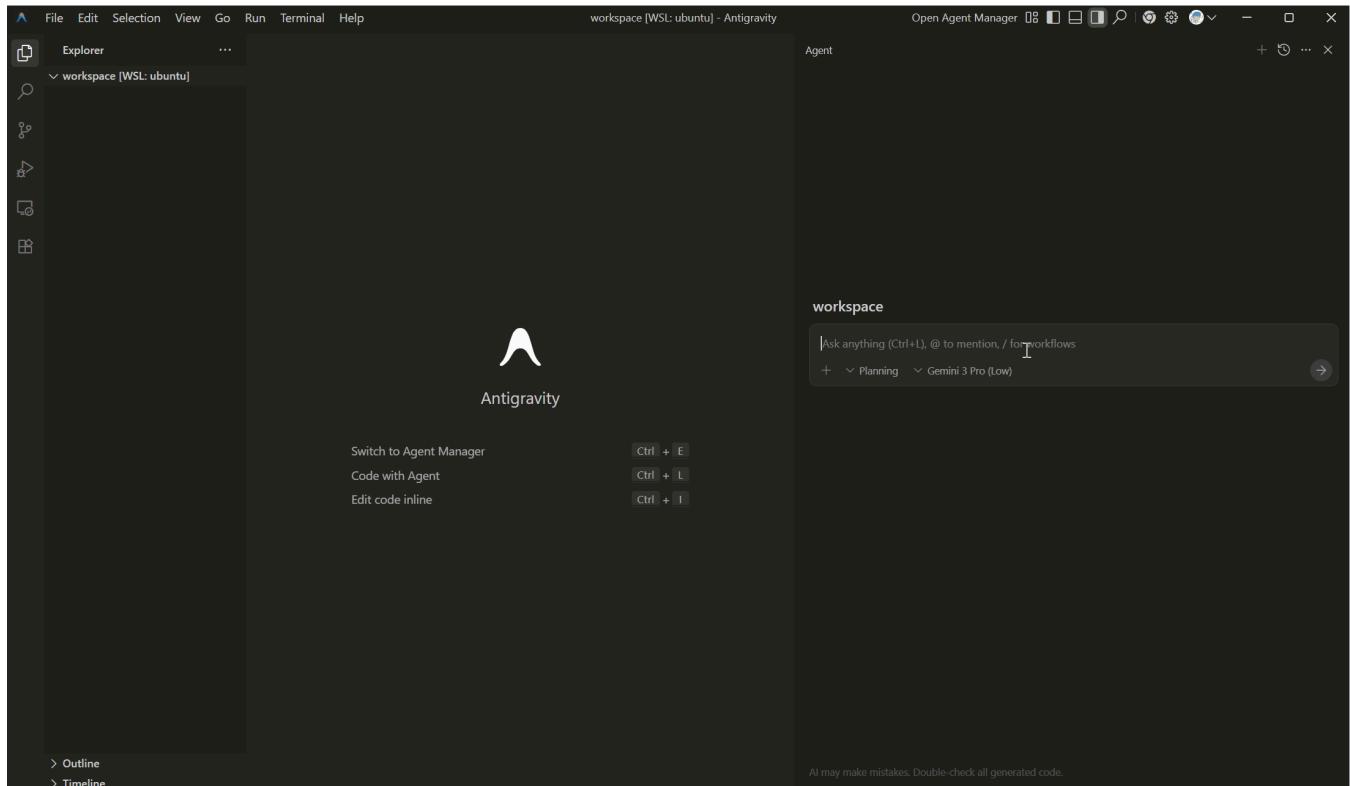
createDoc();
```

## Sample 2: Spreadsheet Manipulation

This example tests the agent's ability to retrieve mathematical outputs from Google Sheets created in the script.

### Prompt:

Create a new Google Spreadsheet by putting a formula `=GOOGLEFINANCE("CURRENCY":



**Implementation Plan:** The Antigravity agent analyzes the request and produces the following plan:

```
# Create and Read Google Spreadsheet

## Goal Description
Create a new Google Spreadsheet, set the formula `=GOOGLEFINANCE("CURRENCY:USD")` in cell A1, and log the value to the console.

## Proposed Changes
### Script Generation
#### [NEW] [create_sheet.js](file:///temp/workspace/create_sheet.js)
- Create a script that:
  - Creates a new spreadsheet using `SpreadsheetApp.create()` .
  - Gets the first sheet using `getSheets()[0]` (as `getActiveSheet()` is unavailable).
  - Sets the formula in A1.
  - Flushes the spreadsheet.
  - Gets and logs the value of A1.

## Verification Plan
### Automated Tests
- Run the script using `mcp0_run-gas-by-gas-fakes` .
- Check the output for the logged value.
```

**Walkthrough:** The agent executes the plan and confirms the results.

```
# Walkthrough - Create and Read Google Spreadsheet
```

I have successfully created a Google Spreadsheet, set a currency conversion for

```
## Execution Results
```

The script [[create\\_sheet.js](#)](file:///temp/workspace/create\_sheet.js) was execut

```
```javascript
function createAndReadSheet() {
  var ss = SpreadsheetApp.create("CurrencySheet");
  var sheet = ss.getSheets()[0];
  var cell = sheet.getRange("A1");
  cell.setFormula('=GOOGLEFINANCE("CURRENCY:USDJPY")');
  SpreadsheetApp.flush();
  var value = cell.getValue();
  console.log("Value in A1: " + value);
}
```

```

```
### Output
```

The script outputted the following value for USD/JPY:

```
```

```

```
Value in A1: 157.21
```

```

```
## Verification
```

The value `157.21` confirms that:

1. The spreadsheet was created.
2. The formula `=GOOGLEFINANCE("CURRENCY:USDJPY")` was set and calculated.
3. The value was correctly retrieved and logged.



## Sample 3: Push Google Apps Script to Google Drive

This example tests the agent's ability to push the generated Google Apps Script to Google Drive.

**Prompt:**

```
Create a new Google Apps Script project in Google Drive and push the script `cr
```



```

File Edit Selection View Go Run Terminal Help
workspace [WSL: ubuntu] - Antigravity - create_doc.js
Open Agent Manager Agent
Explorer workspace [WSL: ubuntu]
JS create_doc.js > ...
JS create_doc.js > ...
1 function createDoc() {
2   var doc = DocumentApp.create
3     ("temp document");
4   var body = doc.getBody();
5   body.appendParagraph("sample");
6   console.log("Document created:
7     " + doc.getId());
8 }
9 createDoc();

```

workspace

Ask anything (Ctrl+L), @ to mention, / for workflows

+ Planning Gemini 3 Pro (High)

> Outline > Timeline

AI may make mistakes. Double-check all generated code.

## Implementation Plan:

```

# Create GAS Project and Push Script

## Goal Description
Create a new Google Apps Script project in the current directory and push the entire project to a repository.

## Proposed Changes
### Google Apps Script Project
#### [NEW] [.clasp.json](file:///temp/workspace/.clasp.json)
- Created automatically by `clasp create`.

#### [MODIFY] [create_doc.js](file:///temp/workspace/create_doc.js)
- Will be pushed to the GAS project.

## Verification Plan
### Automated Tests
- Use `mcp0_list_projects` to verify the project exists.
- Use `mcp0_pull_files` to verify the file was pushed (optional, or just trust

```

## Walkthrough:

```
# Walkthrough - Create GAS Project and Push Script
```

I have successfully created a new Google Apps Script project and pushed the [cr]

```
## Changes
```

```
### Google Apps Script Project
```

```
#### [NEW] [.clasp.json](file:///temp/workspace/.clasp.json)
```

- Created automatically by `clasp create`.

```
#### [MODIFY] [create_doc.js](file:///temp/workspace/create_doc.js)
```

- Pushed to the GAS project.

```
## Verification Results
```

```
### Automated Tests
```

- Verified that the project `create\_doc\_project` exists using `clasp list`.

- Verified that [create\_doc.js](file:///temp/workspace/create\_doc.js) was pushed.



## Summary

This article presented a paradigm shift in Google Apps Script (GAS) development by combining the “Agent-first” capabilities of **Google Antigravity** (powered by **Gemini 3.0**) with the local emulation power of **gas-fakes**.

Key takeaways from this next-generation workflow include:

- **The MCP Bridge:** By utilizing the **Model Context Protocol (MCP)**, we successfully transformed the `gas-fakes` CLI and `clasp` into accessible tools for the IDE’s AI agents. This creates a seamless bridge between the local development environment and the GAS cloud runtime.
- **Autonomous Lifecycle Management:** We demonstrated that Gemini 3.0 agents can autonomously handle the entire development lifecycle:
- **Planning:** Analyzing requirements to create implementation strategies.
- **Coding & Testing:** Writing scripts and verifying logic locally (e.g., handling spreadsheet formulas) without needing to deploy to the cloud first.
- **Deployment:** Utilizing `clasp` to push verified code to production environments.

- **Self-Healing Workflows:** The workflow allows agents to identify environment limitations (such as the absence of `getActiveSheet()` in the local emulator) and autonomously refactor code to ensure successful execution.

By moving from simple code completion to fully agentic verification and deployment, developers can significantly reduce context switching and accelerate the delivery of robust Google Apps Script solutions.

[Google Apps Script](#)[Gemini](#)[Antigravity](#)[Mcp Server](#)[Google Workspace](#)[Following](#)

## Published in Google Cloud - Community

70K followers · Last published 6 hours ago

A collection of technical articles and blogs published or curated by Google Cloud Developer Advocates. The views expressed are those of the authors and don't necessarily reflect those of Google.

[Following](#)

## Written by Kanshi Tanaike

678 followers · 15 following

Physicist / Ph.D. in Physics / Google Developer Expert

No responses yet



Stéphane Jaubert

What are your thoughts?

## More from Kanshi Tanaike and Google Cloud - Community



In Google Cloud - Community by Kanshi Tanaike

### Workaround: Smart Chips with Google Apps Script

Google Apps Script lacks direct smart chip access; workarounds exist to retrieve their data despite current limitations.

Mar 15 3



...

 In Google Cloud - Community by Thomas Chong

## Building AI Agents Visually with Google ADK Visual Agent Builder

Google ADK v1.18.0 introduced the Visual Agent Builder—a browser-based interface that lets you design, configure, and test complex...

Nov 10  152  6



| product_name                                      | description                                       | gifiability_ |
|---|---|--------------|
| AquaClear 50 Gallon Aquarium                      | This stunning aquarium provides a spacious and... |              |
| Cozy Naps Cat Bed                                 | This luxurious pet bed provides a cozy and com... |              |
| AquaClear Aquarium Background                     | This vibrant underwater scene creates a captiv... |              |
| AquaClear 20 Gallon Aquarium                      | This spacious glass enclosure provides a comfo... |              |
| Cozy Naps Cat Scratching Post with Condo          | This cat tree provides a cozy resting spot and... |              |
| ...   | ...   | ...          |
| AquaClear Aquarium Water Clarifier                | This aquarium water clarifier helps to remove ... |              |
| AquaClear Aquarium Thermometer with Digital Di... | This aquarium thermometer allows you to accura... |              |
| Ocean Bites Kitten Dry Food                       | This delicious recipe of crunchy kibble will k... |              |
| Top Dog Lamb & Brown Rice Formula                 | This recipe helps support healthy digestion an... |              |
| AquaClear Aquarium Cleaning Kit                   | This complete pool cleaning kit allows you to ... |              |

 In Google Cloud - Community by Alicia Williams

## Analyze anything with AI-powered SQL in BigQuery

A first look at AI.SCORE, AI.CLASSIFY, and AI.IF, the new Gemini-powered functions for semantic analysis in BigQuery.

Oct 30 63 1



SampleTable1

| B2 | C2 | D2 | E2 | F2 |
|----|----|----|----|----|
| B3 | C3 | D3 | E3 | F3 |
| B4 | C4 | D4 | E4 | F4 |
| B5 | C5 | D5 | E5 | F5 |
| B6 | C6 | D6 | E6 | F6 |

SampleTable2

| B9  | C9  | D9  | E9  | F9  | G9  | H9  |
|-----|-----|-----|-----|-----|-----|-----|
| B10 | C10 | D10 | E10 | F10 | G10 | H10 |
| B11 | C11 | D11 | E11 | F11 | G11 | H11 |
| B12 | C12 | D12 | E12 | F12 | G12 | H12 |
| B13 | C13 | D13 | E13 | F13 | G13 | H13 |

In Google Cloud - Community by Kanshi Tanaike

## Workaround: Using Google Sheets Tables with Google Apps Script

This is a current workaround for using tables in Google Sheets with Google Apps Script.

Nov 2, 2024 6



See all from Kanshi Tanaike

See all from Google Cloud - Community

## Recommended from Medium

The Era of Action Model with Gemini 3 Pro & Google Antigravity

A deep dive into the launch of new foundation model Gemini 3 Pro and new IDE Google Antigravity

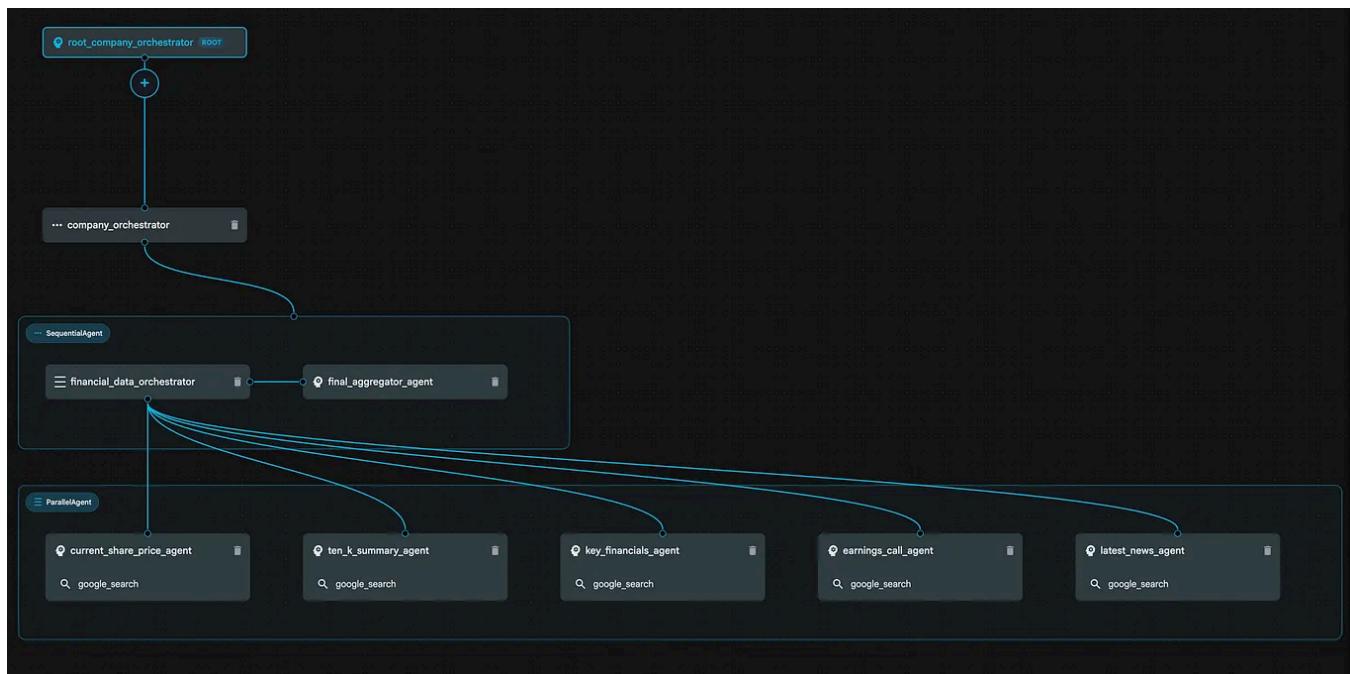
Thomas Chong      Google Developer Expert in AI

In Google Cloud - Community by Thomas Chong

## The Era of Action with Gemini 3 Pro & Google Antigravity

For the last two years, the AI industry has been obsessed with “reasoning”—how well a model can think through a problem. Today, with the...

2d ago    45    1



In AI Cloud Lab by Arjun Prabhulal

## Build AI Agents with Google ADK Visual Agent Builder

No Code agent building using ADK Agent Builder

Nov 14

9

2



...



Veer Muchandi

## The Developer's Toolkit: Speed, Customization, and Reusability with the ADK

Welcome to Post 2, where we unlock the Agent Development Kit (ADK)—the essential tool for developers who need to move beyond the no-code...

Oct 9

1



...



Anthony S

## Breaking Free: The Strategic Case for Open-Source IaC and Accelerated Codification

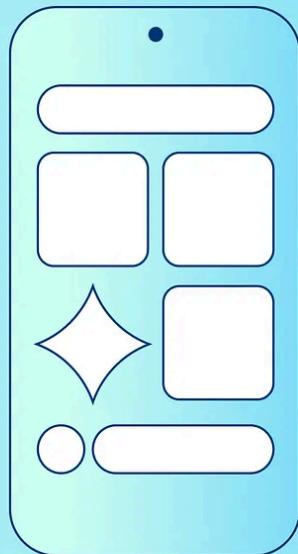
Breaking Free: The Strategic Case for Open-Source IaC and Accelerated Codification

Nov 12 · 1



...

# Introducing the GenUI SDK for Flutter



Flutter

In Flutter by Abdallah W Shaban

## Rich and dynamic user interfaces with Flutter and generative UI

We're thrilled to introduce GenUI SDK for Flutter, now available in alpha on pub.dev.

2d ago · 509



...





Lew C

## Google's Antigravity is anti-good

A slow start for promising software

1d ago 7 2



...

See more recommendations