

# GitHub Spark, il «vibe coding» secondo GitHub!



Massimo Bonanni

~~Senior~~ Technical Trainer @ Microsoft

VINTAGE



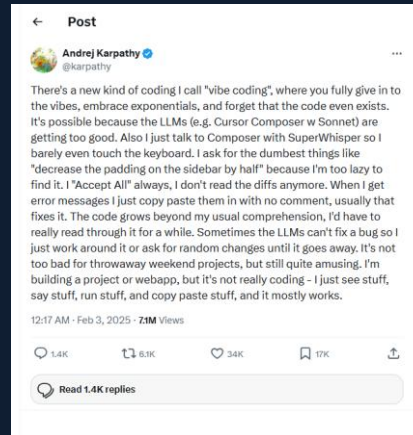
# Vibe Coding.... Where the term came from.

FEB 2, 2025 · X / TWITTER

“

*There's a new kind of coding I call 'vibe coding', where you fully give in to the vibes, embrace exponentials, and forget that the code even exists.*

— **Andrej Karpathy** · co-founder of OpenAI, ex-AI lead at Tesla



## Collins Word of the Year 2025

### Definition of 'vibe coding'

#### vibe coding

in British English

(ˈvaɪb ˌkəʊdɪŋ) ⓘ ⓘ

**noun** slang

the use of artificial intelligence prompted by natural language to assist with the writing of computer code

Collins English Dictionary. Copyright © HarperCollins Publishers

#### Derived forms

vibe coder **noun**

vibe-code **verb**

JAN 2023

'English is the new programming language'

FEB 2025

Karpathy coins 'vibe coding' on X

MAR 2025

Merriam-Webster: slang & trending

2025

Collins Word of the Year

# What is GitHub Spark?

**GitHub's platform for building and shipping full-stack intelligent apps — starting from a conversation.**

*Describe your app · Spark generates frontend, backend, AI and data · You publish in one click.*

## Natural language first

Describe what you want.  
The agent builds it.

## Live preview

See your changes  
in real time.

## One-click publish

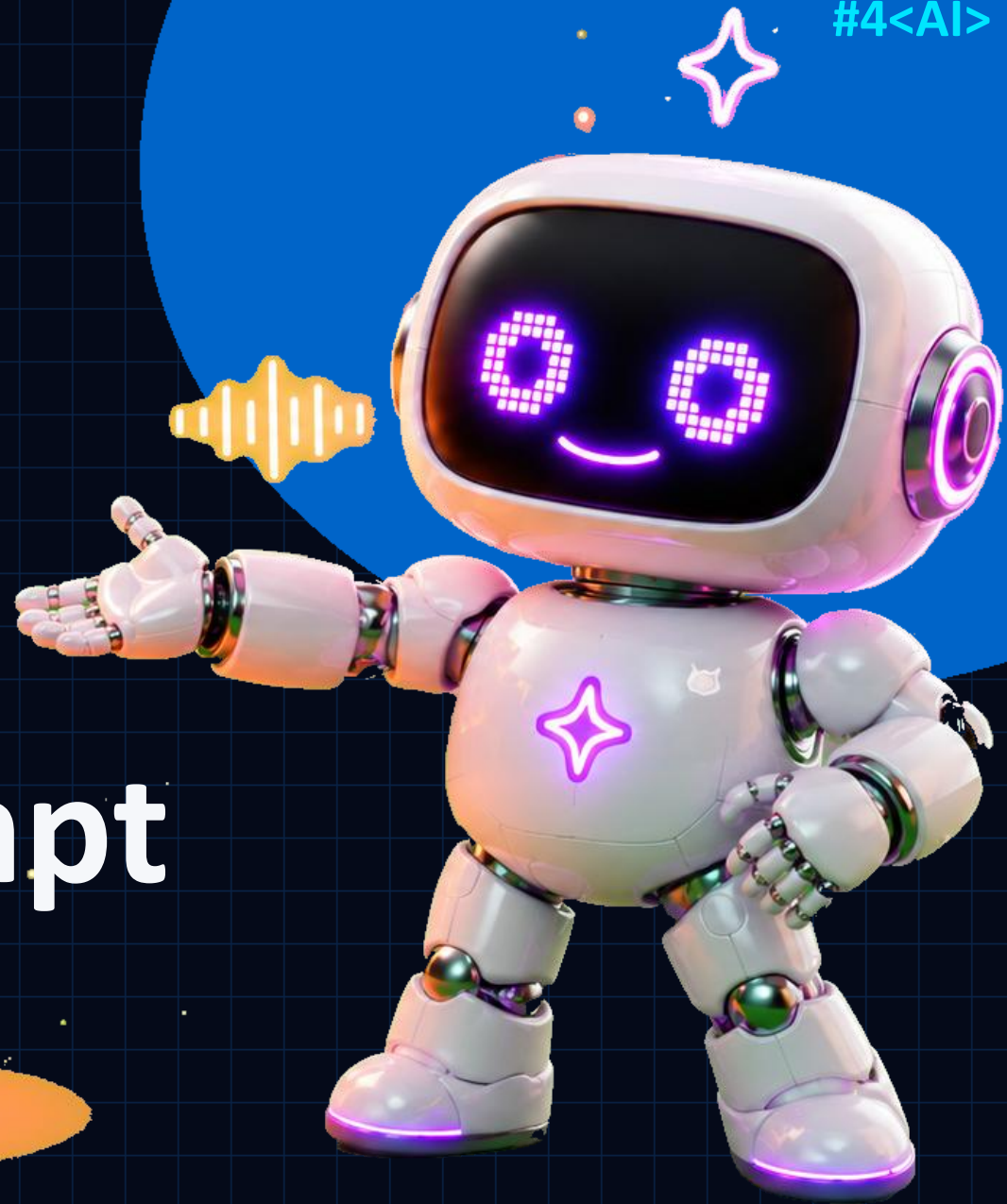
Hosting, auth and runtime  
ready out of the box.

#4<AI>



*a habit tracker  
with dark mode  
and a chart*

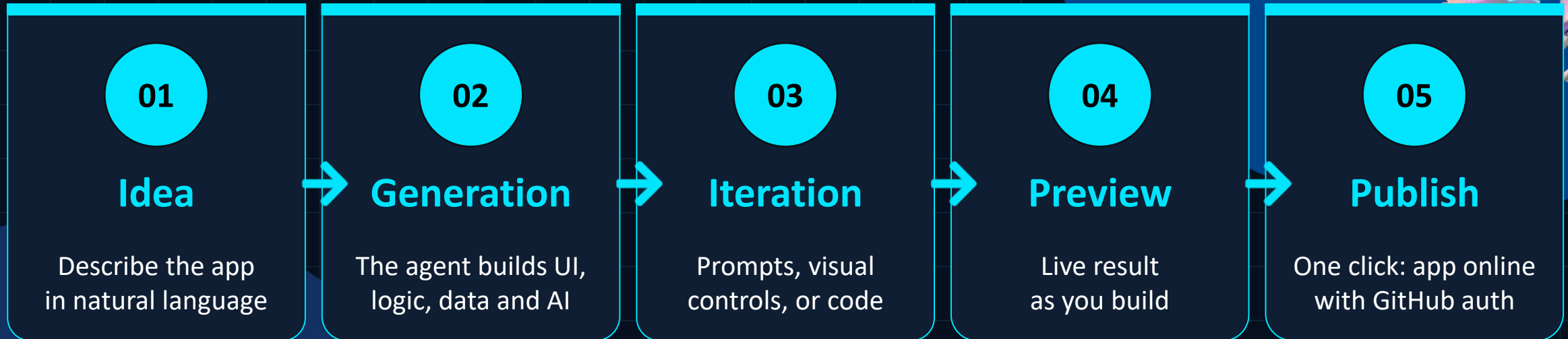
#4<AI>



DEMO

First Vibe Prompt

# How Spark works, from idea to deploy



**The loop closes in minutes — and it can reopen with every new idea.**

Code, preview and deploy stay in sync: no copy-paste, no tool-switching.

# What a Spark app is made of

## Frontend

React · TypeScript

## AI layer

GitHub Models (native inference)

## Data

Managed key-value store · Azure Cosmos DB

## Identity

Built-in GitHub authentication

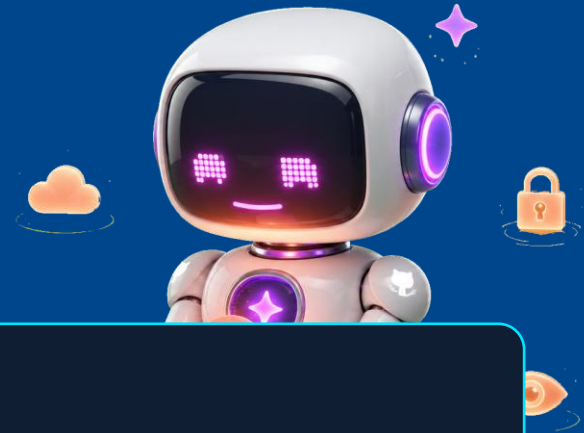
## Runtime

Azure Container Apps · integrated environment

## Built for developers

- ✓ **Copilot completions** — right inside the Spark editor
- ✓ **GitHub Codespaces** — cloud IDE with agent mode
- ✓ **Two-way repo sync** — every change becomes a commit
- ✓ **Copilot Cloud Agent** — delegate tasks in the background
- ✓ **GitHub Actions** — CI/CD when you need to scale





# What makes Spark different



## Natural language first

Build and tweak by talking to your app.  
Targeted visual edits too.



## Built-in AI

Chatbots, content generation, smart  
automation: prompt and Spark wires  
the inference.



## Managed data store

When data needs to live somewhere,  
Spark provisions a Cosmos DB key-value  
store.



## Auth & security

GitHub-authenticated access. Privacy  
modes: private, org-wide, or public.



## One-click deploy

Spark provisions hosting, runtime and  
domain. No YAML, no pipelines.



## GitHub-native

Repos, branches, PRs, Actions,  
Codespaces: a click away.

## GitHub Models, inside the app

AI isn't an integration you wire up:  
it's a building block of the app.

- **Inference ready to go**

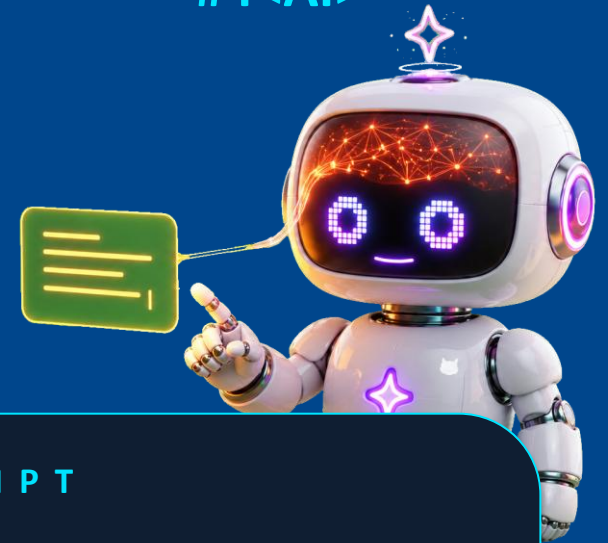
Spark wires up the components it needs the moment you describe an AI feature.

- **Editable system prompts**

Step into the prompts that drive behavior and tune them yourself.

- **Typical use cases**

Summaries, automatic tags, semantic search, conversational assistants.



### SAMPLE PROMPT

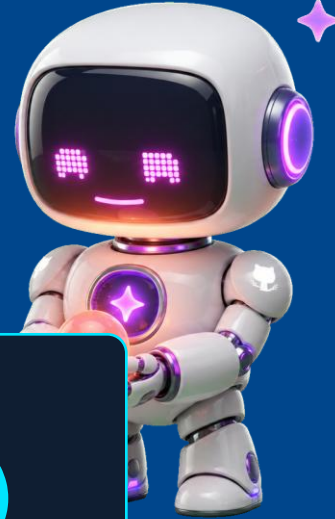
*“Add a summary field to my notes, generated automatically on save, in three sentences max.”*

---

Spark generates the button, the GitHub Models call, the state management and the error fallback.

**You stay free to tweak the system prompt.**

Everything is managed. You focus on the product.



#### HOSTING



### Azure Container Apps

Integrated runtime. No servers to set up, no manual deploys.

#### PERSISTENCE



### Cosmos DB (key-value)

Auto-provisioned, up to 512 KB per record. Designed for small, fast data.

#### IDENTITY



### GitHub Auth

Users sign in with their GitHub account. No login form to build.

#### VISIBILITY



### Private · Org · Public

You decide who sees the app. Read-only mode to showcase without edits.

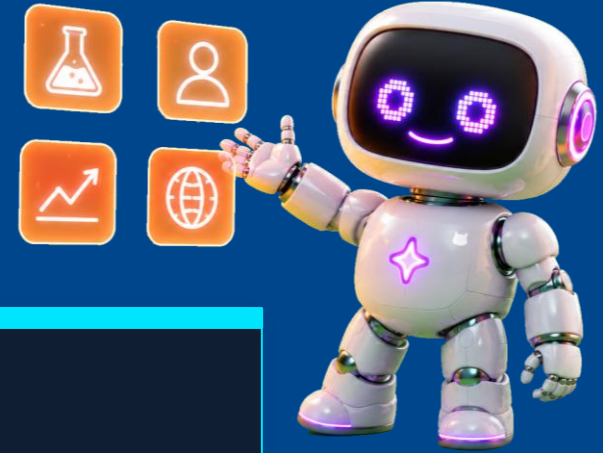
#4<AI>



DEMO

Refine your app

## Four scenarios where Spark really pays off



### Prototyping

Stop explaining the idea — show it. Working prototypes in minutes, instant feedback.



### Personal apps

Trackers, planners, custom assistants. The things no off-the-shelf app covers.



### SaaS launchpad

Validate an idea with real users before investing. Scale only when traction proves it.



### Web essentials

Portfolios, landing pages, marketing sites — but living, with AI features built in.

## Where Spark isn't (yet) the answer



### Opinionated stack

React + TypeScript. A deliberate reliability choice: other stacks aren't on the menu.



### External libraries with care

You can add them, but compatibility with the Spark SDK isn't guaranteed. Test thoroughly.



### Lightweight data store

Key-value, records up to 512 KB. Great for small apps. Not a replacement for a relational OLTP DB.



### Public preview

Capabilities and limits can evolve quickly. Keep that in mind for critical projects.



### Shared storage by default

The published data store is shared across all users of the app. Curate it before going public.

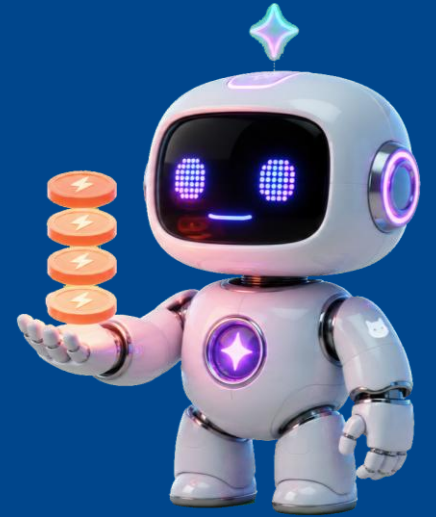


### Plan availability

Available on Copilot Pro+ and Enterprise. Admins can restrict access.



#4<AI>



## How Spark is paid for

### Copilot Pro+

**\$39** /user/month

For the individual power user.  
Includes Spark and the rest of Copilot.

### Copilot Enterprise

**\$39** /user/month

Spark included for every user.  
Centralized governance and policies.

### What's included

Hosting · Compute · AI inference · Storage — all in the integrated runtime.  
Unlimited manual editing. Multiple apps in parallel. Pay-as-you-go coming soon.

# Spark vs traditional development

#4<AI>

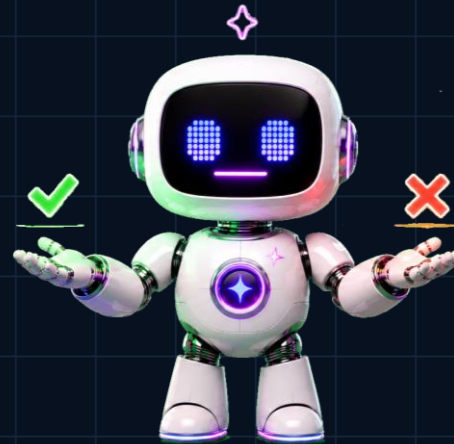


|                         | With Spark                        | Traditional development             |
|-------------------------|-----------------------------------|-------------------------------------|
| Starting point          | Idea in plain English             | Empty repo + stack decisions        |
| Time to first prototype | Minutes                           | Days or weeks                       |
| Frontend                | Generated (React/TS)              | Hand-written and wired up           |
| Backend & data          | Abstracted, managed               | Designed and provisioned            |
| AI features             | Prompt → integration in place     | SDKs, keys, fallbacks, cost mgmt    |
| Deployment              | One click, hosting included       | Pipelines, infra, DNS, monitoring   |
| Deep control            | Possible via Codespaces + Copilot | Total                               |
| Where it wins           | Speed, ideas, prototypes          | Complex systems, performance, scale |

# Spark yes / Spark no

## Spark is the right call when...

- ✓ you want to prototype an idea fast and gather feedback
- ✓ you're building internal tools or personal apps
- ✓ you need an MVP with AI features ready to go
- ✓ you want to validate a SaaS before investing in infra
- ✓ a mixed team (PM, designer, dev) needs to iterate together
- ✓ speed from idea-to-link matters more than deep control



## Look elsewhere if...

- ✗ you need a stack other than React + TypeScript
- ✗ the domain demands a relational or transactional DB
- ✗ the app handles large data volumes or files
- ✗ it's enterprise-critical with bespoke compliance needs
- ✗ the team needs full control over build, deploy and network
- ✗ the app is the value proposition: deep engineering pays off

#4<AI>

# Any questions?



Massimo Bonanni

*Senior Technical Trainer @ Microsoft*

*[aka.ms/maxlinkedin](https://aka.ms/maxlinkedin)*



GitHub Spark, il vibe coding secondo GitHub!