

Know what your change **breaks** before you merge.

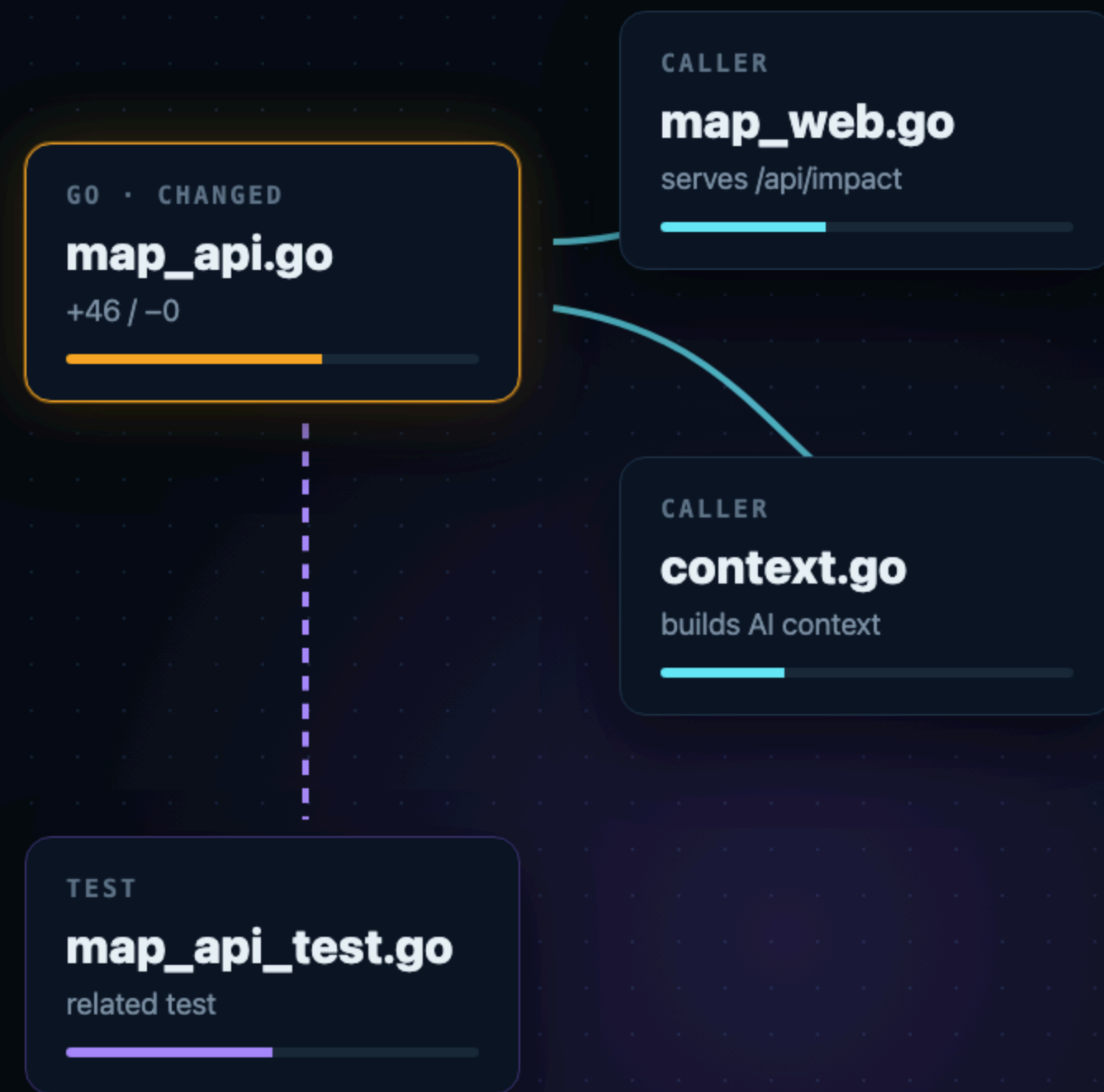
A live map of your Git repo. IBM Bob explains the impact of a change, grounded in real diffs, callers, dependencies and tests.

✦ IBM Bob 2.0 Hackathon

Go

Svelte

Bob Shell



THE PROBLEM

"What else could this change break?"

It is the hardest question in code review, and today it is answered by guessing.

01

Diffs lack context

A diff shows lines, not the callers and tests that depend on them.

02

Tests get missed

Reviewers rarely know which tests cover the code that changed.

03

AI chat guesses

Pasting code into a chatbot gives confident answers that ignore how the repo is actually connected.

THE IDEA

Facts from codemap. Reasoning from Bob.

DETERMINISTIC

codemap

- Scans the Git repo and builds a file graph
- Detects Go calls and TypeScript imports
- Tracks working-tree and branch diffs
- Finds callers, dependencies and related tests

IBM BOB

Semantic explanation

- What changed, in plain English
- Which downstream code may be affected, and why
- Which tests to review
- How selected files work together

Bob never has to guess the architecture: every answer is grounded in evidence codemap can show you.

HOW IT WORKS

From a changed file to an explanation



```
bob run --mode ask --format json --max-turns 3 --max-cost 1 --disable-mcp --workspace <repo> ← context on stdin
```

FEATURE 1

Analyze impact with Bob

WHAT BOB RETURNS

- **Summary** of the change
- **Affected areas** downstream
- **Reasoning** for each one
- **Relevant tests** to run

Shown next to codemap's evidence: callers, dependencies and related tests.

IBM BOB • IMPACT

Diff storage changed in map_api.go

Snapshots now keep capped diff text server-side. Code that builds context and serves the API is affected.

- 1 map_web.go reads snapshot diffs for /api/impact
- 2 context.go output is unchanged; verify limits

map_api_test.go

map_web_test.go

FEATURE 2

Explain the flow with Bob

Drag a box around several files. Bob explains how they work together, in the order the code runs.
Useful for onboarding onto unfamiliar code.

IBM BOB

How these 4 files work together

- 1 main.go parses options and starts the web map server
- 2 map_web.go builds the first snapshot and serves the API
- 3 map_api.go collects diffs, activity and graph edges
- 4 context.go turns a selection into AI-ready context

BUILT WITH IBM BOB

Bob as partner **and** engine

BOB IDE • DEVELOPMENT

- **/init**: project rules in AGENTS.md and .bob/
- **Runner + API**: bob_runner.go, POST /api/impact, typed errors, tests
- **UI**: Analyze impact button and result panel
- **Release**: safety checks and 7 reviewable commits

Session screenshots in [bob_sessions/](#)

BOB SHELL • RUNTIME

- Headless **bob run** powers both features
- **Ask mode**: explains, never edits code
- Turn and cost caps per request
- Login or BOB_API_KEY from the environment; no stored secrets

Fast, honest, local



0s

Cached answers

Keyed by the exact code evidence. Re-checks are instant and free; any edit triggers a fresh analysis.

No fakes

Clear failures

Bob missing, signed out or slow? The UI says so. It never invents an answer.

1 cmd

Runs locally

`make dev`: one Go binary serves UI and API. No database, no cloud.

Review changes with **evidence**, not guesses.

CODE

github.com/roshan-ican/codemap

DEMO VIDEO

tella.tv/video/codemap-understand-code-git-changes-297a

BOB USAGE EVIDENCE

github.com/roshan-ican/codemap/tree/main/bob_sessions