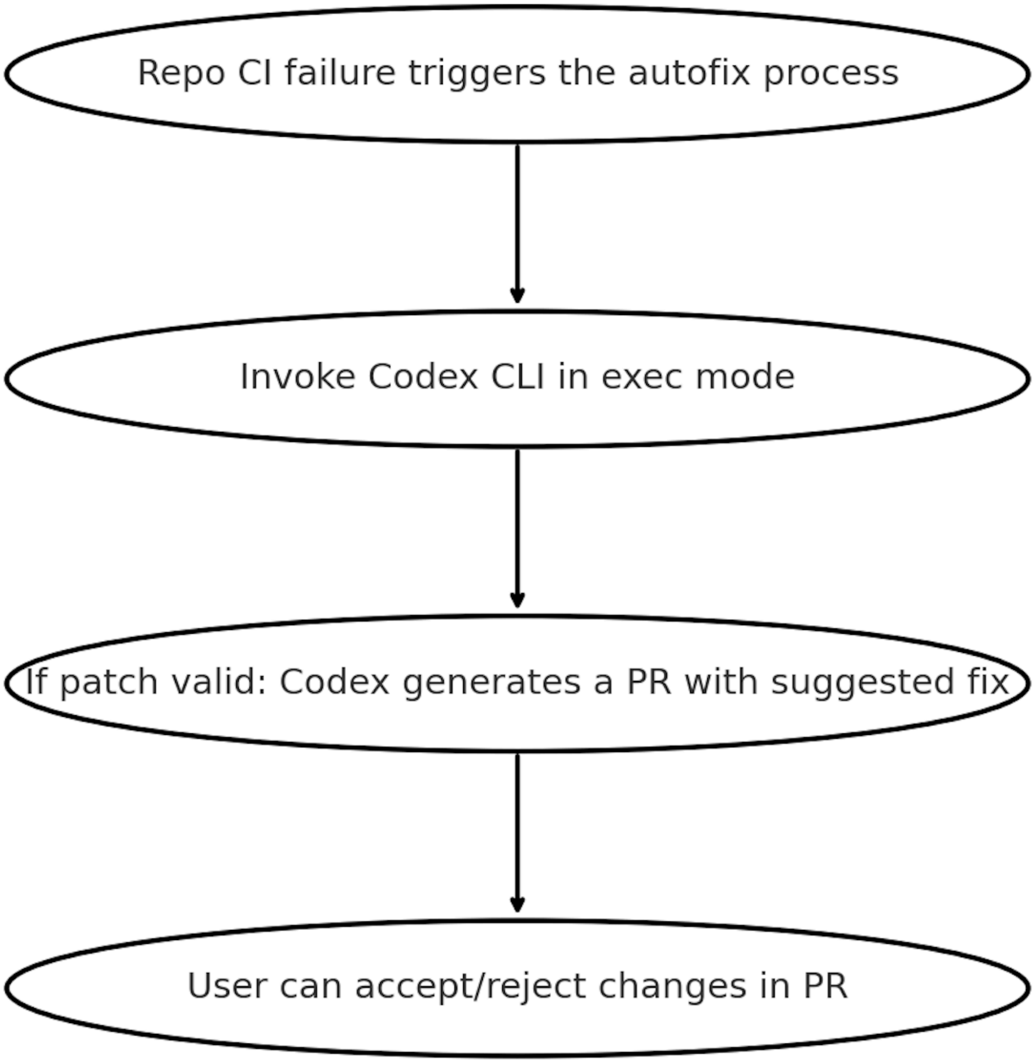
Autofix CI failures on GitHub with Codex CLI

Purpose of this cookbook

This cookbook shows you how to embed the OpenAI Codex CLI into your CI/CD pipeline so that when your builds or tests fail, codex automatically generates & proposes fixes. The following is an example in a node project with CI running in GitHub Actions.

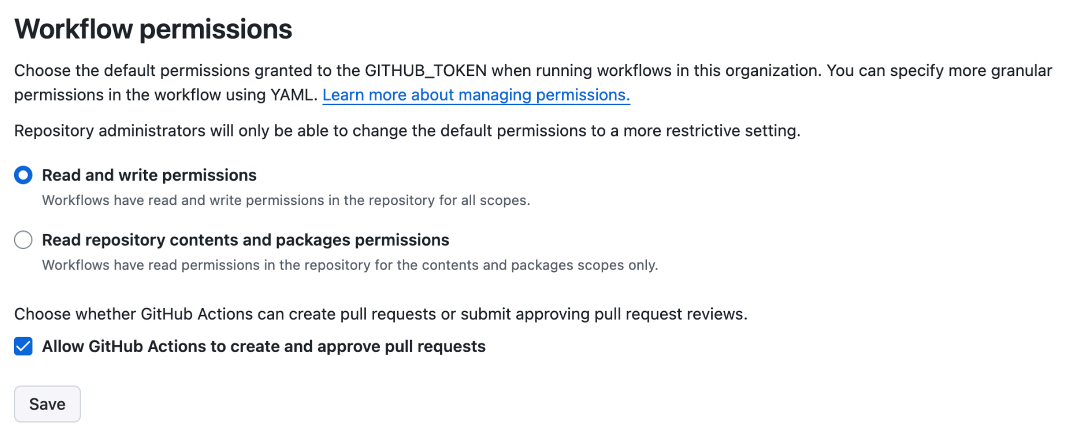
End to End Flow

Below is the pipeline flow we'll implement:


Repo CI failure triggers the autofix process
Invoke Codex CLI in exec mode
If patch valid: Codex generates a PR with suggested fix
User can accept/reject changes in PR

Prerequisites

- A GitHub Repo with Actions workflows
- You'll need to create `OPENAI_API_KEY` as an environment variable in GitHub settings under https://github.com/{org-name}/{repo-name}/settings/secrets/actions. You can also set this at org level(for sharing secrets across multiple repos)
- Codex requires python as a prerequisite to use `codex login`
- You'll need to check the setting to enable actions to create PRs on your repo, and also in your organization:



Workflow permissions

Choose the default permissions granted to the GITHUB_TOKEN when running workflows in this organization. You can specify more granular permissions in the workflow using YAML. Learn more about managing permissions.

Repository administrators will only be able to change the default permissions to a more restrictive setting.

● **Read and write permissions**
Workflows have read and write permissions in the repository for all scopes.

○ **Read repository contents and packages permissions**
Workflows have read permissions in the repository for the contents and packages scopes only.

Choose whether GitHub Actions can create pull requests or submit approving pull request reviews.

☑ **Allow GitHub Actions to create and approve pull requests**

Step 1: Add the Github Action to your CI Pipeline

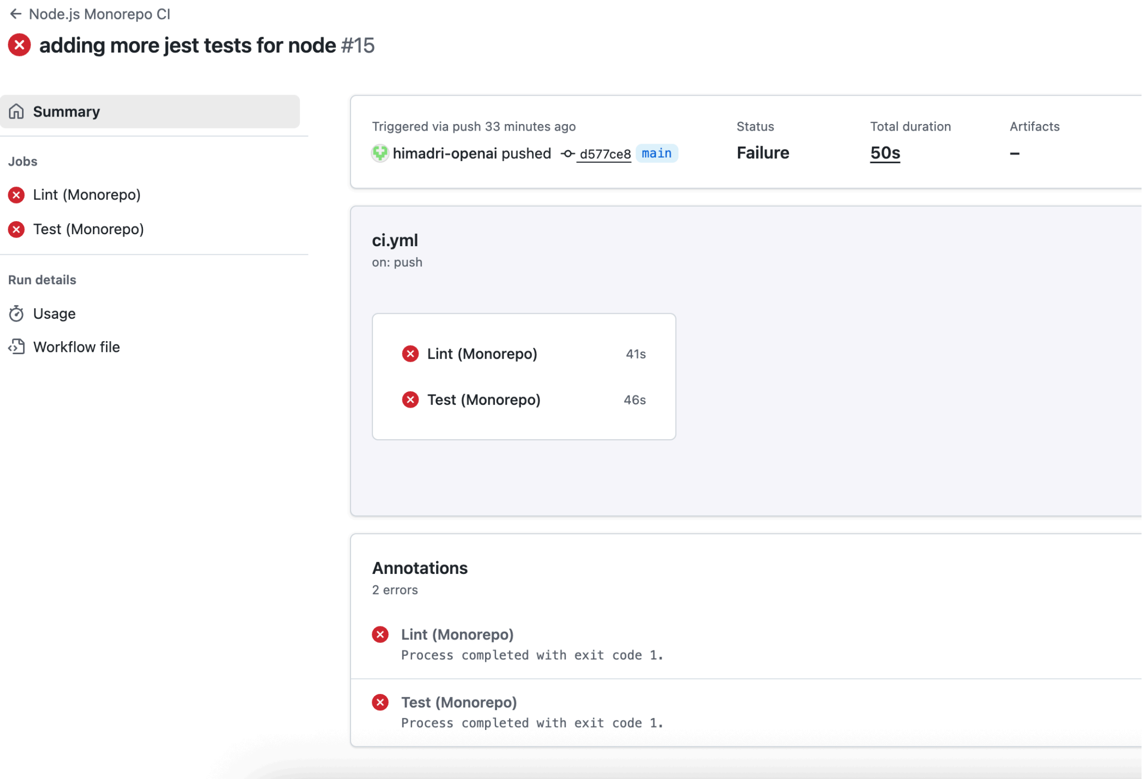
The following YAML shows a GitHub action that auto triggers when CI fails, installs Codex, uses codex exec and then makes a PR on the failing branch with the fix. Replace “CI” with the name of the workflow you want to monitor.

```yaml
name: Codex Auto-Fix on Failure

on:
  workflow_run:
    # Trigger this job after any run of the primary CI workflow completes
    workflows: ["CI"]
    types: [completed]

permissions:
  contents: write
  pull-requests: write

jobs:
  auto-fix:
    # Only run when the referenced workflow concluded with a failure
    if: ${{ github.event.workflow_run.conclusion == 'failure' }}
    runs-on: ubuntu-latest
    env:
      OPENAI_API_KEY: ${{ secrets.OPENAI_API_KEY }}
      FAILED_WORKFLOW_NAME: ${{ github.event.workflow_run.name }}
      FAILED_RUN_URL: ${{ github.event.workflow_run.html_url }}
      FAILED_HEAD_BRANCH: ${{ github.event.workflow_run.head_branch }}
      FAILED_HEAD_SHA: ${{ github.event.workflow_run.head_sha }}
```

```yaml
steps:
  - name: Check OpenAI API Key Set
    run: |
      if [ -z "$OPENAI_API_KEY" ]; then
        echo "OPENAI_API_KEY secret is not set. Skipping auto-fix." >&2
        exit 1
      fi
  - name: Checkout Failing Ref
    uses: actions/checkout@v4
    with:
      ref: ${{ env.FAILED_HEAD_SHA }}
      fetch-depth: 0

  - name: Setup Node.js
    uses: actions/setup-node@v4
    with:
      node-version: '20'
      cache: 'npm'

  - name: Install dependencies
    run: |
      if [ -f package-lock.json ]; then npm ci; else npm i; fi
  - name: Run Codex
    uses: openai/codex-action@main
    id: codex
    with:
      openai_api_key: ${{ secrets.OPENAI_API_KEY }}
      prompt: "You are working in a Node.js monorepo with Jest tests and GitHub Actions. Read the repository, run the test suite, identify the minimal change needed to make all tests pass, implement only that change, and stop. Do not refactor unrelated code or files. Keep changes small and surgical."
      codex_args: '["--config","sandbox_mode=\"workspace-write\""]'

  - name: Verify tests
    run: npm test --silent

  - name: Create pull request with fixes
    if: success()
    uses: peter-evans/create-pull-request@v6
    with:
      commit-message: "fix(ci): auto-fix failing tests via Codex"
      branch: codex/auto-fix-${{ github.event.workflow_run.run_id }}
      base: ${{ env.FAILED_HEAD_BRANCH }}
      title: "Auto-fix failing CI via Codex"
      body: |
        Codex automatically generated this PR in response to a CI failure on workflow `${{ env.FAILED_WORKFLOW_NAME }}`.
        Failed run: ${{ env.FAILED_RUN_URL }}
        Head branch: `${{ env.FAILED_HEAD_BRANCH }}`
        This PR contains minimal changes intended solely to make the CI pass.
```

Step 2: Actions Workflow kicked off

You can navigate to the Actions tab under Repo to view the failing jobs in your Actions workflow.



← Node.js Monorepo CI

adding more jest tests for node #15

Summary

Jobs

- Lint (Monorepo)
- Test (Monorepo)

Run details

- Usage
- Workflow file

Triggered via push 33 minutes ago	Status	Total duration	Artifacts
himadri-openai pushed d577ce8 main	Failure	50s	–

ci.yml
on: push

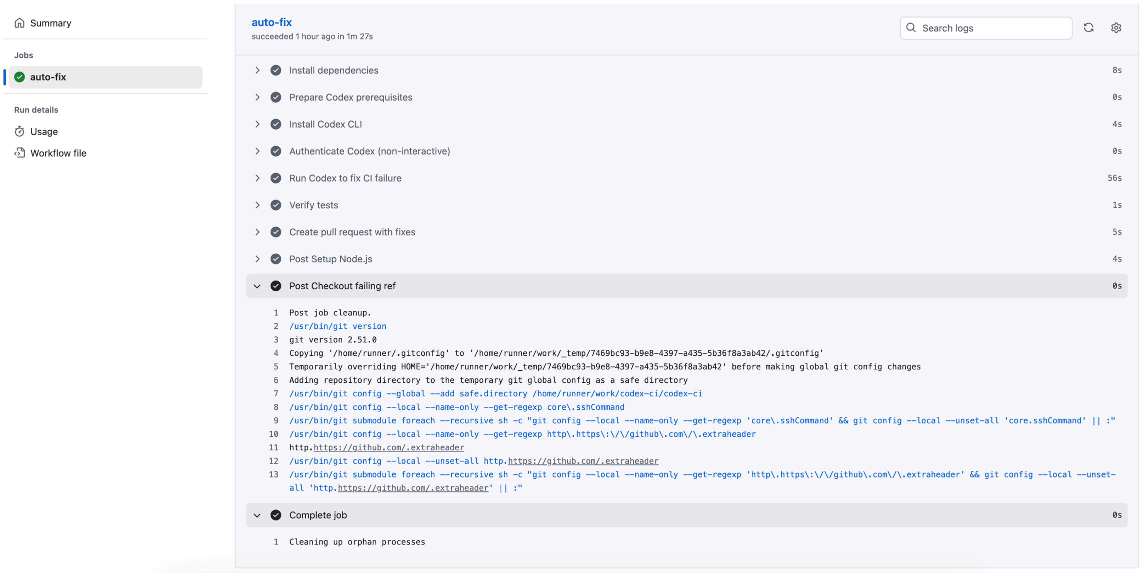
Lint (Monorepo)	41s
Test (Monorepo)	46s

Annotations
2 errors

Lint (Monorepo)
`Process completed with exit code 1.`

Test (Monorepo)
`Process completed with exit code 1.`

The Codex workflow should be triggered upon completion of the failed workflow.



Summary

Jobs

auto-fix

Run details

Usage

Workflow file

auto-fix

succeeded 1 hour ago in 1m 27s

Search logs

```
Install dependencies                     8s
Prepare Codex prerequisites              0s
Install Codex CLI                        4s
Authenticate Codex (non-interactive)     0s
Run Codex to fix CI failure              56s
Verify tests                             1s
Create pull request with fixes           5s
Post Setup Node.js                       4s
Post Checkout failing ref                0s
```

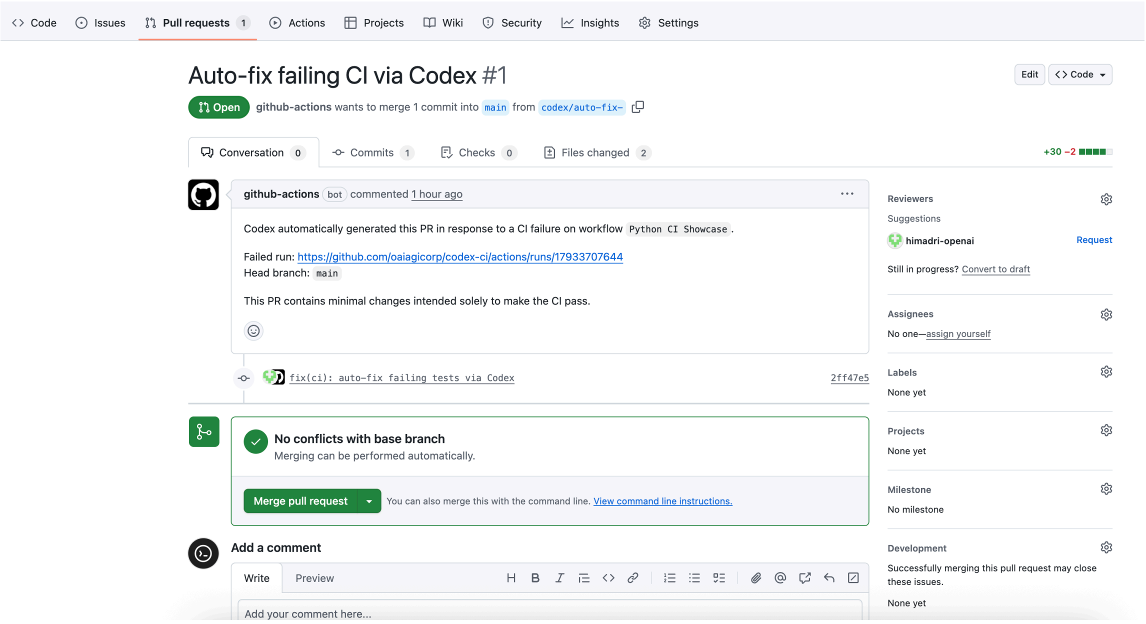
```
Post job cleanup.
/usr/bin/git version
git version 2.51.0
Copying '/home/runner/.gitconfig' to '/home/runner/work/_temp/7469bc93-b9e8-4397-a435-5b36f8a3ab42/.gitconfig'
Temporarily overriding HOME='/home/runner/work/_temp/7469bc93-b9e8-4397-a435-5b36f8a3ab42' before making global git config changes
Adding repository directory to the temporary git global config as a safe directory
/usr/bin/git config --global --add safe.directory /home/runner/work/codex-ci/codex-ci
/usr/bin/git config --local --name-only --get-regexp core\.sshCommand
/usr/bin/git submodule foreach --recursive sh -c "git config --local --name-only --get-regexp 'core\.sshCommand' && git config --local --unset-all 'core.sshCommand' || :"
/usr/bin/git config --local --name-only --get-regexp http\.https\:\/\/github\.com\/\.extraheader
http.https://github.com/.extraheader
/usr/bin/git config --local --unset-all http.https://github.com/.extraheader
/usr/bin/git submodule foreach --recursive sh -c "git config --local --name-only --get-regexp 'http\.https\:\/\/github\.com\/\.extraheader' && git config --local --unset-
all 'http.https://github.com/.extraheader' || :"
```

```
Complete job                             0s
```

```
Cleaning up orphan processes
```

Step 3: Verify that Codex Created a PR for Review

And after the Codex workflow completes execution, it should open a pull request from the feature branch codex/auto-fix. Check to see if everything looks good and then merge it.



Auto-fix failing CI via Codex #1

Open github-actions wants to merge 1 commit into `main` from `codex/auto-fix-`

Conversation 0 Commits 1 Checks 0 Files changed 2 +30 −2

github-actions bot commented 1 hour ago

Codex automatically generated this PR in response to a CI failure on workflow `Python CI Showcase`.

Failed run: https://github.com/oaiagicorp/codex-ci/actions/runs/17933707644
Head branch: `main`

This PR contains minimal changes intended solely to make the CI pass.

`fix(ci): auto-fix failing tests via Codex` `2ff47e5`

No conflicts with base branch
Merging can be performed automatically.

Merge pull request ▾ You can also merge this with the command line. View command line instructions.

Add a comment

Write Preview

Add your comment here...

Reviewers
Suggestions
himadri-openai Request
Still in progress? Convert to draft

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.
None yet

Conclusion

This automation seamlessly integrates OpenAI Codex CLI with GitHub Actions to automatically propose fixes for failing CI runs.

By leveraging Codex, you can reduce manual intervention, accelerate code reviews, and keep your main branch healthy. The workflow ensures that test failures are addressed quickly and efficiently, letting developers focus on higher-value tasks. Explore more about codex-cli and its capabilities here.