



Leveraging the GitHub API

@JasonEtco



Jason Etcovitch
@JasonEtc

Engineering Lead at GitHub

What we'll cover

The GitHub API

Authentication, common queries

GitHub Actions

What it is, how it works

Real JAMstack examples

Using GitHub as a “CMS”

Using Actions as a DB

This makes no sense, bear with me

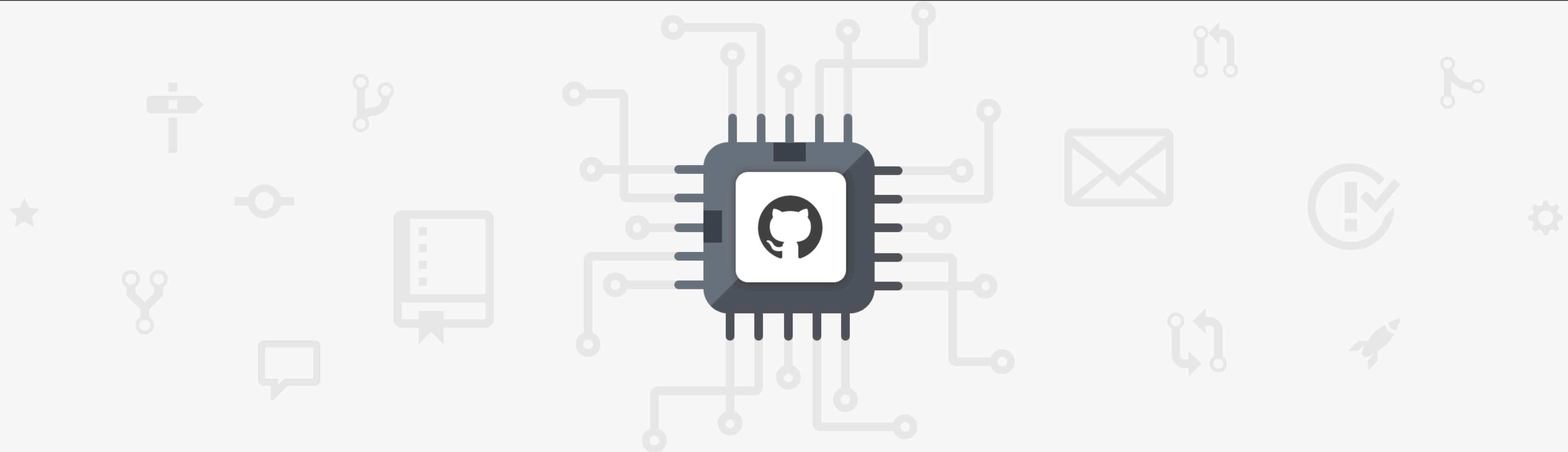
The GitHub API



GitHub Developer

Docs ▾ Blog Forum Versions ▾

Search...

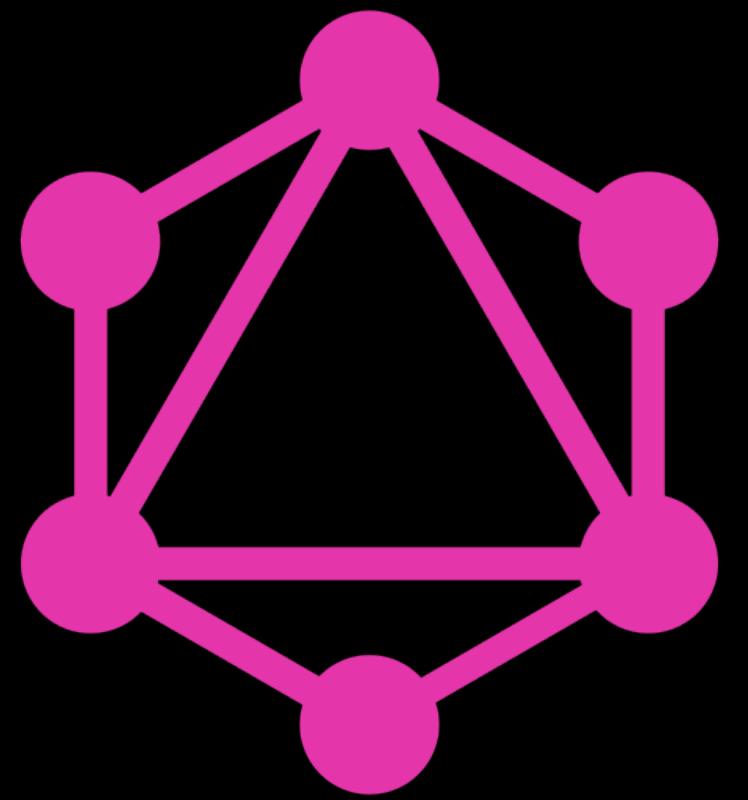


Build your app on GitHub's platform

Not sure where to start? We've put together some handy guides and reference documentation you can use to start building.



REST API



GraphQL API

```
fetch('https://api.github.com/zen')  
  .then(res => res.text())  
  .then(console.log)
```

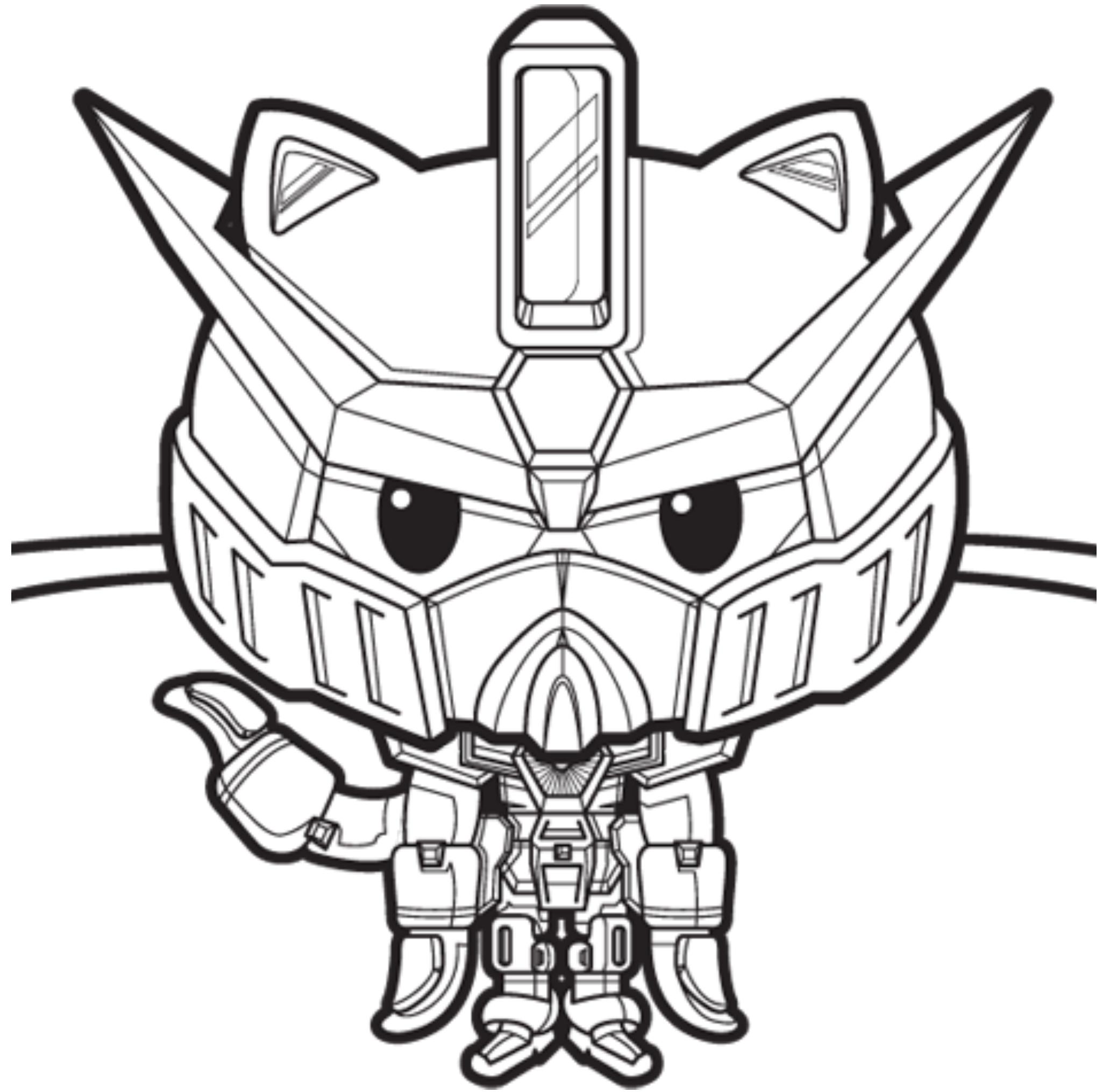
// -> Favor focus over features.

```
{  
  headers: {  
    Authorization: `token ${token}`  
  }  
}
```

Octokit

GitHub API SDKs in many languages

JavaScript: `@octokit/rest.js`



Real JAMstack Examples

Get a list of issues in a repository

GET /repos/:owner/:repo/issues

```
const owner = 'github'  
const repo = 'hub'  
  
fetch(`https://api.github.com/repos/${owner}/${repo}/issues`)  
.then(response => response.json())  
.then(console.log)
```

Read a file from a repository

GET /repos/:owner/:repo/contents/:path

```
const owner = 'github'  
const repo = 'hub'  
  
fetch(`https://api.github.com/repos/${owner}/${repo}/contents/README.md`)  
  // Parse the response as JSON  
  .then(response => response.json())  
  // Convert the base64 encoded text  
  .then(json => atob(json.content))  
  .then(console.log)
```

See recently active repositories

Code: git.io/fjjxr

jasonet.co



My personal site and blog

● JavaScript

★ 17

⌚ 3

```
{  
  viewer {  
    repositories (  
      ownerAffiliations: OWNER,  
      first: 2,  
      orderBy: { field: PUSHED_AT, direction: DESC },  
      isFork: false,  
      privacy: PUBLIC  
    ) {  
      nodes {  
        url  
        name  
        description  
        stargazers {  
          totalCount  
        }  
        forkCount  
        primaryLanguage {  
          color  
          name  
        }  
      }  
    }  
  }  
}
```

Bonus!

Get a user's profile picture



<https://github.com/hubot.png>

GitHub Actions



Search or jump to...



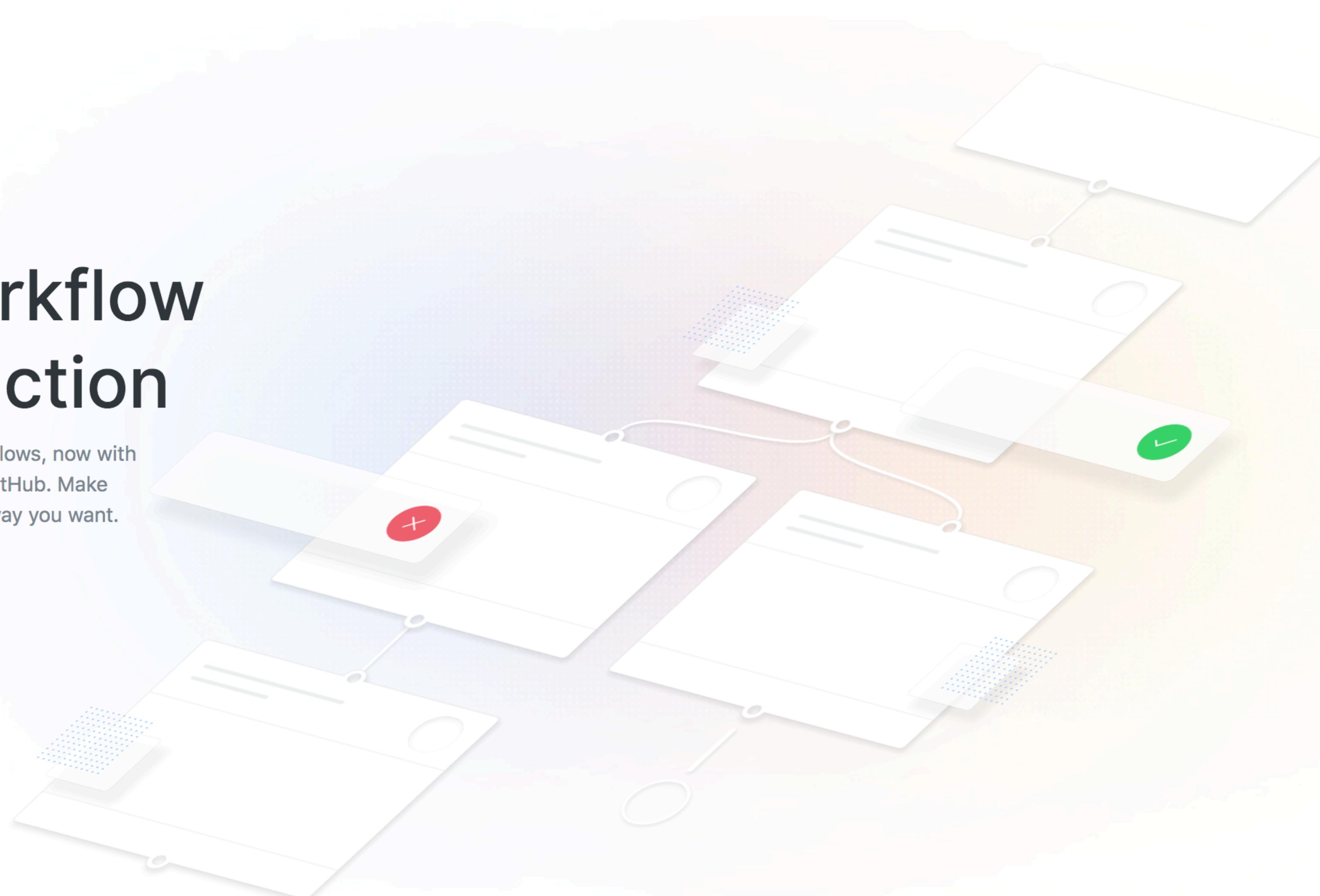
Pull requests Issues Marketplace Explore



Automate your workflow from idea to production

GitHub Actions makes it easy to automate all your software workflows, now with world-class CI/CD. Build, test, and deploy your code right from GitHub. Make code reviews, branch management, and issue triaging work the way you want.

[Sign up for the beta](#)



.github/workflows/ci.yml

Workflows are configured by adding a file to your repository.

When code is pushed, GitHub spins up a Ubuntu virtual machine.

It clones the repo, sets up Node, installs dependencies, and runs our tests.

```
name: Node.js CI

on: [push]

jobs:
  build:
    # Define the kind of VM
    runs-on: ubuntu-latest

    steps:
      # Tell the VM to clone the repo
      - uses: actions/checkout@v1
        # Setup Node.js at a specific version
      - name: Use Node.js 10
        uses: actions/setup-node@v1
        with:
          node-version: 10
        # Install dependencies and run tests
      - name: npm ci, and test
        run:
          npm ci
          npm test
```

On push...

The screenshot shows a GitHub Actions run log for a repository named "action-record". The run was triggered by a push event on the "master" branch. The log details the execution of a job named "build (10.x)" which includes steps for setting up the job, running actions/checkout@v1, using Node.js 10, and executing npm ci, build, and test. The logs show the command output for each step, including the compilation of index.js using ncc and the execution of Jest tests. The entire process completed successfully in 32 seconds.

```
16
17 ncc: Version 0.20.4
18 ncc: Compiling file index.js
19 ncc: Using typescript@3.5.3 (local user-provided)
20 0kB dist/action-record.d.ts
21 0kB dist/find-event-function.d.ts
22 0kB dist/index.d.ts
23 0kB dist/octokit.d.ts
24 0kB dist/register-models.d.ts
25 0kB dist/run-event.d.ts
26 1kB dist/instance.d.ts
27 1kB dist/model.d.ts
28 711kB dist/index.js
29 713kB [5826ms] - ncc 0.20.4
30
31 > @jasonetco/action-record@0.0.9 test /home/runner/work/action-record/action-record
32 > jest
33
34 PASS tests/model.test.ts (6.065s)
35 PASS tests/instance.test.ts
36
37 Test Suites: 2 passed, 2 total
38 Tests: 8 passed, 8 total
39 Snapshots: 3 passed, 3 total
40 Time: 6.575s
41 Ran all test suites.

42
43 ✓ Complete job
```



Search or jump to...

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)



[actions / starter-workflows](#)

[Watch](#) 25

[Star](#) 528

[Fork](#) 215

[Code](#)

[Issues 20](#)

[Pull requests 18](#)

[Actions](#)

[Wiki](#)

[Security](#)

[Insights](#)

Accelerating new GitHub Actions workflows <https://github.com/features/actions>

[actions](#)

100 commits

2 branches

0 packages

0 releases

26 contributors

MIT

Branch: master ▾

[New pull request](#)

[Create new file](#)

[Upload files](#)

[Find File](#)

[Clone or download](#) ▾



jclem Merge pull request #55 from tsloughter/erlang-rebar3 ...

Latest commit dcaa948 2 days ago



automation

Run daily not hourly

10 days ago



ci

Merge pull request #55 from tsloughter/erlang-rebar3

2 days ago



icons

Use the newest Gradle icon

18 days ago



CONTRIBUTING.md

add in contributing.md

last month



LICENSE

Update LICENSE

last month



README.md

Improve the readme

12 days ago



README.md



Recap

GitHub Actions is a *thing-doer*.

- Runs ephemeral “jobs”
- Free for open source (currently in beta)
- Can run any language, in a Windows/MacOS/Linux virtual machine
- Making API requests to GitHub is super easy from within Actions

GitHub Actions supports secrets, with an API token included.

```
steps:  
  - run: echo $GITHUB_TOKEN  
env:  
  GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
```

```
const { context, GitHub } = require('@actions/github')

// Create an authenticated instance of @octokit/rest
const octokit = new GitHub(process.env.GITHUB_TOKEN)

// Create a new issue
await octokit.issues.create({
  ...context.repo,
  title: 'Welcome!'
})
```

Using Actions as a DB

So... GitHub as a database?

Jason you're being silly.

An ORM is the SDK for your DB

3 acronyms in one sentence!

Why this is even possible

- GitHub data is permanent. Once you create a file, it's there forever (until you delete it).
- GitHub's API supports basically everything you'd need to do to read and write data.
- Authentication is built-in; users can have different permissions, like read-but-not-write to files.



README.md

Secret Edit Mode



ActionRecord

⚠ This is extremely WIP. Please don't use it or open issues just yet! ⚠

📄 An "ORM" for storing data in a GitHub repository using GitHub Actions

[Usage](#) • [FAQ](#)

Usage

ActionRecord works by running JavaScript functions in the repository to decide how and where to store the provided raw data.

Including it should be as simple as using the action in your `.github/workflows` file with the `GITHUB_TOKEN` secret:

```
steps:  
  - uses: JasonEtco/action-record  
    env:  
      GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
```

This will tell ActionRecord to run a JavaScript file called `/action-record/events/<EVENT>.js`, where `<EVENT>` is the event name that triggered the workflow.

Event handlers

These should be JavaScript files that export a function that takes one argument, an instance of the [ActionRecord class](#):

git.io/fjjp8

ActionRecord, the “GitHub Actions ORM”

An “ORM” for interacting with GitHub as a database inside of GitHub Actions.

It looks for special files in the **action-record**/directory in your repository.

```
// action-record/models/user.js
module.exports = ({ Joi }) => ({
  name: 'user',
  schema: {
    login: Joi.string()
  }
})

// action-record/events/push.js
module.exports = async action => {
  await action.models.user.create({
    login: action.context.payload.sender.login
  })
}
```


How it works

Storing data in GitHub

- When something happens on GitHub, we can run a job to do “database things”
- Data is stored as JSON in issues
- “Models” map to issue labels
- ActionRecord runs a script from your repository
- It makes API requests to GitHub under the hood

```
on: [push]

jobs:
  store-data:
    steps:
      - uses: JasonEtco/action-record@master
        env:
          GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
```

```
// action-record/events/push.js
module.exports = async ({ models, context }) => {
  const record = await models.user.create({
    login: context.payload.sender.login
  })

  console.log(record)
}
```

```
// action-record/models/user.js
module.exports = ({ Joi }) => ({
  name: 'user',
  schema: Joi.object({
    login: Joi.string()
  }),
  hooks: {
    async beforeCreate (candidateRecord) {
      const existingRecord = await this.findOne({ login: candidateRecord.login })
      if (existingRecord) {
        throw new Error(`User with ${existingRecord.login} already exists!`)
      }
    }
  }
})
```

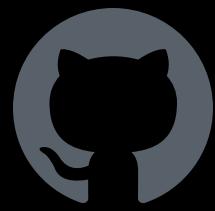
```
/**  
 * Call the search API to return all issues with this model's label.  
 */  
async searchForIssues (model) {  
  // Search for issues by this label  
  const issues = await octokit.search.issuesAndPullRequests({  
    q: `is:issue is:open repo:${process.env.GITHUB_REPOSITORY} label:${model.name}`  
  })  
  return issues.data.items  
}
```

How else can we tell GitHub to store data?

on: repository_dispatch

Use-cases

- Performance isn't important
- You want something free
- You like GitHub's UI
- You feel like doing something silly



Questions? Concerns? Comments?

Ping @JasonEtco on Twitter