



GitHub Actions

...in action!

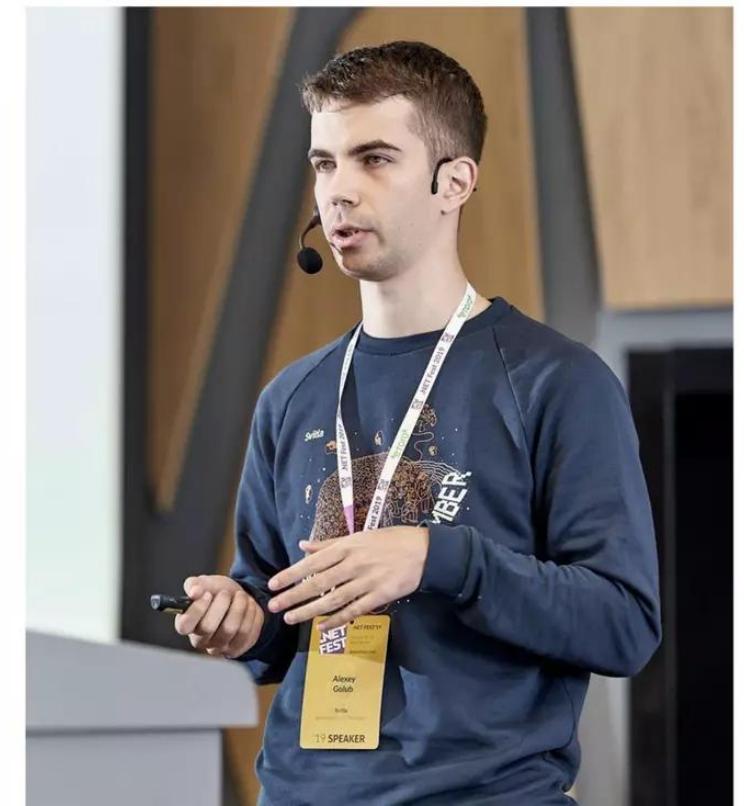


Speaker: Alexey Golub

  @Tyrrrz

/whois \${speaker}

- Open-source developer ⚡
- Conference speaker & blogger 🌐
- C#, F#, JavaScript 💻
- Cloud & web ☁
- Automation & DevOps 🛠



Speaker: Alexey Golub

  @Tyrrrz

GitHub Actions at a glance

- Globally available since November 13, 2019
- Based on Azure Pipelines
- Native integration with GitHub API
- YAML-based configuration
- Modular architecture & community-driven
- Runners on Windows, Linux, macOS, or self-hosted
- Available with Free, Pro, Team, Enterprise Cloud, One

Public repositories

Free! 

Private repositories

Product	Storage	Minutes (monthly)
GitHub Free	500 MB	2,000
GitHub Pro	1 GB	3,000
GitHub Team	2 GB	10,000
GitHub Enterprise Cloud	50 GB	50,000

Operating system	Minute multiplier	Operating system	Per-minute rate
Windows	2	Windows	\$0.016
Linux	1	Linux	\$0.008
macOS	10	macOS	\$0.080

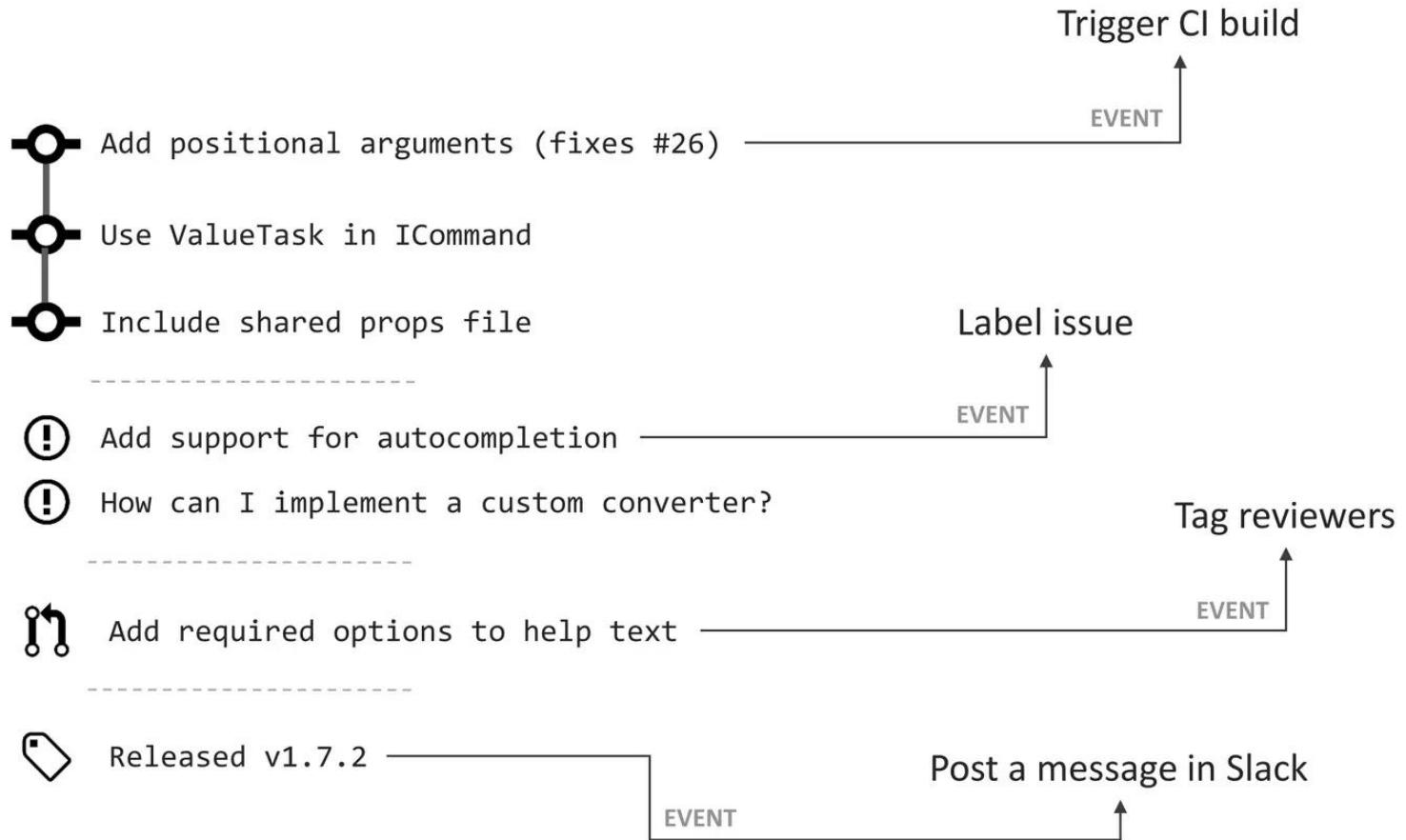
GitHub Actions

=

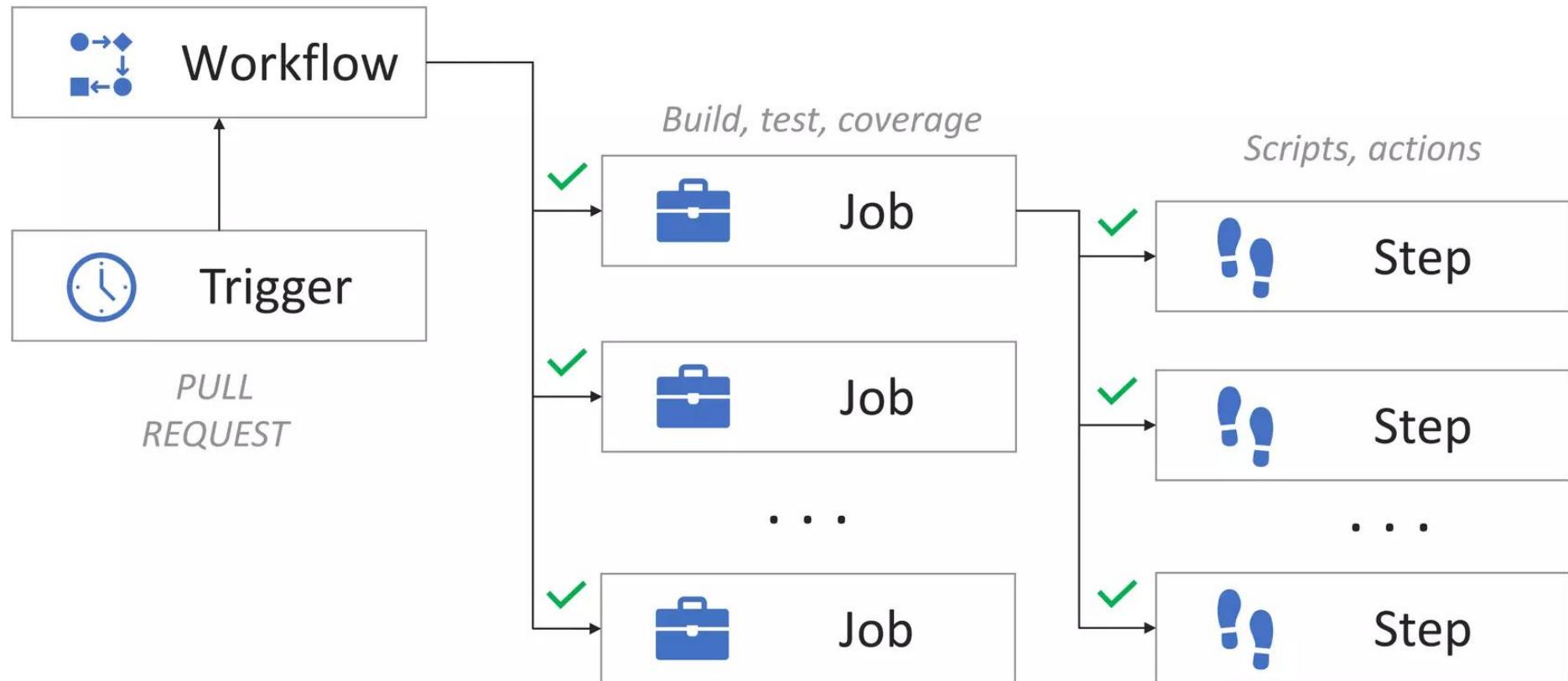
IFTTT for GitHub repositories

Speaker: Alexey Golub

  @Tyrrrz



Continuous integration



The screenshot shows the GitHub repository page for `Tyrrrz/CliFx`. The repository has 15 issues, 1 pull request, 0 actions, 0 projects, a wiki, security information, insights, and settings. It's a declarative framework for building command line interfaces. The repository has 147 commits, 1 branch, 0 packages, 8 releases, and 5 contributors. The latest commit was 6 days ago. The repository has a GitHub icon, a screenshots folder, a CliFx.Benchmarks folder, a CliFx.Demo folder, a CliFx.Tests folder, and a CliFx folder.

Octotree >

Tyrrrz / CliFx

Search or jump to... / Pull requests Issues Marketplace Explore

Sponsor Unwatch 13 Star 327 Fork 20

Code Issues 15 Pull requests 1 Actions Projects 0 Wiki Security Insights Settings

Declarative framework for building command line interfaces Edit

command-line cli net-standard net-core net-framework terminal Manage topics

147 commits 1 branch 0 packages 8 releases 5 contributors View license

Branch: master New pull request Create new file Upload files Find file Clone or download

dgarcia202 and Tyrrrz Add required options to the usage help text (#35) ✓ Latest commit 6344168 4 days ago

.github	Use ValueTask	last month
.screenshots	Move screenshots to repository	5 months ago
CliFx.Benchmarks	Use ValueTask	last month
CliFx.Demo	Add positional arguments (#32)	8 days ago
CliFx.Tests	Add required options to the usage help text (#35)	4 days ago
CliFx	Add required options to the usage help text (#35)	4 days ago

Speaker: Alexey Golub

@Tyrrrz

```
name: CI
on: [push, pull_request] ← Trigger on new commits and pull requests
jobs:
  build:
    runs-on: windows-latest
    steps:
      - name: Checkout
        uses: actions/checkout@v1 ← Clone repository and checkout HEAD
          (commit hash is passed as env var)
      - name: Install .NET Core
        uses: actions/setup-dotnet@v1
        with:
          dotnet-version: 3.1.100 ← Install .NET Core v3.1.100
      - name: Build & test
        run: dotnet test --configuration Release
      - name: Build & publish
        run: dotnet publish LightBulb/ -o LightBulb/bin/Publish/ --configuration Release
      - name: Upload build artifacts
        uses: actions/upload-artifact@master
        with:
          name: LightBulb
          path: LightBulb/bin/Publish/ ← Upload specified directory as
            a ZIP artifact
    - name: Run custom shell scripts
      run: echo "Hello, world!"
```

Workflows [New workflow](#)

[All workflows](#)

CD

CI

Octotree >

All workflows

Filter workflows

Event	Status	Branch	Actor	Timestamp	Duration	More
✓ CI on: push (416e253)	master	triggered by Tyrrrz	10 days ago	2m 36s	...	
✓ CI on: push (830dc1c)	master	triggered by Tyrrrz	12 days ago	3m 10s	...	
✓ CI on: push (83d8c3f)	master	triggered by Tyrrrz	12 days ago	3m 19s	...	
✓ CI on: push (89d70e9)	master	triggered by Tyrrrz	15 days ago	3m 17s	...	
✗ CI on: push (e53588d)	master	triggered by Tyrrrz	15 days ago	1m 53s	...	
✗ CI on: push (6877226)	master	triggered by Tyrrrz	15 days ago	1m 58s	...	
✓ CI on: push (695d29c)	master	triggered by Tyrrrz	18 days ago	3m 2s	...	
✓ CI on: push (2cb9e8c)	master	triggered by Tyrrrz	18 days ago	2m 58s	...	

Speaker: Alexey Golub

  @Tyrrrz

A screenshot of a GitHub pull request interface. At the top, a commit from **thorhj** is shown, adding two commits 18 days ago:

- Eliminated warnings related to nullable in CliFx project (commit `3eb170d`)
- Eliminated nullable warnings from CliFx.Demo and CliFx.Tests (commit `aaa75a3`)

Below this, a comment from **Tyrrrz** is displayed, posted 5 days ago:

Hi, thanks for the PR.
Can you please clarify the purpose of `ICommandSchema` in these changes?

Owner + ...

At the bottom of the pull request page, there is a message: "Add more commits by pushing to the `pr/nullable` branch on `thorhj/CliFx`".

The main content area shows the status of the pull request's CI build:

- All checks have passed** (1 successful check)
- CI / build (pull_request)** Successful in 35s

Below the CI status, a warning message is displayed: "This branch has conflicts that must be resolved". It provides instructions: "Use the web editor or the command line to resolve conflicts." and a "Resolve conflicts" button.

Under the "Conflicting files" section, it lists `CliFx/Services/HelpTextRenderer.cs`.

At the bottom, there is a "Squash and merge" button with a dropdown menu, and a note: "You can also [open this in GitHub Desktop](#) or view command line instructions."

Speaker: Alexey Golub

@Tyrrrz

```
67     }
68     return buffer.ToString();
69   }
70 }

72   public partial class CommandSchema
73   {
74     internal static CommandSchema StubDefaultCommand { get; } =
75       new CommandSchema(null, null, null, new CommandArgumentSchema[0], new CommandOptionSchema[0]);

⚠ Check warning on line 75 in CliFx/Models/CommandSchema.cs
GitHub Actions / build
CliFx/Models/CommandSchema.cs#L75
Cannot convert null literal to non-nullable reference type.

⚠ Check warning on line 75 in CliFx/Models/CommandSchema.cs
GitHub Actions / build
CliFx/Models/CommandSchema.cs#L75
Cannot convert null literal to non-nullable reference type.

⚠ Check warning on line 75 in CliFx/Models/CommandSchema.cs
GitHub Actions / build
CliFx/Models/CommandSchema.cs#L75
Cannot convert null literal to non-nullable reference type.

76 }
```

Speaker: Alexey Golub

  @Tyrrrz

Triggers

- GitHub API events 
push, pull_request, issues, release, and 20 others
- Schedule 
Cron syntax, e.g.: */15 * * * *
- Manual 
POST to /repos/:owner/:repo/dispatches

```
# Trigger on push events on specific branches
on:
  push:
    branches:
      - 'master'
      - 'release/*'
```

```
# Trigger on manual dispatch
on: repository_dispatch
```

```
# Trigger every midnight UTC
on:
  schedule:
    - cron: '0 0 * * *'
```

```
# Trigger when an issue is opened or labeled
on:
  issues:
    types: [opened, labeled]
```

Referencing actions

- By GitHub repository 

{owner}/{repo}@{ref}
{owner}/{repo}/{path}@{ref}

jessfraz/branch-cleanup-action@master
johndoe/my-actions/push-image@v1

- By file path 

./path/to/dir

./.github/actions/my-action

- By Docker image 

docker://{image}:{tag}

docker://hello-world:latest

Advanced configurations

Speaker: Alexey Golub

  @Tyrrrz

Matrices

```
name: Matrix Reloaded  
on: [push]
```

```
jobs:  
  build:  
    runs-on: ${{ matrix.os }}  
  
  strategy:  
    max-parallel: 4  
    matrix:  
      os: [ubuntu-16.04, ubuntu-18.04]  
      node-ver: [6, 8, 10]
```

```
steps:  
  - uses: actions/setup-node@v1  
    with:  
      node-version: ${{ matrix.node-ver }}
```

Ubuntu-16.04	Ubuntu-18.04
Node v6	Node v6
Node v8	Node v8
Node v10	Node v10

Reference variables from the matrix

Docker containers

```
jobs:  
  build:  
    runs-on: ubuntu-latest  
  
    services:  
      redis:  
        image: redis  
        ports:  
          - 6379/tcp  
        options: --entrypoint redis-server  
  
    steps:  
      - uses: actions/checkout@v1  
  
      - run: node client.js  
    env:  
      REDIS_HOST: localhost  
      REDIS_PORT: ${{ job.services.redis.ports[6379] }}  
  
Image from the Docker registry  
Bind port 6379 in container to a random port on host  
Custom arguments passed to docker create  
Exposed port is resolved dynamically
```

Secrets

steps:

```
# ...
```

```
- name: Collect coverage report
  run: |
    choco install codecov --no-progress
    codecov -f LtGt.Tests/bin/Release/Coverage.xml -t ${{secrets.CODECOV_TOKEN}}
```



Secret variable

Speaker: Alexey Golub

  @Tyrrrz

Code Issues 16 Pull requests 0 Actions Projects 0 Wiki Security Insights

Settings

Options

Collaborators

Branches

Webhooks

Notifications

Integrations & services

Deploy keys

Secrets

Actions

Moderation

Interaction limits

Secrets

Secrets are environment variables that are encrypted and only exposed to selected actions. Anyone with collaborator access to this repository can use these secrets in a workflow.

Secrets are not passed to workflows that are triggered by a pull request from a fork. [Learn more](#).

CODECOV_TOKEN

Remove

NUGET_TOKEN

Remove

Add a new secret



CD / build

succeeded 11 days ago in 1m 49s

▶ ✓ Set up job

▶ ✓ Checkout

Secrets are automatically obfuscated in logs

▶ ✓ Install .NET Core

▶ ✓ Pack

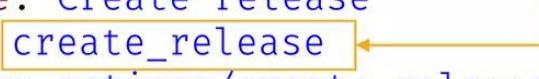
▼ ✓ Deploy

```
1  ► Run dotnet nuget push LtGt\bin\Release\*.nupkg -s nuget.org -k ***
6  info : Pushing LtGt.2.1.0.nupkg to 'https://www.nuget.org/api/v2/package'...
7  info : PUT https://www.nuget.org/api/v2/package/
8  info : Created https://www.nuget.org/api/v2/package/ 759ms
9  info : Your package was pushed.
10 info : Pushing LtGt.2.1.0.snupkg to 'https://www.nuget.org/api/v2/symbolpackage'...
11 info : PUT https://www.nuget.org/api/v2/symbolpackage/
12 info : Created https://www.nuget.org/api/v2/symbolpackage/ 242ms
13 info : Your package was pushed.
```

▶ ✓ Complete job

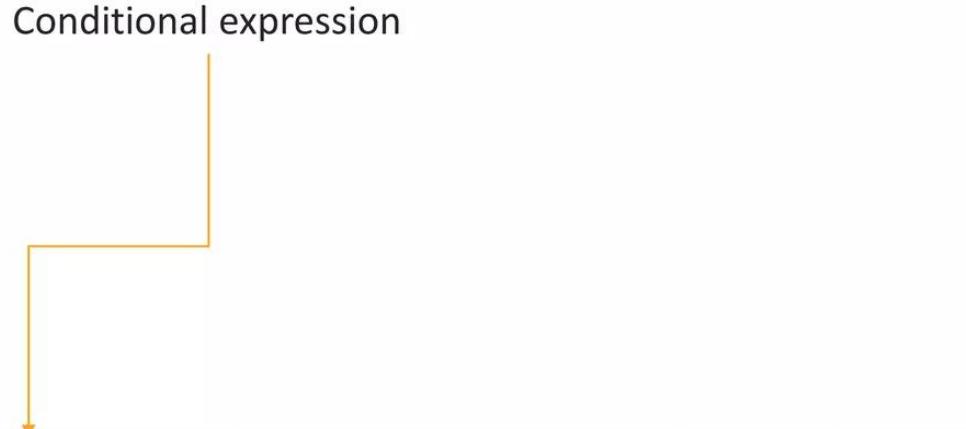


Action to action I/O

- name: Create release
 - id: `create_release`  Actions can be referenced by their ID
 - uses: `actions/create-release@v1`
 - env:
 - `GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}`
 - with:
 - `tag_name: ${{ github.ref }}`
 - `release_name: ${{ github.ref }}`
- name: Upload release asset
 - uses: `actions/upload-release-asset@v1.0.1`
 - env:
 - `GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}` Resolved from the outputs of another action
 - with:
 - `upload_url: ${{ steps.create_release.outputs.upload_url }}` 
 - `asset_path: DiscordChatExporter/bin/Publish/Archive.zip`
 - `asset_name: DiscordChatExporter.zip`

Conditionals

```
jobs:  
  build:  
    runs-on: ubuntu-18.04  
  
    steps:  
      - uses: actions/checkout@v1  
      - uses: actions/setup-dotnet@v1  
        with:  
          dotnet-version: 3.1.100  
  
      - run: dotnet test src  
      - run: dotnet pack src  
  
      - if: github.event_name == 'push' && startsWith(github.ref, 'refs/tags/v')  
        run: dotnet nuget push src/**.nupkg -k ${{secrets.NUGET_TOKEN}}
```



The diagram illustrates a conditional expression in a GitHub Actions workflow. A yellow bracket labeled "Conditional expression" points to the line of code that begins with "- if:". An orange arrow points from this bracket down to the specific line of code: "if: github.event_name == 'push' && startsWith(github.ref, 'refs/tags/v')".

Things we can do with GitHub Actions

- Run tests on every commit
- Publish Docker image when a tag is pushed
- Label an issue by content when it's created
- Run nightly E2E tests
- Automatically close stale issues every week
- Invite new contributors to sign the CLA when a PR is opened
- Automatically format code on push
- ...

Examples of actions you can use in your workflows

Speaker: Alexey Golub

  @Tyrrrz

actions/github-script

```
on:  
  issues: {types: opened}  
  
jobs:  
  comment:  
    runs-on: ubuntu-latest  
    steps:  
      - uses: actions/github-script@0.4.0  
        with:  
          github-token: ${{secrets.GITHUB_TOKEN}}  
          script: |  
            github.issues.createComment({  
              issue_number: context.issue.number,  
              owner: context.repo.owner,  
              repo: context.repo.repo,  
              body: '👋 Thanks for reporting!'  
            })
```

Runs inline JS code that uses the GitHub API

actions/stale

```
on:  
  schedule:  
    - cron: "0 0 * * *"  
  
jobs:  
  stale:  
    runs-on: ubuntu-latest  
    steps:  
      - uses: actions/stale@v1  
        with:  
          repo-token: ${{ secrets.GITHUB_TOKEN }}  
          stale-issue-message: 'Issue closed due to inactivity.'  
          stale-pr-message: 'PR closed due to inactivity.'  
          days-before-stale: 30  
          days-before-close: 5
```

Closes stale issues
and PRs

sonarsource/sonarcloud-github-action

```
on: push                                Sends code to
jobs:                                     SonarCloud to scan for
  sonarCloudTrigger:                      issues
    name: SonarCloud Trigger
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@master
      - name: SonarCloud Scan
        uses: sonarsource/sonarcloud-github-action@master
        env:
          GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
          SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
```

vsoch/pull-request-action

```
on: push

jobs:
  auto-pull-request:
    name: Create pull request for a new branch
    runs-on: ubuntu-latest
    steps:
      - name: pull-request-action
        uses: vsoch/pull-request-action@master
    env:
      GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
      BRANCH_PREFIX: "features/"
      PULL_REQUEST_BRANCH: "master"
```

Automatically creates a
pull request when
publishing a new
branch

Discovering actions

The screenshot shows the GitHub Marketplace search results for the term 'Actions'. The interface includes a search bar at the top, followed by sections for 'Types' (with a 'Search for apps and actions' input), 'Actions' (selected), and 'Categories' (including API management, Chat, Code quality, etc.). Below these are two columns of action cards. The first column contains 'Assignee to reviewer', 'ClearlyNoticed Action', 'Hugo Actions', 'Jest Snapshots', and 'Publish to Rubygems'. The second column contains 'Automatic Revert', 'GraphQL query', 'Jekyll Action', 'Node App Helper Actions', and 'Python Style Checker'. Each card includes a small icon, the action name, the author, a brief description, and the number of stars.

The screenshot shows the GitHub Marketplace page for the 'Actions for Discord' GitHub Action. It features a large header with the action's name, version (0.0.2), and a 'Use latest version' button. Below the header is a description: 'Sends a Discord notification message. Simple as that. Supports all workflow event types by using the Discord GitHub webhooks.' A diagram illustrates how the action integrates with GitHub workflows. To the right are sections for 'Stars' (67), 'Contributors', 'Categories' (Chat, Desktop tools), and 'Links' (Open issues, Pull requests, Report abuse). At the bottom, a note states: 'Actions for Discord is not certified by GitHub. It is provided by a third-party and is governed by separate terms of service, privacy policy, and support documentation.'

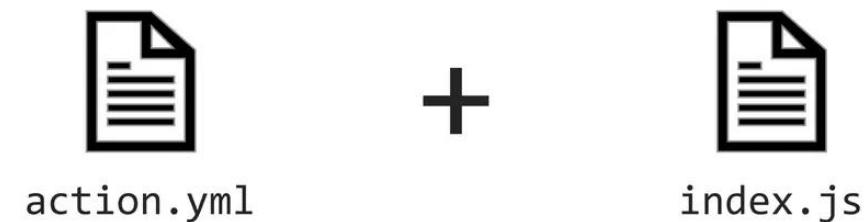
Speaker: Alexey Golub

@Tyrrrz

Can't find an action you
need? Make your own!

JavaScript-based actions

- Supported by all runners
- Can only use JavaScript
- Requires a single self-contained .js file as the entry point
- Node modules @actions/core & @actions/github



```
~/action.yml
```

```
name: 'Hello World'  
description: 'Greet someone and record time'  
inputs:  
  who-to-greet:  
    description: 'Who to greet'  
    required: true  
    default: 'World'  
outputs:  
  time:  
    description: 'The time we greeted you'  
runs:  
  using: 'node12'  
  main: 'index.js'
```

Metadata associated with an action

Inputs & outputs

Entry point

~/index.js

```
const core = require('@actions/core');
const github = require('@actions/github');

try {
    // `who-to-greet` input defined in action metadata file
    const nameToGreet = core.getInput('who-to-greet');
    console.log(`Hello ${nameToGreet}!`);
    const time = (new Date()).toTimeString();
    core.setOutput("time", time);

    // Get the JSON webhook payload for the event that triggered the workflow
    const payload = JSON.stringify(github.context.payload, undefined, 2)
    console.log(`The event payload: ${payload}`);
}

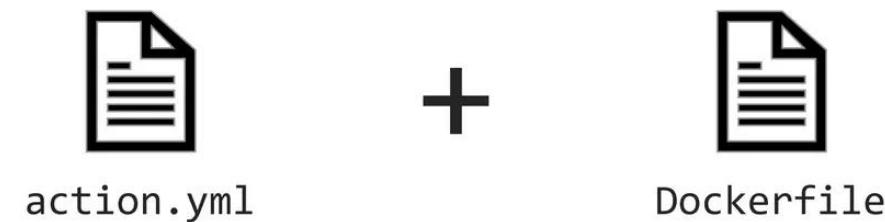
} catch (error) {
    core.setFailed(error.message);
}
```

Speaker: Alexey Golub

  @Tyrrrz

Docker-based actions

- Supported only by Linux-based runners (for now)
- Can use any language or runtime
- Typically runs slower



```
~/action.yml
```

```
name: 'Hello World'  
description: 'Greet someone and record time'  
inputs:  
  who-to-greet:  
    description: 'Who to greet'  
    required: true  
    default: 'World'  
outputs:  
  time:  
    description: 'The time we greeted you'  
runs:  
  using: 'docker'  
  image: 'Dockerfile'  
  args:  
    - ${{ inputs.who-to-greet }}
```

```
~/Dockerfile
```

```
FROM ubuntu:18.04  
COPY entrypoint.sh ./  
ENTRYPOINT ["/entrypoint.sh"]
```

```
~/entrypoint.sh
```

```
#!/bin/sh -l  
  
echo "Hello $1"  
time=$(date)  
echo ::set-output name=time::$time
```

Special commands

- ::set-output name=action_fruit::strawberry
- ::set-env name=action_state::yellow
- ::add-path::/path/to/dir
- ::debug file=app.js,line=1::Entered method
- ::warning file=app.js,line=1,col=5::Missing semicolon
- ::error file=app.js,line=10,col=15::Oops
- ::add-mask::Mona The Octocat

Publishing actions to marketplace

The screenshot shows a GitHub repository page for 'Tyrrrz / DevOpsFestAction'. The repository was generated from 'actions/javascript-action'. The page includes standard GitHub navigation links like Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation, there's a commit message 'Test test please ignore' with an 'Edit' button. A 'Manage topics' section is present. Key statistics are shown: 1 commit, 1 branch, 0 packages, 0 releases, 1 contributor, and MIT license. A prominent yellow-bordered button labeled 'Publish this Action to Marketplace' is displayed, with a sub-instruction 'Make your Action discoverable on GitHub Marketplace and in GitHub search.' To the right of this button are 'Draft a release' and a close ('X') button. At the bottom, there are buttons for Branch: master, New pull request, Create new file, Upload files, Find file, and Clone or download. A commit history shows 'Tyrrrz Initial commit' and 'Latest commit 17595aa 1 minute ago'.

Speaker: Alexey Golub

@Tyrrrz

The screenshot shows a GitHub Marketplace page for a GitHub Action named "DevOpsFestAction". The page includes a yellow circular icon with a play button, the action's name, its version (v1), and a "Latest version" button. It also displays a green badge indicating "units-test passing". Below this, there's a section titled "Create a JavaScript Action" with instructions and links to a template and a "Hello World JavaScript Action". A "Code in Master" section provides details on dependencies and a command-line interface (\$ npm install). On the right side, there are sections for "Stars" (0 stars), "Contributors" (one contributor), "Categories" (Utilities), and "Links" (repository, issues, pull requests, report abuse). A note at the bottom states that the action is not certified by GitHub.

Marketplace / Actions / DevOpsFestAction

GitHub Action

DevOpsFestAction

v1 Latest version

units-test passing

Create a JavaScript Action

Use this template to bootstrap the creation of a JavaScript action.

This template includes tests, linting, a validation workflow, publishing, and versioning guidance.

If you are new, there's also a simpler introduction. See the [Hello World JavaScript Action](#)

Create an action from this template

Click the [Use this Template](#) and provide the new repo details for your action

Code in Master

Install the dependencies

```
$ npm install
```

Run the tests

Stars 0

Contributors

Categories Utilities

Links

- Tyrrrz/DevOpsFestAction
- Open issues 0
- Pull requests 0
- Report abuse

DevOpsFestAction is not certified by GitHub.
It is provided by a third-party and is governed
by separate terms of service, privacy policy,
and support documentation.

Speaker: Alexey Golub

@Tyrrrz

Summary

- GitHub Actions is an automation platform (not just CI/CD)
- Can trigger workflows on various events
- Workflows are based on actions which are sourced by the community
- Free for all public repos, pay-as-you-go for private repos
- Easy to set up and configure to your needs

For the curious

- Awesome Actions by Sarah Drasner
<https://github.com/sdras/awesome-actions>
- GitHub Actions Advent Calendar by Edward Thomson
[https://edwardthomson.com/blog/github actions advent calendar.html](https://edwardthomson.com/blog/github_actions_advent_calendar.html)
- Comprehensive Introduction to GitHub Actions by Tierney Cyren
<https://dev.to/bnb/an-unintentionally-comprehensive-introduction-to-github-actions-ci-blm>
- Official documentation
<https://help.github.com/en/actions>

Thank you!

<https://github.com/Tyrrrz>

<https://twitter.com/Tyrrrz>

<https://tyrrrz.me>