

# GitHub ActionsのActionを作る

Mobile Act OSAKA #11

# 自己紹介

- Twitter: @penguin\_sharp
- GitHub: MeilCli
- Blog: <https://blog.meilcli.net/>
- Skill: C#, Kotlin, Android, Azure Pipelines
- Work: Fenrir Inc.
  - Android Application Engineer
  - 発言は個人の見解であり所属する組織の公式見解ではありません

# GitHub Actions軽くまとめ

- ベータ版
  - <https://twitter.com/natfriedman/status/1174158823877042177>
- Azure Pipelinesのクローン
- Windows, Linux, macOSで動作
- 1つの構成単位としてworkflow、1つの実行ステップ単位としてactionがある

# Actionを作るには

- 作り方は2種類
  - Docker container
  - JavaScript(Node.js) ← 今回試すのはこっち
- Docker containerはLinuxでのみ対応

注意:

今回作るものはあくまで学習目的です

そして、Dangerのactionを作ります

# そもそもDangerとは

- コードレビューを自動化するためのツールみたいなもの
  - <https://github.com/danger/danger>
- 今回はRuby版を使用

# Actionを作る

- Node.jsをインストール
- Visual Studio Codeを用意
- リポジトリの作成
  - `npm init -y`
  - `npm install -g typescript`
  - `tsc -init`
  - `npm install @actions/core --save`
  - `npm install @actions/io --save`
  - `npm install @actions/exec --save`
  - etc..

# Actionを作る

まずはactionの実行コードを書いていく

```
import * as core from '@actions/core';

async function run() {
  try {

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

run();
```



# Actionを作る

action外でDangerを動かすための環境がセットアップされている前提にする

```
import * as io from '@actions/io';

async function checkEnvironment() {
  await io.which("ruby", true);
  await io.which("bundle", true);
}
```

# Actionを作る

actionに渡された引数を取得する

```
interface Option {  
  readonly dangerVersion: string;  
  readonly pluginsFile: string | null;  
  readonly dangerFile: string;  
  readonly dangerId: string;  
}  
  
async function getOption(): Promise<Option> {  
  return {  
    dangerVersion: core.getInput( "danger_version", { required: true  
  } ),  
    pluginsFile: core.getInput( "plugins_file" ),  
    dangerFile: core.getInput( "danger_file", { required: true } ),  
    dangerId: core.getInput( "danger_id", { required: true } )  
  }  
}
```

# Actionを作る

## Dangerをインストール

```
import * as exec from '@actions/exec';

async function installDanger(option: Option) {
    await exec.exec(
        `gem install danger --version
        "${option.dangerVersion}"`,
        undefined,
        { failOnStdErr: true }
    );
    if (option.pluginsFile != null) {
        await exec.exec(
            `bundle install --gemfile= ${option.pluginsFile}`,
            undefined,
            { failOnStdErr: true }
        );
    }
}
```

# Actionを作る

## Danger実行

```
async function runDanger(option: Option) {  
    await exec.exec(  
        `danger  
--dangerfile=${option.dangerFile}  
--danger_id=${option.dangerId}`,  
        undefined,  
        { failOnStdErr: true }  
    );  
}
```

# Actionを作る

action.ymlを書く

```
name: 'Danger action'
description: 'run danger'
author: 'MeilCli'
inputs:
  danger_version:
    description: 'danger version'
    default: '~> 6.0'
  plugins_file:
    description: 'gemfile for danger plugin'
  danger_file:
    description: 'dangerfile for danger'
    required: true
  danger_id:
    description: 'danger id, set identifier string'
    required: true
runs:
  using: 'node12'
  main: 'lib/main.js'
```

# Actionを使う

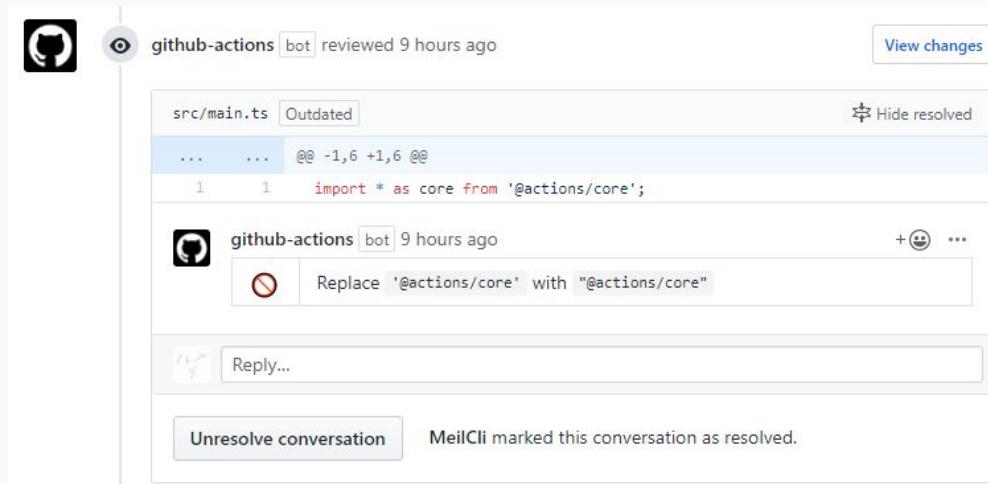
## workflowのymlを書く

- action.ymlのinputsに対してwithで値を渡す
- usesにアカウント名 /レポジトリ名を指定
  - @のあとはバージョン情報
  - ここではTagを指定

```
jobs:
  danger:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v1
      - uses: actions/setup-ruby@v1
        with:
          ruby-version: '2.6'
      - uses: MeilCli/danger-action@v1
        with:
          plugins_file: 'Gemfile'
          danger_file: 'Dangerfile'
          danger_id: 'danger-pr'
    env:
      DANGER_GITHUB_API_TOKEN: ${ secrets.GITHUB_TOKEN }
```

# Actionを使う

WorkflowやDangerfileなどをセットアップしておく  
と、PullRequestでこのようなコメントが送られてくる  
ようになる



The screenshot shows a GitHub interface with a comment from the `github-actions` bot. The comment is titled "reviewed 9 hours ago" and includes a "View changes" link. The code snippet shown is from `src/main.ts` and is marked as "Outdated". The code is:

```
... @@ -1,6 +1,6 @@  
1 1  import * as core from '@actions/core';
```

Below the code, there is a suggestion to replace `@actions/core` with `"@actions/core"`. The suggestion is shown with a red circle and a slash icon, indicating it is a warning or error. The suggestion text is: "Replace '@actions/core' with '@actions/core'".

At the bottom of the comment, there is a "Reply..." input field and a button labeled "Unresolve conversation". A message at the bottom right states: "MeiliCli marked this conversation as resolved."

# Actionを公開する

- 基本的にリポジトリを公開しておくだけでOK
  - 注意点として.jsファイルとそれが参照する node\_modulesも公開しておく必要がある
- Marketplaceにも公開できる
  - <https://github.com/marketplace?type=actions>
  - Marketplaceに公開する場合は action.ymlに必要な情報を記述しておく必要あり
    - 名前、説明、アイコンなど



# おわりに

- まとめ
  - GitHub ActionのactionはTypeScriptで簡単に作ることができる
  - だからみんなもどんどん作ってってくれ～～ 🙏🙏🙏
- 今回作ったもの
  - <https://github.com/MeilCli/danger-action>
  - <https://github.com/marketplace/actions/danger-action>
- 解説
  - <https://blog.meilcli.net/2019/09/15/155447>