

NÍVEL INTERMEDIÁRIO



PROJETO 12

(CONTEÚDO DISPONÍVEL) {
GITHUB;
ACTION;
CUSTOMIZÁVEL;
(end);
})();

#PORTFÓLIOBOOSTPROGRAM

CONHECIMENTOS REQUIRIDOS:



BACK - END

GITHUB ACTION CUSTOMIZÁVEL

Que tal automatizar um pouco os seus processos de envio de código? Vamos criar **actions** customizadas para o **GitHub** nesse projeto.

TECH STACK

→ Github

LIBRARIES

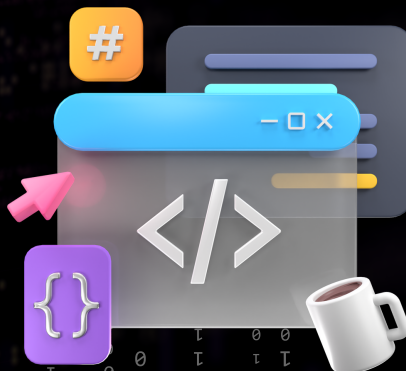
→ slack API



BRIEFING

O **GitHub Actions** é uma ferramenta poderosa que permite **automatizar** tarefas de código e realmente qualquer coisa de forma **agendada** ou **acionada**.

Um ótimo uso é enviar **notificações** para o **Slack** sempre que uma nova **solicitação pull** for feita.



NÍVEL 1

Um dos **hooks** que podemos usar com o **GitHub Actions** é a capacidade de executar algum código quando as **pull requests** são criadas.

Também podemos obter informações **contextuais** sobre essa **pull request** para obter detalhes como quem criou a solicitação.

Crie uma ação do **GitHub** que **registre informações** sobre uma **solicitação pull** quando ela é criada.

NÍVEL 2

O **registro de informações** no console do **GitHub Action** é útil para garantir que funcione, mas não é útil na prática ao tentar **descobrir** essas informações conforme elas acontecem.

Configure a ação do **GitHub** para **postar** a mensagem no **Slack** com um link para a **pull request**.

NÍVEL 3

Para tornar uma ação do **GitHub** útil entre diferentes projetos, ela ajuda a fornecer a capacidade de passar **opções de configuração** para o fluxo de trabalho.

Adicione opções de configuração de **entrada** para que outros projetos possam usar a ação com sua própria **instância Slack** e **git**.

REQUISITOS DETALHADOS



Crie um novo arquivo chamado `action.yml` na pasta `.github/workflows` do seu repositório [GitHub](#).



No arquivo `action.yml`, adicione o seguinte código:

```
name: Notify Slack Channel on Pull Request
Updates

on:
  pull_request:
    types:
      - opened
      - closed
      - approved
      - rejected

jobs:
  notify-slack:
    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v2

      - name: Install dependencies
        run: |
          echo "Installing dependencies..."
          npm install

      - name: Send Slack notification
        run: |
          echo "Sending Slack notification..."
          node notify.js
```




Arquivos **YML** são muito conhecidos entre os devops e desenvolvedores que criam arquivos de **configuração**, caso você não saiba o que significa. Veja mais em:

REDHAT



O código acima define um **workflow** chamado **Notify Slack Channel on Pull Request Updates**.



O **workflow** é executado quando um **pull request** é **aberto**, **fechado**, **aprovado** ou **rejeitado**. O workflow executa duas etapas:



A primeira etapa faz o **checkout** do seu **repositório GitHub** e instala as **dependências** do projeto.



A segunda etapa envia uma **notificação** para o canal do **Slack**.



Para enviar a notificação para o canal do Slack, você precisará criar um arquivo chamado **notify.js** na mesma pasta que **action.yml**.



Esse **notify** será um arquivo que irá conter informações como **channelId**, **message**, **title**, **author** e de fato um **client.chat.postMessage** para enviar a mensagem via **slack**.



A função **notify** recebe dois argumentos: o **token do Slack** e o **ID do pull request**. A função **notify** então usa o **token do Slack** para enviar uma mensagem para o **canal do Slack**.



A mensagem contém o **ID do pull request**, o **título do pull request**, o **autor do pull request** e o **status do pull request**.

➡ Para **configurar** o **GitHub Action**, você precisará adicionar as seguintes variáveis de ambiente ao seu repositório **GitHub**:

➡ **SLACK_API_TOKEN**: O token do seu canal do Slack.

➡ **GITHUB_PR_ID**: O ID do seu pull request.

➡ **GITHUB_PR_TITLE**: O título do seu pull request.

➡ **GITHUB_PR_AUTHOR**: O autor do seu pull request.

➡ Depois de configurar o **GitHub Action**, você pode ativar o **workflow** executando o seguinte comando:

```
gh workflow run notify-slack
```

➡ O **workflow** será executado e enviará uma notificação para o canal do Slack.

