Développement front avancé

MMI 3 – TP#4 S6

IUT CERGY-PONTOISE

Danielo **JEAN-LOUIS**

Au début du déploiement de sites web

- Planification du déploiement
  - HTML, CSS, PHP, dépendances…
- Upload de fichiers divers : traductions...
- Action effectuée par un être humain
  - Gros risque d’erreurs / oublis

Avec la complexité des projets, chaque mise en production est une tâche fastidieuse, risquée et longue. Elle peut faire perdre beaucoup d’argent à des entreprises en cas de mauvaise manipulation

Intégration continue / Livraison continue

- Appelé communément CI/CD
  - **C**ontinuous **I**ntegration/**C**ontinuous **D**elivery ou **D**eployment
- Automatisation de tâches sur un serveur :
  - Compilation, déploiement, tests unitaires…
- Facilite le déploiement de projets

La CI/CD consiste à créer une chaîne de commandes du développement au déploiement

Intégration continue / Livraison continue

- Prévient les bugs en production et lors du déploiement
- Entre dans la logique de SCRUM : livraison régulière d’une nouvelle itération

Intégration continue / Livraison continue

- S’articule souvent autour d’un VCS (Version Control System)
- Existe dans toute typologie de projet : site web, application mobile...

Intégration continue / Livraison continue

- Géré par un(e) DevOps
  - Métier combinant le développement (dev) et l’administration système (ops)
  - Profil très recherché
  - Connaît le terminal et les commandes linux de base : git, cd, touch, ssh...

Intégration continue / Livraison continue
Grandes étapes

1) Compilation
2) Test : qualité, unitaires, e2e, sécurité...
3) Déploiement

Source(s) :
- https://about.gitlab.com/fr-fr/topics/ci-cd/cicd-pipeline/

Intégration Continue (CI)

- Vérifie le code à chaque modification du code source. Ex : quand on effectue une pull request
- Permet de détecter les problèmes en amont

Source(s) :

- https://developer.mozilla.org/fr/docs/Web/HTML/Element/script/type/importmap

Livraison / Déploiement Continue (CD)

- Gère les environnements intermédiaires :
  - Stage, preprod… (delivery)
- Déploie sur le serveur de production (deployment)
- Création de versions
  - Permet un rollback prompt en cas de problème

Source(s) :

- https://github.com/WICG/import-maps?tab=readme-ov-file#installation

git

- VCS le plus populaire
- Présent par défaut sous linux et macOS
- Pierre angulaire du CI/CD

gitignore

- Fichier permettant d’exclure des fichiers du versionning
- Permet d’alléger les dépôts
- Pensez toujours à en mettre un dans vos projets

Source(s) :

- https://github.com/github/gitignore

Github Actions

- Repose sur un système de tâches appelée “Actions”
- Solution freemium permettant la CI/CD sur n’importe quel dépôt sur github
  - Automatisation de tâches

Source(s) :
- https://docs.github.com/fr/actions
- https://github.com/actions

Github Actions - Tarification

Plan	Stockage	Minutes (par mois)
GitHub Free	500 Mo	2 000
GitHub Pro	1 Go	3 000
GitHub Free pour les organisations	500 Mo	2 000
GitHub Team	2 Go	3 000
GitHub Enterprise Cloud	50 Go	50 000

Voici les accès par défaut, passé ces limites, vous serez facturé(e) à la minute (temps d'exécution d'une tâche)

Source(s) :

- https://docs.github.com/fr/billing/managing-billing-for-your-products/managing-billing-for-github-actions/about-billing-for-github-actions

Github Actions

- Vous pouvez créer vos propres actions
  - Ou télécharger ceux de la communauté : Attention tout de même à la qualité
  - Github propose des Actions clés en main

Source(s) :
- https://docs.github.com/fr/actions
- https://github.com/actions

Github Actions

- Fonctionne avec des conteneurs Docker
- Envoie un e-mail, si échec
- Permet d’effectuer des cron (tabà
  - Tâches planifiées
- Gère des fichiers YAML placés dans le dossier “.github/workflows”

Source(s) :

- https://docs.github.com/fr/actions

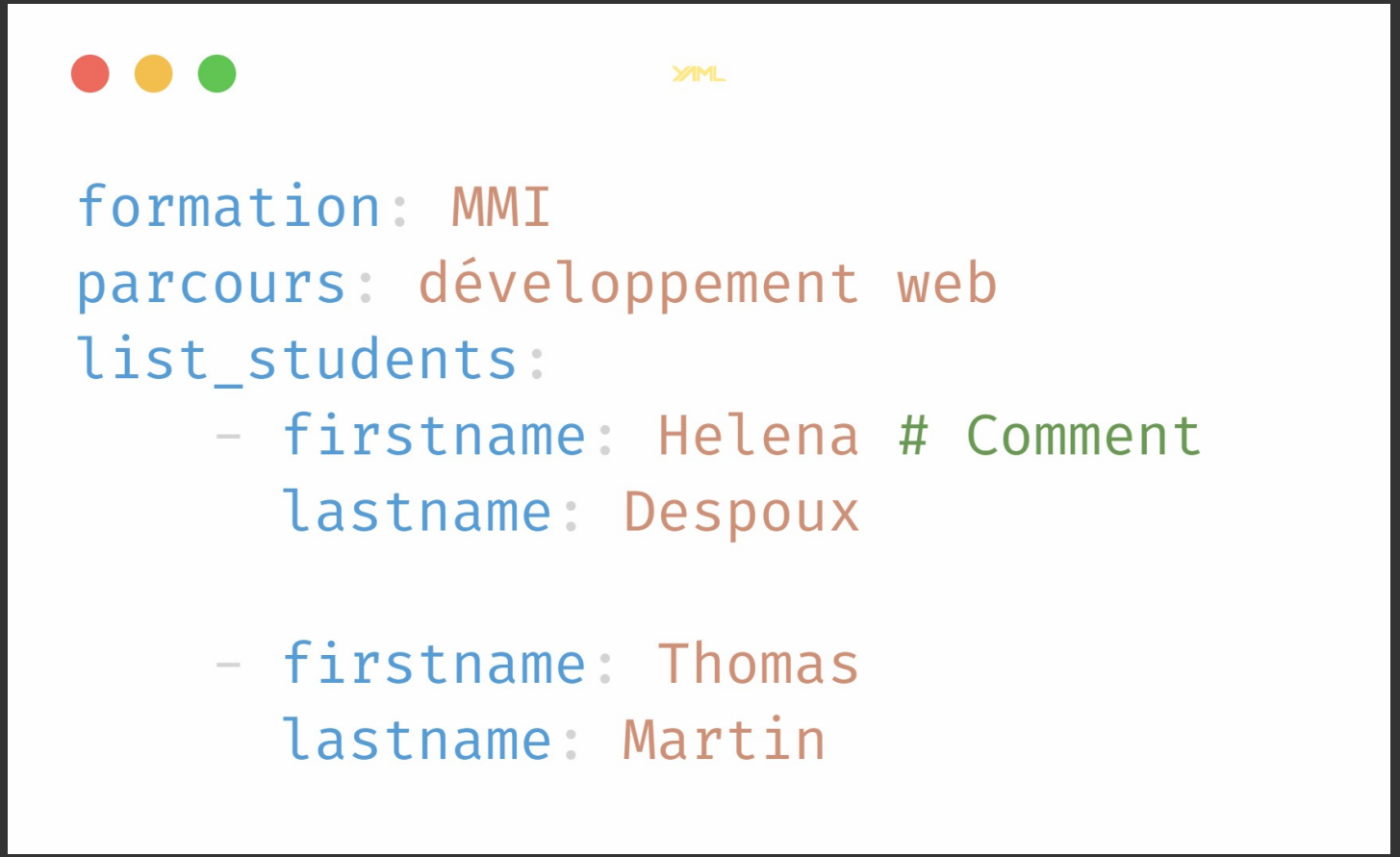
Fichier .yml / .yaml

- Format souvent utilisé pour la configuration
  - Utilisé notamment par Symfony
- Inspiré par le format CSV
  - YAML utilise des indentations pour générer une hiérarchie
- Permet la gestion de données complexes
  - Tout en gardant une lisibilité

Source(s) :

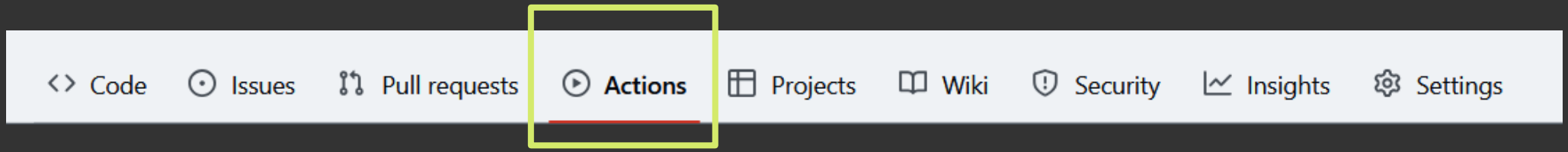
- https://fr.wikipedia.org/wiki/YAML

Fichier .yml / .yaml



Exemple de fichier YAML

Github Actions

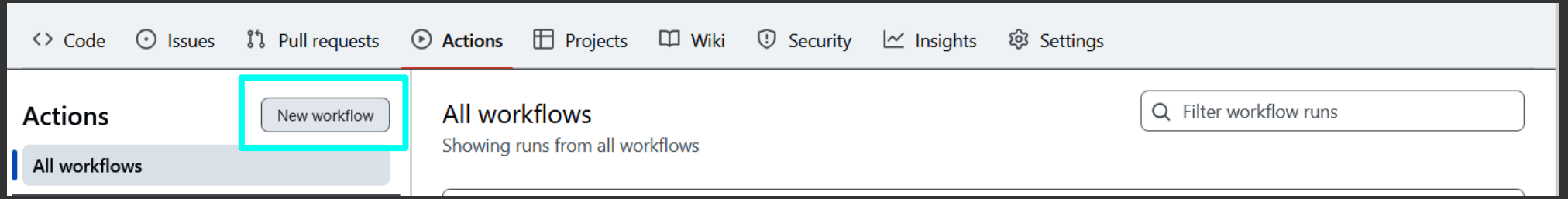


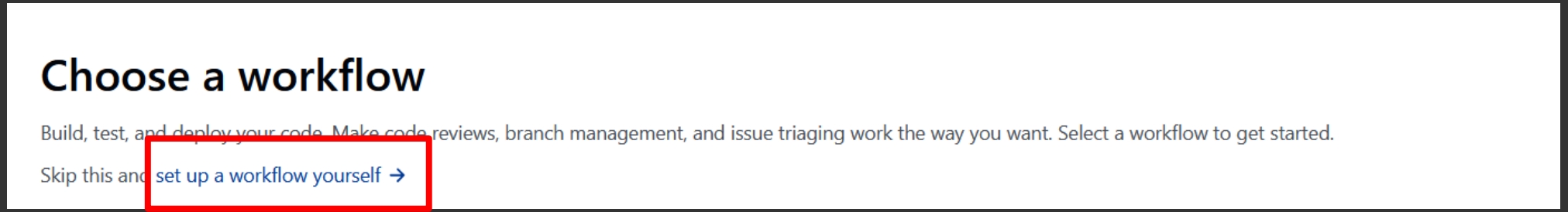
L’onglet “Actions” peut être désactivé

Source(s) :

- https://docs.github.com/fr/actions

Github Actions





Github permet d’éditer des Actions depuis le site web

Source(s) :

- https://docs.github.com/fr/actions

Pratiquons ! - Github actions (Partie 1)

Pré-requis :

- Avoir la ressource ressources/github-actions

A télécharger ici :
https://github.com/DanYellow/cours/raw/refs/heads/main/developpement-web-et-dispositif-interactif-s6/travaux-pratiques/numero-3/developpement-web-et-dispositif-interactif-s6_travaux-pratiques_numero-4.ressources.zip

Act

- Outil gratuit permettant de faire tourner vos Github Actions en local
- Nécessite Docker

Source(s) :
- https://github.com/nektos/act

Github Actions - Gabarit

- Chaque fichier d’actions doit contenir au moins trois clés racines :
  - name : Nom de la tâche
  - on : quand l’Action est exécutée ? (pull, push…) - Valeurs définies (voir source)
  - jobs : Tâches à effectuer

Source(s) :

- https://docs.github.com/fr/actions/writing-workflows/choosing-when-your-workflow-runs/events-that-trigger-workflows

Github Actions - Gabarit

```yaml
name: My first action
on: [my_event]
jobs:
  job_name:
    runs-on: volume_name
    steps:
      - name: step_name (optional)
      - run: command
```

Exemple de base d’un fichier d’actions. Pour “on”, le tableau n’est pas obligatoire.

Source(s) :

- https://docs.github.com/fr/actions/writing-workflows/choosing-when-your-workflow-runs/events-that-trigger-workflows

Github Actions - Gabarit

- Le même fichier peut contenir plusieurs jobs
  - Il est préférable de séparer votre pipeline en plusieurs jobs
    - Un job : une grande tâche

Source(s) :
- https://docs.github.com/fr/actions/writing-workflows/choosing-when-your-workflow-runs/events-that-trigger-workflows

Github Actions – Actions définies

- Ensemble d’actions déjà définies. Ex : Pull le dépôt
  - Développés par la communauté et Github
- S’utilise avec la clé “uses” (à la place de “run”) dans le fichier yml
  - Possibilité d’avoir plusieurs uses au sein du même job

Source(s) :

- https://docs.github.com/fr/actions/writing-workflows/choosing-when-your-workflow-runs/events-that-trigger-workflows
- https://github.com/sdras/awesome-actions?tab=readme-ov-file#official-actions

Github Actions – Actions définies

```yaml
name: Node Continuous Integration

on:
  push:
    branches: [ master ]

jobs:
  create_build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - uses: actions/setup-node@v1
        with:
          node-version: 20
      - name: Install dependencies
        run: npm ci
      - name: Create build
        run: npm run build
```

Source(s) :

- https://docs.github.com/fr/actions/writing-workflows/choosing-when-your-workflow-runs/events-that-trigger-workflows

Pratiquons ! - Github Actions (Partie 2)

Pré-requis :

- Avoir la ressource ressources/github-actions

A télécharger ici :
https://github.com/DanYellow/cours/raw/refs/heads/main/developpement-web-et-dispositif-interactif-s6/travaux-pratiques/numero-3/developpement-web-et-dispositif-interactif-s6_travaux-pratiques_numero-4.ressources.zip

Github Actions – Variables

- Permettent de réutiliser une valeur
- Plusieurs portée possibles :
  - Globale, job, tâche
- Préfixée par “$” pour être affichée

Github Actions – Variables

```yaml
name: Display a variable

on:
  workflow_dispatch

env:
  UNIVERSITY: CY Paris Université # Global scope

jobs:
  display_student_infos:
    runs-on: ubuntu-latest
    env:
      FORMATION: BUT MMI # Job scope
    steps:
      - name: "Presentation"
        run: echo "I'm $FIRST_NAME, I'm a student in $FORMATION at $UNIVERSITY"
        env:
          FIRST_NAME: John Doe # Step scope
```

Ici, nous avons trois variables avec trois portées différentes

Github Actions – Context

- Objets par défaut permettant d’accéder à diverses informations : état du job, nom de l’utilisateur courant…
- S’affiche "${{ <context> }}"

Source(s) :

- https://docs.github.com/en/actions/writing-workflows/choosing-what-your-workflow-does/accessing-contextual-information-about-workflow-runs

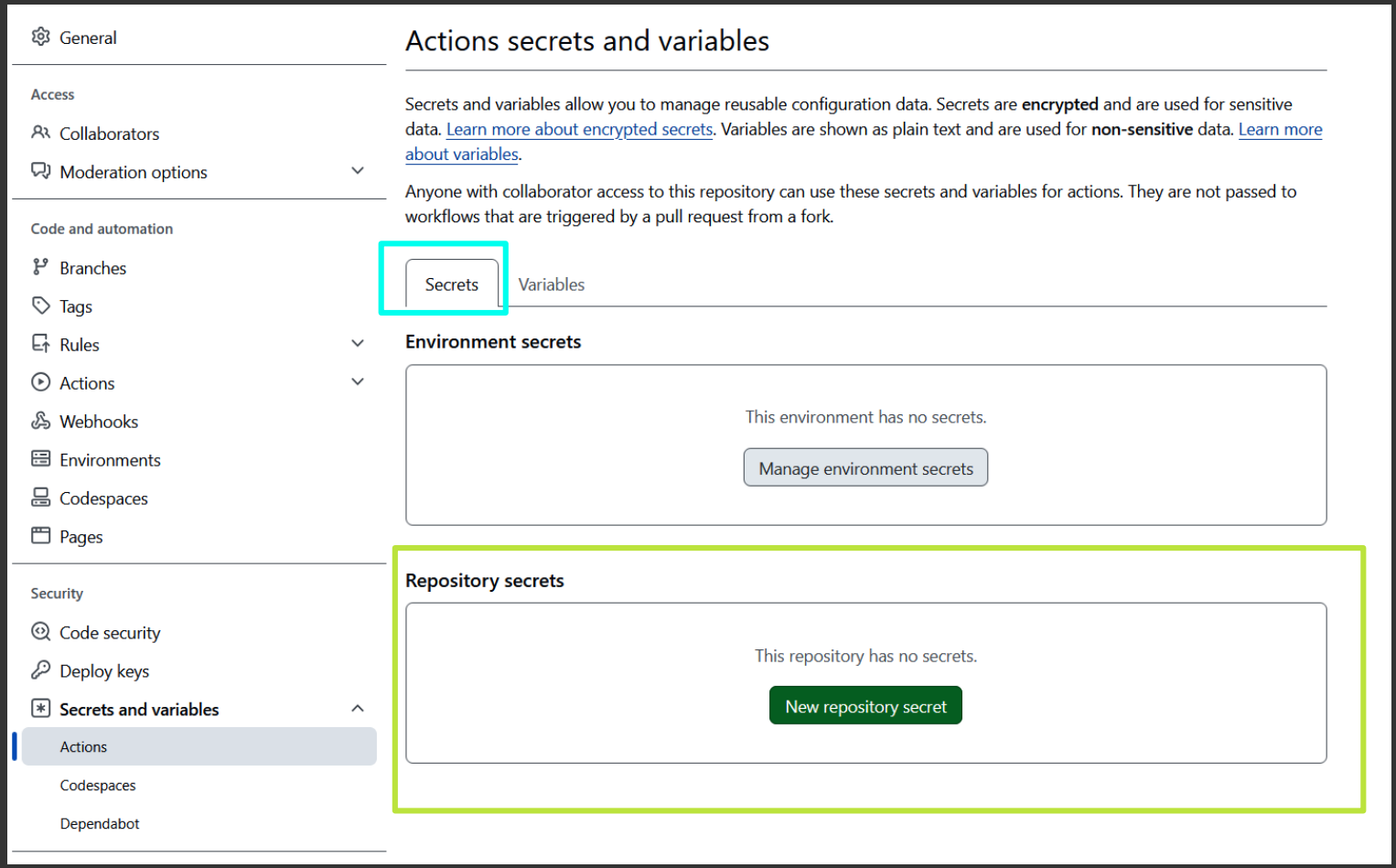
Github Actions – Secrets

- Définissent les variables d’environnement
  - Variables qui ne doivent pas être publiques…
  - ...mais qu’on veut utiliser dans ses Actions
- Données chiffrées
- Ne doivent pas commencer par un nombre
  - Caractères alphanumériques et underscore seulement

Github Actions – Secrets

- Valeurs non sensible à la casse
- Chargés depuis les paramètres du dépôt
  - Settings > Secrets and variables > Actions
- S’affichent comme les variables de contexte
  - ${{ <secrets.SECRET_KEY> }}

Github Actions – Secrets



Github Actions – Artifact

- Représente un dossier **persistant** sur le serveur
  - Souvent un build. Ex : npm run build
  - Durée de vie par défaut : 90 jours
- Peut être lu par d’autres jobs
  - Ex : Tâche de déploiement

Source(s) :

- https://docs.github.com/fr/actions/writing-workflows/choosing-what-your-workflow-does/storing-and-sharing-data-from-a-workflow

Github Actions – Artifact

- Nécessite l’action “action/upload-artifact@master” pour être partagé

```yaml
steps:
  - name: Generate artifact
    uses: actions/upload-artifact@master
    with:
      name: bundle # artifact / directory name on the server
      path: ./dist # source directory
```

On copie le contenu du dossier “dist” dans un artifact nommé “bundle”

Source(s) :

- https://docs.github.com/fr/actions/writing-workflows/choosing-what-your-workflow-does/storing-and-sharing-data-from-a-workflow

Github Actions – Inter-dépendances

- Permet d’attendre l’exécution d’un job avant l’exécution d’un autre
  - Multiple dépendances possibles
- Utilisation de la clé “needs”

Source(s) :

- https://docs.github.com/en/actions/writing-workflows/choosing-what-your-workflow-does/using-jobs-in-a-workflow#defining-prerequisite-jobs

Github Actions – Inter-dépendances

```yaml
deploy:
  runs-on: ubuntu-latest
  needs: [build]

  steps:
    - name: # ...
```

Notre job “deploy” ne peut s’exécuter que si et seulement si le job “build” est terminé

Source(s) :

- https://docs.github.com/en/actions/writing-workflows/choosing-what-your-workflow-does/using-jobs-in-a-workflow#defining-prerequisite-jobs

Pratiquons ! - Github Actions (Partie 3)

Pré-requis :

- Avoir la ressource ressources/github-actions

A télécharger ici :
https://github.com/DanYellow/cours/raw/refs/heads/main/developpement-web-et-dispositif-interactif-s6/travaux-pratiques/numero-3/developpement-web-et-dispositif-interactif-s6_travaux-pratiques_numero-4.ressources.zip

Github Actions – Inputs

- Permet de définir des valeurs depuis github qui seront utilisées dans votre workflow
  - Ex : définir le serveur de stage
- Plusieurs types de données possibles : choice (équivalent select en HTML), boolean, string et environnement

Source(s) :

- https://github.blog/changelog/2021-11-10-github-actions-input-types-for-manual-workflows/

Github Actions – Inputs

- Ne fonctionne qu’avec les workflows lancés manuellement
  - on : workflow_dispatch

Source(s) :
- https://github.blog/changelog/2021-11-10-github-actions-input-types-for-manual-workflows/

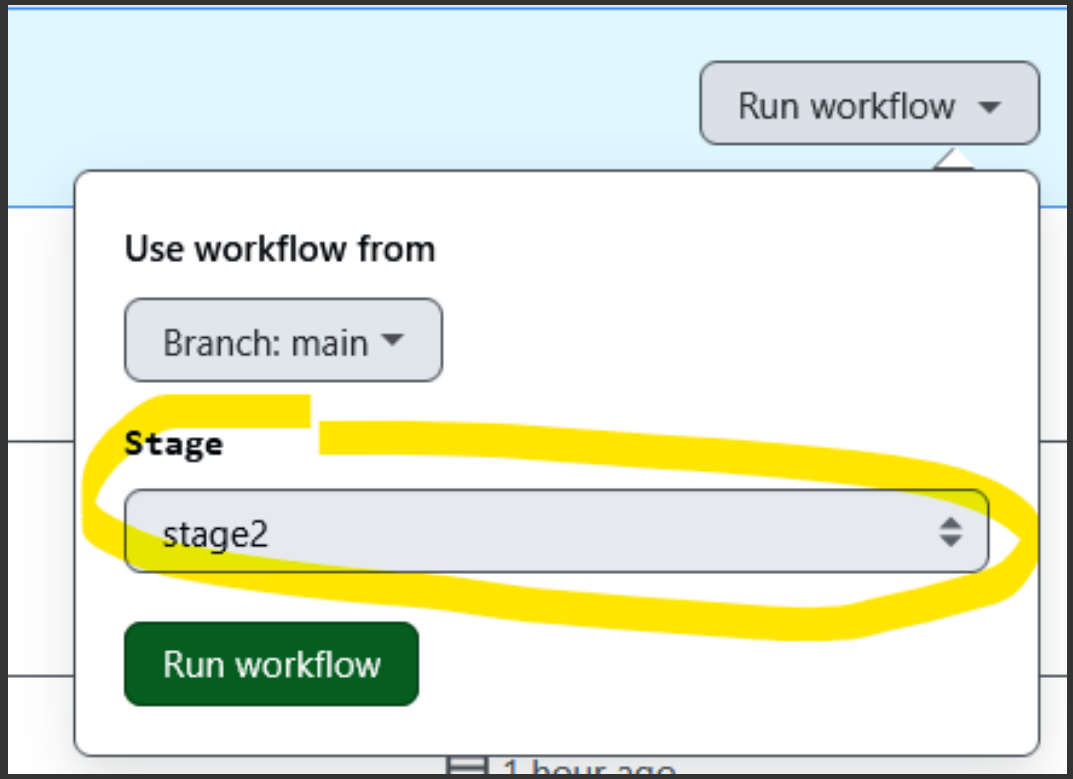
Github Actions – Inputs

```yaml
name: Github inputs

on:
  workflow_dispatch:
    stage:
      type: choice
      description: Select stage env
      default: stage2
      options:
      - stage1
      - stage2

jobs:
  my_input:
    # [ ... ]
    steps:
    - name: Use my input value
      run: echo "${{ github.event.inputs.stage }}"
```

Source(s) :

- https://github.blog/changelog/2021-11-10-github-actions-input-types-for-manual-workflows/

Github Actions – Inputs



Depuis l'interface des actions, je peux changer à la volée des variables de mon workflow

Source(s) :

- https://github.blog/changelog/2021-11-10-github-actions-input-types-for-manual-workflows/

Github Actions – Alternatives – Liste non exhaustive

- Travis CI
- Gitlab
- Circle CI
- Jenkins : Nécessite beaucoup de configuration
- TeamCity
- ...

Questions ?