

Développement front avancé

MMI 3 – TP#4 S6

Danielo **JEAN-LOUIS**

Au début du déploiement de sites web

- Planification du déploiement
 - HTML, CSS, PHP, dépendances...
- Action effectuée par un être humain
 - Gros risque d'erreurs / oublis

Avec la complexité des projets chaque mise en production peut être très complexe et 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

- Tourne souvent autour d'un VCS (Version control system)
- Existe dans plusieurs typologies 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
- 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

- Solution freemium permettant la CI/CD sur n'importe quel dépôt sur github
- Propose de nombreuses Actions
 - Vous pouvez créer les vôtres

Source(s) :

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

Github Actions

- Fonctionne avec des conteneurs Docker
- Envoie un e-mail après chaque exécution
- Gère des fichiers YAML placés dans le dossier “.github/workflows”

Source(s) :

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

Fichier .yaml / .yml

- Format souvent utilisé pour la configuration
- Utilisé notamment par Symfony
- Inspiré par le format CSV
- Permet la gestion de données complexes

Fichier .yaml / .yml



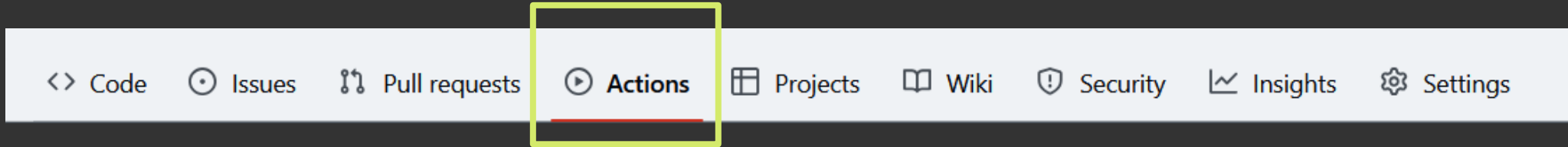
YML

```
formation: MMI
parcours: développement web
list_students:
  - firstname: Helena # Comment
    lastname: Despoux

  - firstname: Thomas
    lastname: Martin
```

Exemple de fichier YAML

Github Actions



L'onglet "Actions" peut être désactivé

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/developement-web-et-dispositif-interactif-s6/travaux-pratiques/numero-3/developpement-web-et-dispositif-interactif-s6_travaux-pratiques_numero-4.ressources.zip

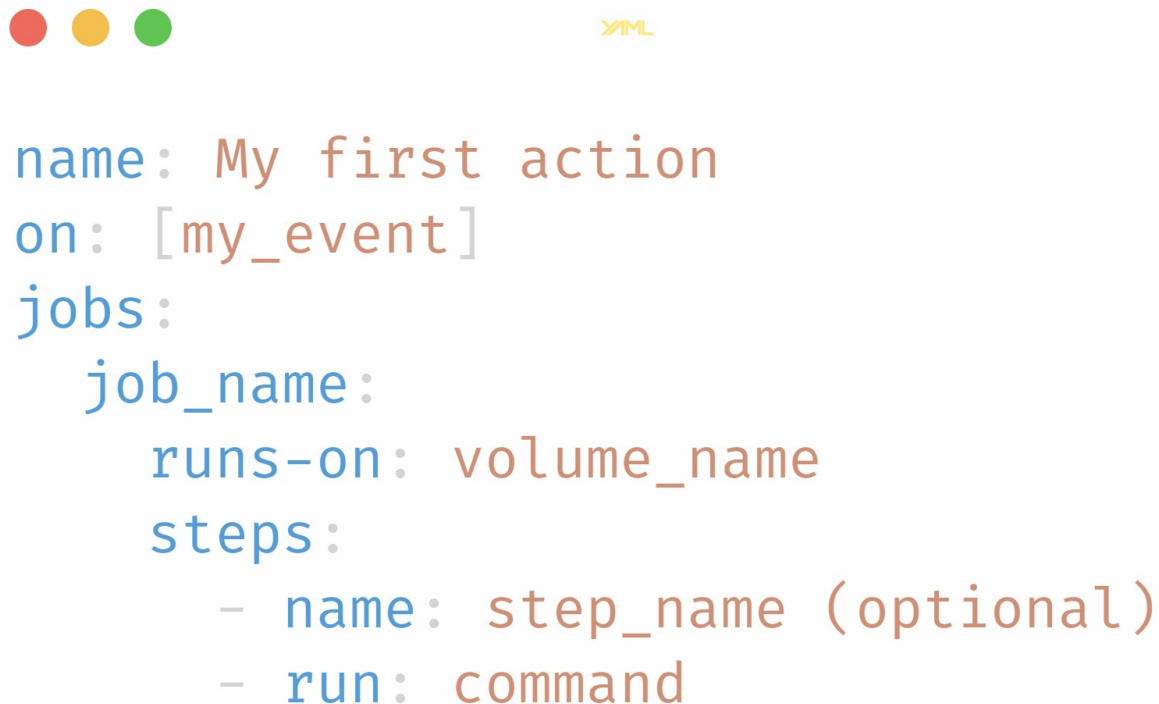
Github Actions - Gabarit

- 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



```
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 définie par Github. Ex : Pull le dépôt
 - Action “actions/checkout@v4”
- S'utilise avec la clé “uses” (à la place de “run”) dans le fichier yml

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/developement-web-et-dispositif-interactif-s6/travaux-pratiques/numero-3/developpement-web-et-dispositif-interactif-s6_travaux-pratiques_numero-4.ressources.zip

Importmap – Fichier externe

- Fichier au format json
- Doit avoir le MIME type
“application/importmap+json”
 - Pas géré par tous les navigateurs
(10/2024)

Source(s) :

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

importmap – Fichier externe



```
1 <script type="importmap" src="importmap.json"></script>
```

On charge un importmap externe si son MIME type est “application/importmap+json”.

Il est possible, à l’avenir, que l’extension “.importmap” arrive.

Source(s) :

- <https://github.com/WICG/import-maps?tab=readme-ov-file#installation> - anglais
- <https://www.honeybadger.io/blog/import-maps/> - anglais

importmap - Scopes

- Possibilité de charger les modules en fonction de l'emplacement du module qui les charge
- Définition d'une clé "scopes" dans l'importmap

Source(s) :

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

Importmap - Scopes



```
1 <script type="importmap">
2 {
3   "imports": {
4     "lodash": "https://cdn.jsdelivr.net/npm/lodash@4.17.21/+esm"
5   },
6   "scopes": {
7     "/test/": {
8       "utils": "../test/utils.js"
9     }
10  }
11 }
12 </script>
```

Le module “utils” ne sera disponible que si le module est dans un dossier “test”

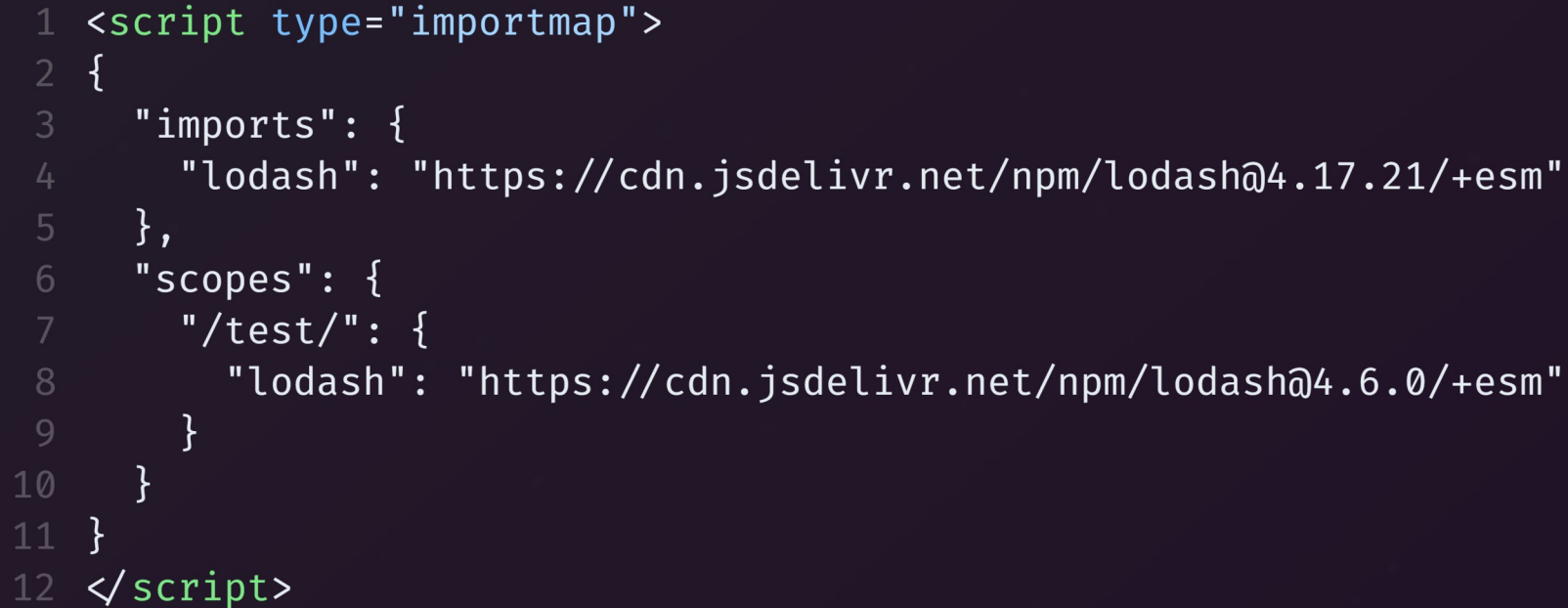
Importmap - Scopes

- Si un module matche plusieurs modules identiques, le navigateur utilisera celui avec le chemin le plus précis

Source(s) :

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

Importmap - Scopes



```
1 <script type="importmap">
2 {
3   "imports": {
4     "lodash": "https://cdn.jsdelivr.net/npm/lodash@4.17.21/+esm"
5   },
6   "scopes": {
7     "/test/": {
8       "lodash": "https://cdn.jsdelivr.net/npm/lodash@4.6.0/+esm"
9     }
10  }
11 }
12 </script>
```

Si un module est contenu dans le dossier “test”, il chargera la version 4.6.0 de lodash, les autres la version 4.17.21

Pratiquons ! - importmaps (Partie 3)

Pré-requis :

- Avoir la ressource `ressources/importmap`

A télécharger ici :

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

Importmap - Limites

- Pas adapté pour les gros projets
 - A préférer pour le prototypage
- Impossibilité de :
 - Faire du tree shaking ou du hot reloading
 - Charger plusieurs importmap par page

Source(s) :

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

Questions ?

