

# HTMX, une alternative aux SPAs ?

Ou React: en as-tu vraiment besoin ?

# Qui je suis

Développeur logiciel chez Dimonoff

Conception 3D et impression: [adaptations3d.ca](http://adaptations3d.ca)

Implication dans BivouaQ: [bivouaq.ca](http://bivouaq.ca)

# Développement frontend

- Frameworkism

Options:

- Framework: React, Vue, Angular, Svelte, ... -> client riche
- Server-side rendering -> client pauvre

Y a-t-il une alternative ?

# HTMX

Option pour enrichir le server-side rendering:

- Chaque élément de la page peut émettre une requête
- Rechargement de page partiel par remplacement d'un élément

# Présentation de l'application démo de liste de tâches

## Fonctionnalités:

- Ajout de tâches
- Suppression de tâches
- Édition in-place
- Ré-ordonnement de tâches via drag-n-drop
- Barrure de la liste et cochage des tâches

# Fonctionnalités HTMX démontrées

- hx-get, hx-post, hx-put et hx-delete: verbes HTTP
- hx-swap: stratégie de remplacement de l'élément
- hx-target: sélecteur CSS de l'élément
- hx-include: sélecteur CSS d'éléments dont la valeur sera ajoutée dans la requête
- hx-trigger: événements javascript pouvant lancer la requête
- Transitions CSS entre les états

# Fonctionnalités HTMX non démontrées

- Out-of-band updates
- Indicateur de requête
- Debounce
- Gestion d'erreurs
- Synchronisation de requêtes (hx-sync)
- Confirmation de requête (hx-confirm)
- Websockets et SSE (server-side events)
- Intégration avec les Web Components ou des framework client riches
- et plus

# Intérêt de HTMX

- Développer un front-end interactif sans quitter le backend
- Locality of Behavior
- Éviter le changement de contexte
- Garder ça simple (vs un framework qui demande un skillset complet)



# Test

- Options de debug (logging) dans HTMX
- Snapshot testing
- Tests js avec mocha
- e2e avec des trucs comme Cypress ou Playwright

# Petite side note sur Go et Templ

- typage des templates

Questions?

# Pour finir

- Code de l'application: <https://github.com/francois-roseberry/go-todo-app>