

SESSION THEME

HTMX in Drupal 入門

AJAX APIをシンプルに置き換える

Studio Umi 小林



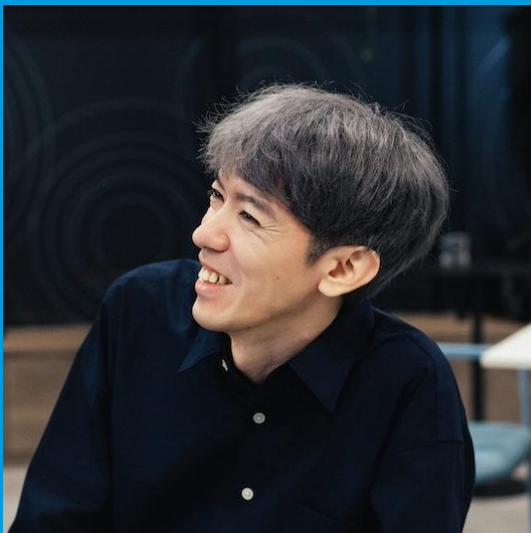
DrupalCamp
Tokyo 2026

AGENDA

1. 自己紹介
2. HTMX とは
3. HTMX の使い方
4. Drupalで利用する方法
5. まとめ



PROFILE



スタジオ・ウミ株式会社

小林 基樹

- バックエンド担当
- Drupal歴: 2016年～ (Drupal7 -)
- <https://www.drupal.org/u/otofu>



SECTION

HTMXとは





HTMX の概要

- 公式サイト: <https://htmx.org/>
- 依存関係が無い約16KBの軽量ライブラリ(Reactと比較し67%削減)
- HTMLのみでAjaxリクエストが可能 (hx- で始まる専用の属性)
- サーバーからHTMLを受け取りページの一部を更新
- WebSocketやSSEにも対応している為、リアルタイム通信も可能



HTMX の原則

- すべての要素がHTTPリクエストを行うことを可能にする
- クリックやフォームのsubmitだけでなくあらゆるイベントをトリガーに出来る
- GET/POSTだけではなくPUT/DELETEといったHTTPメソッドが利用可能になる
- ページ全体ではなく指定した任意の要素を更新する
- HTMXのリクエストに対するレスポンスはhtmlを返す必要がある



DrupalにおけるHTMX

- Drupal 11.2でコアに依存関係として含まれ11.3で正式にサポート
- AJAX APIをHTMXに置き換えるという計画 ※1
- AJAX APIと比較しHTMX関連のJSは71%削減
- Drupalで利用するには data- プレフィックスが必要(data-hx-*)
- AJAX APIを利用するにはDrupal固有APIの学習が必要だがHTMXはその知識があれば良い
- コントローラーでHTMXを受け付ける為アクセス制御やキャッシュ機構はDrupalの知識があれば問題ない

※1: <https://www.drupal.org/project/drupal/issues/3404409>

SECTION

HTMXの基本





AJAXリクエスト送信

- hx-get / hx-post / hx-put / hx-patch / hx-delete でHTTPメソッドに応じたリクエストが可能
- hx-target で置き換える要素を指定可能
- hx-swap で要素の置き換え方法を指定可能

```
<body>
  <button hx-get="/clicked" hx-target="#result" hx-swap="outerHTML">
    Click Me
  </button>
  <div id="result"></div>
</body>
```



hx-swap

- innerHTML : 子要素として追加 (デフォルト)
- outerHTML : 対象要素全体を置き換え
- afterbegin / beforeend : ターゲット内の最初の子要素の前(後)に追加
- beforebegin / afterend : ターゲットの前(後)に追加
- delete : ターゲット要素を削除
- none : 何もしない (HX-Redirect等リダイレクトのみ行いたい場合)



hx-trigger

- once : 1度のみ
- load : 画面読み込み時
- revealed : 要素がビューポートに現れたタイミング
- changed : 要素の値が変更された場合
- delay:<timing> : 指定した時間経過後 (changed等と組み合わせる)
- every <timing> : 指定時間ごと (ex: every 1s だと1秒間隔)

```
<input hx-get="/search" hx-target="#result" hx-trigger="keyup changed delay:300ms" />
```

SECTION



Drupalでの利用方法



data- プレフィックス

- 既存属性への影響を考慮しDrupalにおいてはdata-プレフィックスが必要
- プレフィックス以外はHTMXと同じ構文を指定

```
<!-- HTMX -->
```

```
<button hx-get="/clicked" hx-target="#result">Click Me</button>
```

```
<!-- Drupal HTMX -->
```

```
<button data-hx-get="/clicked" data-hx-target="#result">Click Me</button>
```



Drupal 専用機能

- HTMX ファクトリクラス
 - Drupal\Core\Htmx\Htmx
 - HTMX属性(data-hx-*)、レスポンスヘッダーを生成
- 専用レンダラー
 - ルーティングに `_htmx_route: true` を指定
 - HTMX用に最適化されたhtmlレスポンスを返す

SECTION



Drupalにおけるユースケース

構成の同期 - エクスポート (シングルアイテム)

- 構成タイプを選択すると構成名がHTMXによって置き換わる
- 構成名を変更すると構成の定義が表示される

ホーム > 管理 > 環境設定 > 開発 > 同期

単一の構成項目のエクスポート ☆

同期 インポート エクスポート

フルアーカイブ シングルアイテム

YAML 構造を表示する設定アイテムを選択します。

構成タイプ

簡易構成

構成名

- 選択 -

あなたの構成は下記のとおりです:

```
$form['config_type'] = [  
  '#title' => $this->t('Configuration type'),  
  '#type' => 'select',  
  '#options' => $config_types,  
  '#default_value' => $config_type,  
];  
  
$form_url = Url::fromRoute('config.export_single');  
(new Htmx())  
  ->post($form_url)  
  ->onlyMainContent()  
  ->select('*:has(>select[name="config_name"]')  
  ->target('*:has(>select[name="config_name"]')  
  ->swap('outerHTML')  
  ->applyTo($form['config_type']);
```


構成の同期 - エクスポート (シングルアイテム)

- AJAX API を利用していた際は以下

```
$form['config_type'] = [  
  '#title' => $this->t('Configuration type'),  
  '#type' => 'select',  
  '#options' => $config_types,  
  '#default_value' => $config_type,  
  '#ajax' => [  
    'callback' => '::updateConfigurationType',  
    'wrapper' => 'js-config-form-wrapper',  
  ],  
];  
  
public function updateConfigurationType($form, FormStateInterface $form_state) {  
  $form['config_name']['#options'] = $this->findConfiguration($form_state->getValue('config_type'));  
  $form['export']['#value'] = NULL;  
  return $form;  
}
```

SECTION

まとめ





- HTMXを利用すると利用するJavaScriptの容量が減り効率化
- サーバー側はHTMLを返せば良いので既存処理が使い回せる
- data-hx-* 属性でどのような処理が行われているか明確になる
- HTMX構文の学習コストが低く段階的に利用可能
- コアのAJAXも徐々にHTMXに置き換わる予定

積極的にHTMXを活用していこう！



ご清聴ありがとうございました