


Lightning Web Componets について

 Slides are here

スライド:<https://powerninja.github.io/SSCLWC/ja/index.html>

リポジトリ:<https://github.com/powerninja/SSCLWC>

はじめに

- TODO:勉強会環境にdemoを用意する
- Summer '19 に登場し、約4年立ちました。
- フロントの開発では、Aura や Visualforce よりも選択される機会が増えてきていると思うので、興味のある方は触ってみてください。
- 今回Lightning Web Components の開発を行うための環境構築等の詳細な説明はしません。

トピックス

1. 画面開発の歴史
2. Lightning Web Components と Aura の違いは？
3. なぜ Lightning Web Components を選択するのか
4. 案件紹介
5. 作成した Lightning Web Components の紹介
6. おまけ

画面開発の歴史

- Visualforce
 - Summer '08 くらい？
- Lightning Aura Components
 - Auraと記載される
 - 2014くらいに発表？
- Lightning Web Components
 - LWCと記載される
 - Summer '19

Lightning Web Components と Aura の違いは？

- 共通点
 - Salesforce上での見た目はほぼ同じ([LDS](#)を標準で使用)
 - classic未対応
 - JavaScriptを用いた開発
 - 外部 JavaScript ライブラリの[使用可能](#)(静的リソースで読み込ませるnpmは不可)
 - [Tailwind CSS](#)のようにクラス名でスタイルを与えることができる
 - そのため、CSSファイルはLDSでよければ不要

Lightning Web Components と Aura の違いは？

- 相違点
 - Auraは開発者コンソールで作成可能だが、LWCはVisual Studio Codeが必要(chromeの拡張機能で開発は可能)
 - LWCはユニットテスト [Jest](#)に対応している
 - LWCで対応していない機能がまだある、その場合はAuraを作成する必要あり(モバイル対応など)
 - Auraは開発がアーカイブ化されている(サポートはしている)
[Aura開発リポジトリ](#)
[LWC開発リポジトリ](#)

Lightning Web Components と Aura の違いは？

- Visualforceとの比較
 - 共通点
 - あまりない
 - 相違点
 - コントローラーがLWCはJavaScript(ブラウザ動作),Visualforceとの比較はApex(サーバ動作)
そのため、LWCのパフォーマンスが良い

なぜ Lightning Web Components を選択するのか

新しいから

なぜ Lightning Web Components を選択するのか

- 開発コミュニティが活発なため、新機能などに期待できる
- 標準的なJavaScriptを使用することができるため、JavaScriptの開発経験がある方は開発しやすい
 - そのため、学習コストが低い
- ファイルの数が少なく、初期段階の理解が早い(個人差あり)
- (Auraと比べると)パフォーマンスが良い

なぜ Lightning Web Components を選択するのか

- LWCで作成されるファイル数

プロジェクト名(任意で設定可能)

| -HTML

| -JavaScript

| -xml

| -css(任意)

なぜ Lightning Web Components を選択するのか

- Auraで作成されるファイル数
 - 全部が必要なわけではないが。。。

プロジェクト名(任意で設定可能)

- | -auradoc
- | -cmp(HTML)
- | -cmp-meta.xml
- | -css
- | -design
- | -svg
- | -Controller.js
- | -Helper.js
- | -Render.js

なぜ Lightning Web Components を選択するのか

```
<template>
  <div slds-p-left_xx-large>{hello}</div>
</template>
```

```
import { LightningElement } from 'lwc';

export default class Test extends LightningElement {
  hello = 'Hello,World!'
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
  <apiVersion>56.0</apiVersion>
  <isExposed>true</isExposed>
</LightningComponentBundle>
```

LWCの始め方 → 案件紹介に変更する

1. salesforce CLIのインストール
2. vsCodeのインストール
3. vsCode内でSalesforce Extension Packをインストールする
4. ctrl + shift + P を押下し、新規プロジェクトを作成
5. ctrl + shift + P を押下し、新規LWCを作成する

作成した Lightning Web Components の紹介

- 勉強会の環境に一部デプロイあり

おまけ

- こちらのスライドはgithub actionを使用して、vscodeでmarkdownを記載するだけでスライドを作成できるように。

または

- LWCのOSS版紹介

楽しくスライドを書こう！ 🙌