electron JS

S1 = 개요1

일렉트론은 웹 기술로 앱 개발을 할 수 있게 해주는 소프트웨어 프레임 워크입니다.

즉, 일반적으로 Window app 개발에 사용되는 C++, C# 이나 macOS에서 사용하는 swift등이 아닌

node.js 기반으로 JS, HTML, CSS를 사용하여 Desktop Application을 만듭니다.

S2 = 개요2

또한 Electron 은 JavaScript, HTML, CSS를 활용하는 전통적인 웹개발 방식을 따르기 때문에

React, vue, Angular 와 같은 프레임워크(또는 라이브러리)와 함께 사용할 수 있다.

S3 = 활용사례

일렉트론으로 개발한 데스크톱 애플리케이션은 시중에 다양하게 출시되어 있습니다.

대표적으로 VS코드, 슬랙(Slack) 등이 있으며 완성도가 뛰어나

성능과 개발 편의성은 검증되었다고 보면 됩니다.

S4 = 장점

1. 낮은 진입장벽

자바스크립트 웹 개발자가 데스크톱 애플리케이션을 만들기 위해

새로운 기술이나 언어를 배울 필요가 없습니다., 방대한 커뮤니티

2. 개발속도 향상

일렉트론으로 개발된 애플리케이션은 일반적으로 웹 애플리케이션의

구조를 재사용하기 때문에

개발에 소요되는 시간과 비용이 줄어드는 효과를 제공합니다.

3. 크로스플랫폼 지원

각 OS마다 최적화 필요가 없고 세가지 OS에 대한 각각 빌드를 지원, 설치 가능

윈도우, 맥, 리눅스 별로 js, html, css를 활용해 애플리케이션 구성 가능

4. 써드파티 지원

일렉트론은 nodeJS의 모든 빌트인 모듈 사용가능,

npm에 등록된 대부분의 모듈도 사용가능하다

5. 빌드 툴과 인스톨러 제공

일렉트론은 빌드 툴을 제공하여 순식간에 패키징이 가능하고 인스톨러도 만든다.

또한 autoUpdater라는 구성요소로 앱 자동 업데이트 기능도 사용가능하다.

S5 = 단점

1. 큰 설치 파일 용량

크로미움, 노드 JS, npm모듈들이 패키징 되어있어 기본 앱 용량이 100mb이상일만큼

무겁고, 파일 용량이 크다

2. 상대적으로 느린 속도

하나의 플랫폼을 위해 개발된 네이티브 앱 보다는 효율성이 떨어지고

상대적으로 속도가 느림

3. 보안

사용자가 앱소스코드를 역컴파일 가능

실제로 정말 간단하게 설치 및 실행을 해보겠습니다.

S6 = 설치 및 빌드

각종 설정 예시입니다.

S7 = electron 설정

마지막으로 실제로 electron을 사용해

앱을 만들고 빌드를 해봤습니다

상당히 간편하죠?

S9 = 크로스 플랫폼

OS간 호환, 처음에 만들때 윈도우, 발표 준비하면서 생각없이

macOs로 빌드 했는데 오류는 없었고

icon이 출력되지 않는 이슈가 있었는데,

vue.config.js => win, mac icon설정도 간편하게 할수있음.

이상~~!

특징=

1.가장 빠르고 쉽게 개발하기는 electon이 최고라고 생각함 ( 에코시스템, 쉽고 친숙한 javascript or react기반으로 개발 가능) ,

뷰, 리액트로 된 코드를 패키징만 해도 데스크탑 앱으로 만들수 있는거 같음.

2.다만 여기에 일렉트론이 웹 애플리케이션을 구동하도록

진입점 코드를 작성하는 약간의 수고가 필요합니다.

장점=

- 방대한 커뮤니티를 보유하고 있다.

- Microsoft, Discord, Slack 등에서 사용 중이다.

- 낮은 러닝 커브로 배우기 쉽다.

- 웹 기술을 사용할 수 있다.

- Node.js를 사용하므로 Javascript의 모든 기능을 사용 할 수 있다.

단점=

- 보안은 사실상 node.js의 큰 단점, 난독화 툴 사용해야함, 보안이 중요시되는 업계에서는 적절치 않다고 생각됨

- 큰 사이즈(~120MB)의 설치 파일 사이즈를 가지고 있다.

- 앱이 무겁다. 많은 RAM과 CPU를 사용한다.

- 모바일 지원은 없다.

- 보안에 취약하다. 사용자가 앱 소스코드를 디컴파일해서 볼 수 있다.