



Z przeglądarki na desktop

Aplikacje typu SPA w środowisku desktopowym za pomocą Node-Webkit

Single Page Applications

- Jeden plik “html”
- Wszystkie pliki ładowane w tle bez przeładowania
- Interfejs reaguje na akcje użytkownika
- Często bez frameworka JavaScript się nie obejdzie

Przeprowadzka na desktop

Dlaczego się przeprowadzać?

- Nasza aplikacja zyskała by skrzydła na desktopie
- Przyzwyczajenie użytkowników

Przeprowadzka na desktop

Jakie daje to nam korzyści?

- Jedna aplikacja - wiele platform
- Zaoszczędzony czas
- Ten sam Look and Feel - spójność



nodewebkit

Czym jest?

Node-Webkit

Node.js - szybka charakterystyka

- Środowisko programistyczne
- Do zastosowań serwerowych i nie tylko
- Szybki silnik JavaScript od Google - V8
- Duża ilość dostępnych modułów

Node-Webkit

Webkit - szybka charakterystyka

- Silnik wielu przeglądarek internetowych
- Odpowiada za renderowanie HTML/CSS/JS
- Obecnie najpopularniejszy silnik

Node-Webkit

Możliwości

- Wszystkie moduły Node.js
- HTML5, CSS3, WebGL etc.
- Aplikacja web w oknie systemowym
- Windows, Linux, Mac
- Wsparcie dla menu aplikacji oraz zasobnika systemowego
- Obsługa ikony aplikacji
- Tryb kiosk, fullscreen
- Dialogi wyboru oraz zapisu plików

Node-Webkit

Istnieją jednak drobne minusy...

- Brak natywnych kontrolek
- Dodatkowe bajery na desktopie - dodatkowe warunki w kodzie lub warstwa abstrakcji
- Większy rozmiar aplikacji
- W niektórych wypadkach wydajność mniejsza od natywnej

Node-Webkit

Budowa aplikacji Node-Webkit

- package.json - główny plik konfiguracyjny
- index.html lub inny plik html zdefiniowany w package.json
- pliki modułów Node.js standardowo w node_modules
- odpalamy: **nw.exe path/to/file.html**

Node-Webkit

Budowa aplikacji Node-Webkit

Czas na DEMO!

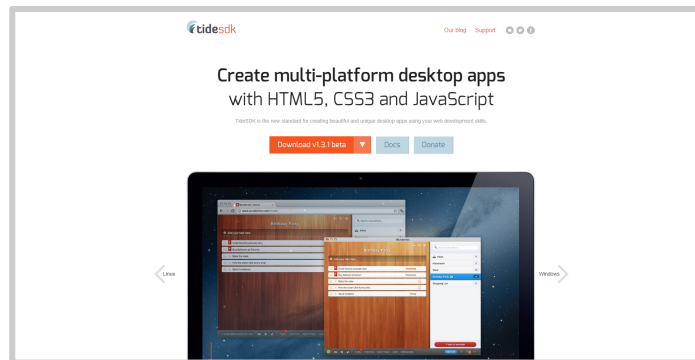
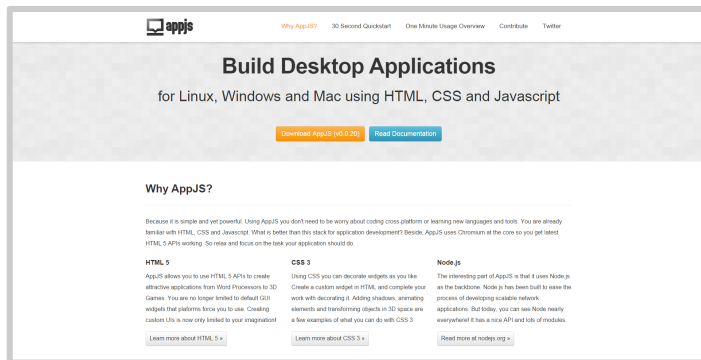
Node-Webkit



<https://github.com/sebastianrosik/RSTNoise>

Node-Webkit

Alternatywy



Node-Webkit

Dzięki :)