

Rozszerzenia Chrome z Angular

Krótki wstęp do tworzenia rozszerzeń z Angular



Wprowadzenie

Angular nierozłącznie kojarzony jest z dużymi aplikacjami webowymi, które obsługujemy w naszych przeglądarkach. Framework, który dzięki swojej jasnej strukturze i sztywnym ramom, ułatwia deweloperom budowanie rozbudowanych aplikacji.

W tym artykule przedstawię Ci inne przeznaczenie dla tego framework'a - rozszerzenie do przeglądarki Chrome. Warto zauważyć, że aplikacja,

z której korzystasz na codzień, w postaci rozszerzenia może znacznie ułatwić i przyspieszyć Twoją pracę. Natomiast samo poznanie procesu tworzenia rozszerzeń będzie kolejną cenną umiejętnością w Twoim arsenale.

Czego potrzebujemy

Na samym wstępie potrzebujemy dwóch rzeczy, bez których nie będziemy w stanie zacząć:

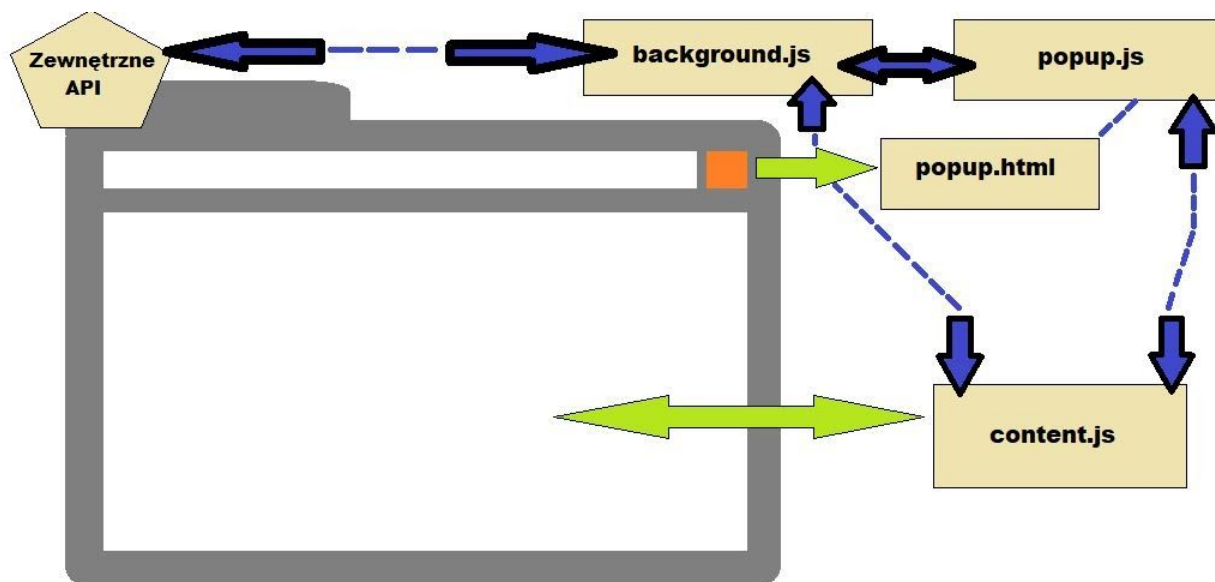
- naszą aplikację napisaną w Angular
 - poznanie podstaw budowania rozszerzeń. Tutaj warto skorzystać z oficjalnej dokumentacji Google ([Overview of Google Chrome Extension](#) oraz [Getting Started Tutorial](#)). Zachęcam mocno do zapoznania się z dokumentacją i stworzenie swojego prostego rozszerzenia od podstaw. To znacząco ułatwi Tobie zrozumienie dalszych kroków oraz poruszanie się w środowisku rozszerzeń Chrome.
 - oczywiście przeglądarki Google Chrome :-)
-

Zapoznanie się z elementami rozszerzenia

Budowa rozszerzeń Chrome jest zbliżona do wzorca MVC, co znacząco ułatwia proces zrozumienia ich działania.

Podstawowe elementy:

- **popup.js/popup.html** - to nasz widok. Tutaj znajduje się cała struktura HTML rozszerzenia oraz skrypty działające na widok. Ma pełen dostęp do API rozszerzeń Chrome.
- **background.js** - część rozszerzenia odpowiedzialna za przekazywanie żądań oraz odpowiedzi. Działa tle i tylko wtedy kiedy jest to potrzebne. Można ją porównać do kontrolera w modelu MVC - w dużej mierze jest odpowiedzialna za obsługę zdarzeń. W background.js również wysyłamy zapytania do zewnętrznego API.
- **content.js** - część, która działa w kontekście strony internetowej, która jest wyświetlana w przeglądarce. Pobiera oraz modyfikuje elementy DOM strony internetowej. W przypadku aplikacji Angular, nie będziemy poświęcać uwagi tej części rozszerzenia.



Zaczynamy!

Na początku zaczynamy od utworzenia manifestu dla rozszerzenia.

W folderze `/src/`, naszej aplikacji, tworzymy plik `manifest.json`. Następnie w pliku umieszczamy podstawowy obiekt json do rozpoczęcia tworzenia rozszerzenia.

```
{
  "name": "Getting Started Example",
  "version": "1.0",
  "description": "Build an Extension!",
  "manifest_version": 2
}
```

W tym momencie powinniśmy dodać jakieś akcje aby nasze rozszerzenie wyświetlało się po kliknięciu w ikonę (popup.html). W tym celu edytujemy nasz plik manifest.json:

```
{
  "name": "Getting Started Example",
  "version": "1.0",
  "description": "Build an Extension!",
  "manifest_version": 2,
  "browser_action": {
    "default_popup": "popup.html"
  }
}
```

Nasze rozszerzenie będziemy załadowywać z folderu `dist`, w związku z czym musimy wykonać dwa kroki:

- w pliku `angular.json` naszej aplikacji, umieszczamy manifest.json w tablicy `assets`, która stanowi część obiektu `options`

```
"assets": [
  "src/favicon.ico",
  "src/assets",
  "src/manifest.json"
],
```

-
- uruchamiamy operację `ng build` w celu załadowania *angular.json* i utworzenia folderu *dist*.

Na sam koniec musimy użyć AOT w naszej kompilacji. W tym celu musimy ponownie edytować nasz plik *angular.json*:

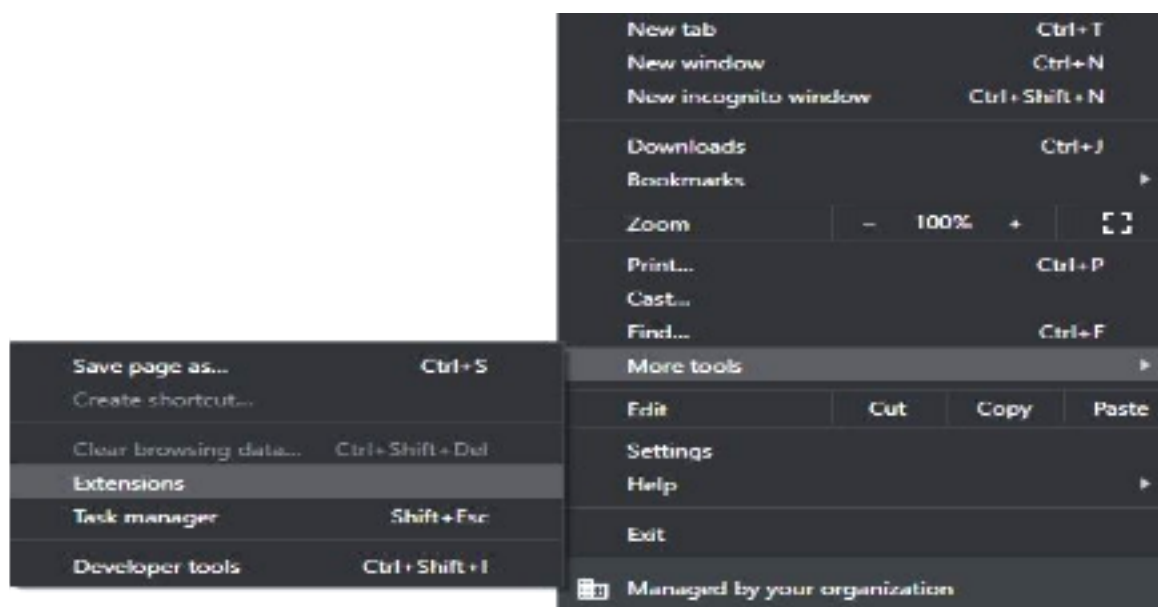
```
"projects": {  
  "angular-chrome-extension": {  
    ...  
    "architect": {  
      "build": {  
        "builder": "@angular-devkit/build-angular:browser",  
        "options": {  
          ...  
          "aot": true  
          ...  
        },  
        ...  
      },  
      ...  
    },  
    ...  
  },  
  ...  
}
```

Teraz wystarczy ponownie skompilować naszą aplikację przy użyciu polecenia `ng build`.

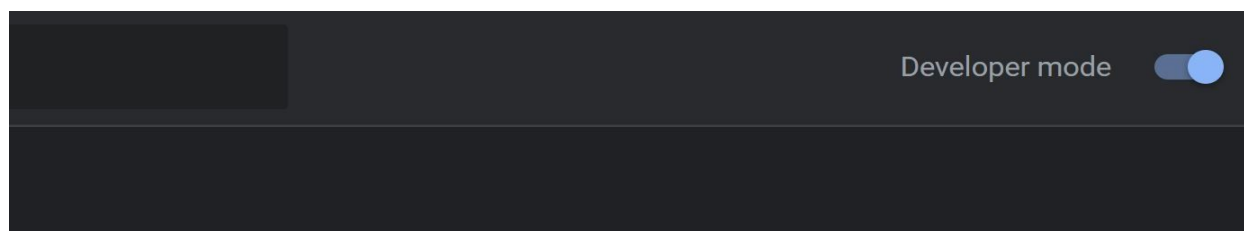
W tym momencie mamy gotowy plik *dist* do załadowania w naszej przeglądarce Chrome.

Czas na Chrome

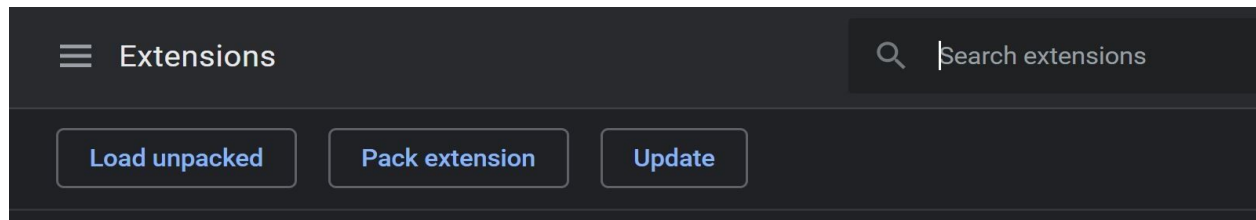
Jeśli już mamy plik *dist* wraz ze znajdującym się *manifest.json*, możemy przystąpić do załadowania naszej aplikacji do Chrome. W tym celu, klikamy trzy kropki w prawym górnym rogu i wybieramy *More Tools* (więcej narzędzi) a następnie *Extensions* (rozszerzenia):



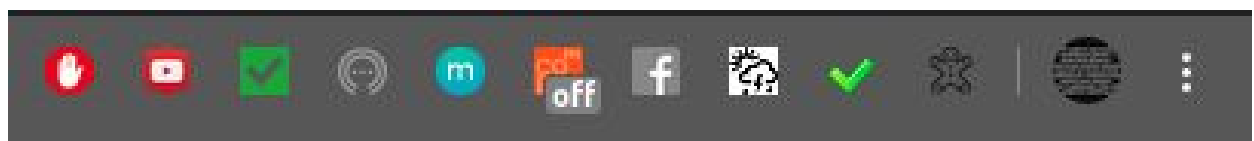
W rozszerzeniach, w prawym górnym rogu włączamy opcję *Developer mode* (tryb developera):



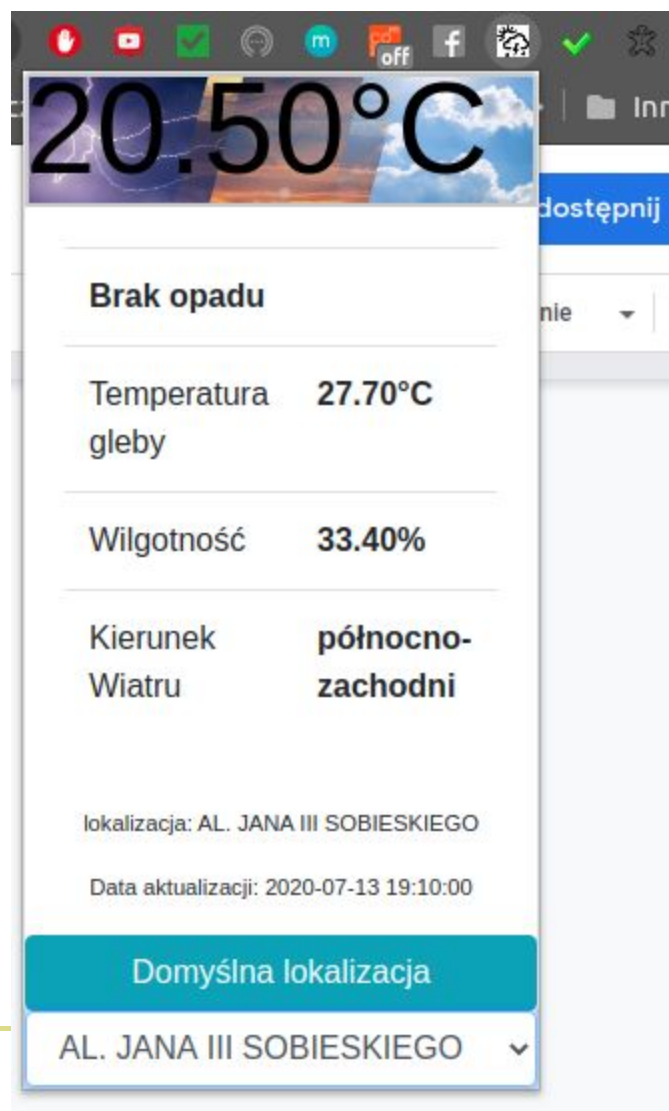
Następnie wybieramy opcję po lewej stronie *Load Unpacked* (rozpakuj rozszerzenie):



Wybieramy ścieżkę z naszym rozszerzeniem i ładujemy ją. W prawym górnym rogu, gdzie znajdują się rozszerzenia Chrome, powinniśmy zobaczyć logo naszego rozszerzenia:



Po kliknięciu w logo, nasze rozszerzenie uruchomi plik popup.html, który z kolei uruchomi aplikację angularową.



Bibliografia oraz przydatne linki:

- [Building a chrome extension with angular](#)
- [Chrome extension with Angular — from zero to a little hero](#)
- [Getting Started Tutorial - Google Chrome](#)
- [Manifest File Format](#)