

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

## FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



### Projekt WAP

Tým 14: Sledování dopadu meteoritů

Zpracoval: Vladislav Halva  
Datum: 7. dubna 2020

# 1 Úvod

Tento dokument popisuje implementaci *SPA* (Single Page Application) pro sledování dopadů meteoritů s použitím API [1] poskytovaným NASA. SPA umožňuje kromě zobrazení dat také jejich filtrování a vyhledávání v nich, podle lokality, data dopadu, apod. K vytvoření aplikace byl použit JavaScriptový framework *Vue.js* [2]. Aplikace je dostupná na adrese <http://meteoritelandings.8u.cz/>.

## 2 Použitý dataset a API

K získání dat o dopaden meteoritů bylo získáno z datasetu [1], který poskytuje NASA prostřednictvím veřejného API. Jedná se o *Socrata Open Data API* [3], které umožňuje jednoduché dotazování nad daty pomocí *SoQL*. Data je možné získat v několika formátech, pro tuto aplikaci byl zvolen formát *JSON*.

Dataset aktuálně sestává z 45716 záznamů o dopadech meteoritů. Každý záznam sestává z deseti hodnot viz [1], z nichž nejdůležitějšími jsou id, jméno, váha, rok dopadu/nalezení a poloha.

## 3 Implementace

### 3.1 Použité technologie

#### 3.1.1 Vue, Vue CLI, Webpack

Aplikace byla vytvořena v jazyce *JavaScript* s využitím frameworku *Vue.js* [2]. Projekt pro implementaci aplikace byl inicializován pomocí *Vue CLI* [4], které umožňuje jednoduché nastavení projektu, integraci knihoven a balíčků, a také automaticky konfiguruje *Webpack* [5], který umožňuje práci s moduly a při nasazení aplikace svazuje závislosti, provádí optimalizace a vytváří balíček pro nasazení na web. Umožňuje také obecně import jiné "assets", jako Sass nebo obrázky. *Vue CLI* dále umí pracovat v "development módu", ve kterém se všechny provedené změny v souborech živě propisují do aplikace při uložení (není tedy třeba obnovovat okno prohlížeče pro načtení změn), což velice zjednodušuje práci.

#### 3.1.2 Vue Router

Pro navádění mezi jednotlivými stránkami SPA byl použit *Vue Router* [7], který tento proces velice zjednodušuje. *Vue Router* také disponuje "HTML5 history módem", který využívá *history.pushState API* pro navigaci mezi URL bez znovunačítání stránky. Oproti standardnímu "hash módu" není třeba používat křížek pro simulaci přechodu mezi stránkami.

#### 3.1.3 Vuex

Pro uložení dat získaných z databáze přes veřejné API byl použit *Vuex* [8], který slouží jako centralizované úložiště pro všechny komponenty aplikace a zaručuje korektní mutace jeho stavu.

### 3.1.4 Axios

Pro komunikaci s veřejným API, skrze které jsou dotazována data o dopadech meteoritů byl použit HTTP klient *Axios* [9] založený na *Promise API*. Který umožňuje vytvoření XMLHttpRequest z prohlížeče, zachycení odpovědi, automatickou transformaci do JSON a jiné.

### 3.1.5 Google Maps JavaScript API

*Google Maps JavaScript API* [6] bylo použito pro zobrazení polohy meteoritů a jejich selekci dle polohy. Umožňuje obohacení map o vlastní obsah, navázání naslouchačů na události vyvolávané mapami a podobně. Pro zjednodušení práce z prostředí Vue.js byl použit balíček *vue2-google-maps*.

Klíč použitý pro přístup k API je omezen pouze pro doménu *http://meteoritelandings.8u.cz/*, proto nebude zobrazení v mapě z jiné domény funkční.

### 3.1.6 Chart.js

Pomocí *Chart.js* [10] byly vytvořeny grafy v sekci statistiky. Jedná se o JavaScript knihovnu, s jejíž pomocí lze vytvářet grafy založené na HTML5. Knihovna také implementuje animace, přechody, překreslování při změně velikosti okna a podobně.

## 3.2 Implementační detaily

### 3.2.1 Členění projektu

Celá aplikace je rozčleněna do několika komponent. Adresář *views* obsahuje komponenty reprezentující jednotlivé stránky aplikace, mezi kterými probíhá routování. Adresář *components* obsahuje všechny zbylé dílčí komponenty.

Adresář *router* obsahuje soubor s nastavením Vue Routeru pro navádění mezi jednotlivými stránkami (*views*).

V adresáři *store* se nachází soubor s nastavením centrálního úložiště pro data. Jsou zde také implementovány metody pro získání dat v určitém formátu (např. jako histogram pro vykreslení grafu).

Adresář *styles* obsahuje soubory se styly v jazyce Sass.

Adresář *assets* obsahuje všechny dodatečné soubory jako pomocné JavaScriptové kódy nebo obrázky.

### 3.2.2 Načítání dat o meteoritech z API

K načítání dat o meteoritech dochází v aplikaci na několika místech.

Při prvním přístupu na libovolnou stránku aplikace je zahájeno asynchronní stahování vybraných informací o datech (pouze id, jména, váhy a roku dopadu). Tyto data jsou následně uložena do centrálního úložiště (Vuex) a jsou využívána komponentami pro vyhledávání (*searchPanel*) a kom-

ponentami vytvářející grafy na stránce se statistikami. Pro tyto účely je třeba pouze omezených informací, ale o všech datech, která jsou dostupná.

Dále jsou data po stránkách (100 objektech) stahována v komponentě TableView, kde jsou všechna data zobrazena ve skrolovatelné tabulce. Tato tabulku na svém konci obsahuje komponentu Observer, která při svém zviditelnění emituje událost dotazující další stránku. Tento přístup byl zvolen, jelikož nedochází k vykreslení celé tabulky najednou a zároveň není třeba dotazovat velký objem dat najednou. Při změně způsobu řazení je pouze vynulováno počítadlo stránek, upraveny parametry dotazu (SoQL) a jsou vykreslena nová data.

Data přímo z API také dotazuje komponenta Mapview pro zobrazení meteoritů v mapě. Důvodem je, že data dotazuje ve shlucích tak, že vždy do stejného hluku patří meteority náležící danému obdelníku danému zeměpisnou šířkou a délkou. Zemský povrch je takto rozdělen mřížkou, a při kliknutí na daný obdelník v mapě (Google Maps JS API) se zobrazí do něj náležící meteority. Objekty jsou vráceny pouze jako *Promise* prakticky jsou tedy staženy až při kliknutí na obdelník v mapě.

Informace o jednom objektu specifikovaném předaným ID jsou dotazována v komponentě Detail, kde je potřeba načíst všechny informace, ale pouze o vybraném objektu. Na stránku detailu se lze dostat po kliknutí na řádek objektu v tabulce či pomocí vyhledávacího panelu.

## 4 Odevzdané soubory

Archiv *xhalva04.zip* obsahuje všechny zdrojové soubory a konfigurační soubory projektu potřebné k sestavení.

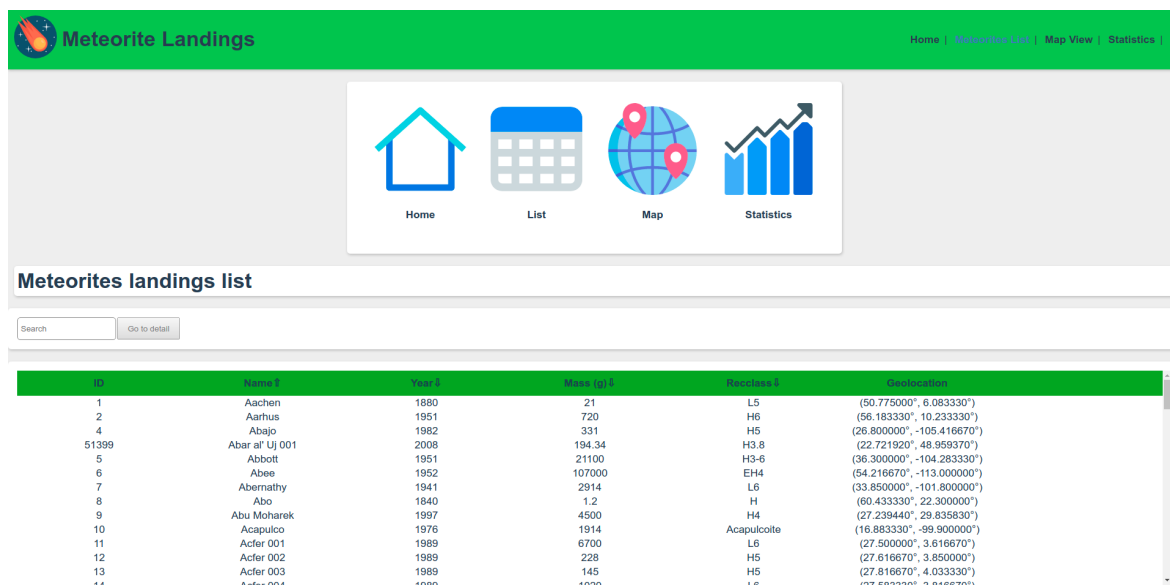
Všechny zdrojové soubory se nachází v adresářích *src* a *public*. Vytvořené zdrojové kódy jsou v adresáři *src*.

Adresář *dist* obsahuje již sestavenou aplikaci připravenou pro nasazení na web.

Soubor *README.md* (taktéž v uvedeném archivu) obsahuje návod k sestavení projektu.

Soubor *xhalva04\_doc.pdf* (tento dokument) obsahuje popis aplikace.

## 5 Ukázka aplikace



Obrázek 1: Ukázka aplikace.

## 6 Shrnutí

Framework Vue.js byl vybrán z několika důvodů. Dle dostupných zdrojů je nejvhodnější pro menší projekty. Vzhledem k tomu, že jsem dosud neměl zkušenost s žádným frontendovým frameworkem ani tvorbou SPA, tak mě zaujala také údajně nejkratší učící křivka z populárních používaných frameworků. Použití Vue CLI se ukázalo jako markantní zjednodušení inicializace projektu, práce s moduly a ostatními zdroji, a použití dalších balíčků. Pomocí Vue Router je poměrně jednoduché a intuitivní vytvořit navigaci mezi jednotlivými stránkami SPA. Dalšími výhodami je existence balíčku pro práci s Google Maps JS API a Chart.js, které také značně usnadnily práci. V neposlední řadě Vue.js umožňuje přehledné členění aplikace na komponenty. Celkově tedy hodnotím volbu toho frameworku kladně a nenarazil jsem na žádný zásadní nedostatek. Za drobnou vadu bych považoval pouze velice pomalé vykreslování komponent vytvořených iterací přes list hodnot, se kterým jsem se setkal při implementaci tabulek zobrazujících informace o meteoritech.

## Reference

- [1] NASA's Open Data Portal: Meteorite Landings.  
URL: <https://data.nasa.gov/Space-Science/Meteorite-Landings/gh4g-9sfh>.  
[Online Accessed: 2020-04-07].
- [2] Vue.js: Javascript framework. URL: <https://vuejs.org/>. [Online Accessed: 2020-04-07].
- [3] Socrata Open Data API. URL: <https://dev.socrata.com/>. [Online Accessed: 2020-04-07].
- [4] Vue CLI. URL: <https://cli.vuejs.org/>. [Online Accessed: 2020-04-07].
- [5] Webpack. URL: <https://webpack.js.org/>. [Online Accessed: 2020-04-07].
- [6] Google Maps JavaScript API.  
URL: <https://developers.google.com/maps/documentation/javascript/tutorial>. [Online Accessed: 2020-04-07].
- [7] Vue Router. URL: <https://router.vuejs.org/>. [Online Accessed: 2020-04-07].
- [8] Vuex. URL: <https://vuex.vuejs.org/>. [Online Accessed: 2020-04-07].
- [9] Axios. URL: <https://github.com/axios/axios>. [Online Accessed: 2020-04-07].
- [10] Chart.js. URL: <https://www.chartjs.org/>. [Online Accessed: 2020-04-07].