

Semestrální práce z KIV/DBM2

SPARQL Tool

Martin Matas
A18N0095P
martinm@students.zcu.cz

1. prosince 2019

Obsah

1	Zadání	1
1.1	Data:	1
2	Řešení	1
2.1	Struktura aplikace	1
2.2	Struktura úložiště	2
3	Uživatelská dokumentace	2
3.1	Sestavení	2
3.2	Spuštění	3
3.3	Práce s knihovnou	3
3.3.1	Modifikace knihovny	4
3.3.2	Nasazení	4
4	Závěr	4

1 Zadání

Vytvořit Javascriptový nástroj, který bude snadno vložitelný do jiných webových projektů (ideálně jeden javascript soubor k načtení v html hlavičce) a bude umět:

- odeslat SPARQL dotaz na zadaný libovolný SPARQL endpoint (ajax)
- vypsat výsledek dotazu do triviální webové tabulky
- načíst/uložit dotaz do místní paměti prohlížeče (browser local storage)
- spravovat uložené dotazy
- vyhledávat v uložených dotazech
- verzovat dotazy

1.1 Data:

- dle svého uvážení, případně na vyžádání dodám SPARQL datazy a příslušné datasety

2 Řešení

Nástroj byl po zvážení vypracován pomocí jazyka **TypeScript**, který umožňuje přeložení zdrojového kódu do Javascriptu ve standardu ECMAScript 5 (ES5). Standard ES5 byl zvolen především pro svoji rozšířenost, protože v dnešní době je podporován většinou prohlížečů a knihoven. Další nespornou výhodou jazyka TypeScript byla možnost napsat nástroj s využitím typové kontroly a možností OOP programování, díky čemuž je kód snadno pochopitelný a přehledný.

Pro správu závislostí a sestavení projektu byl využit nástroj **npm**, který se velmi často využívá ve spojení s klientskými aplikacemi. Pomocí tohoto nástroje byl založen projekt, nadefinovány potřebné závislosti pro vývoj a především skripty, které slouží k sestavení projektu a vygenerování dokumentace kódu. Zprovoznění projektu je poté velmi jednoduché, stačí mít pouze nainstalovaný nástroj **npm** a následně sestavit projekt jedním příkazem.

2.1 Struktura aplikace

Popis jednotlivých tříd a jejich využití je podrobně rozepsáno v přiložené dokumentaci v adresáři **docs/**.

2.2 Struktura úložiště

Aby bylo možné efektivně pracovat s dotazy uloženými v lokálním úložišti prohlížeče (localStorage), byla vytvořena jednoduchá interní struktura viz zdrojový kód níže.

```
{
  queries: [
    {
      name: string,
      queryString: Array<string>,
      currentVersion: number
    }
  ]
}
```

Každý dotazy typu `Query` je uložen uvnitř pole `queries`. Aby bylo možné dotazy verzovat, každý dotaz obsahuje pole všech verzí dotazu `queryString` a ukazatel definující aktuální verzi `currentVersion`.

3 Uživatelská dokumentace

Examples

SELECT ?s ?p ?o ((?s as ?i) WHERE { ?s ?p ?o } LIMIT 10
PREFIX foaf <http://xmlns.com/foaf/0.1> SELECT ?name WHERE { ?person foaf:name ?name . }
SELECT DISTINCT ?concept WHERE { ?s a ?concept . } LIMIT 50

Send SPARQL query

SELECT ?s ?p ?o ((?s as ?i) WHERE { ?s ?p ?o } LIMIT 10

Odeslat na Fuseki endpoint Odeslat na Virtuoso endpoint

Results

s	p	o	i
http://www.openlinksw.com/urdf-data-formats/default-id	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.openlinksw.com/schemas/urdf#QuadMapFormat	5
http://www.openlinksw.com/urdf-data-formats/default-id-nullable	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.openlinksw.com/schemas/urdf#QuadMapFormat	5
http://www.openlinksw.com/urdf-data-formats/default-id-nonblank	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.openlinksw.com/schemas/urdf#QuadMapFormat	5
http://www.openlinksw.com/urdf-data-formats/default-id-nonblank-nullable	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.openlinksw.com/schemas/urdf#QuadMapFormat	5
http://www.openlinksw.com/urdf-data-formats/default	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.openlinksw.com/schemas/urdf#QuadMapFormat	5
http://www.openlinksw.com/urdf-data-formats/default-nullable	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.openlinksw.com/schemas/urdf#QuadMapFormat	5

Obrázek 1: Ukázka UI aplikace.

3.1 Sestavení

Nástroj obsahuje již přeložené zdrojové kódy a vygenerovanou dokumentaci, ale v případě potřeby (např. při změně zdrojového kódu knihovny) lze projekt sestavit pomocí nástroje `npm`, který má definováno několik skriptů:

clean - vyčištění projektu od přeložených souborů a generované dokumentace

tsc - přeložení knihovny

minify - minifikace a znečitelnění zdrojových souborů

docs - vygenerování dokumentace kódu

build - sestavení projektu

build:min - sestavení projektu + minifikace a znečitelnění zdrojových souborů

Jednotlivé skripty se spustí příkazem:

```
... \> npm run název_skriptu
```

3.2 Spuštění

Program je možné spustit pomocí souboru `index.html`, který se nachází v kořenovém adresáři projektu. Nicméně je nejlepší si program sestavit (viz sekce 3.1) pro ujištění se, že je skutečně použita aktuální verze nástroje.

Po otevření `index.html` se zobrazí webová stránka viz obrázek 1. V horní části jsou připraveny 3 ukázkové dotazy, které je možné spustit. Spuštění se provede kliknutím na jeden z dotazů a následným odesláním dotazu pomocí jednoho z tlačítek pod formulářem. Pro Fuseki endpoint je potřeba mít lokálně spuštěný daný server. V případě endpointu Virtuoso se jedná o server třetí strany, který by měl být dostupný, takže odpověď na daný dotaz by se měla zobrazit během okamžiku ve spodní sekci **Results** jako je tomu v ukázce na výše zmíněném obrázku.

V případě, že by server nefungoval nebo se odpověď nezobrazila, pravděpodobně mohla být chyba v dotazu. V tomto případě bude chyba vypsána v konzoli.

Může dojít k situaci kdy nástroj nebude fungovat, typicky tento problém nastával v prohlížeči IE11. Důvodem byl problém s lokálním úložištěm, které nefungovalo správně při spuštění stránky přímo z projektu. Pro správnou funkčnost bylo potřeba umístit `index.html` s adresářem `dist/` na lokální server (např. Apache) a tam stránku otevřít.

3.3 Práce s knihovnou

Popis práce s knihovnou a klíčovými třídami je podrobně popsán v souboru `README.md`.

3.3.1 Modifikace knihovny

Pokud by bylo potřeba knihovnu nějakým způsobem upravit, všechny změny je potřeba provést v souboru `src/sparqljs.ts` a následně nástroj sestavit viz sekce 3.1.

Pokud by tedy například bylo potřeba upravit objekt `Query`, stačí nalézt příslušnou třídu a tu modifikovat. V případě této třídy bude ale potřeba také upravit formát JSON objektu, který přijímá konstruktor třídy `QueryList`. Konstruktor této třídy totiž zpracovává předaný JSON objekt z lokálního úložiště a převádí jej na interní struktury a pravděpodobně by došlo při sestavení k chybě kvůli neodpovídající struktuře objektu na vstupu do konstruktoru.

V každém případě je doporučeno nejprve nastudovat přiloženou dokumentaci, kterou lze otevřít v prohlížeči souborem `docs/index.html`. Pokud dokumentace neexistuje, je možné ji vygenerovat viz sekce 3.1.

3.3.2 Nasazení

Pro nasazení aplikace do provozního prostředí je potřeba (v případě ukázkové aplikace) zkopírovat `index.html` s adresářem `dist/` na daný server. Pro tento případ je možné sestavit nástroj pomocí skriptu `build:min`, který sestaví projekt, minifikuje kód a znečitelní zdrojový kód knihovny.

4 Závěr

Nástroj splňuje požadavky zadání a je kompatibilní se všemi prohlížeči. Při implementaci jsem nenarazil na žádný problém. Problémy nastaly až při sestavení projektu, kdy nebylo jednoduché nastavit konfigurace tak, aby se generoval jeden soběstačný JS soubor.