

Универзитет „Св.Кирил и Методиј”- Скопје

Факултет за информатички науки и компјутерско инженерство

ФИНКИ



Проектна задача по Веб Базирани Системи

Наоѓање рецепти според на состојките што веќе ги имате дома користејќи Linked Data

Ментор:
Милош Јовановиќ

Кандидат:
Данило Поповиќ

Скопје,
Март 2019

1. Вовед

Колку пати до сега сте сакале да зготвите храна според некој рецепт, но сте требале да одите во продавница со листа од состојќи што се потребни за рецептот, па да ги барате едно по едно додека да на ги најдете сите, па да се вратите дома и да го изгубите цел ден во набавување состојќи и припремање храна. Идејата позади овој труд е да овозможи начин да го избегнете оној дел на барање на состојќи според рецептот, туку рецептот да го пронајдете според состојките кои веќе ги имате во својот дом. Како обратен пристап на традиционалниот начин на готвење според рецепт.

2. Технологии

Во овој труд ќе користам JSON-LD[1] како формат за податоците, и бидејќи лична омилена технологија ми е .NET, пронајдов рамка за манипулирање со Linked Data наречена dotnetrdf[2] и ќе се потрудам да ја употребам.

3. Податоци

Ќе започнам со податоците и нивната структура, кои се потребни за создавање множеството рецепти, кое подоцна ќе може да се пребарува за да се оствари целта на идејата. Поради неуспех да пронајдам постоечко множество од рецепти во Linked Data структура, за овој проект ќе се обидам да создадам свое. Најпрвин успеав да најдам множество рецепти во JSON формат, како регуларно објектно-ориентирано множество наречено “Epicurious – Recipes with rating and nutrition”[3]. Битното во ова множество е што секој рецепт содржи име на рецептот, краток опис, состојќи и чекори за припремање на рецептот – се што ми е потребно за мојата идеја. Следно на schema.org успеав да најдам онтологија наречена “Recipe”[4], која ги содржи сите атрибути кои ми се потребни. И за крај целта е да најдам начин постоечките податоци од регуларното множество да ги претворам во JSON-LD форматот, користејќи ја онтологијата за рецепти. Започнав со JSON-LD урнек за еден рецепт, така што ако успешно форматирам еден рецепт, ќе можам да направам таква низа, и да го оформам множеството кое ми е потребно.

```
{
  "@context": {
    "name": "https://schema.org/#name",
    "ingredient": "https://schema.org/#ingredients",
    "recipeInstructions": "https://schema.org/#recipeInstructions",
    "description": "https://schema.org/#description",
    "type": "http://www.w3.org/1999/02/22-rdf-syntax-ns#/type",
    "recipeIngredient": "https://schema.org/#recipeIngredient"
  },
  "type": "Recipe",
  "description": "{{Description}}",
  "recipeIngredient": [{{Ingredients}}],
  "name": "{{Name}}",
  "recipeInstructions": [{{Instructions}}]
}
```

Слика од урнекот за еден рецепт

Во урнекот обколени се вредностите на атрибутите со големи загради, за подоцна преку код да можам да ги заменувам со вредности од постоечкото множество и да ги запишам во новиот формат. Следно, создадов класа `Recipe` во апликацијата за да го де-серијализирам постоечкото множество од рецепти. Успеав да го доведам до листа од “Recipe” во код. Остана уште рецепт по рецепт листата да ја пројдам, заменувајќи ги вредностите на параметрите за секој рецепт, додавајќи го резултатот во низа од објекти структурирани во JSON-LD формат. Кодот успешно се изврши и после 30 долги минути, завршив со 20.000 рецепти во посакуваниот формат. За крај добиениата листа ја запишав во датотека. Таа зафаќа околу 30MB и кога ќе се вчита во меморија во кодот користејќи “dotnetrdf”, целосно завземаат околу 1GB рам меморија. Целиот код за оваа процедура е достапен на GitHub[5].

4. Добивање на рецепти и нивни инструкции

Последниот чекор е да направам SPARQL прашање каде што на влез ќе прима листа од состојќи, и според податоците во моето множество, да ги врати сите рецепти што може да се направат од тие состојќи и инструкции за истите. Тука наидов на голем проблем бидејќи во мојот модел состојките не се објект туку се “string literal”, сите се во различен формат и не можам конкретно да дојдам до името на состојката на веродостоен начин. Идејата е да проверам дали состојката од множеството односно “literal”-от содржи збор што е дел од внесените состојќи(пр. “1 tea spoon of salt”, содржи “salt”), и на тој начин, доколку сите состојќи на рецептот ги има во внесената листа состојќи, тогаш тој рецепт може да се направи, и треба да се прикаже на корисничкиот интерфејс. “dotnetrdf” нуди

начин на формирање SPARQL прашања преку методи во код, пример за земање на рецепти што содржат една внесена состојка, кодот би бил:

```
var queryBuilder =
    QueryBuilder
    .Select(new string[] { x })
    .Where(
        (triplePatternBuilder) =>
        {
            triplePatternBuilder
                .Subject(x)
                .PredicateUri(new Uri(@"https://schema.org/#recipeIngredient"))
                .Object(y);
        })
    .Filter((builder) => builder.Regex(builder.Variable("y"), input, "i"));
```

Но за да не навлаглегуваме во специфични имплементации, ќе го задржам форматот SPARQL, во кои истото прашање би изгледало вака.

```
var query = @"SELECT ?x WHERE { ?x <https://schema.org/#recipeIngredient> ?object
FILTER (REGEX(STR(?object), 'lettuce', 'i') )}";
```

И уште да се изврши прашањето на веќе вчитаниот граф.

```
var results = graph.ExecuteQuery(query);

//Print out the Results
SparqlResultSet g = (SparqlResultSet)results;
foreach (SparqlResult t in g)
{
    Console.WriteLine(t.ToString());
}
Console.ReadKey();
```

```
Please enter an ingridient:
lettuce
?x = _:b0
?x = _:b1
?x = _:b5
?x = _:b8
?x = _:b148
?x = _:b243
?x = ...
```

На крај успешно успеав да ги добијам сите рецепти кои содржат “lettuce”.

Кодот заедно со дата-сетот и темплејтот се поставени на мојот Github[5].

- [1] - <https://json-ld.org/> - Официјалната страна на JSON-LD
- [2] - <https://www.dotnetrdf.org/> - Официјалната страна на dotnetrdf
- [3] - <https://www.kaggle.com/hugodarwood/epirecipes> - Линк од каде што го превземав иницијалното множество од рецепти
- [4] - <https://schema.org/Recipe> - Онтологија за рецепти
- [5] - <https://github.com/danilopopovikj/WBS-Final-Project> - Линк од изворниот код на проектот