

## Домашно бр. 5 Creating Linked Data

### А) Домашна задача

#### 1. Што е FOAF?

- Онтологија за креирање на web profile со некои лични информации и информации за познанства.

#### 2. Наведете дел од класите и својствата дефинирани во FOAF онтологијата / вокабуларот.

FOAF Core	Social Web
<ul style="list-style-type: none"><li>• <a href="#">Agent</a></li><li>• <a href="#">Person</a></li><li>• <a href="#">name</a></li><li>• <a href="#">title</a></li><li>• <a href="#">img</a></li><li>• <a href="#">depiction</a> (<a href="#">depicts</a>)</li><li>• <a href="#">familyName</a></li><li>• <a href="#">givenName</a></li><li>• <a href="#">knows</a></li><li>• <a href="#">based_near</a></li><li>• <a href="#">age</a></li><li>• <a href="#">made</a> (<a href="#">maker</a>)</li><li>• <a href="#">primaryTopic</a> (<a href="#">primaryTopicOf</a>)</li><li>• <a href="#">Project</a></li><li>• <a href="#">Organization</a></li><li>• <a href="#">Group</a></li><li>• <a href="#">member</a></li><li>• <a href="#">Document</a></li><li>• <a href="#">Image</a></li></ul>	<ul style="list-style-type: none"><li>• <a href="#">nick</a></li><li>• <a href="#">mbox</a></li><li>• <a href="#">homepage</a></li><li>• <a href="#">weblog</a></li><li>• <a href="#">openid</a></li><li>• <a href="#">jabberID</a></li><li>• <a href="#">mbox_sha1sum</a></li><li>• <a href="#">interest</a></li><li>• <a href="#">topic_interest</a></li><li>• <a href="#">topic</a> (<a href="#">page</a>)</li><li>• <a href="#">workplaceHomepage</a></li><li>• <a href="#">workInfoHomepage</a></li><li>• <a href="#">schoolHomepage</a></li><li>• <a href="#">publications</a></li><li>• <a href="#">currentProject</a></li><li>• <a href="#">pastProject</a></li><li>• <a href="#">account</a></li><li>• <a href="#">OnlineAccount</a></li><li>• <a href="#">accountName</a></li><li>• <a href="#">accountServiceHomepage</a></li><li>• <a href="#">PersonalProfileDocument</a></li><li>• <a href="#">tipjar</a></li><li>• <a href="#">sha1</a></li><li>• <a href="#">thumbnail</a></li><li>• <a href="#">logo</a></li></ul>

#### 3. На кои начини можеме да креираме сопствен FOAF профил?

- Рачно, со едноставно креирање на rdf (.ttl, .xml, ...) документ или преку користење на некоја автоматизирана алатка.

#### 4. Наведете неколку онтологии / вокабулари кои можеме да ги искористиме за проширување на сопствениот FOAF профил.

- VCARD, Relationship, сопствена онтологија

#### 5. Како може да се објави еден FOAF профил?

- Линкот до документот треба да биде јавно достапен, токму. Тоа може да се направи и преку користење на Dropbox или Google Drive.

## Б) Практична задача

### I. Креирање сопствен FOAF профил

Со користење на [FOAF-a-Matic](#), креирајте свој сопствен FOAF профил, кој ќе содржи основни (јавни) информации за вас, листа од ваши пријатели, линкови кон ваши профили на социјални мрежи и дополнителни податоци.

За да подоцна можеме да го објавите вашиот FOAF профил, потребно е да имате испланирано веб локација каде ќе можете да го поставите. Една таква опција е *foaf.ttl* датотеката да ја поставите на вашиот [Dropbox](#) акаунт и да генерирате јавен линк до неа. Алтернативно, можете датотеката да ја поставите на вашиот [GitHub](#) акаунт, [GitLab](#) акаунт,

1. Отворете го [FOAF-a-Matic](#) во веб прелистувач. Еден од основните параметри кои треба да ги поставите во генераторот е базичното URI (Base URI) за ентитетите кои ќе бидат дел од вашиот FOAF профил. За таа цел, ќе биде потребно однапред да знаеме на која веб локација ќе ја поставиме нашата *foaf.ttl* датотека.

2. Во некој именикот на вашиот Dropbox акаунт креирајте празна текстуална датотека, која ќе ја снимите под името *foaf.ttl*. Датотеката креирајте ја локално на компјутерот во лабораторија, а потоа поставете ја преку Upload опцијата на веб интерфејсот на Dropbox веб страната.

Алтернатива: Користете GitHub или GitLab, каде ќе го прикачите документот во ваш јавен проект.

3. Откако ќе ја поставите, селектирајте ја *foaf.ttl* датотеката. На веб интерфејсот ќе се појави мени со опции, меѓу кои и “Share”. Кликнете ја оваа опција за да го добиете јавно достапното URL до датотеката.

Напомена: Генерираното URL променете го така што во крајниот дел од него ќе го промените “?dl=0” во “?dl=1”. Со ова линкот ќе покажува директно до датотеката, наместо во Dropbox веб страна која ја содржи датотеката.

Алтернатива: Доколку користите GitHub или GitLab, најдете го URL-то кое дава пристап до raw содржината на документот. Тоа треба да е јавно URL кое дава пристап директно до содржината на документот, наместо до веб страната на сервисот каде е хостиран документот.

4. URL-то од претходниот чекор ќе го искористите малку подоцна, па затоа зачувајте го. Тоа е локацијата на која ќе го објавите и вашиот финализиран FOAF профил.

5. Пополнете ги основните податоци за вас на страната на FOAF-a-Matic.

Напомена: За email вредностите користете го “mailto:” префиксот, согласно дефинициите на [mbox](#) и [mbox\\_sha1sum](#) релациите. Пример: “mailto:user@domain.com”, наместо само “user@domain.com”.

6. Додадете информации за ваши пријатели: притоа, додадете записи само за колеги од групата. Ова е потребно за целите од вториот дел на вежбата, односно FOAF профилите на вашите пријатели мора реално да постојат.

Во листата додадете информации за 2-3 (или повеќе) колеги кои работат со вас во групата, притоа користејќи ги е-mail адресите и URI-ата на нивните FOAF профили кои тие ги користат во нивните FOAF профили во рамките на вежбата.

Ова ќе овозможи понатамошно поврзување на вашите FOAF профили, без разлика што се објавени во различни датотеки и на различни Веб локации.

Напомена: За email вредностите користете го “mailto:” префиксот.

- Откако ќе завршите, во делот “Generate Results” селектирајте ја опцијата за заштита на мејл адресите од спам.
- Со клик на “FOAF me!” ќе добиете RDF/XML содржина, која ќе треба рачно да ја ископирате и префрлите во текст-едитор. Со цел да го дефинирате основното URI (base URI) за ентитетите креирани од FOAF-a-Matic, потребно е во рамките на првиот XML елемент (`<rdf:RDF> ... </rdf:RDF>`), помеѓу постоечките namespaces, да додадете нов `xml:base` атрибут. Во него, треба да го искористите URL-то на вашиот `foaf.ttl` документ од точка 4:

```
xml:base="url-na-vasiot-foaf.ttl-dokument#"
```

(пример: `xml:base="http://mjovanovik.com/foaf.ttl#"`)

Напомена: Овој атрибут има формат `xml:base` и е различен од останатите `xmlns:prefix` атрибути од истиот елемент.

- Модифицираниот RDF/XML профил треба да го трансформирате во Turtle формат. За таа цел, можете да го искористите [RDF Translator](#) (или друг веб-базиран преведувач на RDF синтакса). Трансформирајте го RDF/XML профилот во Turtle (N3) формат.
- Креираната Turtle содржина на вашиот FOAF профил сместете ја во постоечкиот `foaf.ttl` документ од точка 2 и 3. Притоа, искористете ја истата датотека, не ја заменувајте со нова. Во спротивно, URL-то генерирано во точка 3 и 4 нема да биде повеќе валидно.
- Преку нов прозорец / таб во веб прелистувачот, пристапете во јавното URL на вашата `foaf.ttl` датотека, добиено во чекор 4.

Дали добивте пристап до Turtle содржината на вашиот FOAF профил? Доколку не, проверете кој е проблемот и отстранете го.

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

<https://raw.githubusercontent.com/SlamkovDejan/web-based-systems-homework/master/src/main/java/lab5/data/foaf.ttl> a foaf:PersonalProfileDocument ;
foaf:maker <https://raw.githubusercontent.com/SlamkovDejan/web-based-systems-homework/master/src/main/java/lab5/data/foaf.ttl#me> ;
foaf:primaryTopic <https://raw.githubusercontent.com/SlamkovDejan/web-based-systems-homework/master/src/main/java/lab5/data/foaf.ttl#me> .

<https://raw.githubusercontent.com/SlamkovDejan/web-based-systems-homework/master/src/main/java/lab5/data/foaf.ttl#me> a foaf:Person ;
foaf:depiction <https://media-exp1.licdn.com/dms/image/C4E03AQ63DeA9Q1XQeQ/profile-displayphoto-shrink_400_400/0/1607877134290?e=1613606400&v=beta&t=B7LdE6XXaXCouf108E4210> ;
foaf:family_name "Slamkov" ;
foaf:givenname "Dejan" ;
foaf:homepage <https://www.linkedin.com/in/dejan-slamkov/> ;
foaf:knows [ a foaf:Person ;
rdfs:seeAlso <https://raw.githubusercontent.com/StojanovVenko/WBS_FOAF_profile/main/foaf.ttl> ;
foaf:name "Venko Stojanov" ] ;
foaf:mbox_sha1sum "992a6dd100e3872f8082c524219b600f8fa4de87" ;
foaf:name "Dejan Slamkov" ;
foaf:nick "Deki" ;
foaf:phone <tel:73222222> ;
foaf:schoolHomepage <https://finki.ukim.mk/> ;
foaf:title "Student" .
```

## II. Извлекување поврзани податоци за пријатели стартувајќи од вашиот FOAF профил

Во овој дел од вежбата ќе работиме со податоците за вашите пријатели, преку обработка на RDF содржината на вашиот FOAF профил.

Преку Java + Jena апликација ќе го вчитате вашиот FOAF профил и URI-ата на вашите пријатели кое се наоѓаат во него. Овие URI вредности ќе ги искористите за вчитување на нивните FOAF профили и комбинирање на сите во еден Jena Model. Од овој модел, ќе може да селектираме одредени сумарни податоци, кои можат да се искористат како основа за развој на апликации.

12. Креирајте нов Java проект во NetBeans. Вклучете ги во проектот сите .jar библиотеки од lib фолдерот од Jena. Jena преземете ја од [Jena сајтор](#).

13. Во main() методот на главната класа од проектот, креирајте основен Jena model. Во овој модел вчитајте го вашиот FOAF профил преку неговото веб URL.

Внимавајте вториот параметар на model.read() да го поставите за вчитување на Turtle содржина.

14. Од моделот, во кој се наоѓа вашиот FOAF профил, селектирајте ги URI вредностите на сите лица кои ги познавате, т.е. сите со кои сте во *foaf:knows* релација. Отпечатете ја листата на овие вредности.

15. Извршете ја апликацијата (или само Java класата). Проверете дали напишаниот код ги наоѓа и печати сите URI вредности на вашите пријатели од FOAF профилот. Доколку има грешки, вратете се назад и коригирајте ги.

```
public class Task1 {

    private static final String myFoafDocUri = "https://raw.githubusercontent.com/SlamkovDejan/web-based-systems-homework/master/src/foaf/foaf.ttl";

    public static void main(String[] args) {

        Model myFoaf = ModelFactory.createDefaultModel();
        myFoaf.read(myFoafDocUri, lang: "TURTLE");

        Resource me = myFoaf.getResource(String.format("%s#me", myFoafDocUri));

        StmtIterator iter = me.listProperties(FOAF.knows);
        while (iter.hasNext()){
            Resource friend = iter.nextStatement().getResource();
            String friendFoafDocUri = friend.getProperty(RDFS.seeAlso).getObject().toString();
            System.out.println(friendFoafDocUri);
        }
    }
}
```

16. Направете промена на кодот, со тоа што наместо да ги печатите, ќе ги искористите URI вредностите на вашите пријатели за да ги вчитате во апликацијата.

**Напомена:** Поради непознатиот број пријатели во FOAF профилот, за вчитување може да искористите два модели: еден привремен во кој итеративно ќе ги вчитувате FOAF профилите на вашите пријатели и еден сумарен во кој после секое вчитување ќе го додавате привремениот, преку користење на *унија на модели*.

17. Направете уште една унија на модели: моделот на вашиот FOAF профил и сумарниот модел со FOAF профилите на вашите пријатели.
18. Отпечатете го моделот од претходната точка за да ја проверите неговата валидност. Извршете ја апликацијата.
19. Искористете го моделот од претходната точка за да ги селектирате и отпечатите:

- a. Имињата и презимињата на сите лица, односно сите ваши пријатели;
- b. Мејл адресите на сите ваши пријатели;
- c. Линковите кон социјални мрежи на сите ваши пријатели;
- d. Сликите на сите ваши пријатели;

20. Извршете ја апликацијата и проверете дали отпечатените вредности одговараат на вредностите кои знаете дека постојат кај секој од FOAF профилите на вашите пријатели.

```
public class Task1 {  
  
    private static final String myFoafDocUri = "https://raw.githubusercontent.com/SlamkovDejan/web-based-systems-homework/master/src/main/java/lab5/data/foaf.ttl";  
  
    public static void main(String[] args) {  
  
        Model myFoaf = ModelFactory.createDefaultModel();  
        myFoaf.read(myFoafDocUri, "lang: Turtle");  
  
        Resource me = myFoaf.getResource(String.format("%s#me", myFoafDocUri));  
  
        List<String> friends = new ArrayList<>();  
        StmtIterator iter = me.listProperties(FOAF.knows);  
        while (iter.hasNext()){  
            Resource friend = iter.nextStatement().getResource();  
            String friendFoafDocUri = friend.getProperty(RDFS.seeAlso).getObject().toString();  
  
            Model friendModel = ModelFactory.createDefaultModel();  
            friendModel.read(friendFoafDocUri, "lang: Turtle");  
            myFoaf = myFoaf.union(friendModel);  
  
            friends.add(friendFoafDocUri);  
        }  
  
        for(String friendUri : friends){  
            Resource friend = myFoaf.getResource(friendUri.concat("#me"));  
            String name = friend.getProperty(FOAF.givenname).getObject().toString();  
            String surname = friend.getProperty(FOAF.family_name).getObject().toString();  
            String email = friend.getProperty(FOAF.mbox_sha1sum).getObject().toString();  
            String imgURL = friend.getProperty(FOAF.depiction).getObject().toString();  
            String homepage = friend.getProperty(FOAF.homepage).getObject().toString();  
            System.out.printf("%s %s %s %s %s\n", friendUri, name, surname, email, homepage, imgURL);  
        }  
    }  
}
```