

ACCESS CONTROL FOR CLIENT-SIDE GRAPH-BASED QUERIES

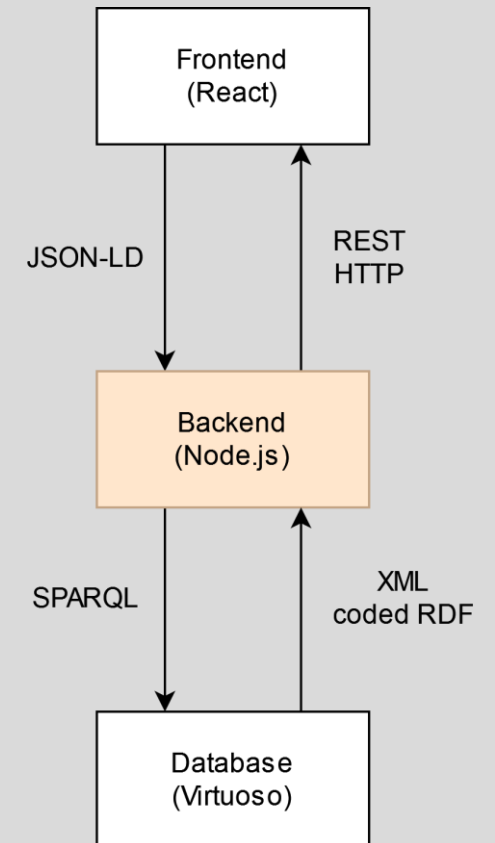
Študent: Miroslav Baluch

Školiteľ: Ján Klúka

Rok: 2022/23

SÚČASNÝ STAV

- Webová aplikácia - podporný kurzový systém Courses 3
- Frontend (React) + Backend (Node.js)
- Bežia nad DB (RDF triple store – Virtuoso server)
- Grafová DB umožňuje ukladať dáta flexibilne a prepojene
- Prístup k dátam riešený cez RESP API
 - Dopyty prekladané do jazyka SPARQL nad DB
 - Jednoduchý prístup pre FE programátorov
- Systém práv na základe ich vzťahov k iným objektom
- FE + BE – vznikli ako predchádzajúce diplomové práce



MOTIVÁCIA

- BE neumožňuje vyberať atribúty entít, ktoré sú podstatné pre FE programátora
- Obmedzená hĺbka prepájania entít
- Nechceme použiť priamo SPARQL:
 - FE programátori nie sú zvyknutí
 - Nevieme zabezpečiť dostatočnú kontrolu

```
GET /data/user/Mn0aN&
```

```
GET /data/user/Mn0aN?_join=memberOf //depth one, memberOf = TeamInstance  
GET /data/team/X&aEn0 //depth two - from TeamInstance we get Team identifier  
//and then we can get the name of the Team
```

CIELE PRÁCE

Expresívnejší a flexibilnejší prístup k dátam ako REST API

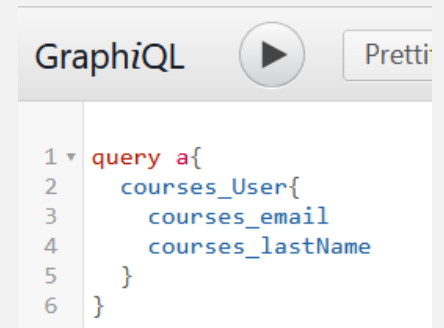
Zachovať kontrolu prístupových práv cez vzťahy entít

Rozšíriť kontrolu práv na prístup k atribútom entít

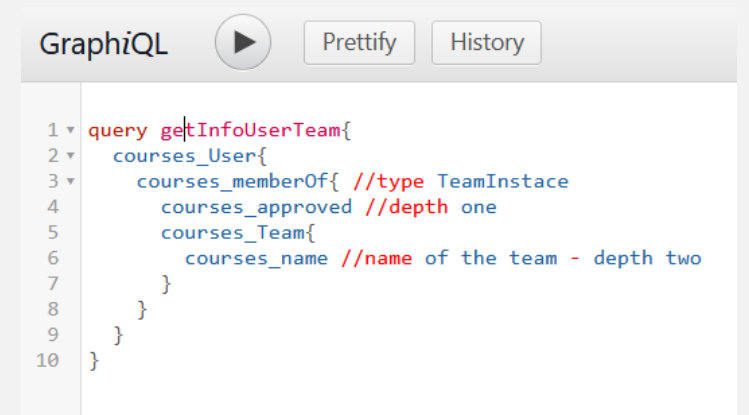
Zachovať výstupný formát (JSON-LD)

NAVRHOVANÉ RIEŠENIE

- Nahradenie REST za GraphQL
 - Konkrétna implementácia UltraGraphQL
- Vytvoriť úplnú štandardnú reprezentáciu schémy dát v RDF
- Overiť či je funkcionálna UltraGraphQL postačujúca pre FE
 - Pridať chýbajúcu funkcionálnu alebo upraviť existujúcu
- Implementovať pre FE prechod z REST na GraphQL
- Vytvoriť mechanizmy na autorizáciu



```
GraphQL ▶ Prettify  
1 query a{  
2   courses_User{  
3     courses_email  
4     courses_lastName  
5   }  
6 }
```



```
GraphQL ▶ Prettify History  
1 query getInfoUserTeam{  
2   courses_User{  
3     courses_memberOf{ //type TeamInstance  
4       courses_approved //depth one  
5       courses_Team{  
6         courses_name //name of the team - depth two  
7       }  
8     }  
9   }  
10 }
```

DOTERAJŠIA PRÁCA

- Pochopenie a práca s existujúcim kódom
 - Riešenie chýb, zabezpečenie funkčnosti na novej databáze
 - Refaktorizácia kódu, aktualizácia knižníc
 - Asistencia FE developerom - pridaná podpora nadtried/podtried pre modely
- Štúdium kódu a nastavenie UltraGraphQL
 - Vytvorenie konfigurácie
- Štandardizácia schémy databázy
- Riešenie problémov knižnice UltraGraphQL
 - Oprava chýb v kóde
 - Doplnenie UPDATE metódy

The screenshot shows the MATFYZ.sk interface for an assignment titled 'Projekt2'. The page is divided into several sections:

- Navigation:** Includes 'Submissions', 'Reviews', 'Code review (initial)', and 'Code review' tabs.
- Assignment Details:** Shows the student's name 'Filip' and the assignment title 'Prioritný rad'. The description asks to define a **PriorRad** class with methods **Insert** and **DeleteMax**, and to implement a priority queue using a heap.
- Documents:** A section for uploading or viewing documents related to the assignment.
- Submissions:** A table comparing two submission attempts:

| Submission Type | Deadline | Action |
|---------------------|------------------|----------|
| Improved submission | 23:59 16.04.2022 | Download |
| Initial submission | 22:00 06.04.2022 | Download |
- Score:** A progress bar showing a score of 10/10.
- Comment:** A text input field for providing feedback on the submission, with a 'Save score' button below it.

ŠTANDARDIZÁCIA SCHÉMY DATABÁZY

- Dáta vo forme JS objektov
- Zabezpečenie mapovanie JS objektov na RDF triples.
- Vpravo hore je možné vidieť časť insert query, ktorá je vytvorená a pridáva dáta do konkrétneho grafu (ktorý sa dá špecifikovať).
- Vpravo dole je zobrazená časť kódu z exportera – mapovanie atribútov modelu na RDF triples.

```
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>
prefix dcterms: <http://purl.org/dc/terms/>
prefix op: <local:operation:>
prefix courses: <http://www.courses.matfyz.sk/ontology#>
prefix coursesData: <http://www.courses.matfyz.sk/data>
prefix schema: <http://schema.org/>
```

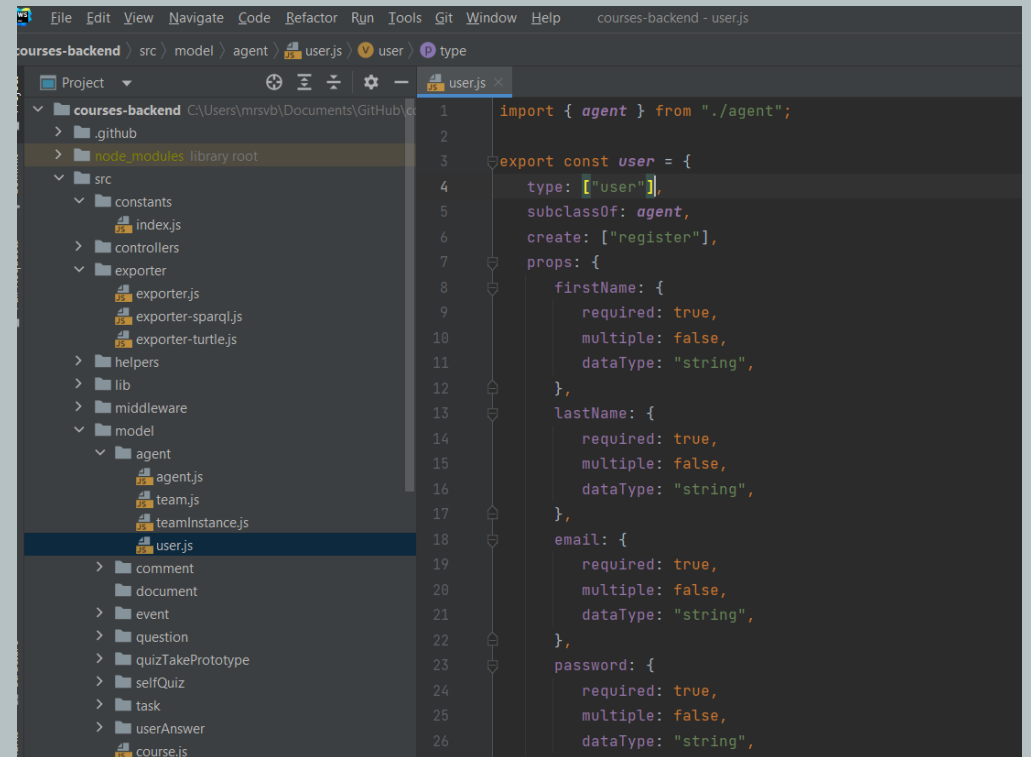
```
INSERT IN GRAPH <http://www.courses.matfyz.sk/> {
```

```
<http://www.courses.matfyz.sk/ontology#AGENT> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/2000/01/rdf-schema#Class> .
<http://www.courses.matfyz.sk/ontology#User> <http://www.w3.org/2000/01/rdf-schema#subClassOf> <http://www.courses.matfyz.sk/ontology#AGENT> .
<http://www.courses.matfyz.sk/ontology#Team> <http://www.w3.org/2000/01/rdf-schema#subClassOf> <http://www.courses.matfyz.sk/ontology#AGENT> .
<http://www.courses.matfyz.sk/ontology#Avatar> <http://schema.org/domainIncludes> <http://www.courses.matfyz.sk/ontology#AGENT> .
<http://www.courses.matfyz.sk/ontology#Avatar> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/2002/07/owl#DatatypeProperty> .
<http://www.courses.matfyz.sk/ontology#ASSIGNMENT> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/2000/01/rdf-schema#Class> .
<http://www.courses.matfyz.sk/ontology#ASSIGNMENT> <http://www.w3.org/2000/01/rdf-schema#subClassOf> <http://www.courses.matfyz.sk/ontology#TASK> .
<http://www.courses.matfyz.sk/ontology#shortDescription> <http://schema.org/domainIncludes> <http://www.courses.matfyz.sk/ontology#ASSIGNMENT> .
```

```
exporter.js
51
52 getCommonOntology() {
53     let ontologyArray = [];
54
55     const properties = new Set();
56
57     Object.values(models).map((model) => {
58         let className;
59         if (model.type) {
60             className = this.firstLetterToUppercase(model.type);
61             ontologyArray.push(this.getTriple(PREFIXES.courses, className, PREFIXES.rdf, p: "type", PREFIXES.rdfs, o: "Class")
62         }
63         if (model.subclassOf && model.subclassOf.type) {
64             ontologyArray.push(this.getTriple(PREFIXES.courses, className, PREFIXES.rdfs, p: "subClassOf", PREFIXES.courses, t
65         }
66
67         if (model.subclasses) {
68             for (let subclass of model.subclasses) {
69                 ontologyArray.push(this.getTriple(PREFIXES.courses, this.firstLetterToUppercase(subclass), PREFIXES.rdfs, p: "
70             }
71         }
72         if (model.props) {
73             Object.entries(model.props).map(([propertyName, propertyObject]) => {
74                 ontologyArray.push(this.getTriple(PREFIXES.courses, propertyName, PREFIXES.schema, p: "domainIncludes", PREFIX
75                 if (propertyObject) {
76
77                     properties.add(propertyName);
78
79                     if (propertyObject.objectClass) {
80                         ontologyArray.push(this.getTriple(PREFIXES.courses, propertyName, PREFIXES.schema, p: "rangeIncludes",
```

MOMENTÁLNE ROZPRACOVANÉ ČASTI

- Prispôsobenie vytvárania UltraGraphQL schémy na základe RDF schémy
- Dokončenie integrácie UltraGraphQL do Courses Systému
- Písanie textu k diplomovej práci

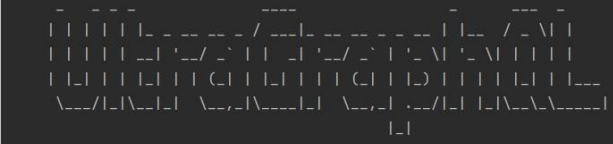


```
1 import { agent } from "./agent";
2
3 export const user = {
4   type: ["user"],
5   subclassOf: agent,
6   create: ["register"],
7   props: {
8     firstName: {
9       required: true,
10      multiple: false,
11      dataType: "string",
12    },
13     lastName: {
14       required: true,
15       multiple: false,
16       dataType: "string",
17     },
18     email: {
19       required: true,
20       multiple: false,
21       dataType: "string",
22     },
23     password: {
24       required: true,
25       multiple: false,
26       dataType: "string",
```


ULTRAGRAPHQL I.

- Vyextrahovanie schémy z grafu a následne dopyt nad konkrétnymi modelmi a atribútmi.
- UltraGraphQL spracuje zo schémy v DB tie triedy a vzťahy, ktoré majú inštancie

```
type courses_User implements courses_User_Interface @service(id: "matfyz-graphql") {
  courses_nicknameTeamException: [String] @service(id: "matfyz-graphql")
  courses_email: [String] @service(id: "matfyz-graphql")
  courses_showBadges: [String] @service(id: "matfyz-graphql")
  rdf_type: [rdfs_Class] @service(id: "matfyz-graphql")
  courses_allowContact: [String] @service(id: "matfyz-graphql")
  courses_lastName: [String] @service(id: "matfyz-graphql")
  courses_password: [String] @service(id: "matfyz-graphql")
  courses_firstName: [String] @service(id: "matfyz-graphql")
  courses_publicProfile: [String] @service(id: "matfyz-graphql")
  courses_useNickname: [String] @service(id: "matfyz-graphql")
  courses_isSuperAdmin: [String] @service(id: "matfyz-graphql")
  courses_nickname: [String] @service(id: "matfyz-graphql")
  courses_description: [String] @service(id: "matfyz-graphql")
  courses_showCourses: [String] @service(id: "matfyz-graphql")
}
```



```
2022-12-04 21:56:05 INFO Application:81 - Starting controller...
HGQL service name: matfyz-graphql-config
GraphQL server started at: http://localhost:8080/graphql
GraphQL UI available at: http://localhost:8080/graphql
```

GraphQL UI interface showing a query and its result.

Query:

```
query getAllUsers {
  courses_User {
    _id
    courses_firstName
    courses_lastName
    courses_publicProfile
  }
}
```

Result:

```
{
  "extensions": {},
  "data": {
    "courses_User": [
      {
        "courses_firstName": [
          "Admin"
        ],
        "courses_publicProfile": [
          "0"
        ],
        "courses_lastName": [
          "Admin"
        ],
        "_id": "http://www.courses.matfyz.sk/data/user/s3MzY"
      }
    ],
    "@context": {
      "courses_firstName": "http://www.courses.matfyz.sk/ontology#firstName",
      "courses_User": "http://www.courses.matfyz.sk/ontology#User",
      "courses_publicProfile": "http://www.courses.matfyz.sk/ontology#publicProfile",
      "courses_lastName": "http://www.courses.matfyz.sk/ontology#lastName",
      "_type": "@type",
      "_id": "http://hypergraphql.org/query/_id"
    }
  },
  "errors": []
}
```

ULTRAGRAPHQL II.

- UltraGraphQL v základe podporuje INSERT a DELETE metódy.
- V rámci vývoja nového BE som pridal UPDATE metódu.
 - Potrebuje tiež schému

```
1 mutation insertUser {
2   insert_courses_User(
3     _id: "http://www.courses.matfyz.sk/data/user/newUserIdentifier",
4     courses_firstName: "Janko",
5     courses_lastName: "Hraško",
6     courses_email: "janko@hrasko.courses",
7     courses_showCourses: "true",
8     courses_publicProfile: "false") {
9     _id
10    courses_firstName
11    courses_lastName
12  }
13 }
14
15 mutation deleteUser {
16   delete_courses_User(_id: "http://www.courses.matfyz.sk/data/user/s3MzY") {
17     _id
18   }
19 }
20
21 mutation updateUser {
22   update_courses_User(
23     _id: "http://www.courses.matfyz.sk/data/user/s3MzY",
24     courses_nickname: "Janicko") {
25     _id
26     courses_nickname
27   }
28 }
29 }
```

Documentation Explorer

courses_user

- courses_User
- courses_User_Interface
- input_courses_User
- Query.courses_User
- Mutation.insert_courses_User
- Mutation.update_courses_User
- Mutation.delete_courses_User
- input_Query.courses_User

update_courses_User

Autogenerated mutation function for the object courses_User

TYPE

[courses_User]

ARGUMENTS

- courses_showCourses: [String]
Autogenerated mutation function for the object courses_User
- courses_password: [String]
Autogenerated mutation function for the object courses_User
- courses_publicProfile: [String]
Autogenerated mutation function for the object courses_User
- courses_firstName: [String]
Autogenerated mutation function for the object courses_User
- courses_nicknameTeamException: [String]
Autogenerated mutation function for the object courses_User
- courses_email: [String]
Autogenerated mutation function for the object courses_User

TO DO

- Autorizácia
 - Navrhnuť model
 - Implementovať ho
- Implementovať na FE prechod z REST na GraphQL (čiastočne)
- BE sa bude testovať so FE developermi
- Zapracovať feedback od developerov

LITERATÚRA

- Milan Cifra. Semantic data model for a course management system, July 2020.
- Guus Schreiber, VU University Amsterdam, Yves Raimond, and BBC. Rdf 1.1 primer, 2014.
- The GraphQL Foundation and GraphQL documentation authors. GraphQL, 2021.
- UltraGraphQL team and contributors. Ultragraphql, 2021.
- Lars Christoph Gleim, Tim Holzheim, István Koren, Stefan Decker. Automatic Bootstrapping of GraphQL Endpoints for RDF Triple Stores, 2020.
- Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data – the story so far. In Semantics Services, 2011.

ĎAKUJEM ZA
POZORNOST