

ΜΗΧΑΝΙΚΗ ΛΟΓΙΣΜΙΚΟΥ ΠΑΡΟΥΣΙΑΣΗ ΕΡΓΑΣΙΑΣ

Σχολείο Κώδικα

Θέμα: Χρήση dummy δεδομένων για τη δημιουργία ενός GraphQL server.

Προσωπικά Στοιχεία:

Ονοματεπώνυμο: Ιωάννης Μπαρακλής

AEM: 3685

Εξάμηνο: 5

E-mail: imparaki@csd.auth.gr

Project Repository: <https://github.com/JohnBaraki/GraphQL-Server-Using-Dummy-Video-Club-Data>

ΘΕΜΑ ΕΡΓΑΣΙΑΣ

Χρήση dummy δεδομένων (π.χ πανεπιστήμιο ή video club) για τη δημιουργία ενός GraphQL server.

- Δεδομένα π.χ. σε SQLite ή MongoDB

Χρήση περιβάλλοντος όπως το GraphQL και να μπορούν να γίνουν:

- 1-2 σύνθετα Queries
- 2 Mutations (π.χ. Ένα update και ένα create)

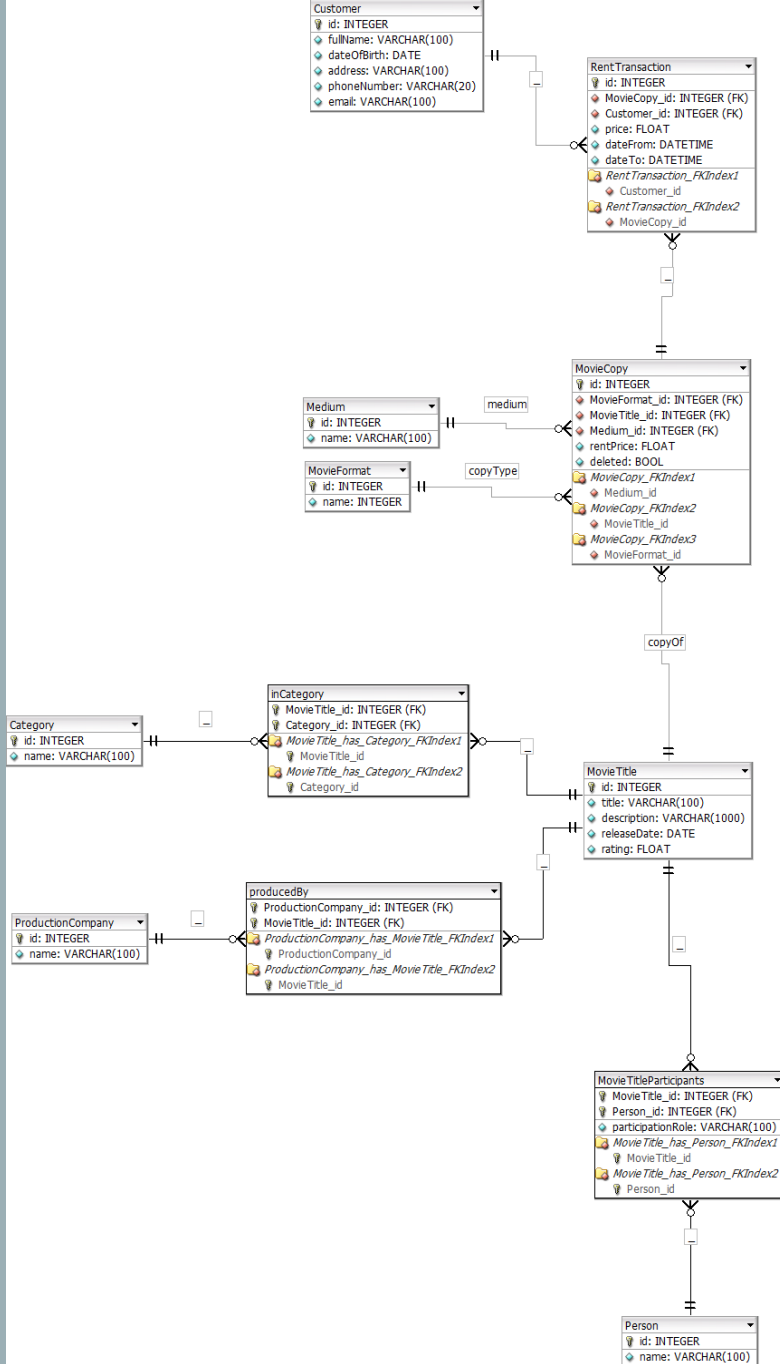
ΤΕΧΝΟΛΟΓΙΕΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ

GRAPHQL SERVER

- Java
- Maven
- GraphQL Java Kickstart
- Spring Boot
- GraphQL Java Extended Scalars

ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

- SQLite
- Dummy δεδομένα video club (δεδομένα ταινιών από IMDb και Wikipedia)
- SQLite JDBC
- Σχεδίαση βάσης με DB-Designer Fork



Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ: ΤΟ ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ

Αρχεία σχετικά με την βάση:

- database/video_club.sqlite
- database/relational_model.bmp
- database/video_club_sqlite_create_script.sql

GRAPHQL SCHEMA:

ΤΥΠΟΙ

- **MovieTitle:** Ταινία (src/main/resources/graphql/videoclub/movietitle.graphqls)
 - Μοναδικό αναγνωριστικό (id)
 - Τίτλος (title)
 - Περιγραφή (description)
 - Ημερομηνία κυκλοφορίας (releaseDate)
 - Σκηνοθέτες (directors)
 - Ηθοποιοί (actors)
 - Κατηγορίες (inCategory)
 - Εταιρία(-ες) παραγωγής (producedBy)
 - Βαθμολογία (rating)
- **MovieCopy:** Αντίγραφο ταινίας που μπορεί να νοικιάσει κάποιος πελάτης (src/main/resources/graphql/videoclub/moviecopy.graphqls)
 - Μοναδικό αναγνωριστικό (id)
 - Ταινία (movieTitle)
 - Φυσικό μέσο (medium)
 - Τύπος (copyType)
 - Τιμή ενοικίασης (rentPrice)
- **Customer:** Πελάτης (src/main/resources/graphql/videoclub/customer.graphqls)
 - Μοναδικό αναγνωριστικό (id)
 - Ονοματεπώνυμο (fullName)
 - Ημερομηνία Γέννησης (dateOfBirth)
 - Διεύθυνση (address)
 - Τηλέφωνο (phoneNumber)
 - e-mail (email)
 - Ενοικιάσεις που έκανε ο πελάτης (rentTransactions)
- **RentTransaction:** Ενοικίαση (src/main/resources/graphql/videoclub/renttransaction.graphqls)
 - Μοναδικό αναγνωριστικό (id)
 - Πελάτης (customer)
 - Αντίγραφο ταινίας (movie)
 - Τιμή ενοικίασης (price)
 - Ημερομηνία ενοικίασης (dateFrom)
 - Ημερομηνία επιστροφής (dateTo)
- **Actor** (src/main/resources/graphql/videoclub/actor.graphqls)
 - Μοναδικό αναγνωριστικό (id)
 - Όνομα (name)
 - Λίστα ταινιών στις οποίες συμμετείχε (moviesPlayedIn)

GRAPHQL SCHEMA: ΤΥΠΟΙ - ΣΥΝΕΧΕΙΑ

- **Director:** Σκηνοθέτης
(src/main/resources/graphql/videoclub/director.graphqls)
 - Μοναδικό αναγνωριστικό (id)
 - Όνομα (name)
 - Λίστα ταινιών στις οποίες συμμετείχε (moviesDirected)
- **Category:** Κατηγορία ταινιών
(src/main/resources/graphql/videoclub/category.graphqls)
 - Μοναδικό αναγνωριστικό (id)
 - Όνομα (name)
 - Λίστα ταινιών στην κατηγορία (moviesInCategory)
- **ProductionCompany:** Εταιρία Παραγωγής
(src/main/resources/graphql/videoclub/productioncompany.graphqls)
 - Μοναδικό αναγνωριστικό (id)
 - Όνομα (name)
 - Λίστα ταινιών τις οποίες παρήγαγε (moviesProduced)

GRAPHQL SCHEMA:

ΤΥΠΟΙ: ΠΕΛΑΤΗΣ - ΤΑΙΝΙΑ - ΕΝΟΙΚΙΑΣΗ

```
# Describes a customer that rents movies from the video club.
type Customer {
  # The customer's unique identifier (it is an integer).
  id: ID!

  # The customer's full name.
  # If the customer has multiple first and/or last names, only one can be registered.
  fullName: String!

  # The customer's date of birth.
  # The format of this date should be of the ISO-8601 calendar system, such as 2007-12-03.
  dateOfBirth: Date

  # The full address where the customer lives. Can include any of: Address, Number, Area,
  City, Country, Area code, etc.
  address: String!

  # The customer's phone number in any format.
  phoneNumber: String!

  # The customer's e-mail address.
  email: String

  # A list of the transactions the customer has participated in.
  rentTransactions(status: TransactionStatus): [RentTransaction!]
}
```

customer.graphqls

```
# Describes a movie in the abstract sense, as in a movie and not a (physical) movie copy. }
type MovieTitle {
  # The movie's unique identifier (it is an integer).
  id: ID!

  # The movie's title.
  title: String!

  # The movie's description. It may span many lines.
  description: String

  # The movie's release date.
  # The format of this date should be of the ISO-8601 calendar system, such as 2007-12-03.
  releaseDate: Date

  # The movie's director(s).
  directors: [Director!]

  # The movie's actor(s).
  actors: [Actor!]

  # The category(-ies) that characterize the movie.
  inCategory: [Category!]

  # The production company(-ies) that produce the movie.
  producedBy: [ProductionCompany!]

  # The movie's rating (arbitrary source).
  rating: Float
}
```

movietitle.graphqls

```
# Describes a rent transaction.
type RentTransaction {
  # The statement's unique identifier (it is an integer).
  id: ID!

  # The customer that made this transaction.
  customer: Customer!

  # The movie copy that the customer rented.
  movie: MovieCopy!

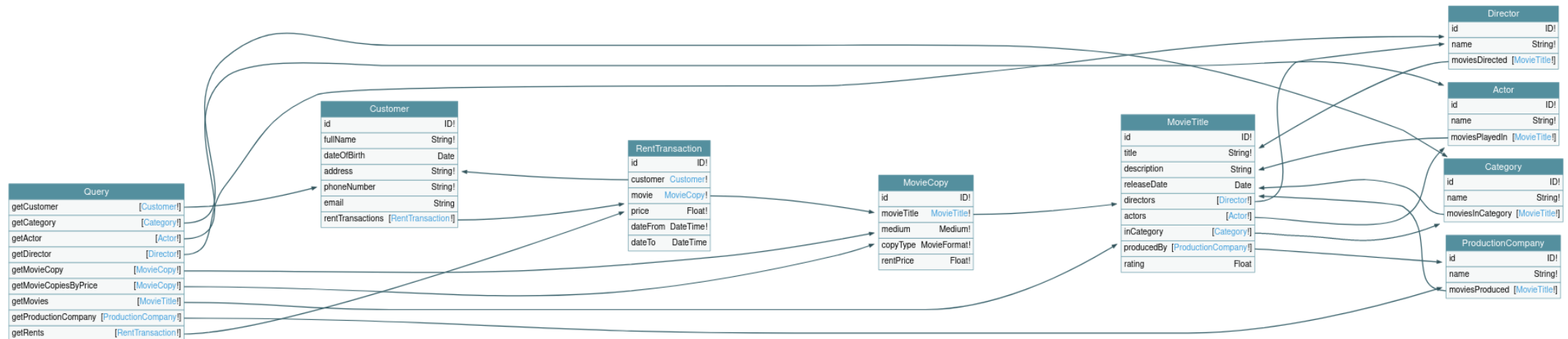
  # The price (per day) of the rent.
  price: Float!

  # The date and time when the movie copy was rented
  # This is a date-time with an offset from UTC/Greenwich in the ISO-8601 calendar system,
  such as 2007-12-03T10:15:30+01:00
  dateFrom: DateTime!

  # The date when the movie copy was returned. If null then it still is being rented.
  # This is a date-time with an offset from UTC/Greenwich in the ISO-8601 calendar system,
  such as 2007-12-03T10:15:30+01:00
  dateTo: DateTime
}
```

renttransaction.graphqls

GRAPHQL SCHEMA: ΟΠΤΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ



<ip>:8080/voyager

GRAPHQL SCHEMA: QUERIES / MUTATIONS

QUERY

- getCustomer
- getCategory
- getActor
- getDirector
- getMovieCopy
- getMovieCopiesByPrice
- getMovies
- getProductionCompany
- getRents
- Βρίσκονται
στο src/main/resources/graphql/query.graph
qls

MUTATION

- createCustomer
- registerNewRenting
- registerReturn
- registerNewMovieCopy
- deleteMovieCopy
- Βρίσκονται
στο src/main/resources/graphql/mutation.gr
aphqls

ΥΛΟΠΟΙΗΣΗ ΣΤΗΝ JAVA

ΚΛΑΣΕΙΣ ΑΝΤΙΣΤΟΙΧΙΣΗΣ ΤΥΠΩΝ

- Actor: Actor.java
- Category: Category.java
- Customer: Customer.java
- Director: Director.java
- MovieCopy: MovieCopy.java
- MovieTitle: MovieTitle.java
- ProductionCompany: ProductionCompany.java
- RentTransaction: RentTransaction.java

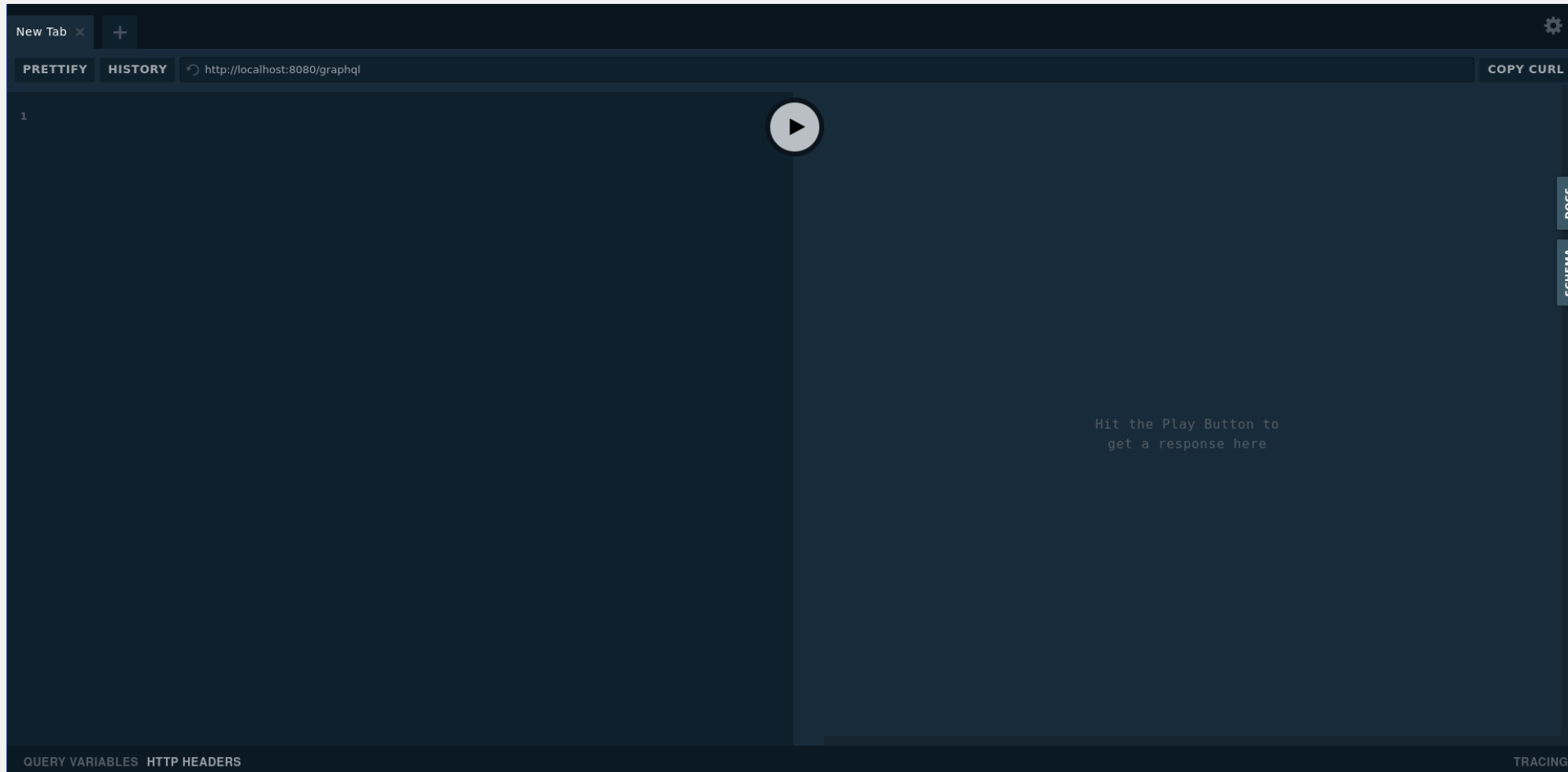
RESOLVERS

- Πακέτο fields (src/main/java/videoclub/graphql/server/resolvers/fields): Απόκτηση δεδομένων για πεδία τύπων που δεν μπορούν να αποκτηθούν εξ' ορισμού (π.χ. το πεδίο moviesProduced του τύπου ProductionCompany)
- Κλάση QueryResolver (src/main/java/videoclub/graphql/server/resolvers/query/QueryResolver.java): Συγκέντρωση δεδομένων για ικανοποίηση αιτημάτων απόκτησης δεδομένων (queries).
- Κλάση MutationResolver (src/main/java/videoclub/graphql/server/resolvers/mutation/MutationResolver.java): Συγκέντρωση δεδομένων για ικανοποίηση αιτημάτων μετάλλαξης δεδομένων (mutations).

ΕΠΙΚΟΙΝΩΝΙΑ ΜΕ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

- Χρήση JDBC
- Πακέτα datastore, DataPoints (src/main/java/videoclub/datastore, src/main/java/videoclub/datastore/DataPoints): Κλάση για κάθε τύπο με αντίστοιχες μεθόδους ανάκτησης/μετάλλαξης δεδομένων.

ΧΡΗΣΗ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ΟΠΩΣ ΤΟ GRAPHIQL ΚΑΙ ΝΑ ΜΠΟΡΟΥΝ ΝΑ ΓΙΝΟΥΝ QUERIES / MUTATIONS: GRAPHQL PLAYGROUND



<ip>:8080/playground

ΖΗΤΟΥΜΕΝΟ Ι : ΣΥΝΘΕΤΑ QUERIES

Query προς Server

```
1 # Complex query showcase: Show movie details of movies in the category "Crime".
2 query {
3   getCategory(name: "Crime") {
4     moviesInCategory {
5       title
6       description
7       directors {
8         name
9       }
10    }
11  }
12 }
```

- `src/main/resources/exampleQueries/exampleI.graphql`
- Playground tab: "Complex Query"

Απόκριση Server

```
{
  "data": {
    "getCategory": [
      {
        "moviesInCategory": [
          {
            "title": "The Godfather",
            "description": "The Godfather follows Vito Corleone, Don of the Corleone family, as he passes the mantel to his unwilling son, Michael.",
            "directors": [
              {
                "name": "Francis Ford Coppola"
              }
            ]
          },
          {
            "title": "The Dark Knight",
            "description": "When the menace known as the Joker wreaks havoc and chaos on the people of Gotham, Batman must accept one of the greatest psychological and physical tests of his ability to fight injustice.",
            "directors": [
              {
                "name": "Christopher Nolan"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

ZHTOYMENO 2 : MUTATIONS (CREATE)

Mutation προς τον Server

```
1 # Create mutation showcase: Create a new customer.
2 mutation {
3   createCustomer(
4     input: {
5       fullName: "Raymond C. Montgomery"
6       address: "211 Lynn Avenue"
7       phoneNumber: "715-533-4965"
8     }
9   ) {
10     id
11     fullName
12     address
13     phoneNumber
14   }
15 }
```

- src/main/resources/exampleQueries/example2.graphql
- Playground tab: "Insert Mutation"

Απόκριση Server

```
{
  "data": {
    "createCustomer": {
      "id": "8",
      "fullName": "Raymond C. Montgomery",
      "address": "211 Lynn Avenue",
      "phoneNumber": "715-533-4965"
    }
  }
}
```

ZHTOYMENO 2 : MUTATIONS (UPDATE)

Mutation προς τον Server

```
1 # Change mutation showcase: Register the rent with id = 4 as returned at the specified date.
2 # After that, show the rent's id and date rented and returned.
3 mutation {
4   registerReturn(input: { rentID: 4, date: "2022-01-18T18:36:00+02:00" }) {
5     id
6     dateFrom
7     dateTo
8   }
9 }
```

- `src/main/resources/exampleQueries/example3.graphql`
- Playground tab: "Change Mutation"

Απόκριση Server

```
{
  "data": {
    "registerReturn": {
      "id": "4",
      "dateFrom": "2020-11-24T17:10:03+02:00",
      "dateTo": "2022-01-18T16:36:00Z"
    }
  }
}
```

ΕΥΧΑΡΙΣΤΩ

Project Repository:

<https://github.com/JohnBarakl/GraphQL-Server-Using-Dummy-Video-Club-Data>