**תרגיל אפולו סרבר:**

1. צריך לעשות clone לפרויקט הבא  
    <https://github.com/Enigmatis/apollo-twitter-server.git>
2. לפתוח אותו ב-webstorm, להריץ npm install
3. להריץ npm run start, ולראות שהשרת מצליח לרוץ, על ידי בדיקה של השרת בכתובת <http://localhost:4000/graphql>

**תיאור המאגר:**

המאגר שנכתוב הוא מאגר ה-tweets, שממנו יצרכו מידע עבור משתמשים והציוצים שלהם, כמו למשל האתר apollo-twitter שנכתוב בהמשך הקורס.

המאגר מכיל מידע עבור הישויות הבאות: משתמש וציוץ.

הוא יאפשר הפעלת שאילתות בסיסיות שיאפשרו לנו לשלוף ולעדכן את הישויות האלו במאגר.

**ה-API של המאגר:**

המאגר שהורדתם מה-git לא מלא, ועליכם להשלים את ה-API שהמאגר מיחצן בעזרת apollo-server, שימו לב שהוא נכתב על בסיס express.

הישויות שנצטרך להגדיר הם:

ציוץ Tweet:

1. id (מזהה של הציוץ)
2. body (הגוף של הציוץ עצמו)
3. date (התאריך שבו פורסם הציוץ)
4. user (המשתמש שכתב את הציוץ)

משתמש User:

1. id (מזהה של המשתמש)
2. username (שם המשתמש שאיתו מתחברים)
3. password (הסיסמא של המשתמש)
4. firstName (השם הפרטי של המשתמש)
5. lastName (שם המשפחה של המשתמש)
6. name ***@deprecated*** (השם המלא של המשתמש, מורכב משם פרטי ומשפחה, לשים לב שזהו שדה עם directive של deprecated)
7. avatarUrl (url של תמונה אשר המשתמש מגדיר כתמונה שלו)
8. tweets (רשימה של כל הציוצים של אותו משתמש)

UserInput:

1. username (שם המשתמש שאיתו מתחברים)
2. password (הסיסמא של המשתמש)
3. firstName (השם הפרטי של המשתמש)
4. lastName (שם המשפחה של המשתמש)

**צריך להגדיר את הפעולות הבאות:**

*Query:*

allUsers – מחזיר את כל המשתמשים

allTweets – תחזיר את רשימת כל הציוצים

userByUsername(username: String) – מחזיר משתמש לפי שם המשתמש שלו

allUsersAndTweets – מחזיר רשימה של כל הציוצים והמשתמשים

tweetsByUserName(username: String) – מחזיר רשימה של כל הציוצים שנכתבו על ידי משתמש מסויים

*Mutation:*

createTweet(body: String) – יוצר ציוץ חדש, מקבל את הusername דרך ה-headers ב-context

createUser(user: UserInput ) – יוצר משתמש חדש

deleteUser(username: String) – מחיקת משתמש מהמערכת

updateUserAvatar(avatarUrl: String) – עדכון התמונה של המשתמש, מקבל את ה-username דרך ה-headers ב-context

**בונוס - Subscriptions:**

יש להוסיף מנגנון Pubsub לשירות ה-GraphQLי שיצרנו.

יש להוסיף לסכמה את השדה הבא:

*Subscription:*

userCreated – שולח עדכון עבור כל משתמש חדש שנוצר