

## Period-4 Geo-location on the backend (GeoJson, MongoDB and geoqueries)

# LAVET AF: Christian Madsen, Frederik Hurup & Jacob Simonsen

Note: This description is too big for a single exam-question. It will be divided up into separate questions for the exam

### MongoDB indexes and Geo-features

**Explain about indexes in MongoDB, how to create them, and *demonstrate* how you have used them.**

Indexes øger effektiviteten af hvor hurtigt queries kan udføres, da den ikke skal køre alle dokumenterne i hele kollektionen igennem. De minder meget om indexes for andre databasesystemer, men i MongoDB kan man have indexes alle felter og underfelter.

Se *makeTestFriends.ts* for hvordan man laver et index (`collection.createIndex("name": key)`). Key værdien bestemmer hvordan mongo sorterer indexet. Vi har brugt det til at lave et unikt index på email.

**Explain, using your own code examples, how you have used some of MongoDB's "special" indexes like TTL and 2dsphere and perhaps also the Unique Index.**

Se *makeTestPositions.ts*:

```
createIndex({ "lastUpdated": 1 }, { expireAfterSeconds: 60 })
```

Bruger TTL til at fjerne dokumentet efter 60 sekunder, da en position hele tiden er dynamisk og vi derfor ikke ønsker en masse forældede dokumenter.

```
createIndex({ location: "2dsphere" })
```

For at bruge queries med geospatial coordinate data skal der oprettes et index med "2dsphere" der hvor koordinaterne kan findes.

```
createIndex({ email: 1 }, { unique: true })
```

Opretter et unikt index.

### Geo-location and Geojson

**Explain and demonstrate basic Geo-JSON, involving as a minimum, Points and Polygons**

Et point består af en enkelt position [Longitude, latitude, elevation]

En polygon definer et område, både inderside og yderside ud fra adskillige positions [..][..][..]

se *geoInterfaces.ts*

## **Explain and demonstrate ways to create Geo-JSON test data**

Opret bruger -> opret position med bruger -> commit til db

Se *positionFacadeTest.ts* & *makeTestPositions.ts*

## **Explain the typical order of longitude and latitude used by Server-Side APIs and Client-Side APIs**

Server-side APIs: Longitude - Latitude / Client-side APIs: Latitude - Longitude

## **Explain and demonstrate a GraphQL API that implements geo-features, using a relevant geo-library and plain JavaScript**

Der er blevet brugt geojson-utils (<https://github.com/maxogden/geojson-js-utils>)

I schema'et er der lavet relevante typer til at holde geo-værdierne.

I resolvers er der brugt gju biblioteket's funktioner til bearbejde informationen.

Se projekt geoStartNoDB -> src/resolvers.ts & schema.ts

## **Explain and demonstrate a GraphQL API that implements geo-features, using MongoDB's geospatial queries and indexes.**

Der skal være oprettet et index i den pågældende collection på denne måde: `createIndex({ location: "2dsphere" })`, da ingen af geo forespørgslerne ellers vil virke.

Der er brugt en geospatial query i `findNearbyFriends` funktionen hvor `$near` og `$maxDistance` er brugt. `$Near` tjekker om noget er i nærheden af det definerede punkt, og `$maxDistance` er radius for det område `$near` skal tjekke på.

Facademetoderne er brugt i resolverne, som er defineret ved brug af schemaet.

Se startcode ->

src/graphql/resolvers.ts & schema.ts

src/interfaces/IPosition.ts & geoInterfaces.ts

src/facades/positionFacade.ts

src/utils/makeTestPositions.ts & geoUtils.ts