Veebiarendus

Back-End arendus

Martti Raavel

martti.raavel@tlu.ee

Tänased teemad

- Meenutame eelmist loengut
- Autentimine ja autoriseerimine
- Bcrypt
- JWT
- Andmete saatmine Express API-le - Header
- Autentimise ja autoriseerimise rakendamine
- Relatsiooniline andmebaas
- MySQL
- MySQL Dockeris

Millest rääkisime eelmises loengus?

Autentimine ja autoriseerimine

Autentimine on protsess, millega üks kasutaja, süsteem või muu olem (objekt) saab kontrollida teise olemi väidetava identiteedi tõesust, tavaliselt mingit tüüpi identsustõendi alusel:

- miski, mida Sa tead (näiteks parool, PIN-kood, robotilõks, turvaküsimus);
- miski, mis Sul on (näiteks ID-kaart, pangakaart, telefoni number, e-mail, riistvarapääsmik, paroolikaart, sertifikaat) või
- miski, mis Sa oled (sõrmejälg, näo pilt või topograafia, iirise struktuur, ...).

Autoriseerimine

`Autoriseerimine` ehk volitamine on protsess, mis annab (või keelab) õiguse ligi pääseda (võrgu)ressurssidele. Näiteks enamik e-kaubanduse turvasüsteeme põhineb kaheastmelisel protsessil. Esiteks toimub `autentimine`, kus kontrollitakse, et kasutaja on tõesti see, kellena ta esineb, seejärel toimub `autoriseerimine`, mis lubab kasutajal juurde pääseda temale ette nähtud ressurssidele.

Kuidas toimub autentimine ja autoriseerimine API-s?

- Kasutaja tuvastamine - kuidas?
- Kasutaja õigused?
- Millal ja kuidas kasutaja õiguseid kontrollitakse?

Paroolide räsimine

Räsimine vs krüpteerimine

Räsimine

- Räsimine on protsess, kus sisendväärtus (nt parool) muudetakse fikseeritud pikkusega väljundiks, mida nimetatakse räsi väärtuseks.
- Räsi väärtus on unikaalne ja seda ei saa tagasi muuta algseks sisendiks.
- Räsi väärtust kasutatakse sageli andmete turvalisuse tagamiseks, kuna see on ainulaadne ja seda ei saa tagasi muuta algseks sisendiks.

> Või kas ikka ei saa???

Bcrypt

`Bcrypt` on räsimisfunktsioon (*hashing function*), mida kasutatakse peamiselt paroolide räsimiseks.

Turvalisus: Erinevalt paljudest teistest räsimisfunktsioonidest on `bcrypt` spetsiaalselt kavandatud aeglaseks, mis teeb ründajate jaoks raskemaks salasõnade jõuga lahtimurdmise. Selle aeglust saab reguleerida nn `soolamise` korduste arvu suurendamise või vähendamisega.

Bcrypt - soolamine

Soolamine: `bcrypt` sisaldab automaatselt soolamise funktsiooni. Soolamine tähendab juhuslike andmete lisamist salasõnale enne räsimist, et vältida räsi väärtuste kordumist samade salasõnade jaoks ja raskendada ründajate tööd, kes kasutavad eelnevalt arvutatud räsi väärtuste tabeleid (nn `rainbow tables`).

Bcrypt kasutamine

Kui meil on andmebaasis kasutajate paroolid räsitud, siis kuidas me saame kontrollida, kas kasutaja sisestatud parool on õige, kui me ei saa räsitud parooli tagasi muuta algseks paroolik?

Bcrypt kasutamine - näide

```
// Impordime bcrypt'i
const bcrypt = require('bcrypt');

const saltRounds = 10;

const hashService = {
  hash: async (password) => {
    const hash = await bcrypt.hash(password, saltRounds);
    return hash;
  },
  compare: async (password, hash) => {
    const match = await bcrypt.compare(password, hash);
    return match;
  },
};

module.exports = hashService;
```

Mis hetkel me peaksime parooli räsima?

JSON Web Token (JWT)

JWT (*JSON Web Token*) on avatud standard (RFC 7519), mida kasutatakse andmete turvaliseks edastamiseks osapoolte vahel JSON formaadis. See on eriti kasulik autentimise ja informatsiooni vahetamise jaoks, kuna see võimaldab teabe kindlat ja tõhusat edastamist. JWT on kompaktne ja hõlpsasti kasutatav mitmesugustes stsenaariumides, sealhulgas veebirakenduste autentimises ja autoriseerimises.

JWT komponendid

JWT koosneb kolmest osast, mis on eraldatud punktidega (.). Need osad on:

- Header (Päis)
- Payload (Koormus)
- Signature (Allkiri)

Header (Päis)

See sisaldab teavet tokeni tüüp ja kasutatava allkirja algoritmi kohta (nt HMAC SHA256 või RSA).

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

Payload (Koormus)

See sisaldab väiteid (*claims*), mis on JWT-sse kodeeritud teave. Väited võivad olla standardiseeritud, avalikud või privaatsed.

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

Allkiri (Signature)

Allkiri genereeritakse kasutades header ja payload osi ning salajast võtit. Allkiri võimaldab kontrollida, kas JWT on kehtiv ja usaldusväärne.

JWT struktuur

JWT struktuur on järgmine:

```
xxxxx.yyyyy.zzzzz
```

kus: xxxxx on päis, yyyyy on koormus ja zzzzz on allkiri, kõik base64-url kodeeringus.

JWT näide

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
```

JWT kasutamine

Kuidas me saame JWT-d kasutada autentimise ja autoriseerimise protsessides?

Andmete saatmine Express API-le - Header

Sageli on vaja saata andmeid API-le, mis on seotud kasutaja autentimise ja autoriseerimisega. Kuidas seda teha?

Authorization Header

`Authorization` päises saab saata autentimiseks vajalikku teavet. Näiteks JWT tokenit.

JWT autentimise ja autoriseerimise rakendamine

Relatsiooniline andmebaas

Relatsioonilised andmebaasid on laialdaselt kasutatav andmehaldusvahend, mis võimaldab andmeid salvestada struktureeritud ja korraldatud kujul. Need andmebaasid kasutavad tabelite süsteemi, et andmeid talletada ja hallata, ning tuginevad relatsioonilisele mudelile.

Relatsioonilise andmebaasi põhikomponendid

- Tabelid
- Primaarvõti (Primary Key)
- Võõrvõti (Foreign Key)
- Indeksid
- ...

Tabelid

Tabelid on relatsiooniliste andmebaaside põhielemendid, mis sisaldavad ridu ja veerge. Iga tabel esindab kindlat andmekogumit (nt kliendid, tellimused, tooted).

- **Rida (Row):** Tabeli andmekirje. Iga rida sisaldab andmeid vastavalt tabeli veergudele. Rida võib olla ka nimetatud kui kirje või olem.
- **Veerg (Column):** Tabeli omadus või atribuut, mida nimetatakse ka väljadeks.

Näide tabelist:

ID	Nimi	Vanus
1	Jaan	25
2	Mari	30

Primaarvõti (Primary Key)

Primaarvõti on unikaalne identifikaator, mis eristab iga tabeli rida. Primaarvõti tagab, et iga kirje on unikaalne.

Võõrvõti (Foreign Key)

Võõrvõti on veerg või veergude kombinatsioon, mis loob seose kahe või enama tabeli vahel. Võõrvõti viitab primaarvõtmele teises tabelis.

Indeksid

Indeksid on struktuurid, mis aitavad kiirendada andmete otsimist ja sorteerimist andmebaasis. Indeksid võimaldavad andmebaasisüsteemil kiiresti leida andmeid, mis vastavad konkreetsetele kriteeriumidele.

MySQL

MySQL on avatud lähtekoodiga relatsiooniline andmebaasisüsteem, mis on laialdaselt kasutatav veebirakenduste arendamisel. MySQL kasutab SQL-i (Structured Query Language) andmete haldamiseks ja päringute tegemiseks.

MySQL Dockeris

Docker on populaarne konteinerite virtualiseerimise platvorm, mis võimaldab arendajatel luua, käivitada ja jagada rakendusi konteinerites. MySQL-i saab käivitada Dockeris, et luua ja hallata andmebaase arendus- ja tootmisrakendustes.

SQLTools VS Code lisandmoodul

SQLTools on Visual Studio Code lisandmoodul, mis võimaldab arendajatel ühendada ja hallata SQL-andmebaase otse VS Code keskkonnas. SQLTools toetab mitmeid andmebaasisüsteeme, sealhulgas MySQL, PostgreSQL, SQLite, SQL Server ja paljud teised.

Kodune töö

- Loe läbi tänase loengu materjalid
- Rakenda Bcrypt ja JWT oma Express API-s
- Veendu, et saad tööle MySQL Dockeris ja sellega ühendust luua