



Syyskuu-Lokakuu 2024

Viikko 38 - Projektin aloitus

E Pääkomponenttien luonti

- Toteutettu projektin perusrakenne:
 - Navbar-komponentti
 - Highway-komponentti (myöhemmin uudelleennimetty)
- Haaste: Biome-tietojen näyttäminen valetietokannasta
- Ratkaisu: Dynaaminen toteutus
 - 1. Sivusto listaa biomet ID:n perusteella
 - 2. ID:tä klikkaamalla pääsee uudelle sivulle
 - 3. Uusi sivu näyttää biomin sisällön
- Opitut asiat:
 - o Dynaaminen reititys Angularissa
 - o Tietorakenteiden suunnittelu
 - o Komponenttien välinen kommunikaatio

Projektin Kehitys



Dunkkuapuri

<u>Dungeon Master Resources</u> Random Encounters



Toiminnallisuuden kehitys

- Ensimmäinen toiminto toteutettu: biomien listaus valetietokannasta
- Toiminnallisuus demonstroitu kahdella näkymällä:

Dunkkuapuri

Dungeon Master Resources
Highway
Dungeon
settlement
wilderness

1.

Dunkkuapuri

Dungeon Master Resources

Highway Encounters

Bacl

Highwaymen Niko Travelling Merchant Group of Pilgrims Wanted Poster Abandoned Cart Mysterious Stranger Roadside Shrine

Rolled Encounter:

Name: Group of Pilgrims

Description: A group of devout pilgrims, traveling to a nearby holy site, are seeking donations or food from the party.

2.

Tietokantarakenne

** Vale tietokanta**

```
export class InMemoryService implements InMemoryDbService {
  createDb() {
    const highwayEncs = [
```

```
id: 1,
      name: 'Highwaymen',
      description: 'A group of bandits, armed to the teeth, are lying in
wait to ambush unsuspecting travelers.'
   },
   // ... other highway encounters
  1;
  const dungeonEncs = [
      id: 9,
      name: 'Injured Dungeoneer',
      description: 'An injured dungeoneer is limping around the dungeon and
needs medical attention'
    },
   // ... other dungeon encounters
  1;
  const wildernessEncs = [
    {
      id: 12,
      name: 'Owlbear',
      description: 'An Owlbear is heard rummaging through the bushes nearby'
    },
    // ... other wilderness encounters
  ];
  return { highwayEncs, dungeonEncs, wildernessEncs };
}
```

```
### Viikko 39
- Valtion virallinen kertausharjoitus

### Viikko 40 - Modaalit ja tietokantaintegraatio
#### 1 MongoDB Atlas -migraatio

#### 1 MongoDB Atlas -migraatio
- Tietokanta siirretty pilveen
```

}

```
- Docker-riippuvuudesta luovuttu
- Yhteys tietokantaan pilvipalvelun kautta

#### & Uudet toiminnallisuudet
- Aloitettu CRUD-operaatioiden toteutus:
- Edit-toiminnallisuus
- Delete-toiminnallisuus

## Marraskuu 2024

## 18.11.2024 - Encountereiden tallennus
---

***Saavutus**: Encountereiden tallennus saatu toimivaksi pitkän työstämisen jälkeen.

#### 

Tekninen oivallus:
Ongelma oli tietokannan `saveEnc`-metodissa. Alkuperäinen koodi:
```javascript
_id: req.params.biomeId,
'enc.name': req.body.name // Tämä esti encounterin nimen päivittymisen
```

#### Miksi tämä oli ongelma?

- Toiminto mätsää tallennettavan encounterin nimen
- Jos nimi on muuttunut, haku ei löydä dokumenttia
- Ehto etsii enciä, jonka nimi vastaa uutta nimeä

#### **☑** Ratkaisu:

```
_id: req.params.biomeId,
'enc._id': req.body._id // Nyt löytyy aina ID:n perusteella
```

#### Loppuviikon tehtävät:

- Toiminnon implementointi
- Demon toiminnallisuuden varmistaminen

#### 22.11.2024 - Demo ja palaute

#### S Demo-arvosana: 3

#### **B** Kehityskohteet seuraavaan demoon:

- 1. Syvällisempi analyysi ongelmatilanteista
  - o Mitä ongelmia kohdattiin?
  - o Miksi ne ilmenivät?
  - o Miten ne ratkaistiin?

- 2. Selkeämpi esittely tehdyistä asioista
- 3. Työskentelytapojen avaaminen
- 4. Loppupohdinta
  - Mikä meni hyvin?
  - o Mikä meni huonosti?

## 25.11.2024 - 📆 Google-kirjautuminen ja tavallinen kirjautuminen

## Tavoitteet Google-kirjautumiselle toteuttamiselle:

- Tehdä login-sivu
- •

# Tavoitteet tavallisen kirjautumisen toteuttamiselle:

\* Viikon aloitus: Positiivinen palaute motivoi työskentelyä!

#### **Q** Pohdinta aiheesta: Google-kirjautuminen:

- Aloitettu Tuikan tutoriaalin pohjalta
- Haaste: Esimerkkikoodin soveltaminen omaan projektiin
- Oppimiskokemus: Koodin soveltaminen vaatii syvällistä ymmärrystä

#### Pohdinta aiheesta: Tavallinen kirjautuminen:

- Kaikki olikin jo valmiina, kun koodi oltiin tehty backend1 tunnilla
- Tarvitsi ainoastaan luoda tunnarit tietokantaan ja homma pelitti

## Google-kirjautumisen toteutus

#### 1. Projektin alustus

- Pohja otettu tutoriaali-tehtävästä
- Asennettu tarvittava paketti: @abacritt/angularx-social-login

#### 2. Abacritt-kirjasto

#### Avoimen lähdekoodin Angular-kirjasto sosiaalisen median kirjautumisille

- Tukee useita kirjautumistapoja (Google, Facebook)
- Automatisoi:
  - o Kirjautumisprosessin
  - o Käyttäjätietojen haun
  - Istunnonhallinnan
- Yksinkertaistaa integraation toteutusta

#### 3. Konfiguraatio (app.config.ts)

```
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { SocialLoginModule, GoogleSigninButtonModule } from '@abacritt/angularx-
social-login';
import { JwtModule } from '@auth0/angular-jwt';
@NgModule({
 imports: [
 FormsModule,
 SocialLoginModule,
 GoogleSigninButtonModule,
 JwtModule.forRoot({
 config: {
 tokenGetter: tokenGetter,
 allowedDomains: ['localhost:4200'],
 disallowedRoutes: []
 }
 })
]
})
export class AppModule {}
```

#### 4. Komponenttien selitykset

#### **Moduulit**

#### • SocialLoginModule

- Hallinnoi Googlen kirjautumisprosessia
- Käsittelee autentikaation logiikan

#### • GoogleSigninButtonModule

- Tarjoaa valmiin Google-kirjautumispainikkeen
- Noudattaa Googlen brändäysohjeita

#### **☆** JWT-autentikaatio

#### JwtModule.forRoot()

- o Hallinnoi JWT-pohjaista autentikaatiota
- o Konfiguraation osat:
  - tokenGetter: Hakee JWT-tokenin
  - allowedDomains: Määrittää sallitut domainit (localhost:4200)
  - disallowedRoutes: Määrittää kielletyt reitit

#### 4. Frontend-toteutus

#### Kirjautumiskomponentit

#### LoginComponent

- Käsittelee sekä Google- että tavallisen kirjautumisen
- o Sisältää kirjautumislomakkeen ja Google-kirjautumispainikkeen
- o Ohjaa käyttäjän dashboardille onnistuneen kirjautumisen jälkeen

#### **⚠** AuthService

```
// Tavallinen kirjautuminen
login(username: string, password: string): Observable
boolean> {
 return this.http.post(this.basicLoginUrl, { username, password })
 .pipe(map((res: any) => {
 const token = res['token'];
 if (token) {
 this.token = token;
 sessionStorage.setItem('accesstoken', token);
 return true;
 return false;
 }));
}
// Google-kirjautuminen
glogin(gtoken: string): Observable<boolean> {
 return this.http.post(this.googleLoginUrl, { gtoken })
 .pipe(map((res: any) => {
 const token = res['token'];
 if (token) {
 sessionStorage.setItem('accesstoken', token);
 return true;
 return false;
 }));
}
```

#### Reittien suojaus

- Toteutettu authGuard-funktionaalisella guardilla
  - 1. Tarkistaa jokaisen suojatun reitin kohdalla, onko käyttäjä kirjautunut sisään
  - 2. Jos käyttäjällä on voimassa oleva token sessionStoragessa, sallii pääsyn reitille
  - 3. Jos tokenia ei ole tai se on vanhentunut, ohjaa käyttäjän automaattisesti login-sivulle
  - 4. Toimii "portinvartijana" kaikille suojatuille reiteille, kuten dashboard, retables ja merchants

Esimerkki suojatuista reiteistä:

```
export const routes: Routes = [
 // Julkiset reitit
 { path: 'login', component: LoginComponent },
 { path: '', redirectTo: 'login', pathMatch: 'full' },
```

```
// Suojatut reitit (vaativat kirjautumisen)
{
 path: 'dashboard',
 component: KaruselliComponent,
 canActivate: [authGuard] // Tarkistaa kirjautumisen
},
{
 path: 'retables',
 component: RetablesComponent,
 canActivate: [authGuard] // Tarkistaa kirjautumisen
}
]
```

#### 5. Backend-toteutus

### MongoDB User-malli

Käyttäjäskeema määrittelee sovelluksen käyttäjän tietorakenteen:

- username: Käyttäjän uniikki tunniste, jota käytetään kirjautumiseen
- password: Käyttäjän salasana (huom: tallennetaan salattuna)
- isadmin: Määrittää käyttäjän ylläpito-oikeudet

```
const UserSchema = new mongoose.Schema({
 username: { type: String, unique: true, required: true },
 password: { type: String, required: true },
 isadmin: { type: Boolean, required: true },
});
```

#### 29.11.2024 - 09.12.2024

- --- Kertausharjoitus, eli olen poissa ja palaan 09.12.2024. takaisin ---
  - Tämän jälkeen, pyritään suorittamaan seuraavat tehtävät:
    - 1. Kirjautumisprosessin toteuttaminen loppuun Kirjautumistietojen tallennus tietokantaan
    - 2. Selvitys LocalStoragen käyttöönotosta
    - 3. PWAn buildaus