



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
  ];
  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

Modaalien toteutus

- Enctable-komponentin modaalit:
 - Nopanheiton tulosten näyttäminen
 - Satunnaisten kohtaamisten esittäminen
 - Yksittäisten kohtaamisten tarkastelu

♥ Backend-integraatio

- · Frontend yhdistetty backendiin
 - o Valetietokannan ongelmat ratkaistu
 - Tiedonkulku optimoitu

MongoDB Atlas -migraatio

- Tietokanta siirretty pilveen
 - Docker-riippuvuudesta luovuttu
 - o Yhteys tietokantaan pilvipalvelun kautta

W Uudet toiminnallisuudet

- Aloitettu CRUD-operaatioiden toteutus:
 - Edit-toiminnallisuus
 - Delete-toiminnallisuus

Viikko 41 - Demoviikko

• Random Encountersin Save/Edit/Delete-toiminnallisuudet

Marraskuu 2024

18.11.2024 - Encountereiden tallennus

Saavutus: Encountereiden tallennus saatu toimivaksi pitkän työstämisen jälkeen.

Parameter Tekninen oivallus:

Ongelma oli tietokannan saveEnc-metodissa. Alkuperäinen koodi:

```
_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

★ Demo-arvosana: 3

凌 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
 - o Mikä meni hyvin?
 - o Mikä meni huonosti?

25.11.2024 - 📆 Google-kirjautuminen ja tavallinen kirjautuminen

Tavoitteet Google-kirjautumiselle toteuttamiselle:

- Tehdä login-sivu
- Integroida Google-kirjautuminen
- Toteuttaa JWT-autentikaatio
- Suojata reitit auth guardilla

Tavoitteet tavallisen kirjautumisen toteuttamiselle:

- Hyödyntää olemassa olevaa backend-toteutusta
- Integroida kirjautuminen frontend-puolelle
- · Varmistaa tietoturva
- * Viikon aloitus: Positiivinen palaute motivoi työskentelyä!

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

4. Frontend-toteutus

Kirjautumiskomponentit

LoginComponent

- o 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

& Loppureflektio ja Ammatillinen Kehitys

Roolit ja Vastuut

- Product Owner
- FullStack-kehittäjä

Osaamisen Kehittyminen

Frontend (Angular)

Lähtötilanne ja Tavoitteet:

Alkutaso: 5/10Tavoitetaso: 8/10Nykyinen taso: 7/10

Konkreettinen Kehittyminen:

- Angular-rakenteen kokonaisvaltainen ymmärtäminen
- Komponenttien välinen kommunikaatio
- Palveluiden (Services) tehokas hyödyntäminen
- Komponenttien elinkaarien hallinta

Backend (Express.js)

Lähtötilanne ja Tavoitteet:

• Alkutaso: 6/10

• Tavoitetaso: 8/10

Nykyinen taso: 7/10

Konkreettinen Kehittyminen:

- Express.js -arkkitehtuurin syvällisempi ymmärrys
- Controller-metodien monipuolinen toteutus
- REST API -toiminnallisuuksien kehittäminen
- Tietokantaintegraatioiden hallinta

Yhteenveto

Vaikka en täysin saavuttanut alkuperäisiä tavoitetasojani (8/10), olen tyytyväinen kehitykseeni sekä frontendettä backend-puolella. Backend-kehitys jäi projektissa vähemmälle huomiolle sen yksinkertaisemman luonteen vuoksi, mutta opin silti arvokkaita taitoja molemmilla osa-alueilla.

Pää- ja sivutyön näkyvyys projektissa

• Päätyö (TC Dungeon Helper)

- Vastuualueeni FullStack -kehityksessä
- o Merchant ja Random Encounters -toiminnallisuudet

Sivutyö (Portfolio)

- Dokumentointi ja oppimispäiväkirjan ylläpito
- o Koodin laadun varmistaminen ja refaktorointi
- o Tiimin sisäinen mentorointi ja tuki
- Autentikaation toteutus (Google-kirjautuminen ja perinteinen kirjautuminen)

Oppimiskokemukset ja oivallukset

• Tekniset taidot

- o Angular-osaamisen syventyminen
- Autentikaation ja käyttäjähallinnan ymmärrys
- REST API -integraatiot
- Git-työskentelyn tehostuminen

Pehmeät taidot

- Tiimityöskentelyn merkitys projektin onnistumisessa
- o Kommunikaatiotaitojen kehittyminen
- Ongelmienratkaisukyvyn vahvistuminen
- o Itsenäisen työskentelyn ja ajankäytön hallinta

Tavoitteiden toteutuminen

• Saavutetut tavoitteet

- o Toimiva autentikaatiojärjestelmä
- o Random Encounters-toiminnallisuuden perustoiminnot
- Merchant-toiminnallisuuden perustoiminnot
- Responsiivinen käyttöliittymä

Saavuttamattomat tavoitteet

- o Quest giver -toiminnallisuuden toteutus
- o Loot table -toiminnallisuuden toteutus
- LocalStoragen toteutus

Jatkokehityskohteet

- Käyttäjäkokemusten kerääminen ja analysointi
- o Saavutattomien tavoitteiden toteutus
- LocalStoragen käyttöönotto

Suunnitelman ulkopuoliset opit

• MongoDB Atlas -integraation haasteet ja ratkaisut

- Product Ownerin tehtävät ja vastuut
- JWT-tokenien käyttö autentikaatiossa

Tulevaisuuden näkymät

Uratoiveet

FullStack -kehittjäksi

Opintojakson tuki tavoitteille

- o Käytännön kokemusta modernista web-kehityksestä
- o Ymmärrys full stack -kehityksen periaatteista
- o Tiimityöskentelyn ja projektinhallinnan taidot

& Kehityskohteet

• Tarvittava lisäosaaminen

- Testaukseen perehdyminen
- o Tilanhallintaan syventyminen (Signal Store)

⊗ Vahvuudet

- Ihmistaidot
- Kyky työskennellä itsenäisesti ja tiimissä
- Ongelmienratkaisukyky ja sinnikkyys
- Dokumentoinnin ja koodin laadun arvostaminen

Portfolio-valintojen perusteet

- Dokumentit osoittavat teknisen osaamisen kehityksen
- Koodiesimerkit demonstroivat ongelmanratkaisutaitoja
- Oppimispäiväkirja kuvastaa reflektointikykyä ja ammatillista kasvua
- Valitut työt edustavat monipuolista osaamista web-kehityksessä

Huomioita

 Portfolio sivulle selkeästi esimerkit toiminnallisuuksista, miten ne ovat alkaneet, miten ne ovat kehittyneet ja miten ne toimivat. Mitkä olivat ongelmat, mikä voisi toimia paremmin ja miksi?

10.12.2023 - Tyyppien ja virhekäsittelyn parannuksia

Nerchants-komponentin korjaukset 🐪

• **Ongelma**: NgFor-direktiivi aiheutti ajonaikaisen virheen yrittäessään iteroida merkkijonoa taulukon sijaan

• Tehdyt muutokset:

- Lisätty tarkka tyypitys merchants-muuttujalle (Merchants[])
- o Alustettu merchants tyhjänä taulukkona välttääksemme undefined-tilanteet
- Parannettu getMerchants-metodin virhekäsittelyä

Miksi muutokset tehtiin:

- Parempi tyyppitarkistus auttaa välttämään ajonaikaisia virheitä
- Tyhjä taulukko alustuksena varmistaa, että NgFor-direktiivillä on aina iteroitava taulukko
- Virhekäsittelyn parannus tekee sovelluksesta vakaamman virhetilanteissa

Opitut asiat:

- o TypeScriptin tyyppijärjestelmän tärkeys Angular-sovelluksissa
- Virhekäsittelyn merkitys käyttäjäkokemuksen kannalta
- Komponenttien alustuksen parhaat käytännöt

10.12.2024 - Google-kirjautumisen korjaus tuotantoympäristössä

Ongelma: Google-kirjautumisen redirect URI -virhe

Ongelman kuvaus:

- Google-kirjautuminen ei toiminut tuotantoympäristössä (EBS)
- Virheviesti: "redirect_uri_mismatch"
- Syy: Tuotantoympäristön URL:ää ei oltu lisätty sallittuihin osoitteisiin Google Cloud Consolessa

Ratkaisu:

- 1. Google Cloud Consolen päivitys:
 - o Lisätty EBS-domain sallittuihin JavaScript-lähteisiin
 - Lisätty EBS-domain ja login-polku sallittuihin redirect URI:hin

Opitut asiat:

- Google OAuth vaatii tarkan URL-konfiguraation eri ympäristöille
- Tuotantoympäristön URL:t pitää erikseen sallia Google Cloud Consolessa
- @abacritt/angularx-social-login -kirjasto hoitaa OAuth-flown automaattisesti ilman erillistä callbacktiedostoa

Google-kirjautumisen CORS-ongelman ratkaisu (10.12.2024)

Ongelma

- Tuotantoympäristössä Google-kirjautuminen epäonnistui CORS-virheen takia
- Frontend yritti käyttää localhost-osoitetta tuotantoympäristössä

Ratkaisu

1. Backend CORS-konfiguraation päivitys

- Lisättiin tuotantoympäristön URL (dunkku.eu-north-1.elasticbeanstalk.com) sallittuihin CORSosoitteisiin
- CORS-asetukset päivitettiin tukemaan sekä kehitys- että tuotantoympäristöä

2. Auth Service päivitys

- o Muutettiin kovakoodatut localhost-osoitteet dynaamisiksi
- o Luotiin baseUrl-muuttuja, joka vaihtuu ympäristön mukaan
- o Kehitysympäristössä käytetään localhost:3000
- o Tuotannossa käytetään dunkku.eu-north-1.elasticbeanstalk.com

Opitut asiat

- CORS (Cross-Origin Resource Sharing) konfigurointi Express.js:ssä
- Ympäristökohtaisten URL-osoitteiden hallinta Angular-sovelluksessa
- Tuotanto- ja kehitysympäristöjen erojen huomioiminen autentikaatiossa