

House Helper-applikaatio

Synopsis

Tehdään kiinteistöongelmien käsittelyyn tarkoitettu applikaatio. Talon asukkaat voivat lisätä valituksia ilman kirjautumista, joita kiinteistöhenkilökunta käsittelee. Valitusten käsittely tapahtuu kirjautuneena applikaatioon. Applikaation etusivulla on formi, jolla asukkaat voivat esittää valituksiaan. Sisäänkirjautumissivu on piilotettu linkin, napin tai menun taakse. Sisäänkirjaututtuaan kirjautunut käyttäjä voi katsella kaikkia valituksia tai filtteröidä huoneiston, valituksen tilan tai molempien kautta. Lisäksi käyttäjä voi ottaa uuden valituksen itselleen käsiteltäväksi tai laittaa omansa käsitellyksi.

Tarvitaan siis etusivu, jossa form valitusten submittaamiseen, login ja registersivu, sivu, jossa on table ja inputteja ja nappeja filtteröimään valituksia tilan ja huoneiston avulla sekä ottamaan valitus haltuun sekä muuttamaan sen tilaa.

Valitus-objekti

```
{  
  id:number //tulee backendiltä,  
  name:string //käyttäjä syöttää input vaiheessa  
  apartment:string //käyttäjä syöttää input vaiheessa. Pitää olla muodossa "b35"  
  date: number //epoch aika millisekunteina valituksen jättövaiheessa. Järjestelmä laittaa.  
  desc:string // ongelman kuvaus. Käyttäjä laittaa input vaiheessa"  
  user:string // "" tai valitusta käsittelevän userin username  
  state: string ("unassigned" | "open" | "completed") // Järjestelmä laittaa. Aluksi "unassigned".  
  Kun rekisteröitynyt käyttäjä ottaa valituksen käsittelyyn se muuttuu tilaan "open" ja käyttäjä voi  
  muuttaa sen tilaan "completed"  
}
```

Eli käyttäjä syöttää formilla name, apartment ja desc. Loput tulee järjestelmältä

API

LOGIN:

POST /register:

bodynä {username,password} objekti. Palauttaa 200, jos rekisteröinti onnistuu ja 409, jos username on jo varattu. Palauttaa 422, jos username (4 merkkiä) tai password (8 merkkiä) puuttuu tai on liian lyhyt

POST /login:

bodynä {username,password} objekti. Palauttaa 200 ja {token:string} objektin. Palauttaa 422, jos username (4 merkkiä) tai password (8 merkkiä) puuttuu tai on liian lyhyt. Tokenia tarvitaan osaan data-apista. Käyttäjät voivat syöttää valituksia ilman loginia

POST /logout:

token headereihin {token:token}. Palauttaa 200 jos logout onnistuu. Muuten 404 tai 409.

COMPLAINT:

POST /complaint:

Bodynä {name,apartment,desc} objekti. Ei vaadi kirjautumista. Palauttaa 422, jos ei ole bodyä, muuten 200

GET /api/complaint:

Palauttaa kaikki valitukset. Query parametreilla voi filtteröidä palautettavaa dataa

?state=unassigned | open | completed palauttaa tietyssä tilassa olevat valitukset

?apartment=string palauttaa tietyn asunnon tekemät valitukset.

Nämä voi yhdistää.

PUT /api/complaint:

Muokkaa valituksen tilaa ja laittaa käyttäjän vastaamaan valituksesta. Sellaisenaan muuttaa userin vastaamaan nykyistä käyttäjää ja jos state on "unassigned" muuttaa sen tilaan "open". Query param:

?state = unassigned | open | completed muuttaa valituksen tähän tilaan.

Muista asentaa node modules:

npm install

Backend käynnistyy:

node app

Jos haluaa vähän blingä helpolla appsiin, niin voi katsella seuraavia vaihtoehtoja

Semantic UI React-kiinnitys: <https://react.semantic-ui.com/>

Bootstrap React-kiinnitys: <https://react-bootstrap.github.io/>

Materia UI React-kirjasto: <https://material-ui.com/>