Kratke smjernice za smještaj NodeJS web-aplikacije na servisu *Render*

Servis *Render* https://render.com, uz određena ograničenja omogućava besplatnu uslugu smještaja web-aplikacija i korištenje PostgreSQL baze podataka, što je dovoljno dobro za isporuku projekata iz ovog kolegija.

PostgreSQL

Za kreiranje baze podataka potrebno je slijediti upute s https://render.com/docs/databases.

Osjetljive i promjenjive postavke vezano uz bazu podataka (adresa, korisničko ime, lozinka, …) pospremite u varijable okruženja. Tijekom lokalnog razvoja, ti podaci se trebaju nalaziti u datoteci *.env,* a u oblaku ćete ih unijeti u varijable okruženje preko za to predviđenog sučelja.

Za spajanje na bazu podataka iz NodeJS aplikacije preporuča se koristiti biblioteku *node-postgres* (*pg*). Detaljnije o korištenju može se pronaći na https://node-postgres.com, a u nastavku je dan primjer konfiguracije klijenta i jednostavnog upita.

```
import { Pool } from 'pg'
import dotenv from 'dotenv'
dotenv.config()

const pool = new Pool({
  user: process.env.DB_USER,
  host: process.env.DB_HOST,
  database: 'web2demo',
  password: process.env.DB_PASSWORD,
  port: 5432,
  ssl : true

})

export async function getComments() {
    const comments : string[] = [];
    const results = await pool.query('SELECT id, comment from comments');
    results.rows.forEach(r => {
        comments.push(r["comment"]);
    });
    return comments;
}
```

Potrebni programski paketi za gore navedeni odsječak su *pg* i *dotenv*, pri čemu ako se radi u *TypeScriptu* za *pg* treba dodati još i *@types/pg*.

```
{
  ...
  "dependencies": {
    "dotenv": "^10.0.0",
    "pg": "^8.8.0",
    ...
  },
  "devDependencies": {
    ...
    "@types/pg": "^8.6.5"
  }
}
```

Preporuka je napraviti biblioteku za manipulaciju s bazom podataka iz konzolne aplikacije u NodeJS-u, a onda je kasnije proširiti i ugraditi u web-aplikaciju.

Priprema web-aplikacije prije isporuke na *Render*

Aplikacija će se na *Renderu* izvršavati unutar posebnog kontejnera, a R*ender* će napraviti preusmjeravanje s javne adrese (dostupne preko HTTPS-a), na lokalnu adresu u kontejneru u kojem server treba pokrenuti preko HTTP-a na portu određenom s varijablom okruženja PORT. Ova vrijednost je unaprijed dostupna i ne možemo je mijenjati. Slično vrijedi i za varijablu okruženja RENDER_EXTERNAL_URL.

Posljedično u programski kod nalik onom s predavanja treba unijeti neke izmjene, npr. u slučaju primjera 08-oidc/src/08-webapp.ts:

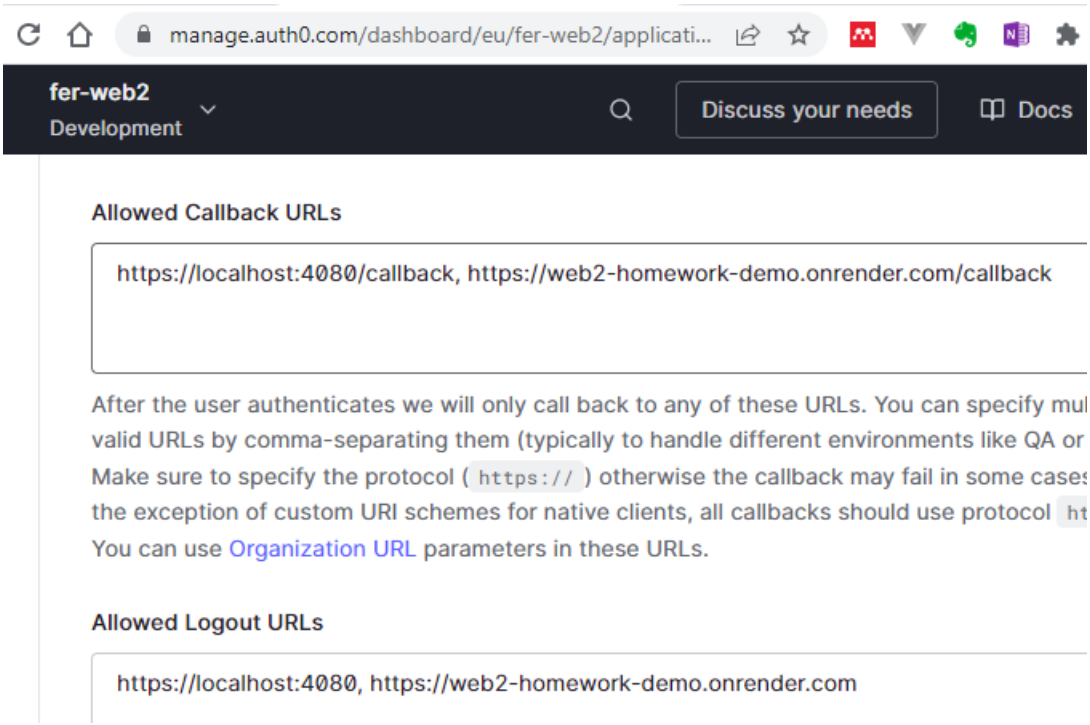
```
const externalUrl = process.env.RENDER_EXTERNAL_URL;
const port = externalUrl && process.env.PORT ? parseInt(process.env.PORT) : 4080;
```

```
const config = {
 ...
 baseURL: externalUrl || `https://localhost:${port}`,
;
```

```
if (externalUrl) {
  const hostname = '127.0.0.1';
  app.listen(port, hostname, () => {
    console.log(`Server locally running at http://${hostname}:${port}/ and from
outside on ${externalUrl}`);
  });
}
```

```
else {
  https.createServer({
    key: fs.readFileSync('server.key'),
    cert: fs.readFileSync('server.cert')
  }, app)
  .listen(port, function () {
    console.log(`Server running at https://localhost:${port}/`);
  });
}
```

pri čemu ne zaboravite na *Auth0* u postavkama za aplikaciju postaviti *Allowed Callback URLs* i *Allowed Logout URLs* tako da sadrže i lokalne adrese i adrese na *Renderu*.

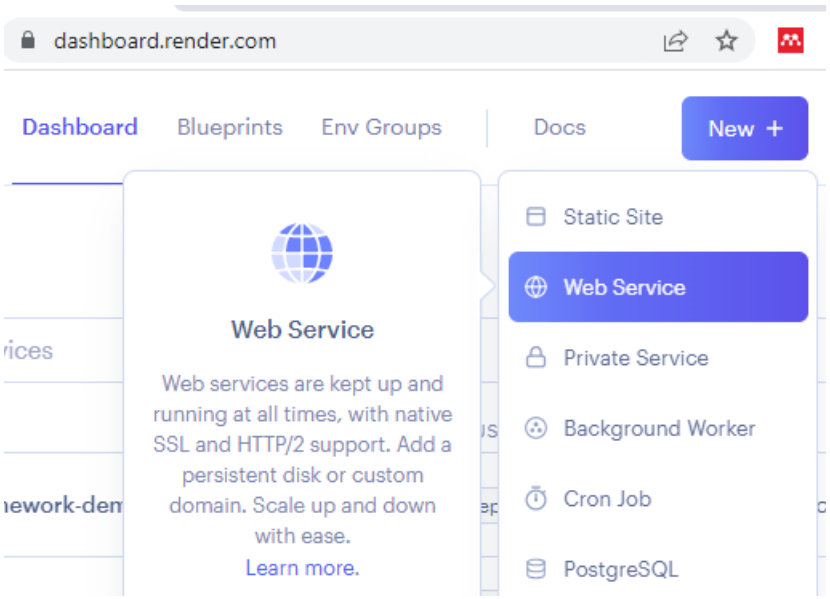


Dodatno, potrebno je izmijeniti i naredbe za skripte u packages.json kako bi bile prilagođene za linux, npr.

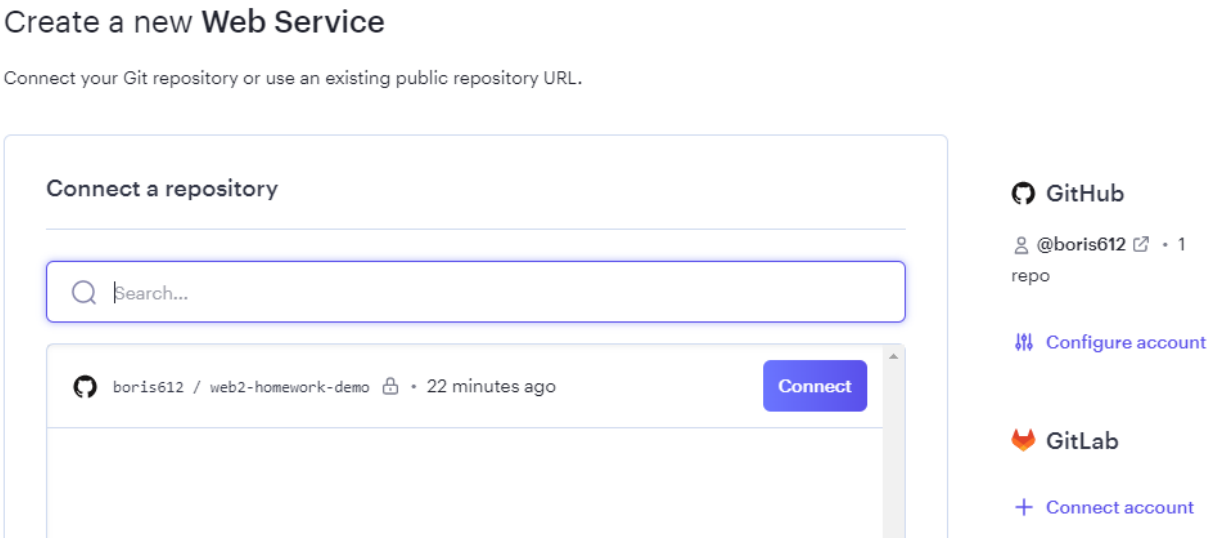
```
{
  "scripts": {
    "build": "npm run clean; tsc; npm run copy-views; npm run copy-views",
    "clean": "rm -rf dist",
    "copy-views": "cp -r src/views/ dist/views/",
    "start": "node dist/08-webapp.js"
  },
```

Izrada nove usluge smještaja web-aplikacije

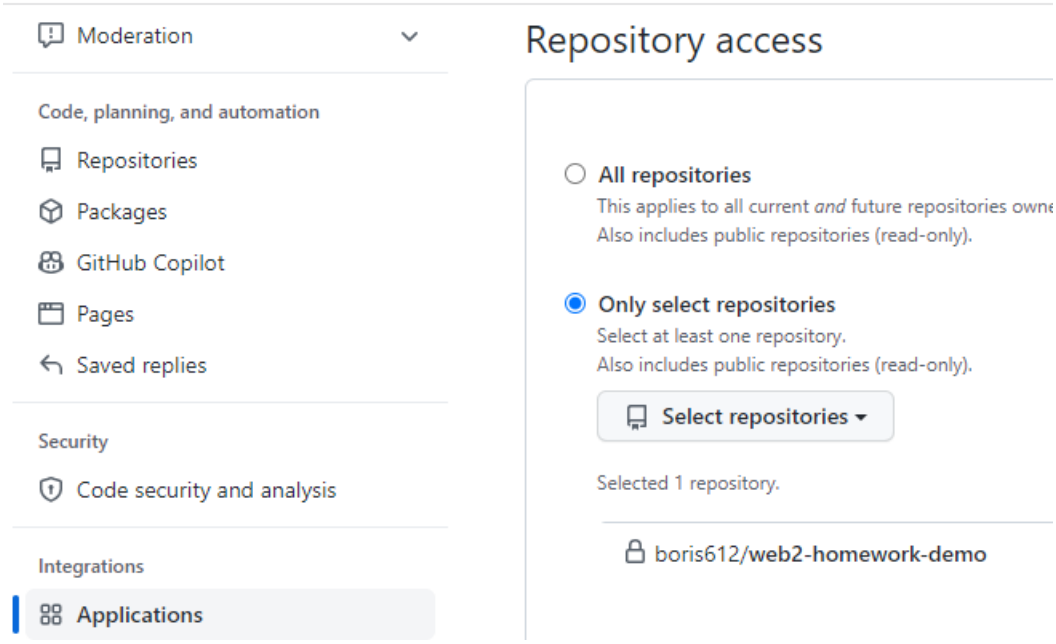
Nakon prijave na *Render* potrebno je na https://dashboard.render.com/ odabrati opciju *New Web Service* i potom odabrati repozitorij za koji će se definirati kontinuirana isporuka aplikacije.



Moguće se povezati na GitHub ili GitLab. Nakon prijave i odobrenja pristupa svojem GitHub ili GitLab računu u listi će se pojavljivati odabrani repozitoriji.



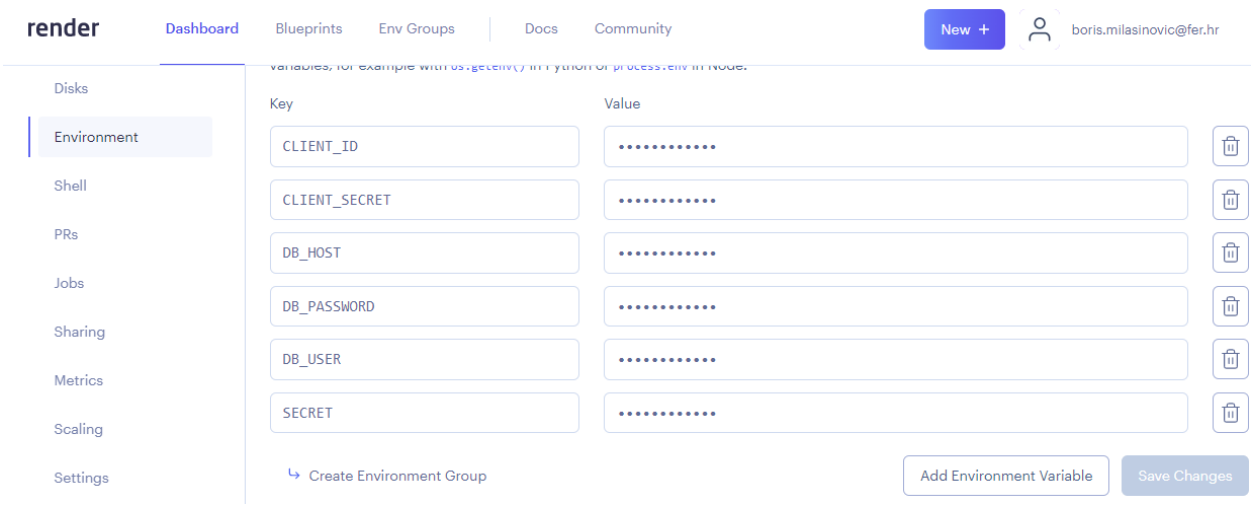
Poželite li naknadno dodati još koji repozitorij u odabir, to se može napraviti u postavkama GitHuba/GitLaba. Primjerice, na donjoj slici se vidi navedeni ekran s postavkama na GitHubu.



Prilikom kreiranja nove usluge u dijelu *General* treba odabrati neko jedinstveno ime i područje (npr. Frankfurt EU Central). U dijelu Build & Deploy potrebno je upisati sljedeće:

- *Branch*: `main` (ili kako je već nazvana glavna grana, odnosno grana za koju se definira kontinuirana isporuka)
- *Root Directory*: ostaviti prazno
- *Build Command*: `npm install;npm install typescript; npm run build`
- *Start Command:* `npm start`

U dijelu pod *Environment* potrebno je definirati varijable okruženja. Nazivi su proizvoljni i odgovaraju onome što je potrebno u programskom kodu. PORT i RENDER_EXTERNAL_URL su automatski definirani i ovdje se ne navode.



Aplikacija ne radi? – Provjeriti što piše pod *Logs*