

SC-T-213-VEFF - Web Programming I

Assignment 3 – RESTful Backend

Háskólinn í Reykjavík

Skilafrestur: **20. mars 2023, 23:59**

Viðfangsefni verkefnis: **Að skrifa RESTful bakenda**

Þetta er **hópverkefni**, sem þið megið leysa ein eða í hóp með allt að **3 nemendum**.

1. Yfirlit

Í þessu verkefni muntu þróa bakenda sem hægt er að nota til að halda utan um lög og lagategundir. Hugmyndin er sú sama og í verkefni 2 og þarf bakendinn þinn að vera samhæfður uppfærðri lausn á verkefni 2. Bakendinn býður upp á tvenns konar afurðir (e. resource) með nokkrum endapunktum sem fylgja REST hugmyndafræðinni og bestu starfsvenjum (e. best practices). Vinsamlegast athugaðu að Postman getur komið sér gríðarlega vel þegar verið er að þróa lausnir eins og þessa.

Athugið: Líkt og verkefni 2 gæti þetta verkefni virst yfirpyrmandi í fyrstu. Byrjaðu á því að hanna API eftir því sem þú lærðir í myndböndum L15/16 (skrifaðu niður HTTP aðferðir og vefslóðir fyrir hvern endapunkt). Settu síðan auðveldu endapunkta fyrst (t.d. er lestur (e. read) venjulega auðveldastur) og notaðu “dummy” gögn í upphafi (t.d. harðkóðaðan JSON eins og í fyrirlestrunum).

2. Afurðir í bakendanum

Fyrir þetta verkefni þurfum við tvær tegundir “afurða” (e. resources): lag (e. tune) og lagategund (e. genre). Lög tákna safn nóta og tímasetningu þeirra. Lagategundir eru leið til að flokka lög í mismunandi flokka.

Lag (tune) hefur eftirfarandi eiginleika:

1. *id* - Einstakt númer eða strengur sem auðkennir hvert lag.
2. *name* - Strengur sem gefur titil fyrir lagið.
3. *genreId* - Auðkenni lagategundarinnar sem þetta lag tilheyrir.
4. *content* - Safn tóna sem lagið inniheldur. Hver hlutur í þessum lista hefur nótu (*note*) á strengjasniði (*string*), lengd (*duration*) á strengjasniði og tíma (*timing*) á heiltölusniði (*int*) miðað við upphaf lagsins. Athugið og tímalengd (*duration*) og nótur (*note*) verða að vera í samræmi við Tone.js pakkann sem notaður er í verkefni 2.

Lagategund (genre) hefur eftirfarandi eiginleika:

1. id - Einstakt númer/strengur sem auðkennir hverja tegund.
2. genreName - Strengur sem lýsir tegundinni / nafn.

Þér er frjálst að innleiða þessar tvær “afurðir” eins og þú vilt í bakendanum þínum. Hins vegar þurfa bakendarnir að skila nauðsynlegum eiginleikum á tilskildu sniði.

3. Endapunktur

Eftirfarandi endapunktur skulu útfærðir fyrir **lög**. Í öllum nema fyrsta endapunkti skal meðhöndla lag sem undirefni (sub-resource) tegundar.

1. Lestu öll lög

Skilar lista allra laga. Fyrir hvert lag eru aðeins auðkennið (*id*), nafnið (*name*) og tegundarnúmerið (*genreId*) innifalið í svarinu. Að auki, bjóða upp á „sú“ (e. filter) sem fyrirspurnarfæribreytu (e. query parameter) sem skilar aðeins lögnum sem eru í tegundinni með uppgefnu nafni. Ef ekkert lag er í tilgreindri tegund, eða engin tegund með uppgefnu nafni er til, er tómum lista (e. empty array) skilað. Það er mikilvægt að skila tómum lista svo framendinn geti búist við lista, sama hver niðurstaðan er.

2. Lesið einstakt lag

Skilar öllum eiginleikum tilgreinds lags.

3. Búðu til nýtt lag

Búðu til nýtt lag. Endapunkturinn gerir ráð fyrir nafni (*name*) og innihaldi (*content*) lagsins. Tvítekin nöfn eru leyfð. *GenreId* er gefið í gegnum vefslóðina. Eigindið innihald (*content*) verður að vera “ótómur” listi af hlutum, sem hver um sig hefur eigindin nóta (*note*), tímasetningu (*timing*) og tímalengd (*duration*). Auðkennið (*id*) skal mynda sjálfkrafa (þ.e.a.s. ekki gefið upp í meginmáli (*body*) beiðinnar). Beiðnin skal mistakast ef lagategund með uppgefnu auðkenni er ekki til. Beiðnin, ef hún tekst, skal skila nýja laginu (allir eiginleikar, þ.mt auðkenni (*id*) og heildar innihald (*content*)).

4. Uppfærðu lag að hluta

Uppfærir lag sem til er, að hluta. Hægt er að uppfæra alla eiginleika nema auðkennið. Allir eiginleikar sem eru gefnir upp í beiðninni eru uppfærðir. Ef innihald (*content*) er til staðar er það alltaf að koma í stað þess sem fyrir er. Beiðnin, ef hún tekst, skilar uppfærðu laginu. Til að breyta tegundinni þarf beiðnin að búast við gamla tegundaauðkenninu (*genreId*) í vefslóðinni og nýja tegundaauðkenninu í meginmáli beiðinnar.

Eftirfarandi endapunktur skulu útfærðir fyrir **lagategundir**:

1. **Lestu** allar tegundir

Skilar lista af öllum lagategundum (með öllum eiginleikum).

2. **Búðu til** nýja tegund

Búðu til nýja tegund. Endapunkturinn gerir ráð fyrir aðeins *genreName* eigindinu í meginmáli beiðninnar. Einkvæma auðkennið skal myndast sjálfkrafa. Tvítekin nöfn fyrir tegundir eru ekki leyfð. Beiðnin, ef hún tekst, mun skila nýju tegundinni (allir eiginleikar, þar á meðal auðkenni).

3. **Eyða** tegund

Eyðir núverandi tegund. Beiðnin mun mistakast ef einhver lög eru til í þessari tegund. Beiðnin, ef hún tekst, skilar öllum eiginleikum tegundarinnar sem var eytt.

4. **Kröfur**

Fylgja skal eftirfarandi kröfum/starfsvenjum:

1. Framendakóðinn (óbreyttur) sem gefinn er í viðbótarefni þarf að virka með forritinu. Athugaðu að þetta verkefni notar mismunandi endapunkta en voru í verkefni 2, þannig að venjuleg lausn (og lausn þín) í verkefni 2 mun ekki virka með lausn þessa verkefnis.
2. Eins og í verkefni 2, ættir þú að velja að nota *arrow functions* þegar mögulegt er, ekki nota „var“ og nota innbyggðar JavaScript aðgerðir í stað venjulegra *for-lykkja* o.s.frv.
3. Verkefnið skal fylgja REST-hugmyndafræðinni.
4. Fylgja skal bestu starfsvenjum frá fyrirlestrum L15/16. Þetta þýðir að
 - Nota skal fleirtölunafnorð um afurðir (resources)
 - Taka skal á sérstökum afurðum með því að nota auðkenni (id) þeirra, sem hluta af vefslóð afurðarinnar.
 - Nota skal undirheimildir (sub-resources) til að sýna tengsl milli tegunda og laga, eins og segir í 3. grein.
 - Nota skal JSON sem meginmál beiðna/svara.
 - Nota skal HTTP sagnirnar til að lýsa CRUD aðgerðum. Fylgja skal eiginleikum “safe” (fyrir GET) og “idempotent” (fyrir DELETE og PATCH).
 - Nota skal viðeigandi HTTP stöðukóða fyrir svör. 200 ætti að nota fyrir árangursríkar GET, DELETE og PATCH beiðnir, 201 fyrir árangursríkar POST beiðnir. Í villutilvikum skal nota 400 ef beiðnin var ekki gild, 404 skal nota ef óskað var eftir auðlind sem ekki er til. 405 skal nota ef beðið er um tilföng með óstuddri HTTP sögn (t.d. reynt að eyða öllum lögum), eða ef kallað er á endapunkt sem ekki er til.
 - Þú þarft EKKI að innleiða HATEOAS/tengla.
5. Forritið/bakendinn skal birtast á <http://localhost:3000/api/v1/>. Ef þú átt í vandræðum með að keyra bakendann þinn á port 3000, hafðu samband við okkur - vinsamlegast ekki breyta portinu í lausnarkóðanum.

6. Lausnin skal vera Node.js forrit. Package.json skrá skal vera hluti af verkefninu, með lista af nauðsynlegum pökkum.
7. Forritið skal ræst með skipuninni npm start (þú mátt ekki fjarlægja möguleikann á að nota npm start).
8. Forritinu er aðeins heimilt að nota innbyggðar einingar af Node.js, sem og Express.js, body parser og cors.
9. Þér er ekki ætlað/leyft að bæta "geymslu" (e. persistence) (gagnagrunni, skráargeymslu eða álíka) við verkefnið.
10. Það eru engar takmarkanir á ECMAScript (JavaScript) útgáfunni.

5. "Start pakkinn"

Í viðbótarefninu er að finna sýnishorn af Node.js verkefni (*package.json* skrá og *index.js* skrá). *Index.js* inniheldur aðeins tvær breytur sem innihalda dæmi fyrir afurðasniðið (e. resource format) sem skilgreint er í 2. grein. Þú getur breytt framsetningu afurðarinnar og/eða bætt við gagnaskipulagi eftir þörfum. Þú ættir að nýta "start pakkann" til að útbúa bakendakóðann þinn, viðbótarskrár eru að sjálfsögðu leyfðar. Pakkarnir sem þurfa að vera til staðar í verkefninu eru nú þegar skilgreindir í *package.json* og hægt er að sækja þá með því að keyra npm install í verkefna möppunni.

6. Að keyra verkefnið í "development mode"

1. Farðu í **prj03_suppl/prj03_starter** möppuna í terminal.
2. Sæktu NPM pakkana sem listaðir eru í **package.json** með því að keyra "**npm install**" skipunina ("**npm i**" gerir það sama). Þetta mun sækja alla pakkana sem þarf til að keyra verkefnið.
3. Til að keyra verkefnið upp í þróunarham þarf að keyra "**npm run dev**", sem í okkar tilfelli þýðir að ferlið mun fylgjast með breytingum á *index.js*. Þegar kóðinn er uppfærður mun ferlið sjálfkrafa endurræsast til að endurspegla breytingarnar sem gerðar eru á kóðanum. Ferlið mun prenta út skilaboð sem segja þér að það hafi byrjað:

```
> prj03_starter@0.0.1 dev
> nodemon index.js
```

```
[nodemon] 2.0.20
```

```
[nodemon] to restart at any time, enter `rs`
```

```
[nodemon] watching path(s): *.*
```

[nodemon] watching extensions: js,mjs,json

[nodemon] starting `node index.js`

Tune app listening on port: 3000

4. Nú er bakendinn keyrandi á <http://localhost:3000> og er aðgengilegur í gegnum Postman, cURL eða vefsíðuna.

7. Verkefnaskil

Verkefninu skal skilað í gegnum Canvas. Sendu inn zip-skrá sem inniheldur allan bakendann þinn. **EKKI** senda *node_modules* möppuna með, nú verður dregið af einkunn fyrir að senda hana með. Ekki er tekið við skilum eftir að skilafrestur er liðinn.