

Tehtävänanto:

Muuttoankkojen paluu on koittanut

Tähän mennessä Vincitin aktiiviset ankkabongaajat ovat toteuttaneet pienen palvelinsovelluksen, joka on ladattavissa osoitteesta <https://github.com/Vincit/summer-2018/> Palvelinsovelluksen rajapinnat ja niiden lähettämät vastaukset ovat alla olevan määrittelyn mukaiset. Toteuta joku neljästä vaihtoehdosta ja toimita se Vincitille hakemuksesi yhteydessä.



Tähän mennessä palvelimen toiminnallisuudet ovat:

GET /sightings	Palauttaa kaikki kirjatut havainnot
POST /sightings	Lisää uuden havainnon <i>Huom! Mikäli käyttäjä lähettää havainnon, jolle hän määrittänyt id:n, palvelinsovellus korvaa annetun id:n oman toteutuksen mukaisella id:llä ja tallentaa havainnon vasta sitten palvelimelle.</i>
GET /species	Palauttaa kaikki tuetut ankkalajit

Palvelinsovelluksen tietomallit:

Sighting (Havainto)	id (yksilöivä tunniste) Havainnot yksilöivä merkkijono dateTime (utc) ISO 8601-formaatin mukainen aika (ilman aika vyöhykettä eli yleisaika) esim. 2016-12-12T10:10:00Z description Vapaamuotoinen tekstikenttä species (mallard, redhead, gadwall, canvasback, lesser scaup, ...) Ankkalaji, jonka pitää vastata jotain /species-rajapinnan palauttamista arvoista. count Lukumäärä, paljonko ankkoja on näkynyt havainnossa. Pitää olla suurempi kuin nolla.
Species (Laji)	name Ankkalajin nimi



Asiakassovellusvaihtoehdot ja asiakassovelluksen yleiset vaatimukset

- Asiakassovelluksella tulee voida listata kaikki havainnot ja lisätä uusia havaintoja. Operaatiot tulee kohdistaa tehtävänannossa kuvattuun palvelinsovellukseen.
- Havainnoista tulee näyttää tarkoituksenmukaisella tavalla kaikki sovelluksen käytön kannalta oleelliset tiedot.
- Lisättäessä uutta havaintoa tulee tarkistaa käyttäjän syöte siten, että palvelimelle lähtevät arvot ovat aina valideja. Sellainen ankkalaji, josta palvelinsovelluksessa ei ole määrittelyä, todetaan olevan virheellinen arvo.
- Havainnot tulee voida järjestää käyttöliittymässä ajallisesti joko laskevaan tai nousevaan järjestykseen.
- Palvelimen lähettämiä vastauksia ei tarvitse erikseen tarkistaa vaan voit luottaa niiden olevan aina määrittelyjen mukaisia.



Vaihtoehto A) Selainsovellus

Toteuta yksinkertainen yhden sivun web-käyttöliittymä, joka toteuttaa yllä kuvatut yleiset asiakassovelluksen vaatimukset. Toteutuksessa on suotavaa käyttää jotain nykyaikaista ohjelmistokehystä (esim. Angular tai React) sekä käyttöliittymäkirjastoa (esim. Bootstrap). Sovelluksen käyttöliittymäratkaisujen tulee olla käyttötarkoituksen mukaisia.

Selainyhteensopivuuteen ei tarvitse suuremmin kiinnittää huomiota, riittää että tekemäsi käyttöliittymä toimii ja näyttää halutunlaiselta tuoreella Google Chrome -työpöytäselaimella.

Vaihtoehto B) Mobiilisovellus

Toteuta mobiilisovellus, joka toteuttaa yllä kuvatut yleiset asiakassovelluksenvaatimukset. Sovelluksen käyttöliittymäratkaisujen tulee olla käyttötarkoituksen mukaisia.

Voit toteuttaa sovelluksen haluamallasi nykyaikaisella tekniikalla, natiivisti tai alustariippumattomasti, Androidille, iOS:lle tai molemmille.

 **HUOM! OBS! PALAUTATHAN TEHTÄVÄN HAKEMUKSEN YHTEYDESSÄ!** 



Palvelinsovellusvaihtoehdot

Palvelinsovellukseen ei tarvitse toteuttaa autentikointia, käyttäjienhallintaa jne.



Vaihtoehto C) Vapaamuotoinen palvelinsovellus

- Luo sovellus, jossa on sama rajapinta kuin vanhalla palvelinsovelluksella, mutta havaintoon kuuluu myös paikkatieto.
- Sovellus tallentaa uudet havainnot myös vanhaan palvelinsovellukseen havainnon lisäyksen yhteydessä niiden tietojen osalta, joita vanha palvelinsovellus tukee.
- Luotava sovellus kykenee havaintojen yleiseen listaamiseen, kuten olemassa oleva palvelinsovellus.
- Luotavan sovelluksen tulee palauttaa vastaukset HTTP-pyyntöihin JSON-muodossa.
- Luotava sovellus voi sisältää muita mielenkiintoisia toiminnallisuuksia saman datan pohjalta toimivina (vapaaehtoinen).

Toteuta itse valitsemallasi sovelluskehysellä palvelinsovellus, joka toteuttaa listatut vaatimukset.

Vaihtoehto D) Olemassa olevan palvelinsovelluksen uudelleentoteutus

- Luo palvelinsovellus, joka toteuttaa kaikki olemassa olevan palvelinsovelluksen toiminnallisuudet jollain modernilla sovelluskehysellä.
- Luo järjestelmälle testit, joilla toimivuus varmistetaan.

Sovelluksen tulee kyetä keskustelemaan HTTP-pyyntöjen avulla samojen sovellusten kanssa, jotka keskustelevat alkuperäisen palvelinsovelluksenkin kanssa. Tarkoituksena on siis luoda korvike vanhalle projektille.

⚠️ HUOM! OBS! PALAUTATHAN TEHTÄVÄN HAKEMUKSEN YHTEYDESSÄ! ⚠️