



AKADEMIJA.IT

INFOBALT IR TECH CITY

PRAKTIŅĖ UŽDUOTIS (PERLAIKYMAS)

Java Technologijos ir Priemonės

Andrius Stašauskas

`andrius@stasauskas.lt`

<http://stasauskas.lt/itpro2017/>

LAIKAS

- Pradžia
 - 8:55 (8:00)
- Pabaiga
 - 12:25
- Patarimas: vėliausiai 12 valandą atlikite git commit ir git push į repozitoriją, kad liktų laiko, jeigu kažkas nepavyksta, net jeigu užduoties dar nepabaigėte
- Vėliau negu 12:25 įkelti pakeitimai nebus vertinami
- Užduotis tikrinimui bus paimama komanda git clone, leidžiamos komandos mvn eclipse:clean idea:clean, mvn clean install, startuojamas serveris mvn org.apache.tomcat.maven:tomcat7-maven-plugin:2.2:run-war



UŽDUOTIS (1/2)

- Sukurti paslaugų registro programą
 - egzistuoja galimybė pridėti paslaugą
- paslaugos atributai: pavadinimas, kaina, asociatyvinė nuotrauka, kategorija, aprašymas
 - galimybė prie paslaugos pridėti ir iš jos pašalinti paslaugos teikėją
- paslaugos kategorijos: IT, maistas, transportas, grožio paslaugos, ugdymo paslaugos
- teikėjo atributai: pavadinimas, miestas, įmonės kodas, įvertinimas
 - nuo paslaugos priklausantys: tos pačios paslaugos kaina gali būti skirtinga kiekvienam teikėjui
- paslaugos teikėjo tipas: įmonė, privatus asmuo, valstybinė
- naudotojas gali pasirinkti paslaugų kategoriją, mato paslaugas pagal kainą ir gali jas užsakyti iš tam tikro paslaugų teikėjo
 - egzistuoja galimybė peržiūrėti užsakytas paslaugas



UŽDUOTIS (2/2)

- Tas pats teikėjas gali būti priskirtas skirtingoms paslaugoms
- Tiek paslaugos, tiek teikėjo pavadinimai turi būti unikalūs
- Paslaugos peržiūroje prie kiekvieno teikėjo parašyta kaina
- Paslaugos peržiūroje rodoma, kiek kartų paslauga suteikta
- Egzistuoja galimybė peržiūrėti visus teikėjus
- Egzistuoja galimybė pakeisti informaciją apie teikėją
- Egzistuoja galimybė pašalinti teikėją
- Egzistuoja galimybė ieškoti teikėjo paslaugos peržiūros lange
 - Teikėjus filtruoti galima pagal teikėjo tipą



PATEIKIMAS

- GitHub/BitBucket Public Repository
 - repository turi būti naujas
- Naudojamas Git
- Nuorodas į repozitoriją įkelti čia
 - <http://stasauskas.lt/vatas>
- Nuoroda turi veikti su git clone
- Projektas turi būti švarus
 - be target/build katalogų, be IDE projekto failų
- repozitorijoje turi būti tik **katalogai** kodas-perlaikymo-app ir kodas-perlaikymo-ui
 - kur kodas - jūsų kodas, įrašytas teste
 - kodas-perlaikymo-app kataloge turi egzistuoti pom.xml failas
 - kodas-perlaikymo-ui kataloge turi egzistuoti package.json failas



STRUKTŪRA

- susideda iš 1 (**vieno**) maven projekto
 - kodas-perlaikymo-app (war tipo)
- ir iš 1 (**vieno**) npm projekto
 - kodas-perlaikymo-ui (npm tipo)
- aplikacija turi būti kataloge kodas-perlaikymo-app
- aplikacijos pavadinimas - kodas-perlaikymo-app
- aplikacijos rezultatas - kodas-perlaikymo-app.war failas
- aplikacijos paketas - lt.perlaikymas
- aplikacija turi kompiliuotis be klaidų
- aplikacija neturi mesti išimčių naršyklėje ir konsolėje
- aplikaciją turi būti galima paleisti naudojant komandą:
 - mvn org.apache.tomcat.maven:tomcat7-maven-plugin:2.2:run-war



SERVISAI

- Turi būti naudojami tik Spring beans'ai
- bent vienas Spring bean turi būti aprašytas xml faile
- Turi egzistuoti bent vienas funkcionalus request apimties bean
- Servisai turi būti singleton apimties, bet juose negali būti saugomi duomenys
- Duomenys gali būti saugomi prototype/request bean (sąrašai, sąrašų elementai, kurie vėliau saugomi DB, ir pan.)
- Duomenimis dalintis tarp Rest ir Controller ir Servisų turi būti naudojamos skirtingos klasės negu dirbant su DB



SERVISAI

- Visi service ir request apimties bean'sai turi turėti logginimą į žurnalą (java, log4j, slf4j arba kitokį):
 - prieš susikurdami turi apie tai informuoti LOG žinute
 - žinutėje turi matytis pirma apimtis (scope), ir tada bean name arba klasė
- System.out naudoti negalima



DUOMENŲ BAZĖ (1/2)

- Duomenų struktūra turi tenkinti užduoties reikalavimus
- Įvesti duomenys turi išlikti aplikaciją netikėtai išjungus
- Įjungus turi užsikrauti prieš tai išsaugoti duomenys
- Aplikacija turi būti sukurta taip, kad veiktų tiek su tuščia, tiek su užpildyta duomenų baze
- Trinant paslaugą, kartu turi išsitrinti ir susieti teikėjai, jeigu jie priskirti tik tai vienai paslaugai



DUOMENŲ BAZĖ (2/2)

- DB reikalavimai
 - turi būti panaudotas JPA darbui su DB
 - aplikacija turi būti sukurta taip, kad veiktų tiek su tuščia, tiek su užpildyta duomenų baze
 - paslaugų/teikėjų filtravimas turi vykti per DB
 - turi būti parašyta bent viena sudėtinga DB SQL užklausa
 - trinant paslaugą, jos sąsajų turi nelikti ir prie teikėjų



NAVIGACIJA

- Svetainė turi būti 1 neatsinaujinantis naršyklės puslapis
- Navigacijai tarp puslapių naudojamas React Router
- Navigacija gali vykti tiek naudojantis nuorodomis, tiek mygtukais
- Pirminis puslapis, į kurį patenkama, turi būti paslaugų sąrašo puslapis
 - į jį patekti turi būti galima tiesiog atidarius aplikaciją naudojant tokią <http://localhost:8080/kodas-perlaikymo-app> ar panašią nuorodą



VAIZDAI (1/2)

- Aplikacija turi turėti bent tokius atskirus vaizdus (js):
 - paslaugų sąrašas
 - paslaugos peržiūros langas su teikėjų sąrašu
 - paslaugos pridėjimo/redagavimo langas
 - teikėjo pridėjimo langas
- Vaizdo komponentas turi leisti nustatyti puslapio title
- Vaizdo (presenatation) komponente turi būti tik:
 - title - puslapio pavadinimo eilutė
 - pats komponento turinys
- Vaizdai neturi būti išdarkyti, informacija aiškiai matytis, elementai išdėstyti struktūriškai, ekrano ribose



VAIZDAI (2/2)

- Vaizduose esančios formos turi leisti dirbti su visais duomenimis, esančiais duomenų bazėje
- Naudojami resursai negali būti išoriniai - viskas turi būti war faile
- Vaizduose privalo būti prijungtas 1 CSS failas, spalvoms ir kitiems UI pakeitimams turi būti naudojamas šis failas, o vaizduose nustatomos CSS klasės
 - style atributai nenaudojami
 - style per JavaScript nenustatomas (išskyrus CSS klases)



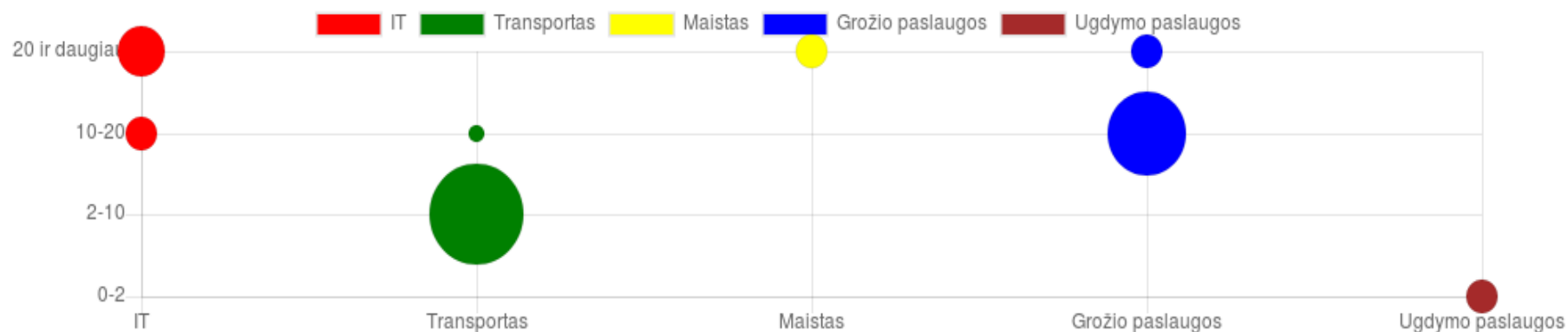
SĄRAŠŲ VAIZDAI (1/2)

- Duomenų sąrašai turi būti išvedami naudojant [Bootstrap table](#)
- Jeigu nėra paslaugų arba jų mažiau negu 5, turi pasirodyti [Bootstrap alert](#) su pranešimu, jog trūksta paslaugų. Jo neturi matytis, pridėjus 5-ą paslaugą (turi iškart pasislėpti) ir jeigu paslaugų yra daugiau
 - tarp paslaugų turi būti bent viena transportavimo paslauga, kitaip rodomas `alert` pranešimas
- Analogiškas reikalavimas teikėjų sąrašui - paslauga turi turėti bent 1 teikėją
- Duomenis trinant turi būti panaudotas [Bootstrap modal](#) ir pasirodyti dialogas su paklausimu, ar tikrai trinti duomenis



SĄRAŠŲ VAIZDAI (2/2)

- Parindiniame lange po paslaugomis turi būti parodyti teikėjų kiekiai pagal paslaugos kategoriją naudojant Bubble diagramą iš [react-chartjs-2](#) ([hint hint2](#)) (burbulo dydis nusako kainą)
- y ašyje - teikėjai kategorijose pagal kiekį: 0-2, 2-10, 10-20, 20 ir daugiau
- x ašyje - paslaugų kategorija
- burbulo dydis - pagal kainų dydį
- Rekomenduojama naudoti react-chartjs-2 versiją 2.6.4 su chart.js versija 2.7.1



REDAGAVIMO VAIZDAI (1/2)

- Viršuje redagavimo lange matomi egzistuojantys teikėjai
 - jame turi matytis visi aplikacijoje esantys teikėjai (pavadinimas ir kaina)
 - naujus teikėjus galima pridėti ir jie atsiras šioje vietoje
- Teikėjus reikia išrinkti iš DB teikėjų lentelės, jie turi būti surūšiuoti DB lygyje pagal paslaugos kainą
- Teikėjai priskiriami prie paslaugos mygtuko paspaudimu
- Redagavimo vaizduose esantys laukai turi validuotis, spalvinimui naudoti [Bootstrap validation states](#):
 - validuojama, ar laukas privalomas
 - validuojamas lauko dydis
 - validuojama, ar įvestas skaičius



REDAGAVIMO VAIZDAI (2/2)

- Redagavimo languose formoms turi būti naudojamas [Bootstrap Form CSS](#)
- Formų laukai ir jų pavadinimai turi tarpusavyje lygiuotis
- Formoje turi būti kuo mažiau kodo (formos laukai ir jų išdėstymas)
 - visas puslapis neturi būti vienoje formoje
- Teikėjo pridėjimo formoje teikėjo įvertinimą galima pasirinkti radio button nuo 1 iki 5
- Šalinant teikėją, jeigu jis priskirtas daugiau negu 1 paslaugai, trynimo pranešime turi būti parodytas įspėjimas, kurioms dar paslaugoms teikėjas priskirtas



REST REIKALAVIMAI

- Rest servisuose turi būti galima:
 - CRUD operacijos su paslaugomis ir teikėjais
 - gauti tiek sąrašą, tiek vieną paslaugą ar paslaugos teikėją
 - paslaugų iš teikėjo užsakymas kiekvienam naudotojui atskirai pagal jo username
 - paslaugų filtravimas pagal pavadinimą
 - paslaugos ir teikėjo susiejimas ir atsiejimas
 - turi būti pasiekiamas Swagger per /swagger-ui.html



SĒKMĒS



AKADEMIJA.IT

INFOBALT IR TECH CITY