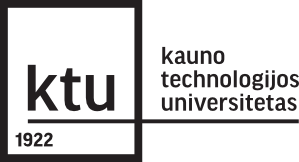
KAUNAS, 2015

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INformatikos fakultetas

Informatikos STUDIJŲ PROGRAMA



Programavimo praktika

Web serviso kūrimo grupė:

Andrius Lukas Narbutas

Danielius Strašunskis

Lukas Kriaučiūnas

Justinas Linkevičius

Karolis Skrupskelis

Darbo vadovas:

Vaidas Semėnas

Turinys

[1. Užduotis 3](#_Toc416721212)

[2. Užduoties analizė ir siūlomo sprendimo metodas 4](#_Toc416721213)

[3. Sistemos architektūra 4](#_Toc416721214)

[4. Duomenų perdavimo aprašymas 6](#_Toc416721215)

[5. Realizuotų sistemų UML diagramos ir jų aprašymai 11](#_Toc416721216)

[6. Diegimo instrukcija 12](#_Toc416721217)

[7. Testavimas ir jo rezultatai 13](#_Toc416721218)

[8. Išvados 14](#_Toc416721219)

[9. Priedai 15](#_Toc416721220)

# Užduotis

**Sistema „Paruošk formą“**

Sistema „Paruošk formą“ bus skirta elektroninių formų kūrimui ir pildymui. Naudodamiesi šia sistema individualūs asmenys ir įmonės galės susikurti įvairių formų (pvz. prekių ar paslaugų užsakymo, atsiskaitymo ir pan.) šablonus ir juos paviešinti. Kiti sistemos vartojai gali užpildyti jiems reikalingas formas. Užpildytų formų duomenis jų savininkai gali naudoti savo reikmėms.

**Funkciniai reikalavimai**

* Sistemoje gali užsiregistruoti naujas naudotojas;
* Sistemos naudotojas turi galėti sukurti ir išsaugoti norimą formos šabloną;
* Sistemos naudotojas turi galėti užpildyti pasirinktą formą;
* Sistemos naudotojas turi galėti peržiūrėti savo sukurtų formų šablonų sąrašą;
* Sistemos naudotojas turi galėti pašalinti norimą formos šabloną;
* Sistemos naudotojas turi galėti peržiūrėti užpildytų formų duomenis.

**Nefunkciniai reikalavimai**

* Prie sistemos gali prisijungti tik registruoti vartotojai;
* Sistema turi turėti 2 vartotojo sąsajos tipus:
  + Darbastalio (Desktop) programą,
  + Mobilią programėlę,
* Duomenys saugomi duomenų bazėje ir prieinami tik per WEB servisus;
* Visos sistemos dalys įgyvendinamos naudojant Java technologijas (jei reikia galima naudoti HTML, CSS, JavaScript ir kitas WEB technologijas);
* Duomenų mainams naudojamas JSON duomenų perdavimo formatas.

# Užduoties analizė ir siūlomo sprendimo metodas

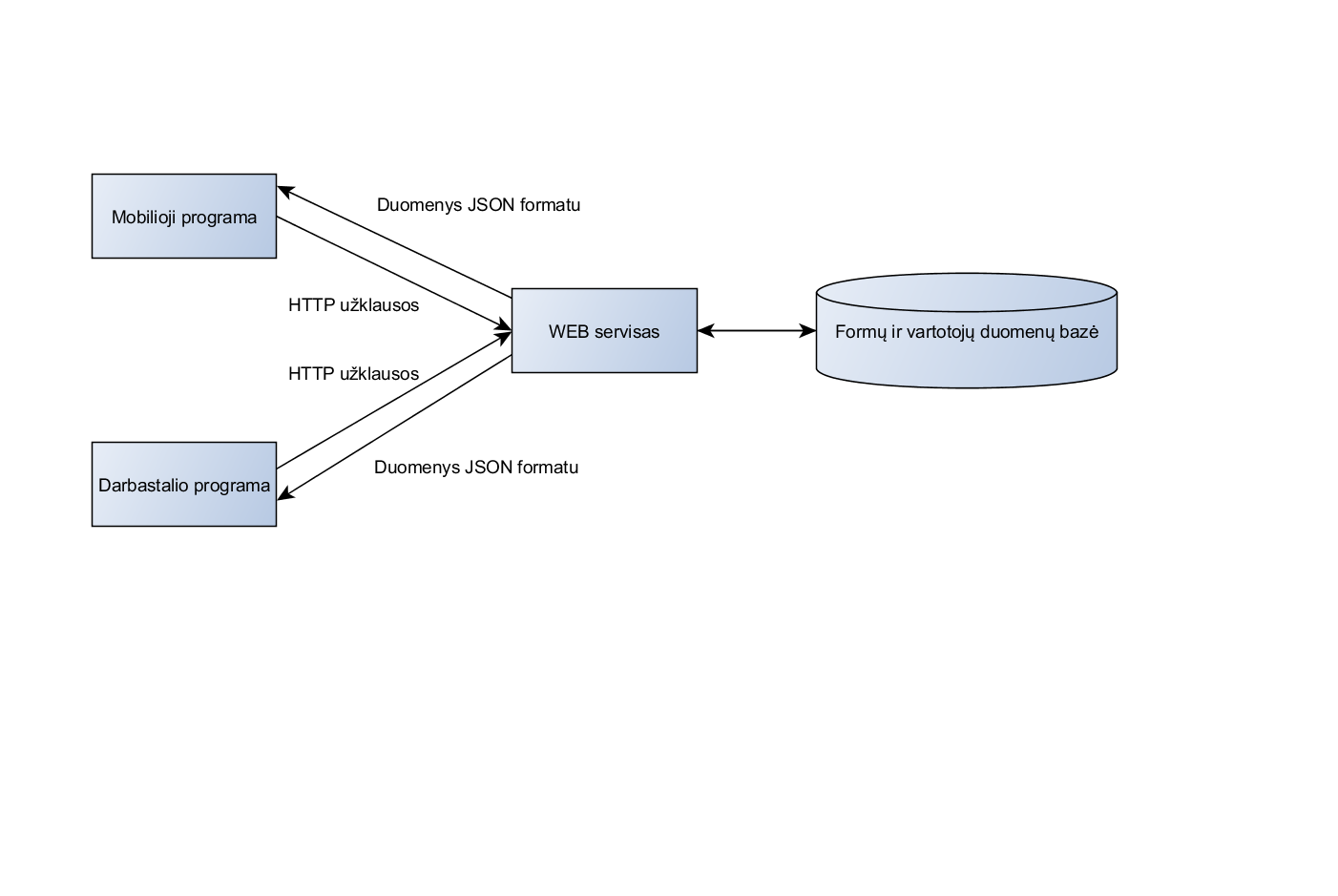
Sistemą sudarys trys atskyros posistemės: darbalaukio, mobilioji ir WEB servisas pateikiantis duomenis pirmoms dviems.

Darbalaukio ir mobilioji programos turės funkcionalumą registruoti vartotojus – neregistruoti vartotojai galės pildyti formas, tačiau negalės kurti naujų.

Registruoti ir prisijungę vartotojai galės kurti naujas formas (formą sudaro 5-20 laukų) ir matyti informacija apie jau sukurtas savo formas, t.y. peržiūrėti pačia forma, matyti kaip ją užpildė kiti žmonės ir informacija apie pildžiusius žmones. Laukų tipai galės būti tokių tipų: skaičius, el. paštas ([\*@\*.\*](mailto:*@*.*) formatu), vieno pasirinkimo klausimas, kelių pasirinkimų klausimas. Formos bus kuriamos dedant po vieną lauką ir pasirenkant to lauko atributus (tipą, ar laukas gali būti tuščias ir panašiai).

WEB servisas bus kuriamas naudojant Jersey karkasą. Vartotojų prisijungimui bei resursų paskirstymui pagal atitinkamus URL bus naudojamos karkaso klasės. Mobilioji bei darbalaukio programos WEB servisui siųs HTTP užklausas kartu su duomenimis JSON formatu, WEB servisas apdorojęs užklausą ir atrinkęs duomenis iš duomenų bazės darbalaukio ar mobiliąjai programai atsakymą siųs taip pat JSON formatu. Užklausos servisui bus siunčiamos REST principu. Formas bus galima filtruoti pagal pavadinima ir tagus. Funkcijoms reikalaujančioms prisijungimo naudosime <http://en.wikipedia.org/wiki/Basic_access_authentication>. Klientas siūs naudotojo vardą ir slaptažodį kiekvienai užklausai. Anoniminiai vartotojai naudos vardą „Anonymous“, ir tuščią slaptažodį. Kadangi HTTP BASIC AUTH metodu siunčiami slaptažodžiai nėra šifruojami, rekomenduojama naudoti HTTPS.

# Sistemos architektūra



# Duomenų perdavimo aprašymas

**BENDRAS PRANEŠIMŲ FORMATAS**

REST standartas nusako, kad kiekviena serverio operacija gali būti apibrėžta 4 operacijomis:

- GET - paimti duomenis iš serverio

- PUT - sukurti/perrašyti duomenis

- POST - padėti duomenis į serverį

- DELETE - trinti duomenis iš serverio

Mes naudosime HTTP protokolą komunikacijai, ir perduosime duomenis JSON formatu.

Testuojant bus naudojamas 17303 portas HTTP protokolui, ir 17304 portas HTTPS protokolui, kadangi jie nereikalauja root teisių unix sistemose. Tiksliai dar nėra nuspręsta kokie portai bus naudojami (standartiniai ar ne), todėl klientas turėtų turėti galimybę juos pasirinkti.

Naudojamos užklausos:

> /user

> metodas POST

> priregistruoja naują vartotoją

> pateikiami duomenys (JSON formatu)

- uname - naudotojo vardas (string)

- pass - slaptažodis (string)

\* company - kompanijos pavadinimas (jei kompanija) (string)

\* name - vardas (jei asmuo) (string)

\* surname - pavardė (jei asmuo) (string)

> /forms/id

> metodas GET

> grąžina pasirinktą formą

> rezultatas: žiūrėti "FORMOS FORMATAS" sekciją, "iš serverio siunčiamos formos formatas" skyrių

> metodas PUT

> išsiunčia formą

> žiūrėti "FORMOS FORMATAS" sekciją, "publikuojamos formos formatas" skyrių

> metodas DELETE

> ištrina formą

> /answers/naudotojovardas/formospavadinimas

> metodas POST

> formos užpildymas

> žiūrėti "FORMOS FORMATAS" sekciją, "atsakymo formatas" sekciją

> /stats/naudotojovardas/formospavadinimas

> metodas GET

> gauna info apie rezultatus

> žiūrėti "FORMOS STATISTIKA" sekciją

> /forms

> metodas GET

> vykdo paiešką

> parametrai yra perduodami per adresą naudojant tokį formatą: /forms?param1=val1&param3=val3+with+spaces ( http://en.wikipedia.org/wiki/Query\_string )

> galimi parametrai:

\* q: paieškos eilutė

\* tags: kableliais atskirti tag'ai, jei norima panaikinti tam tikrus rezultatus, naudojamas "-" ženklas, pvz: -food,gaming

\* sort: rikiavimo tipas, galimos reikšmės: "date" (numatytasis), "relevance", "popularity"; daugiau tipų gali būti pridėta ateityje

\* limit: apriboti rezultatų kiekį, pvz: 10

\* skip: praleisti pirmus x rezultatus

\* order: ascending - didėjančia tvarka, descending - mažėjančia tvarka

\* author: nusako autoriaus username'a

\* searchtags: nusako ar query gali turėti tag'ų (0 arba 1)

\* allow-anon: filtruoti pagal tai, ar leidžiama pildyti anonimams (1 - tik anonimines formas, 0 - tik neanonimines)

\* finished: filtruoti pagal tai, ar užpildėme formą

> rezultatas: žiūrėti "PAIEŠKOS REZULTATŲ FORMATAS" sekciją

> metodas POST

> sukuria naują formą

> žiūrėti "FORMOS FORMATAS" sekciją, "publikuojamos formos formatas" skyrių

> rezultatas: grąžina formos id: { "id": 123 }

> /user/naudotojovardas

> metodas GET

> gražina info apie naudotoją json formatu:

{

"company": "kompanijos pavadinimas"

}

>>> arba

{

"name": "vardas/vardai",

"surname": "pavarde/pavardes"

}

> metodas PUT

> atnaujina informaciją apie klientą

> įrašomi tik tie laukai, kurie bus keičiami

> metodas DELETE

> ištrina vartotoją

**PAIEŠKOS REZULTATŲ FORMATAS**

paieškos rezultatai yra pateikiami kaip objektas turintis objektų masyvą:

{

"count": 10,

"results": [ {...}, {...}, {...}, ... ]

}

Naudojami tokie objektai:

{

"name": "battletoads",

"author": "frogman",

"tags": ["tag1","tag2","tag3"],

"date": "20150125 13:58:12",

"finished": false

}

**FORMOS FORMATAS**

Iš serverio siunčiamos formos formatas:

{

"name": "battletoads",

"author": "frogman",

"desc": "questions about computer games",

"tags": ["tag1","tag2","tag3"],

"date": "20150125 13:58:12",

"finished": false,

"allow-anon": false,

"public": true,

"show-results": true,

"questions": [

{

"name": "do you guys have battletoads?",

"desc": "Battletoads is a platformer video game created by Tim and Chris Stamper and developed by Rare as the first installment of the Battletoads series",

"type": "one-choice",

"allow-empty": false,

"choices": ["Yes", "No"],

},

{

"name": "what kind of games do you like playing?",

"type": "multi-choice",

"allow-empty": true,

"allow-custom": "string",

"choices": ["FPS games","RPG games","2d scrollers"]

},

{

"name": "how old are you?",

"type": "integer",

"min-val": 10,

"max-val": 200,

}

]

}

Publikuojamos formos formatas yra labai panašus, tik neturi „date“, „id“, „author“ ir „finished“ laukų.

> formą sudaro:

- "name" - trumpas pavadinimas (string)

\* "author" - autoriaus usernam'as (string)

\* "desc" - paaiškinimas (string)

\* "tags" - tag'ai (array of strings)

% "date" - publikavimo data (string)

% "finished" - ar užbaigta? (boolean)

- "allow-anon" - ar galima anonimiškai spręsti? (boolean)

- "public" - ar galima viešai rasti? (boolean)

- "show-results" - ar viešai galima rodyti rezultatus? (boolean)

- "questions" - klausimai (array of objects)

> klausimą sudaro:

- "name" - trumpas klausimas (string)

\* "desc" - ilgesnis klausimo paaiškinimas (string)

- "type" - tipas (string)

# papildomi nustatymai...

> galimi klausimų tipai:

> "integer" - skaičius

> "string" - teksto eilutė

> "email" - el paštas (\*@\*.\*)

> "one-choice" - vienas pasirinkimas

> "multi-choice" - keli pasirinkimai

> papildomi nustatymai:

\* "allow-custom" - leisti savus pasirinkimus (argumentas yra bet koks tipas išskyrus "one-choice" ir "multi-choice") (galioja tik "one-choice" ir "multi-choice" tipams) (string)

# pastaba: jei šis argumentas pažymėtas, galima naudoti elementarių tipų apribojimus, ir jie bus pritaikomi "custom" laukui

- "allow-empty" - leisti neatsakyti į šį klausimą (boolean)

\* "default" - numatyta reikšmė (tipas priklauso nuo klausimo tipo)

> "integer" - integer

> "string", "email" - string

> "one-choice" - integer

> "multi-choice" - array of boolean

> papildomi tipų apribojimai:

- "integer":

\* "min-val" - minimali reikšmė (integer)

\* "max-val" - maksimali reikšmė (integer)

- "string":

\* "min-val" - minimalus eilutės ilgis (integer)

\* "max-val" - maksumalus eilutės ilgis (integer)

\* "allow-ws" - ar leisti tarpus (whitespace) (boolean)

\* "allow-nl" - ar leisti newline simbolius (boolean)

- "email":

\* "min-val" - min naudotojo vardo ilgis (integer)

\* "max-val" - max naudotojo vardo ilgis (integer)

\* "allowed-providers" - leidžiami el. pašto tiekėjai (array of strings)

- "multi-choice":

\* "min-choices" - minimalus kiekis pasirinkimų

\* "max-choices" - maksimalus kiekis pasirinkimų

atsakymo formatas:

{

"name": "battletoads",

"author": "frogman",

"answers": [

{ "id": 1 },

{ "selected": [true, false, true], "custom": ["battletoads"] },

{ "val": 20 }

]

}

> atsakymų objektai:

> "integer":

- "val" - atsakymas (integer)

> "string", "email":

- "val" - atsakymas (string)

> "one-choice":

- "id" - asakymo id (pradedant nuo 1). jei 0 - klausimas neatsakytas, arba naudojamas "custom" elementas (tipas: integer)

\* "custom" - jei egzistuoja, klientas pasirinko "custom" atsakymą (tipas: integer arba string)

> "multi-choice":

- "selected" - pasirinkti atsakymai (array of boolean)

\* "custom" - pasirinkti custom atsakymai (array of integer or string (depends on allowed custom values in question))

**FORMOS STATISTIKA**

formos statistikos formatas:

{

"numvotes": 50,

"answers": [

{ "choices": ["ch1", "ch2"], "votes": [25, 20] },

...

]

}

- numvotes - kiek balsų iš viso (integer)

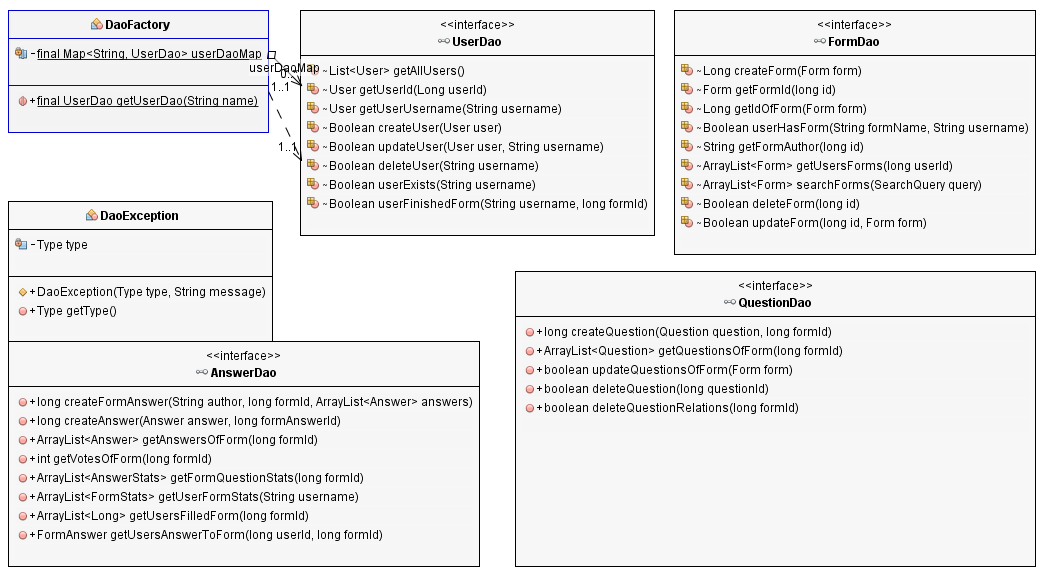
- answers - atsakymai (array of object)

"choices" - galimi pasirinkimai (array of string or integer)

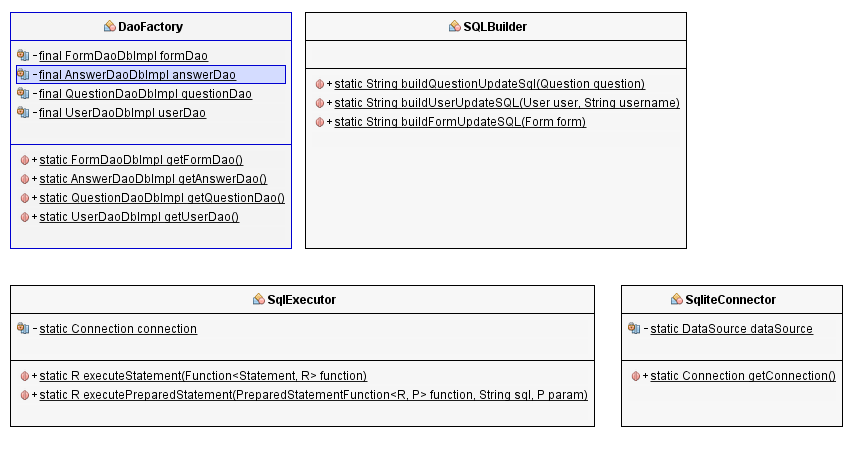
"votes" - atsakymai į tuos pasirinkimus (array of integer)

# Realizuotų sistemų UML diagramos ir jų aprašymai

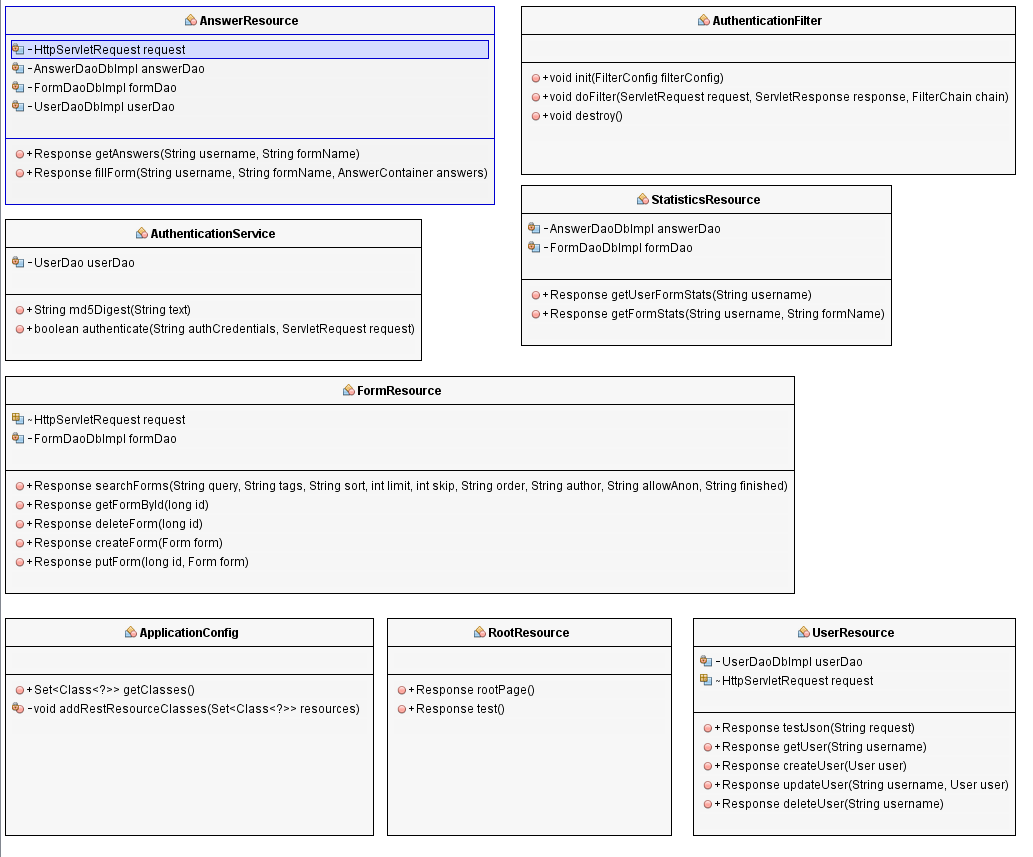
**ktu.formbackend.dao:**



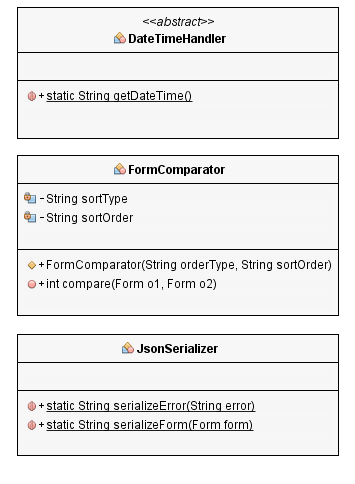
**ktu.formbackend.dao.impl.db:**



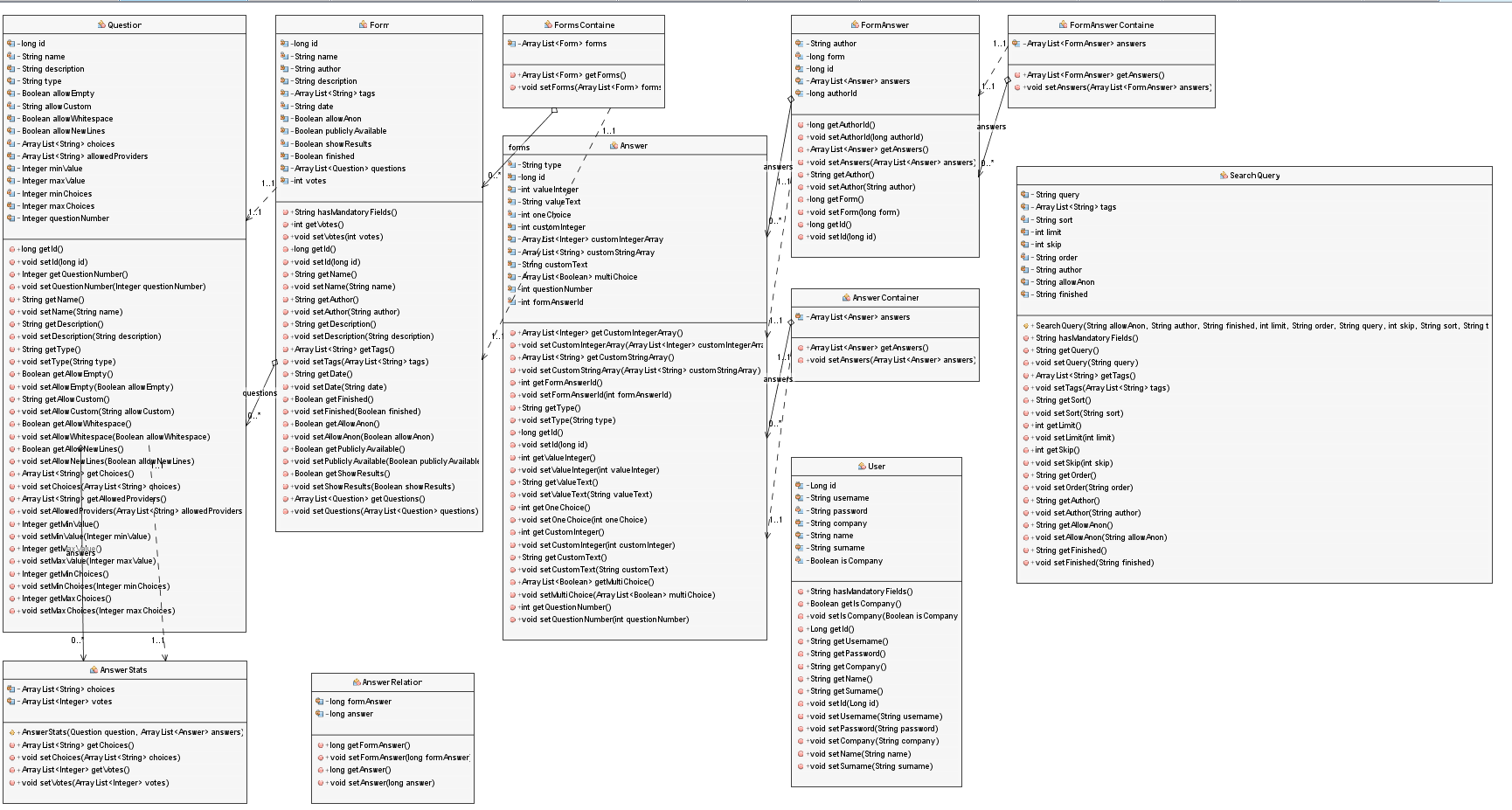
**ktu.formbackend.rest:**



**ktu.formbackend.utility:**



**ktu.formbackend.model:**



# Diegimo instrukcija

1. **Paruošimas**:
   1. Parsisiųskite ir instaliuokite savo kompiuteryje JDK 8 iš: *http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html*
   2. Parsisiųskite ir instaliuokite savo kompiuteryje Netbeans 8 iš: *https://netbeans.org/downloads/*
   3. Parsisiųskite Tomcat 8: *https://tomcat.apache.org/download-80.cgi ir išskleisti į norimą direktoriją({tomcat\_dir})*

Nuėjus į tomcat direktoriją /bin/catalina.bat pakeičiam 196 ir 201 eilutes į:

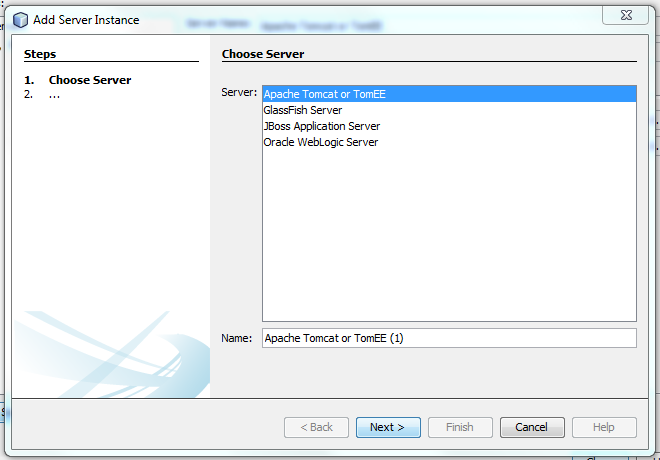
set JAVA\_OPTS=%JAVA\_OPTS% %LOGGING\_CONFIG%

ir

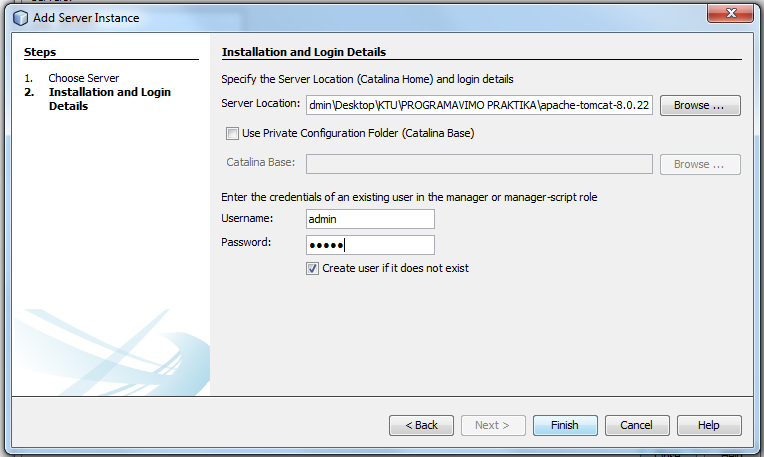
set JAVA\_OPTS=%JAVA\_OPTS% %LOGGING\_MANAGER%

* 1. Parsisiųskite SQLite 3.8.7 iš: https://bitbucket.org/xerial/sqlite-jdbc/downloads/sqlite-jdbc-3.8.7.jar, parsisiųstą failą įkelkite į {tomcat\_dir}lib/ direktoriją.
  2. Duombazės failą („database.db“) įdėkite į {tomcat\_dir}/database/ direktoriją.

1. **Diegimas**
   1. Paleidus Netbeans 8. Viršutiniame meniu pasirinkite Tools->Servers, tada “Add Server”



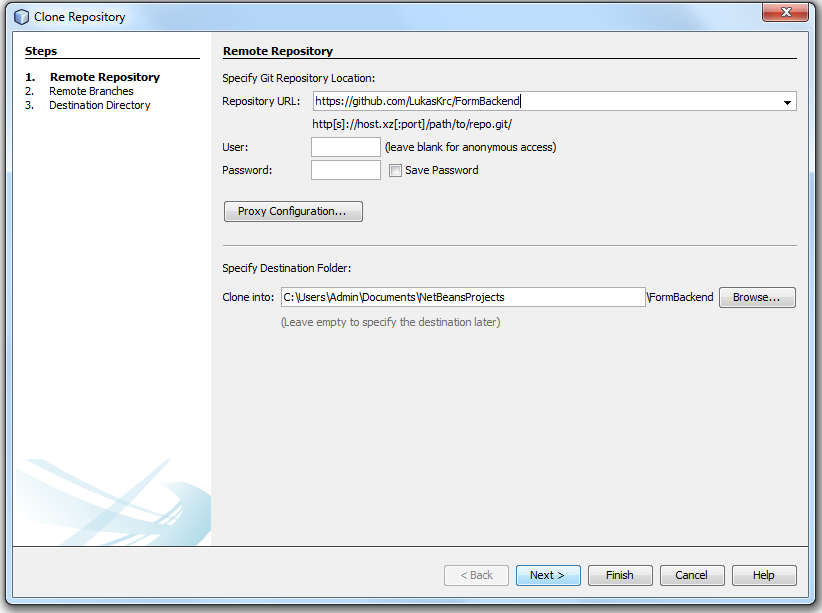
Pasirinkite Tomcat Serverį ir spauskite „Next“



Server Location laukelyje pasirinkite direktoriją kurioje yra išskleistas Tomcat 8 serveris.

Username: „admin“, Password: „admin“. Tada spauskite „Finish“ ir „Close".

* 1. Netbeans programoje iš viršutinio meniu pasirenkame „Team->Git->Clone“

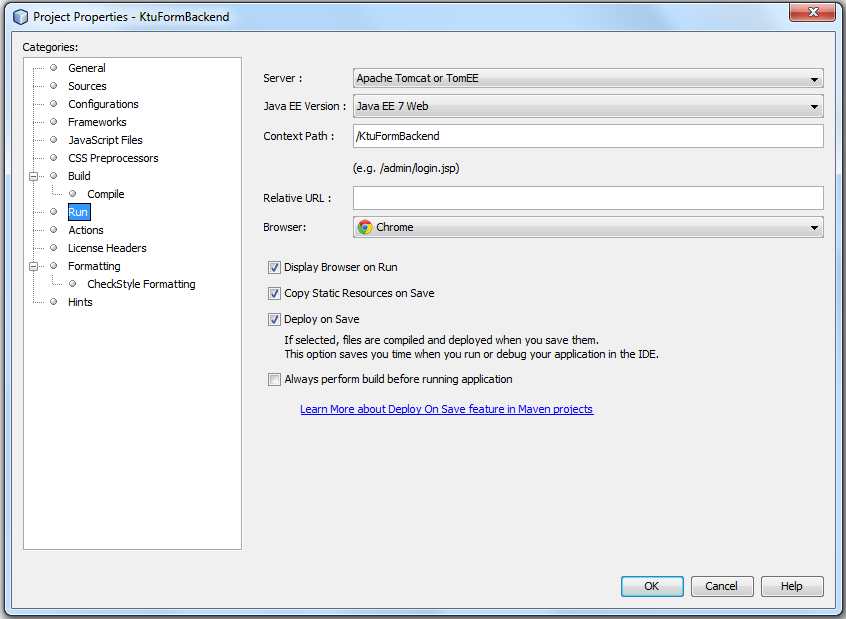


Repository URL nustatykite į: *https://github.com/LukasKrc/FormBackend*

User, Password laukus galite palikti tuščius.

Spaudžiame „Finish“.

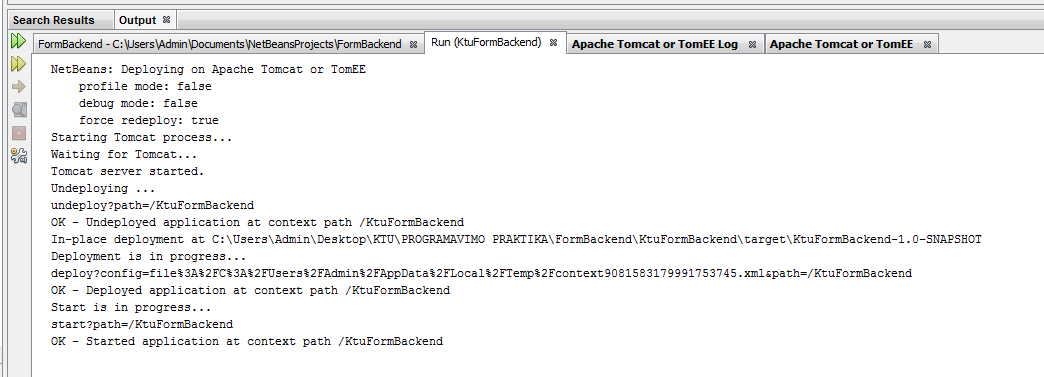
Išsaugome projektą. Spaudžiame dešinį pelės klavišą ant projekto, spaudžiame „Properties->Run“ ir prijungiame prie projekto mūsų prieš tai sukurtą Tomcat 8 serverį.



Spaudžiame „OK“

1. **Paleidimas**

Spaudžiame F6 (Run) ir laukiame kol projektas pasileis.

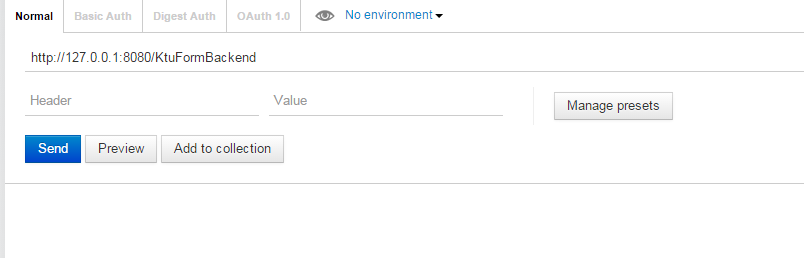


Paleidus projektą galima pradėti testuoti su kokiomis nors priemonėmis pavyzdžiui

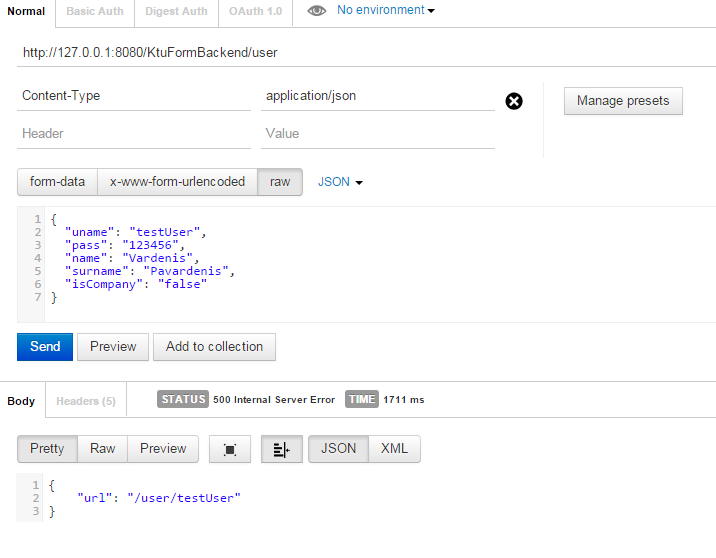
Google Chrome įskiepis „PostMan Rest Client“

# Testavimas ir jo rezultatai

Projektą testuosime su „PostMan REST Client“ papildiniu Chrome naršyklei. Atsidarę papildinį testavimą pradedame įrašę projekto URL:

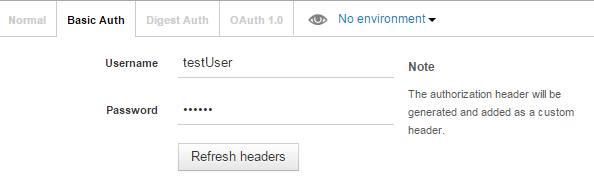


1. Vartotojo kūrimas sitemoje:

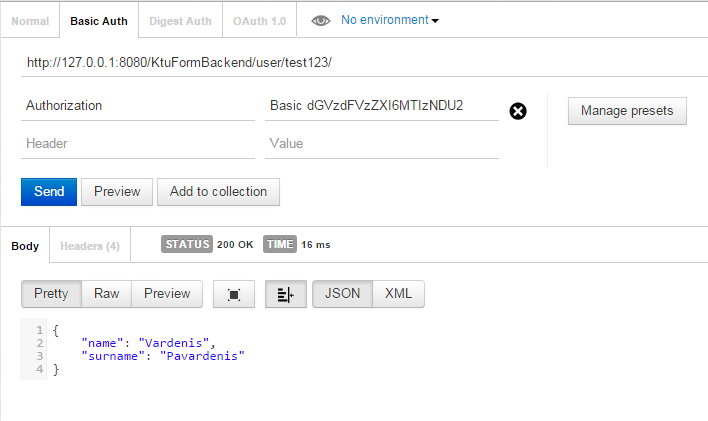


Pateikiame duomenis nurodytu JSON formatu.

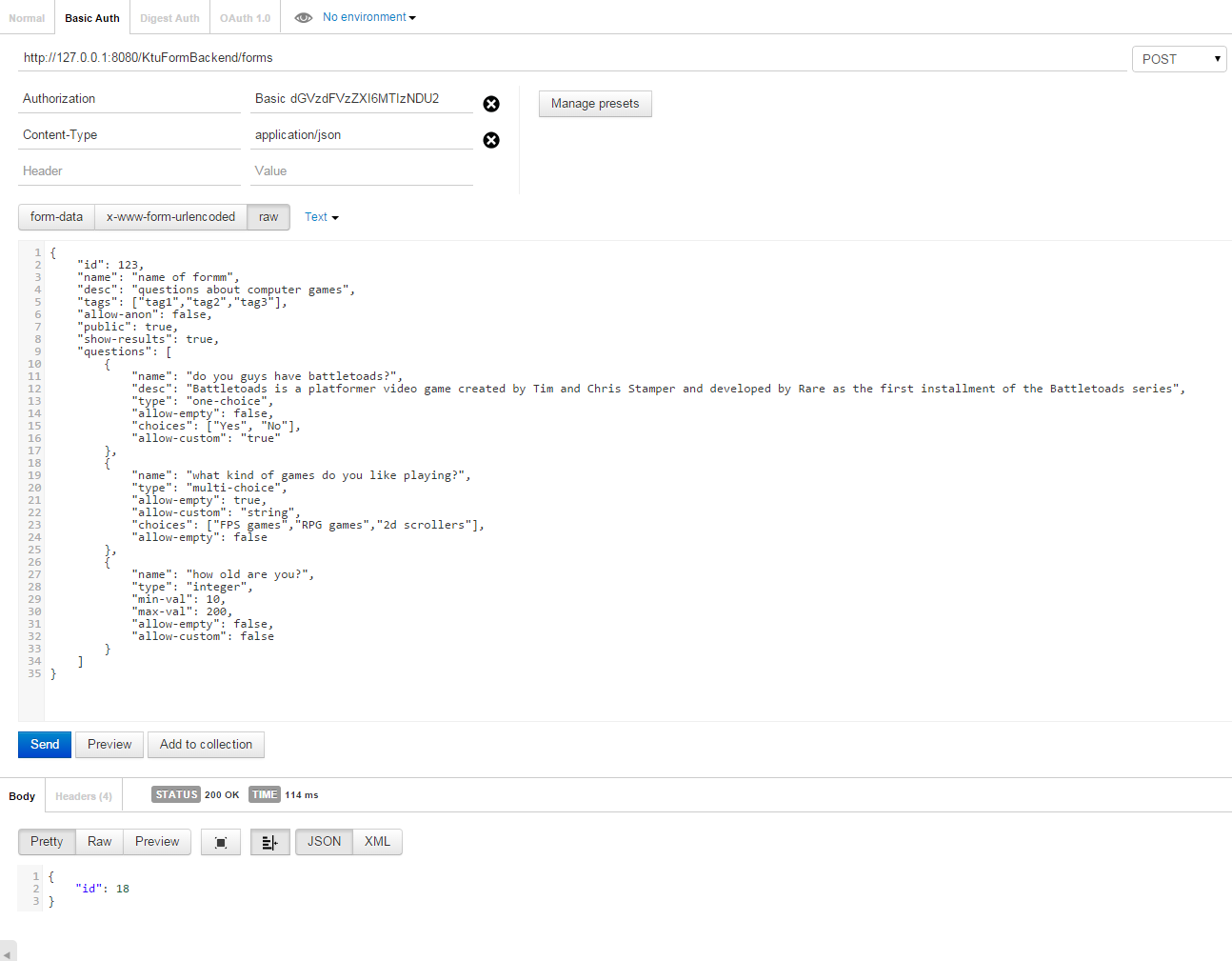
1. Vartotojo peržiūra testavimo sistemoje:



Spaudžiame refresh, ir „Send“



1. Formos sukūrimas:



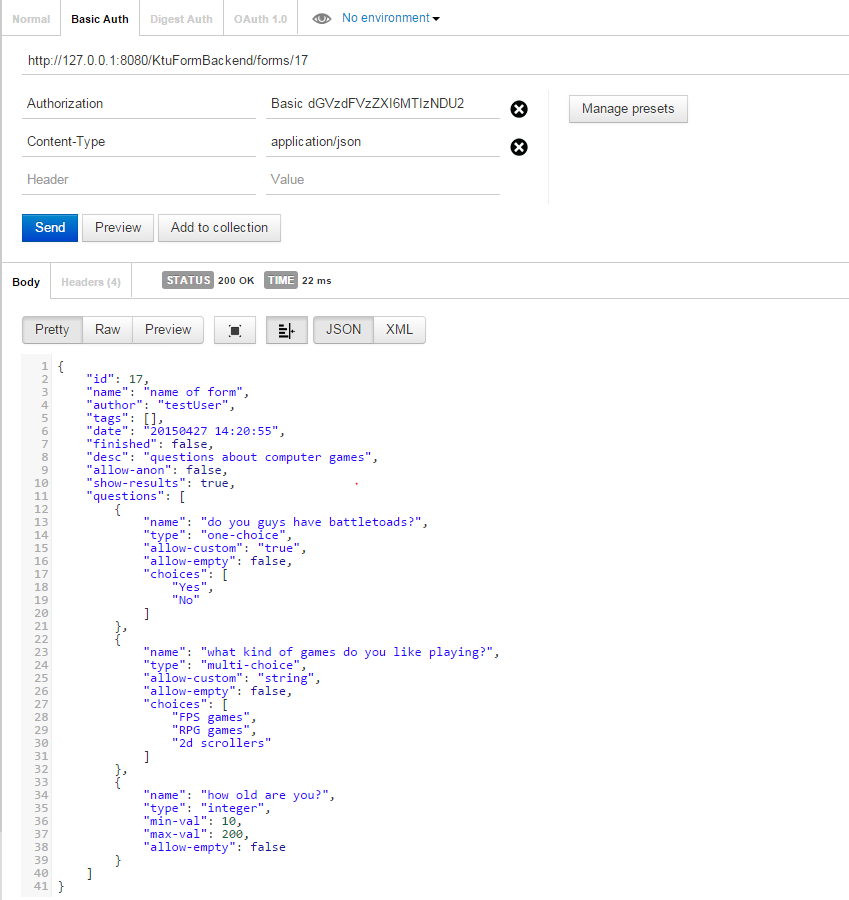
Duomenis serveriui pateikiame JSON formatu:

* {
* "id": 123,
* "name": "name of form",
* "desc": "questions about computer games",
* "tags": ["tag1","tag2","tag3"],
* "allow-anon": false,
* "public": true,
* "show-results": true,
* "questions": [
* {
* "name": "do you guys have battletoads?",
* "desc": "klausimo aprasymas",
* "type": "one-choice",
* "allow-empty": false,
* "choices": ["Yes", "No"],
* "allow-custom": "true"
* },
* {
* "name": "what kind of games do you like playing?",
* "type": "multi-choice",
* "allow-empty": true,
* "allow-custom": "string",
* "choices": ["FPS games","RPG games","2d scrollers"],
* "allow-empty": false
* },
* {
* "name": "how old are you?",
* "type": "integer",
* "min-val": 10,
* "max-val": 200,
* "allow-empty": false,
* "allow-custom": false
* }
* ]

}

Teisingai pateikus duomenis bus grąžinamas formos ID.

1. **Formos peržiūra:**



# Išvados

Darydami šį projektą įgavome patirties dirbti grupėje, susipažinome su JAVA REST karkasu, Tomcat serverių kūrimu, įvairiomis technologijomis, tokiomis kaip JSON, HTTP, taip programų testavimu. Mūsų užduotis buvo sukurti projekto „Backend“ dalį, kuri sudarytų sąlygas kitoms programos gauti, keisti, papildyti duomenis serverio duomenų bazėje. Projektas veikia.

# Priedai

Duomenų bazės struktūra:

