

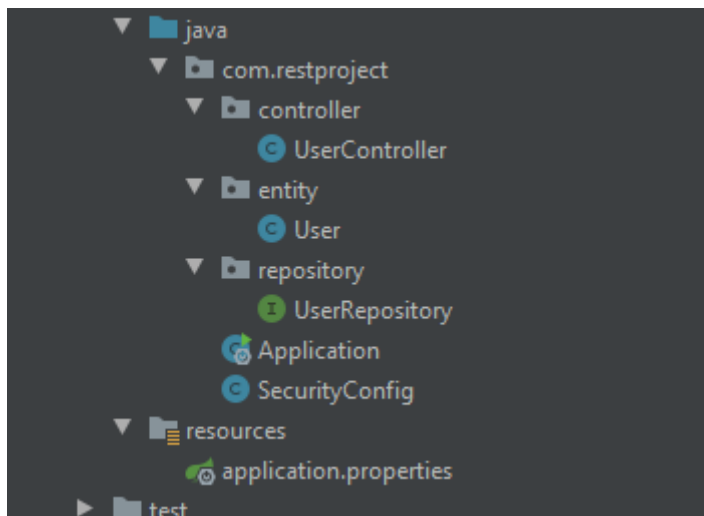
Лабораторная работа №4

Web-службы и REST

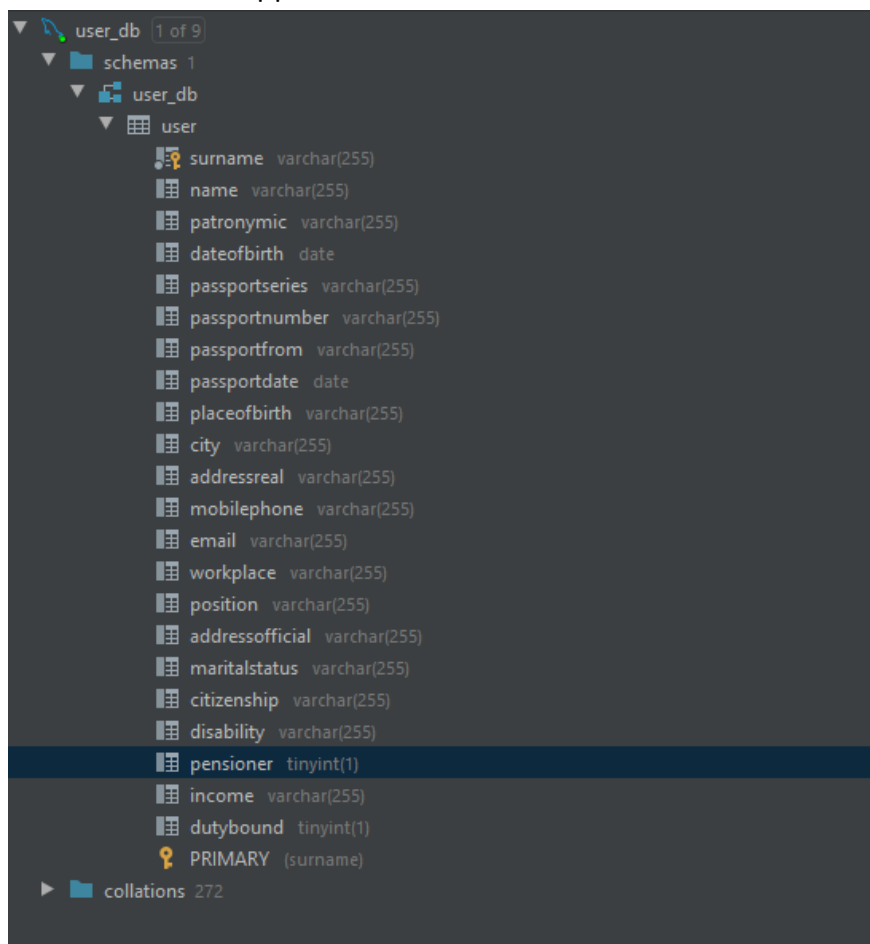
Сидоров Владислав 773901 вариант 23

Работы выполнялась на Spring-проекте с использованием базы данных MySQL и Postman для просмотра CRUD запросов (в методичке нет контрольных вопросов поэтому файл один)

Структура проекта:



Колонки базы данных:



Заполненная база данных:

	👤 surname	👤 name	👤 patronymic	📅 dateofbirth	📄 passportseries	📄 passportnumber	📄 passportfrom	📅 passportdate	📍 placeofbirth	📍 city	📍 addressreal	📞 mobilephone	✉ email
1	Surname1	Name1	Patronymic1	2000-06-06	MP	1234567	Minsk RVD	2016-03-18	Minsk1	Minsk	Address1	111-22-33	mail@gmail.co
2	Surname2	Name2	Patronymic2	1999-05-12	MP	7654321	Minsk RVD	2018-03-12	Minsk2	Minsk	Address2	567-32-11	test@gmail.co
3	Surname3	Name3	Patronymic3	2000-10-10	MP	2765806	Minsk RVD	2019-06-11	Minsk3	Minsk	Address3	653-67-87	querty@gmail.
4	Surname4	Name4	Patronymic4	1995-01-01	MD	1267853	Minsk RVD	2015-09-30	Minsk4	Gomel	Address4	123-45-67	asdf@gmail.co

	📍 city	📍 addressreal	📞 mobilephone	✉ email	🏢 workplace	👤 position	📍 addressofficial	👤 maritalstatus	📄 citizenship	📄 disability	👤 pensioner	💰 income	📄 dutybound
1	Minsk	Address1	111-22-33	mail@gmail.com	Place1	Director	Address1	Not married	Belarus	0	0	\$500	0
2	Minsk	Address2	567-32-11	test@gmail.com	Place2	Cleaner	Address2	Not married	Belarus	0	1	2000 b.r.	1
3	Minsk	Address3	653-67-87	querty@gmail.com	Place3	Manager	Address3	Married	Belarus	0	0	\$1000	0
4	Gomel	Address4	123-45-67	asdf@gmail.com	Place4	Accountant	Address4	In civil marriage	Belarus	Invalid 1st stage	0	\$700	1

Настройки application.properties для Spring:

```
# Automatically update the database
spring.jpa.hibernate.ddl-auto=update

# The database connection URL
spring.datasource.url=jdbc:mysql://localhost:3306/user_db?useSSL=false

# Username
spring.datasource.username=root

# Password
spring.datasource.password=root

# Define the database platform
spring.jpa.database-platform=org.hibernate.dialect.MySQL5InnoDBDialect

# Define the naming strategy
spring.jpa.hibernate.naming-strategy = org.hibernate.cfg.ImprovedNamingStrategy

# Define the default schema
spring.jpa.properties.hibernate.default_schema=schema
```

Pom.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.javahelps</groupId>
    <artifactId>mysql-rest-service</artifactId>
    <version>0.0.1-SNAPSHOT</version>

    <!-- Define the parent pom -->
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>1.5.8.RELEASE</version>
    </parent>

    <!-- Set the Java version -->
    <properties>
        <java.version>1.8</java.version>
    </properties>

    <dependencies>

        <!-- Spring Boot -->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.data</groupId>
            <artifactId>spring-data-rest-core</artifactId>
        </dependency>

        <!-- JPA Data -->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>

        <!-- Data REST -->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-rest</artifactId>
        </dependency>

        <!-- MySQL Connector-J -->
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
        </dependency>

        <!-- Spring Security -->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-security</artifactId>
        </dependency>

    </dependencies>
</project>
```

Класс Application для запуска проекта:

```
package com.restproject;

import com.restproject.entity.User;
import com.restproject.repository.UserRepository;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;

@SpringBootApplication
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
}
```

Класс UserController с описанием CRUD методов:

```
import com.restproject.repository.UserRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import javassist.tools.web.BadHttpRequest;

@RestController
@RequestMapping(path = "/users")
public class UserController {

    @Autowired
    private UserRepository repository;

    @GetMapping
    public Iterable<User> findAll() { return repository.findAll(); }

    @GetMapping(path =("/{surname}")
    public User find(@PathVariable("surname") String surname) { return repository.findOne(surname); }

    @PostMapping(consumes = "application/json")
    public User create(@RequestBody User user) { return repository.save(user); }

    @DeleteMapping(path =("/{surname}")
    public void delete(@PathVariable("surname") String surname) { repository.delete(surname); }

    @PutMapping(path =("/{surname}")
    public User update(@PathVariable("surname") String surname, @RequestBody User user) throws BadHttpRequest {
        if (repository.exists(surname)) {
            user.setSurname(surname);
            return repository.save(user);
        } else {
```

Интерфейс:

```
package com.restproject.repository;

import com.restproject.entity.User;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.rest.core.annotation.RestResource;

@RestResource(exported = false)
public interface UserRepository extends JpaRepository<User, String> {

}
```

Класс SecurityConfig для возможности логиниться:

```
import com.restproject.repository.UserRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
import org.springframework.security.core.authority.AuthorityUtils;
import org.springframework.security.core.userdetails.UsernameNotFoundException;

@Configuration
@EnableWebSecurity
public class SecurityConfig extends WebSecurityConfigurerAdapter {

    @Autowired
    private UserRepository userRepository;

    @Autowired
    protected void configureGlobal(AuthenticationManagerBuilder auth) throws Exception {
        auth.userDetailsService(surname -> {
            User user = userRepository.findOne(surname);
            if (user != null) {
                return new org.springframework.security.core.userdetails.User(user.getSurname(), user.getName(),
                    enabled: true, accountNonExpired: true, credentialsNonExpired: true, accountNonLocked: true, AuthorityUtils.createAuthorityList("ROLE_USER"));
            } else {
                throw new UsernameNotFoundException("Could not find the user '" + surname + "'");
            }
        });
    }

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.authorizeRequests().anyRequest().fullyAuthenticated().and().httpBasic().and().csrf().disable();
    }
}
```

Класс User со всеми переменными (в скриншот не влезят
автосгенерированные геттеры и сеттеры для каждой переменной, но они
есть :)

```
import javax.persistence.Entity;
import javax.persistence.Id;

@Entity
public class User {

    @Id
    private String surname;
    private String name;
    private String patronymic;
    private String dateofbirth;
    private String passportseries;
    private String passportnumber;
    private String passportfrom;
    private String passportdate;
    private String placeofbirth;
    private String city;
    private String addressreal;
    private String mobilephone;
    private String email;
    private String workplace;
    private String position;
    private String addressofficial;
    private String maritalstatus;
    private String citizenship;
    private String disability;
    private String pensioner;
    private String income;
    private String dutybound;

    @Override
    public String toString() {
        return "User{" +
            "surname='" + surname + '\'' +
            ", name='" + name + '\'' +
            ", patronymic='" + patronymic + '\'' +
            ", dateofbirth='" + dateofbirth + '\'' +
            ", passportseries='" + passportseries + '\'' +
            ", passportnumber='" + passportnumber + '\'' +
            ", passportfrom='" + passportfrom + '\'' +
            ", passportdate='" + passportdate + '\'' +
            ", placeofbirth='" + placeofbirth + '\'' +
            ", city='" + city + '\'' +
            ", addressreal='" + addressreal + '\'' +
            ", mobilephone='" + mobilephone + '\'' +
            ", email='" + email + '\'' +
            ", workplace='" + workplace + '\'' +
            ", position='" + position + '\'' +
            ", addressofficial='" + addressofficial + '\'' +
            ", maritalstatus='" + maritalstatus + '\'' +
            ", citizenship='" + citizenship + '\'' +
            ", disability='" + disability + '\'' +
            ", pensioner='" + pensioner + '\'' +
            ", income='" + income + '\'' +
            ", dutybound='" + dutybound + '\'' +
            '}';
    }
}
```

Работоспособность Spring проекта:

```
2020-03-27 20:52:23.834 INFO 12028 --- [main] o.s.d.r.w.RepositoryRestHandlerMapping : Mapped "{[/repository/search],methods=[GET],produces=[application/hal+json || application/json]}" onto public org.springframework.data.rest.webmvc.RepositoryRestController
2020-03-27 20:52:23.834 INFO 12028 --- [main] o.s.d.r.w.RepositoryRestHandlerMapping : Mapped "{[/repository/search/{search}],methods=[GET],produces=[application/hal+json || application/json]}" onto public org.springframework.data.rest.webmvc.RepositoryRestController
2020-03-27 20:52:23.834 INFO 12028 --- [main] o.s.d.r.w.RepositoryRestHandlerMapping : Mapped "{[/repository/search/{search}],methods=[OPTIONS],produces=[application/hal+json || application/json]}" onto public org.springframework.data.rest.webmvc.RepositoryRestController
2020-03-27 20:52:23.834 INFO 12028 --- [main] o.s.d.r.w.RepositoryRestHandlerMapping : Mapped "{[/repository/search/{search}],methods=[HEAD],produces=[application/hal+json || application/json]}" onto public org.springframework.data.rest.webmvc.RepositoryRestController
2020-03-27 20:52:23.835 INFO 12028 --- [main] o.s.d.r.w.RepositoryRestHandlerMapping : Mapped "{[/repository/search/{search}],methods=[GET],produces=[application/hal+json || application/json]}" onto public org.springframework.data.rest.webmvc.RepositoryRestController
2020-03-27 20:52:23.835 INFO 12028 --- [main] o.s.d.r.w.RepositoryRestHandlerMapping : Mapped "{[/ ],methods=[GET],produces=[application/hal+json || application/json]}" onto public org.springframework.data.rest.webmvc.RepositoryRestController
2020-03-27 20:52:23.836 INFO 12028 --- [main] o.s.d.r.w.RepositoryRestHandlerMapping : Mapped "{[/ ],methods=[OPTIONS],produces=[application/hal+json || application/json]}" onto public org.springframework.data.rest.webmvc.RepositoryRestController
2020-03-27 20:52:23.836 INFO 12028 --- [main] o.s.d.r.w.RepositoryRestHandlerMapping : Mapped "{[/ ],methods=[HEAD],produces=[application/hal+json || application/json]}" onto public org.springframework.data.rest.webmvc.RepositoryRestController
2020-03-27 20:52:23.838 INFO 12028 --- [main] o.s.d.r.w.BasePathAwareHandlerMapping : Mapped "{[/profile/{repository}],methods=[GET],produces=[application/schema+json]}" onto public org.springframework.data.rest.webmvc.BasePathAwareHandler
2020-03-27 20:52:23.838 INFO 12028 --- [main] o.s.d.r.w.BasePathAwareHandlerMapping : Mapped "{[/profile/{repository}],methods=[OPTIONS],produces=[application/alps+json]}" onto public org.springframework.data.rest.webmvc.BasePathAwareHandler
2020-03-27 20:52:23.839 INFO 12028 --- [main] o.s.d.r.w.BasePathAwareHandlerMapping : Mapped "{[/profile/{repository}],methods=[GET],produces=[application/alps+json || */]}" onto public org.springframework.data.rest.webmvc.BasePathAwareHandler
2020-03-27 20:52:23.839 INFO 12028 --- [main] o.s.d.r.w.BasePathAwareHandlerMapping : Mapped "{[/profile],methods=[OPTIONS]}" onto public org.springframework.data.rest.webmvc.BasePathAwareHandler
2020-03-27 20:52:23.839 INFO 12028 --- [main] o.s.d.r.w.BasePathAwareHandlerMapping : Mapped "{[/profile],methods=[GET]}" onto org.springframework.data.rest.webmvc.BasePathAwareHandler
2020-03-27 20:52:23.960 INFO 12028 --- [main] o.s.j.e.a.AnnotationBeanExporter : Registering beans for JMX exposure on startup
2020-03-27 20:52:23.995 INFO 12028 --- [main] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat started on port(s): 8080 (http)
2020-03-27 20:52:23.999 INFO 12028 --- [main] com.restproject.Application : Started Application in 3.912 seconds (JVM running for 4.341)
```

Далее идет отчет о работе программы:

Поскольку есть функция логина, без правильного никнейма/пароля в роле которых выступают фамилия/имя будет возвращен код 401

The screenshot shows a REST client interface with a GET request to `http://localhost:8080/users`. The response is a 401 Unauthorized status with a JSON body containing error details.

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body: Cookies (1) Headers (10) Test Results Status: 401 Unauthorized Time: 172ms Size: 468 B Save Response

```
1 {
2   "timestamp": 1585331593038,
3   "status": 401,
4   "error": "Unauthorized",
5   "message": "Bad credentials",
6   "path": "/users"
7 }
```

Работа GET:

GET

http://localhost:8080/users

Send

Save

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

Cookies

Code

TYPE

Basic Auth

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Username

Surname2

Password

Name2

☒ Show Password

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. [Learn more about variables](#)

Body

Cookies (1)

Headers (10)

Test Results

Status: 200 OK

Time: 136ms

Size: 2.37 KB

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

{

"surname": "Surname",

"name": "Name1",

"patronymic": "Patronymic1",

"dateofbirth": "2000-06-06",

"passportseries": "MP",

"passportnumber": "1234567",

"passportfrom": "Minsk RYVD",

"passportdate": "2016-03-18",

"placeofbirth": "Minsk1",

"city": "Minsk",

"addressreal": "Address1",

"mobilephone": "111-22-33",

"email": "mail@gmail.com",

"workplace": "Place1",

"position": "Director",

"addressofficial": "Address1",

"maritalstatus": "Not married",

"citizenship": "Belarus",

"disability": "0",

"pensioner": "0",

"income": "\$500",

"dutybound": "0"

},

{

"surname": "Surname2".

Работа POST:

POST ▼ http://localhost:8080/users Send Save ▼

Params Authorization ● Headers (10) ● **Body** ● Pre-request Script Tests Settings Cookies Code

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL ● JSON ▼ Beautify

```
1 {
2   "surname": "POSTTEST",
3   "name": "POSTTEST",
4   "patronymic": "POSTTEST",
5   "dateofbirth": "2000-06-05",
6   "passportseries": "MP",
7   "passportnumber": "1234567",
8   "passportfrom": "Minsk RYVD",
9   "passportdate": "2016-03-18",
10  "placeofbirth": "Minsk1",
11  "city": "Minsk",
12  "addressreal": "Address1",
13  "mobilephone": "111-22-33",
14  "email": "mail@gmail.com",
15  "workplace": "Place1",
16  "position": "Director",
17  "addressofficial": "Address1",
18  "maritalstatus": "Not married",
19  "citizenship": "Belarus",
20  "disability": "0",
21  "pensioner": "0",
22  "idcard": "05000"
```

Body Cookies (1) Headers (9) Test Results Status: 200 OK Time: 78ms Size: 814 B Save Response ▼

Pretty Raw Preview Visualize JSON ≡ 🔍

```
1 {
2   "surname": "POSTTEST",
3   "name": "POSTTEST",
4   "patronymic": "POSTTEST",
5   "dateofbirth": "2000-06-05",
6   "passportseries": "MP",
7   "passportnumber": "1234567",
8   "passportfrom": "Minsk RYVD",
9   "passportdate": "2016-03-18",
10  "placeofbirth": "Minsk1",
11  "city": "Minsk",
12  "addressreal": "Address1",
13  "mobilephone": "111-22-33",
14  "email": "mail@gmail.com",
15  "workplace": "Place1",
16  "position": "Director",
17  "addressofficial": "Address1",
18  "maritalstatus": "Not married",
19  "citizenship": "Belarus",
20  "disability": "0",
21  "pensioner": "0",
22  "idcard": "05000"
```

Работа PUT:

PUT

http://localhost:8080/users/POSTTEST

Send

Save

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

Cookies

Code

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1 {
2   "surname": "POSTTEST",
3   "name": "POSTTEST---NEW",
4   "patronymic": "POSTTEST",
5   "dateofbirth": "2000-06-05",
6   "passportseries": "MP",
7   "passportnumber": "1234567",
8   "passportfrom": "Minsk RYVD",
9   "passportdate": "2016-03-18",
10  "placeofbirth": "Minsk1",
11  "city": "Minsk",
12  "addressreal": "Address1",
13  "mobilephone": "111-22-33",
14  "email": "mail@gmail.com",
15  "workplace": "Place1",
16  "position": "Director",
17  "addressofficial": "Address1",
18  "maritalstatus": "Not married",
19  "citizenship": "Belarus",
20  "disability": "0",
21  "pensioner": "0",
22  "passport": "65000"
23 }
```

Body

Cookies (1)

Headers (9)

Test Results

Status: 200 OK

Time: 44ms

Size: 821 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "surname": "POSTTEST",
3   "name": "POSTTEST---NEW",
4   "patronymic": "POSTTEST",
5   "dateofbirth": "2000-06-05",
6   "passportseries": "MP",
7   "passportnumber": "1234567",
8   "passportfrom": "Minsk RYVD",
9   "passportdate": "2016-03-18",
10  "placeofbirth": "Minsk1",
11  "city": "Minsk",
12  "addressreal": "Address1",
13  "mobilephone": "111-22-33",
14  "email": "mail@gmail.com",
15  "workplace": "Place1",
16  "position": "Director",
17  "addressofficial": "Address1",
18  "maritalstatus": "Not married",
19  "citizenship": "Belarus",
20  "disability": "0",
21  "pensioner": "0",
22  "passport": "65000"
23 }
```

Работа DELETE (на втором скрине больше нет пользователя POSTTEST):

DELETE

http://localhost:8080/users/POSTTEST

Send

Save

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON

Beautify

```
1 {
2   "surname": "POSTTEST",
3   "name": "POSTTEST---NEW",
4   "patronymic": "POSTTEST",
5   "dateofbirth": "2000-06-05",
6   "passportseries": "MP",
7   "passportnumber": "1234567",
8   "passportfrom": "Minsk RYVD",
9   "passportdate": "2016-03-18",
10  "placeofbirth": "Minsk1",
11  "city": "Minsk",
12  "addressreal": "Address1",
13  "mobilephone": "111-22-33",
14  "email": "mail@gmail.com",
15  "workplace": "Place1",
16  "position": "Director",
17  "addressofficial": "Address1",
18  "maritalstatus": "Not married",
19  "citizenship": "Belarus",
20  "disability": "0",
21  "pensioner": "0",
22  "passport": "df008"
```

GET

http://localhost:8080/users

Send

Save

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL JSON

Beautify

```
1 {
2   "surname": "POSTTEST",
3   "name": "POSTTEST---NEW",
4   "patronymic": "POSTTEST",
5   "dateofbirth": "2000-06-05",
6   "passportseries": "MP",
7   "passportnumber": "1234567",
8   "passportfrom": "Minsk RYVD",
9   "passportdate": "2016-03-18",
10  "placeofbirth": "Minsk1",
11  "city": "Minsk",
12  "addressreal": "Address1",
13  "mobilephone": "111-22-33",
14  "email": "mail@gmail.com",
15  "workplace": "Place1",
16  "position": "Director",
17  "addressofficial": "Address1",
18  "maritalstatus": "Not married",
19  "citizenship": "Belarus",
20  "disability": "0",
21  "pensioner": "0",
22  "passport": "df008"
```

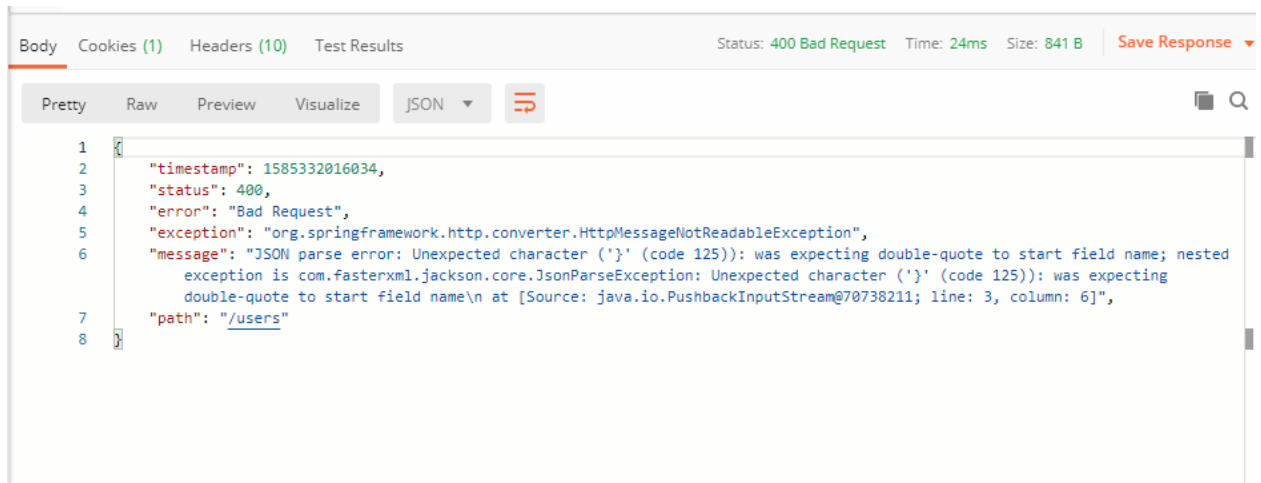
Body Cookies (1) Headers (9) Test Results Status: 200 OK Time: 16ms Size: 2.29 KB Save Response

Pretty Raw Preview Visualize JSON

🔍

```
1 {
2   "surname": "Surname",
3   "name": "Name1",
4   "patronymic": "Patronymic1",
5   "dateofbirth": "2000-06-06",
6   "passportseries": "MP",
7   "passportnumber": "1234567",
8   "passportfrom": "Minsk RYVD",
9   "passportdate": "2016-03-18",
10  "placeofbirth": "Minsk1",
11  "city": "Minsk",
12  "addressreal": "Address1",
13  "mobilephone": "111-22-33",
14  "email": "mail@gmail.com",
15  "workplace": "Place1",
16  "position": "Director",
17  "addressofficial": "Address1",
18  "maritalstatus": "Not married",
```

Обработка кода 400 Bad request:



The screenshot displays the 'Body' tab of a web browser's developer tools. The status bar at the top indicates a '400 Bad Request' error, with a response time of 24ms and a size of 841 B. The response body is shown in JSON format, detailing the error information.

```
1 {
2   "timestamp": 1585332016034,
3   "status": 400,
4   "error": "Bad Request",
5   "exception": "org.springframework.http.converter.HttpMessageNotReadableException",
6   "message": "JSON parse error: Unexpected character ('}') (code 125)): was expecting double-quote to start field name; nested
              exception is com.fasterxml.jackson.core.JsonParseException: Unexpected character ('}') (code 125)): was expecting
              double-quote to start field name\n at [Source: java.io.PushbackInputStream@70738211; line: 3, column: 6]",
7   "path": "/users"
8 }
```