**Stackoverflow**

Sandu Mihai-Alexandru

Group 30433/2

1. Introduction

Stackoverflow is an online platform where users can post questions about certain topics regarding their code and other users of the platform can post their answers and solutions to those problems. The platform serves as an important tool for every programmer regarding their day to day problems, it helps them to find answers to their solutions that other users came across before them.

This project presents a simpler version of the original app. It contains the basic functionality of the platform. The user can ask questions, answer questions, upvote, downvote questions and answers.

1. Technology

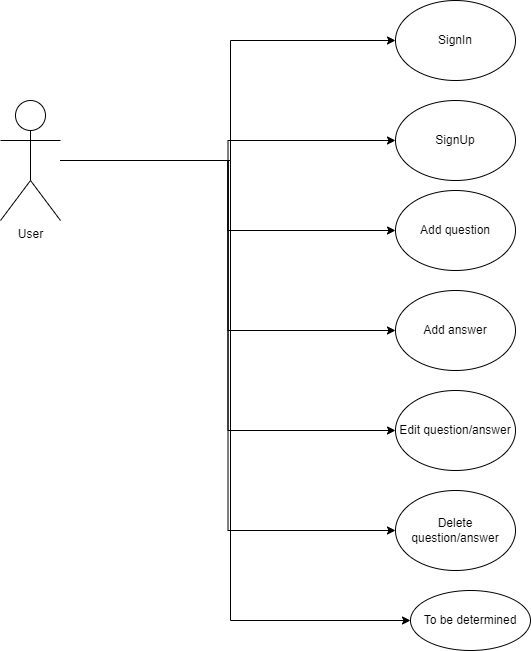
* Spring boot: java framework is used for developing the backend of the application.
* React JS: is a java script framework used for developing the frontend of the application.
* Redux toolkit: is a tool used together with React to manage the states of the application. Together with redux toolkit, redux persist will also be used, its purpose is to persist the data stored using the redux library.
* MySQL: database used in the application.
* IDE’s: IntelliJ Ultimate for the BE, for Java, WebStorm for the JavaScript, for the frontend.

Spring boot:

Java Spring Boot is an open-source framework that provides a fast and easy way to create production-grade Spring-based applications. It aims to simplify the development process by providing a robust set of tools and conventions that make it easy to create and configure Spring-based applications. Spring Boot offers various features such as embedded servers, auto-configuration, and production-ready metrics. It also provides support for various databases, messaging systems, and APIs. With Spring Boot, developers can focus on writing business logic rather than worrying about infrastructure and configuration details, making it a popular choice for building web applications, microservices, and APIs.

React:

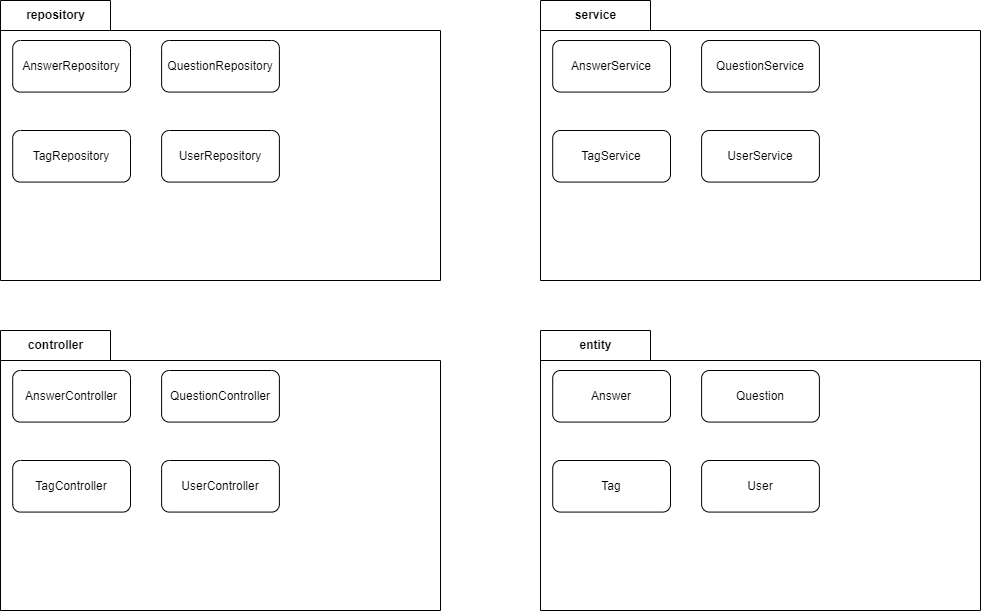
React is an open-source JavaScript library for building user interfaces. It was created by Facebook and is widely used for developing single-page applications and mobile applications. React allows developers to create reusable UI components, which can be composed to build complex and interactive user interfaces. React uses a declarative programming style, which makes it easier to reason about the application's state and behavior. React is often used in combination with other tools and libraries, such as Redux, React Router, and Axios, to build modern web applications. It can also be used to build native mobile applications using tools such as React Native. React has gained popularity due to its flexibility, performance, and large ecosystem of third-party libraries and tools. It is commonly used in modern web development to create dynamic and responsive user interfaces, making it a valuable tool for front-end developers.

1. Use case diagram.
2. Architecture

The layered architecture is a design pattern used in software engineering that organizes an application into separate layers, each with a specific responsibility. The layers typically include presentation, business logic, and data access, with each layer interacting only with the layers directly above and below it. The presentation layer is responsible for handling user input and displaying output to the user. The business logic layer contains the application's logic and rules, and the data access layer is responsible for accessing and manipulating data in the application's storage. The layered architecture promotes separation of concerns and improves maintainability by making it easier to modify or replace individual layers without affecting the others. It also allows for scalability by enabling the addition of new layers or the duplication of existing ones to handle increased load or functionality.

1. Package diagram

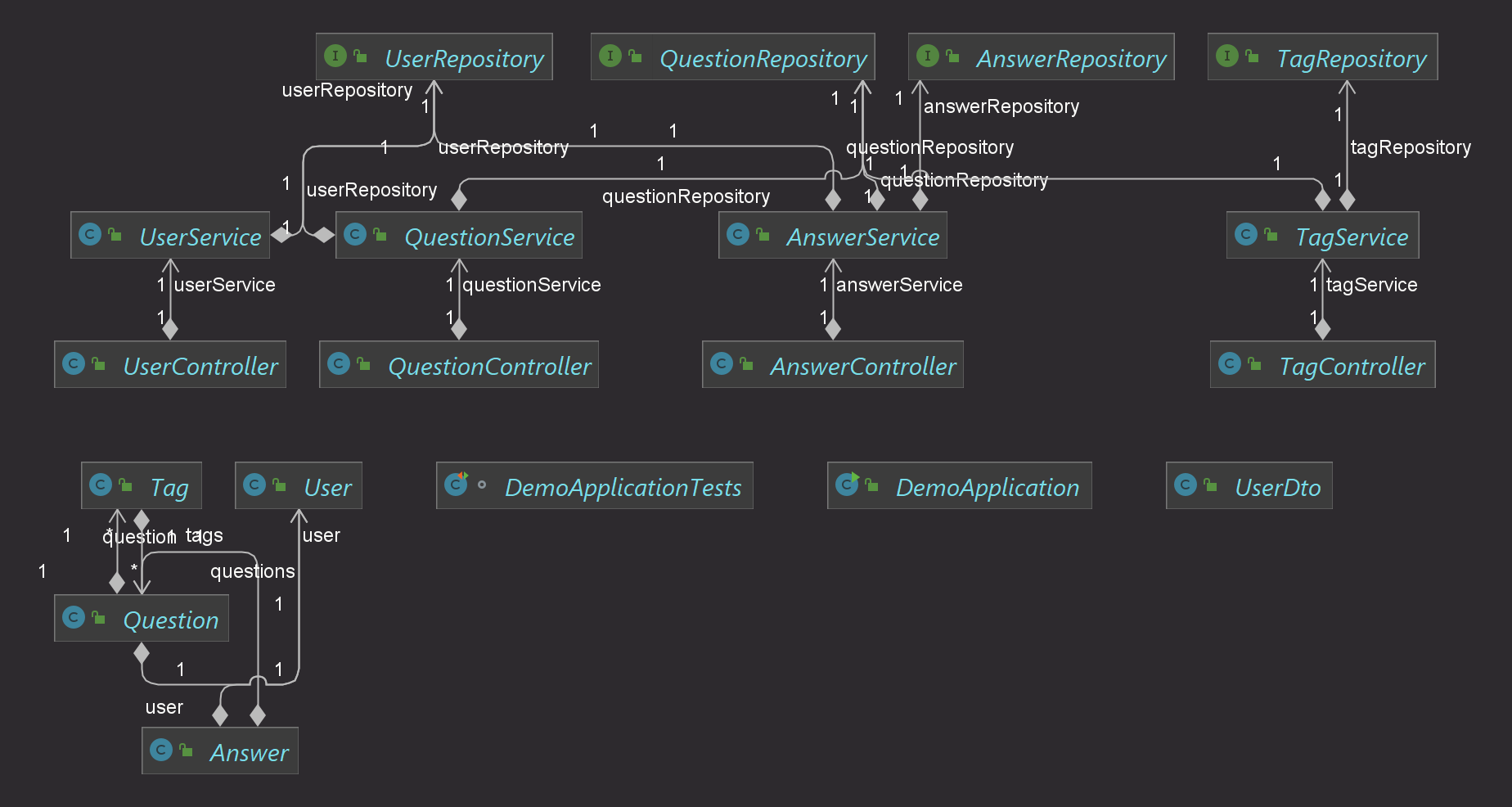
A package diagram is a UML diagram that illustrates the organization of packages and their dependencies in a software system or application. It shows how packages are related to each other and can help developers and architects understand the system's structure and design.



This application is organized into different packages, each serving a specific function. The repository package contains the database operations, while the controller package manages the routes and maps the requests accordingly. The service package contains the business logic and handles the requests. Each controller depends on a service to perform the logic. The entity package represents the tables in our application, including their fields and relationships.

1. Class diagram

A class diagram is a type of UML (Unified Modeling Language) diagram that represents the structure and behavior of objects in a system. It shows the classes in a system, their attributes, methods, and the relationships between them. Class diagrams are useful for modeling the static structure of an application and help developers understand the relationships between different classes and objects in a system. They are commonly used in software engineering to design and document object-oriented systems.



1. Entities

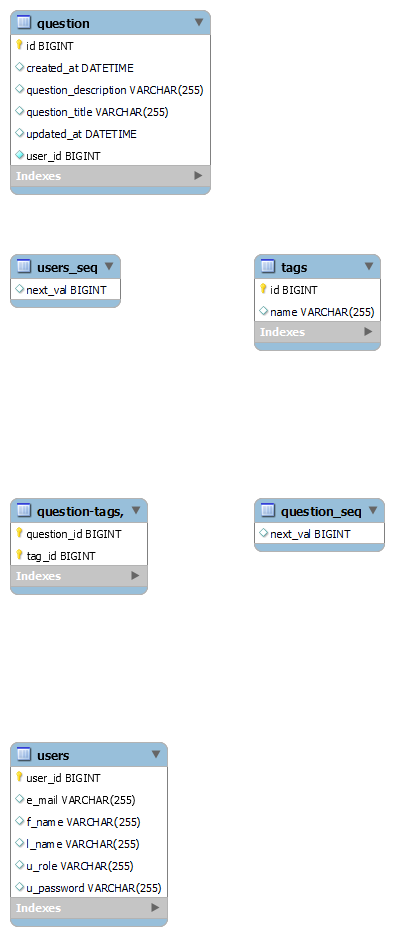
Entities:

* User: fields
  + private Long userId (id of user)
  + private String lastName
  + private String firstName
  + private String email
  + private String password
  + private String role
  + +getters, setters, constructors

Describes the user entity, its representation in the DB (subject to change).

* UserService: fields
  + UserRepository userRepository
  + Public List<User> retrieveUsers(): gets all users and maps them to the DTO
  + Public User getUserByID(Long cnp): gets a specific user by its id, after it maps it
  + Public User findUserByEmail(String email): gets a specific user by its email, after it maps it
  + Public long deleteUserByID(Long id): delete user with that id
  + Public User saveUser(User user): saves an user
  + Public void updateUser(User user): updates an user
* UserControler: fields
  + UserService userService
  + Public List<User> retrieveUsers(): handles mapping and calls the corresponding method from the service
  + Public User getUserById(@PathVariable Long id): handles mapping and calls the corresponding method from the service
  + Public long deleteUserById(@PathVariable Long id): handles mapping and calls the corresponding method from the service
  + Public User addUser(@RequestBody User user): handles mapping and calls the corresponding method from the service
  + Public void updateUser(@RequestBody User user): handles mapping and calls the corresponding method from the service

1. Data base design



The database currently has 4 tables(6 but the \_seq are autogenerated for the ids).

drop schema if exists oop1 ;

create schema oop1;

use oop1;

create table question (id bigint not null, created\_at datetime, question\_description varchar(255), question\_title varchar(255), updated\_at datetime, user\_id bigint not null, primary key (id)) engine=MyISAM;

create table `question-tags,` (question\_id bigint not null, tag\_id bigint not null, primary key (question\_id, tag\_id)) engine=MyISAM;

create table question\_seq (next\_val bigint) engine=MyISAM;

insert into question\_seq values ( 1 );

create table tags (id bigint not null auto\_increment, name varchar(255), primary key (id)) engine=MyISAM;

create table users (user\_id bigint not null, e\_mail varchar(255), f\_name varchar(255), l\_name varchar(255), u\_role varchar(255),u\_password varchar(255), primary key (user\_id)) engine=MyISAM;

create table users\_seq (next\_val bigint) engine=MyISAM;

insert into users\_seq values ( 1 );

alter table question add constraint FK7rnpup7eaonh2ubt922ormoij foreign key (user\_id) references users (user\_id);

alter table `question-tags,` add constraint FKsr008ctva1g0cf69t589ursfq foreign key (tag\_id) references tags (id);

alter table `question-tags,` add constraint FKk286q98bbdpil9lfuk3dbh6nl foreign key (question\_id) references question (id);

1. Endpoints request

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer screen

Description automatically generated with medium confidence

A screenshot of a computer screen

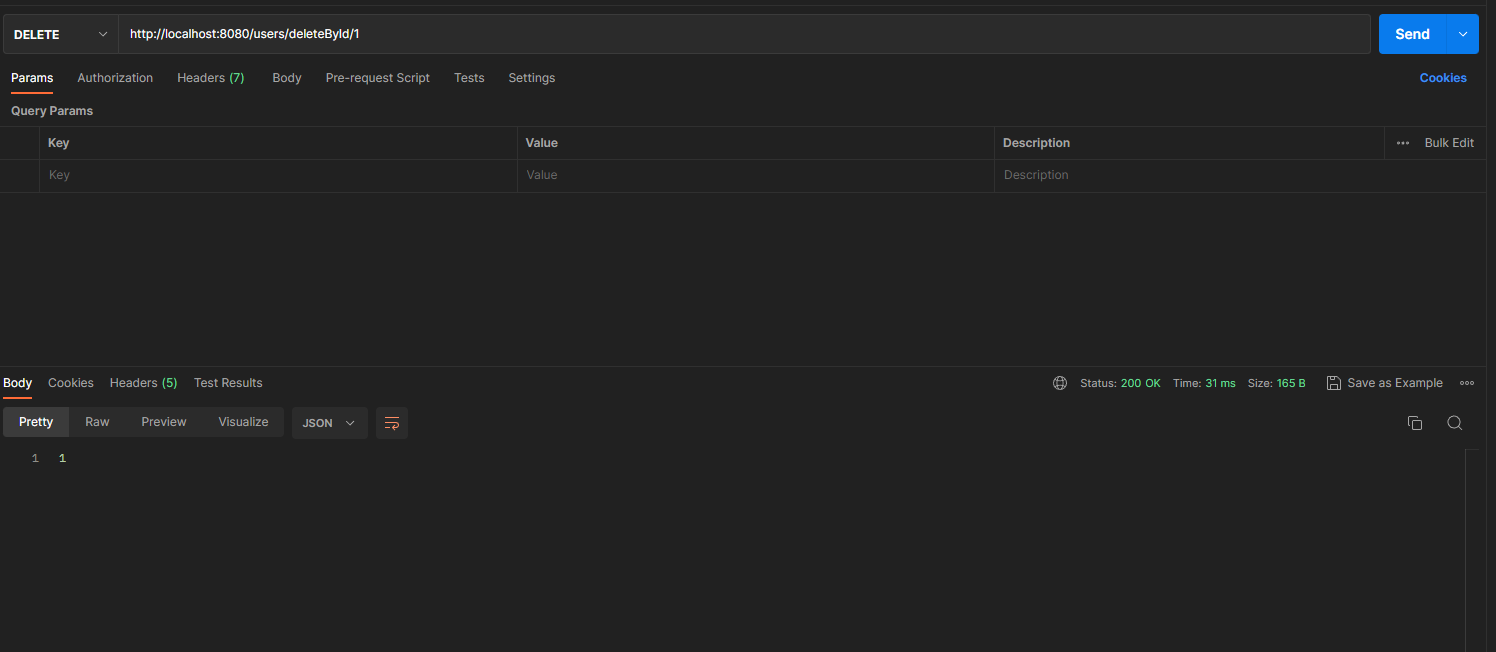
Description automatically generated with medium confidence

A screenshot of a computer screen

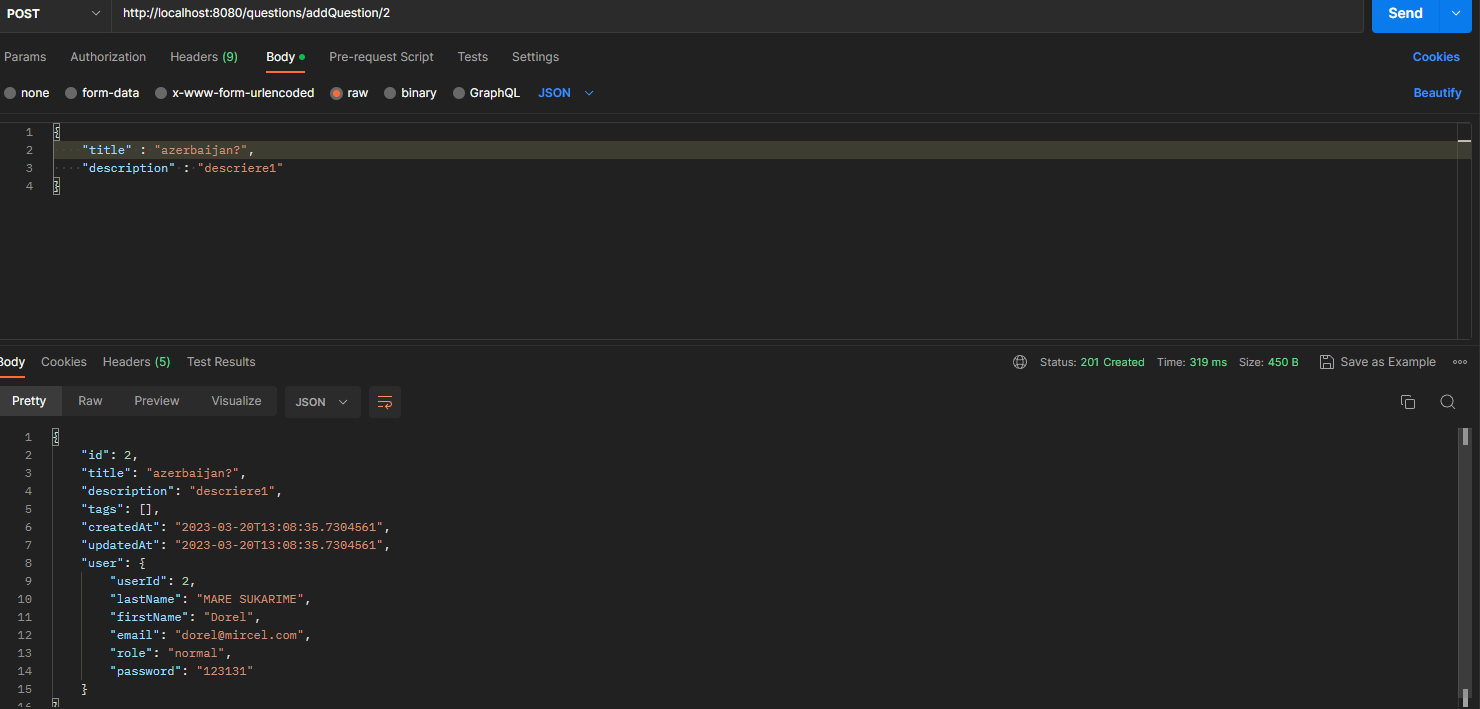
Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence



Question:



A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer screen

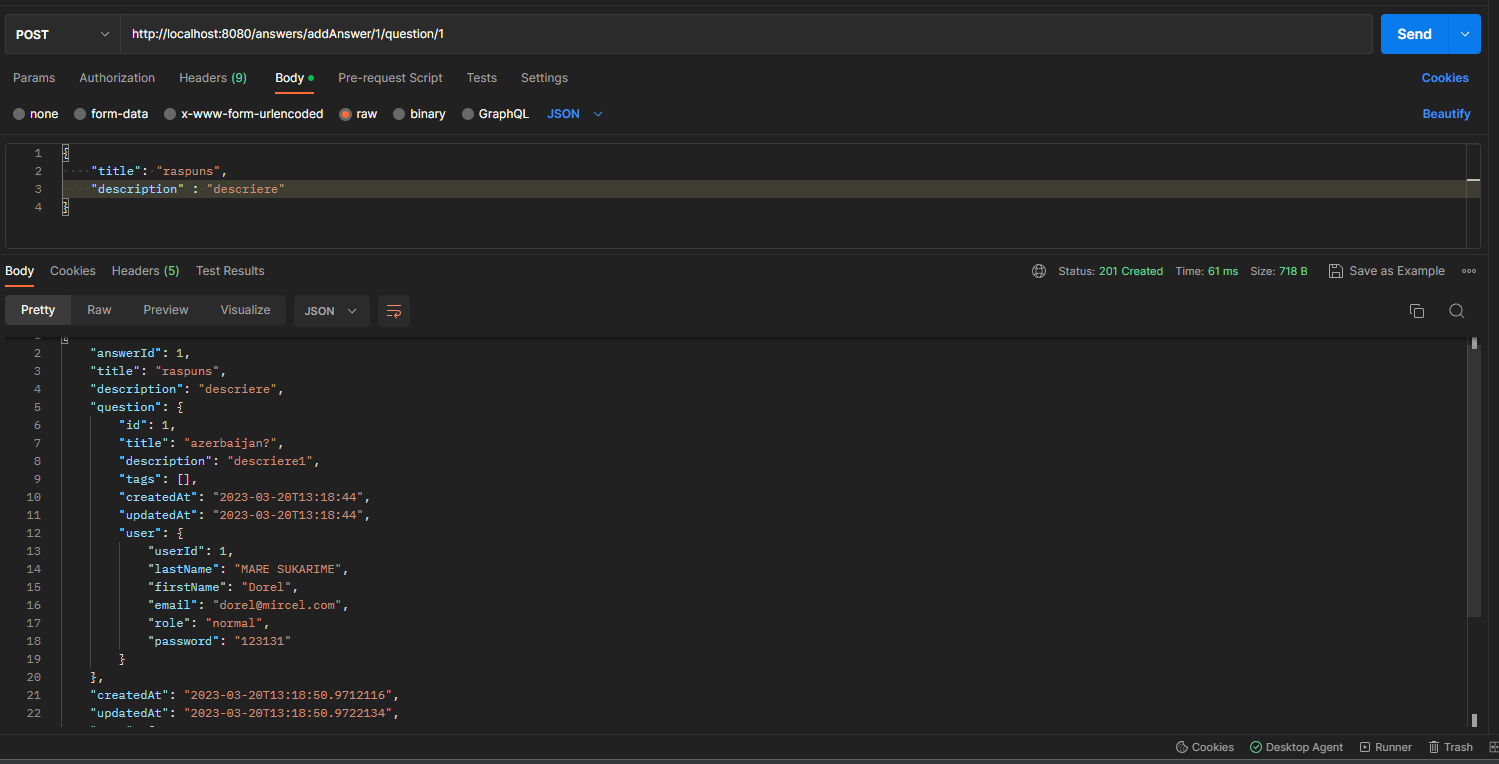
Description automatically generated with medium confidence

A screenshot of a computer screen

Description automatically generated with medium confidence

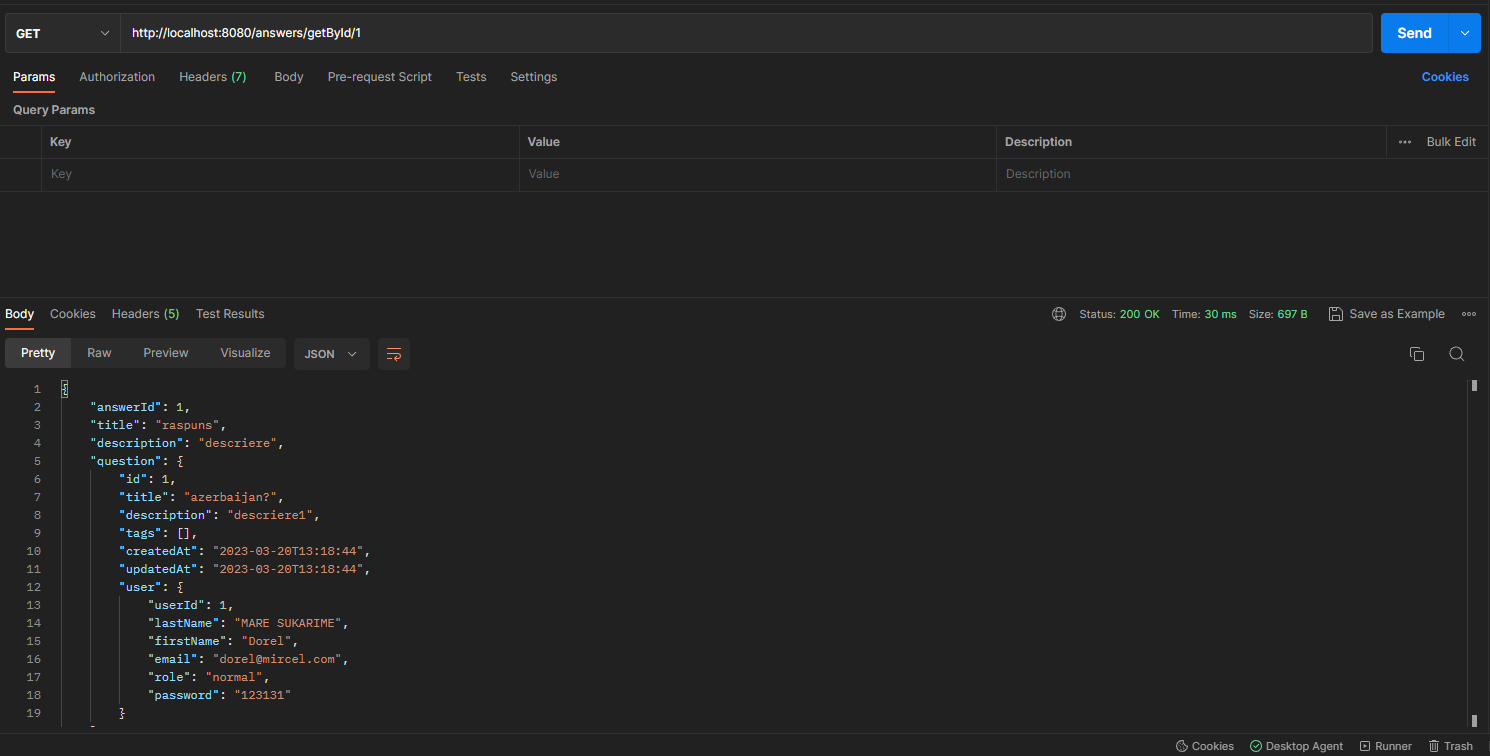
A screenshot of a computer

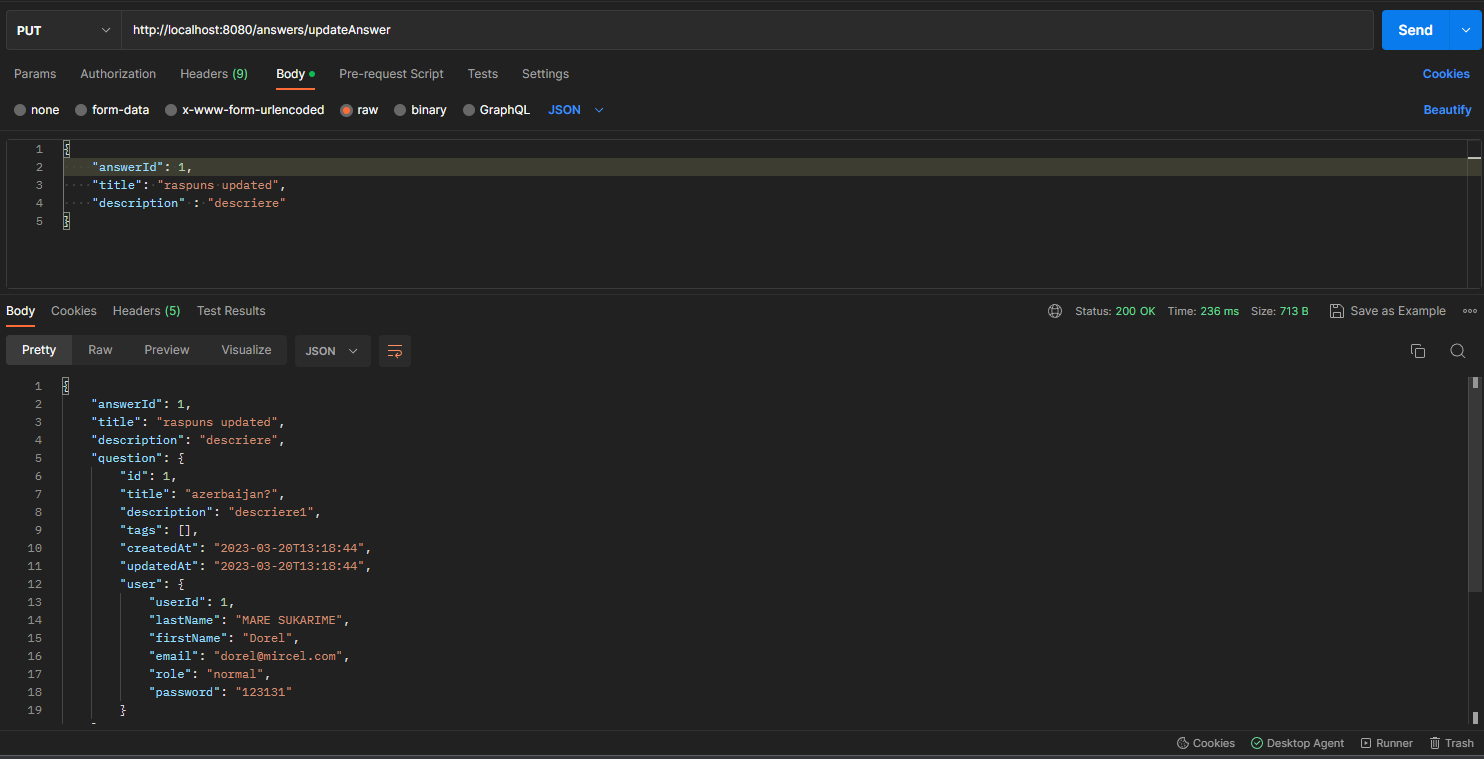
Description automatically generated

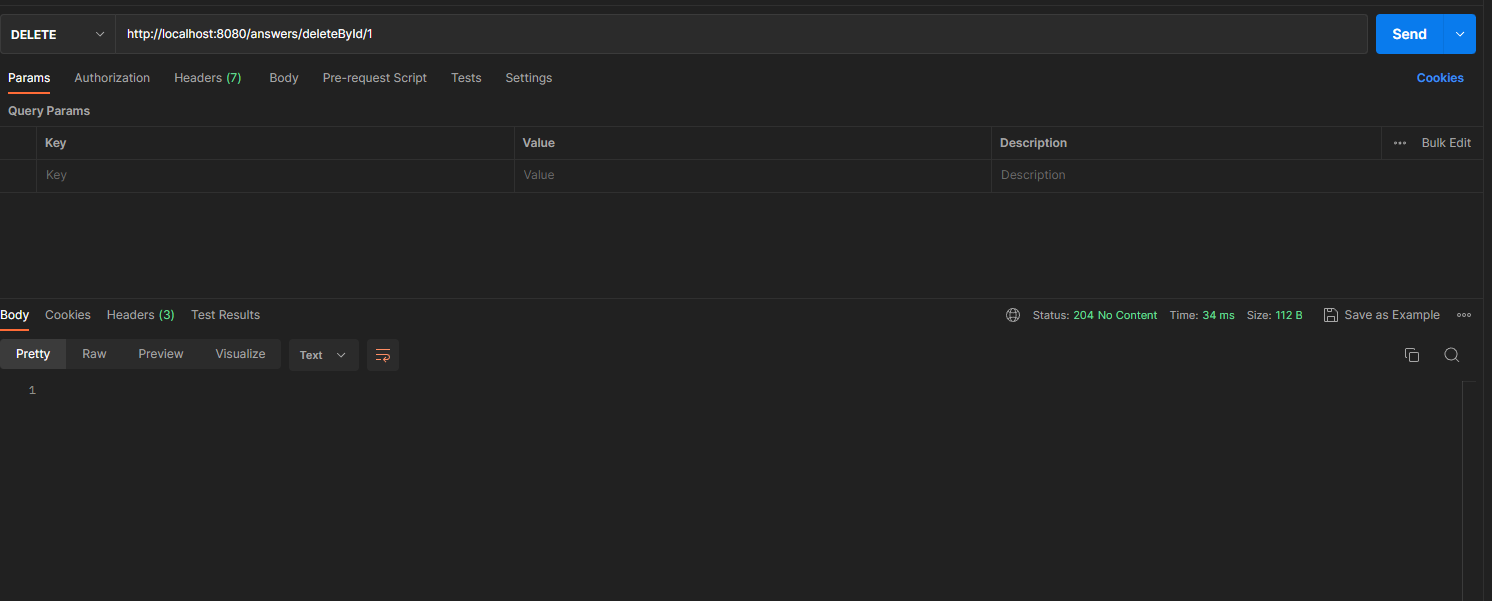


A screenshot of a computer

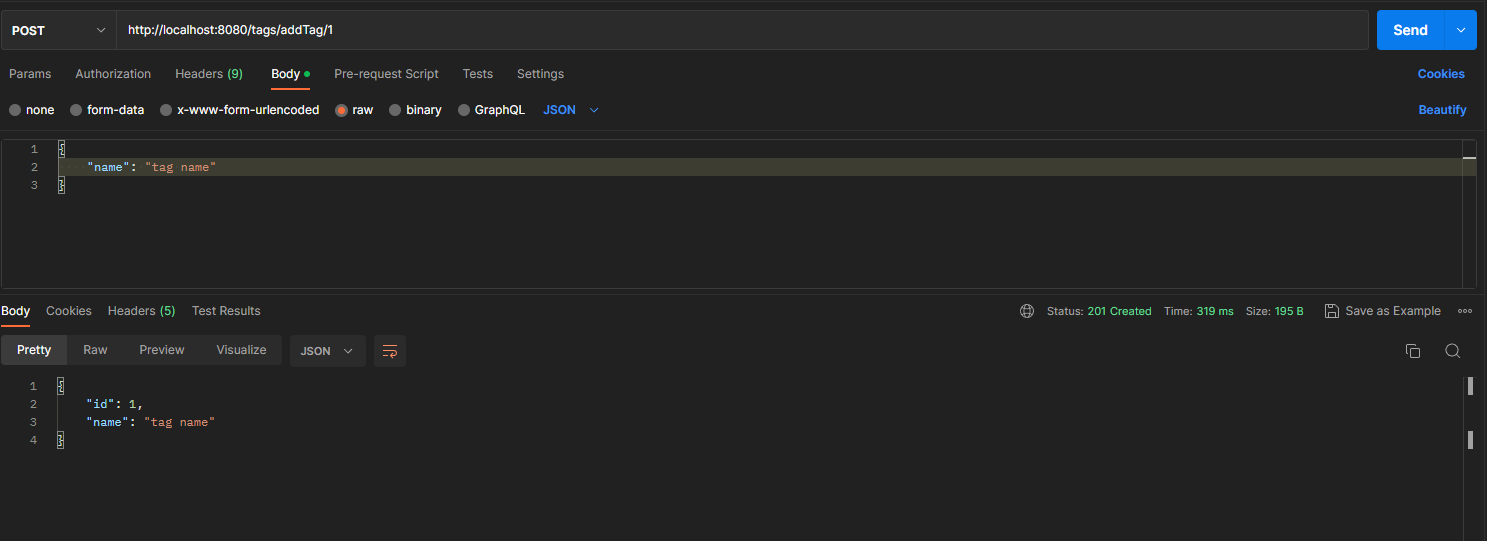
Description automatically generated with medium confidence

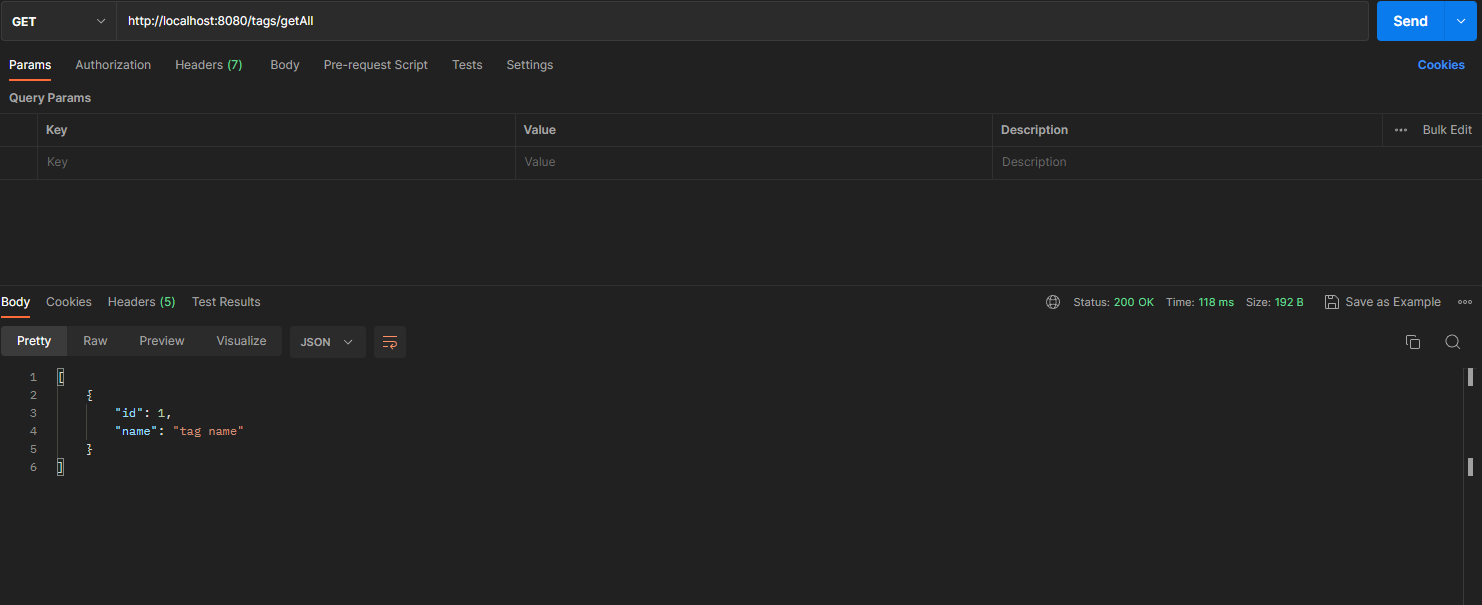






Tag:





A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

