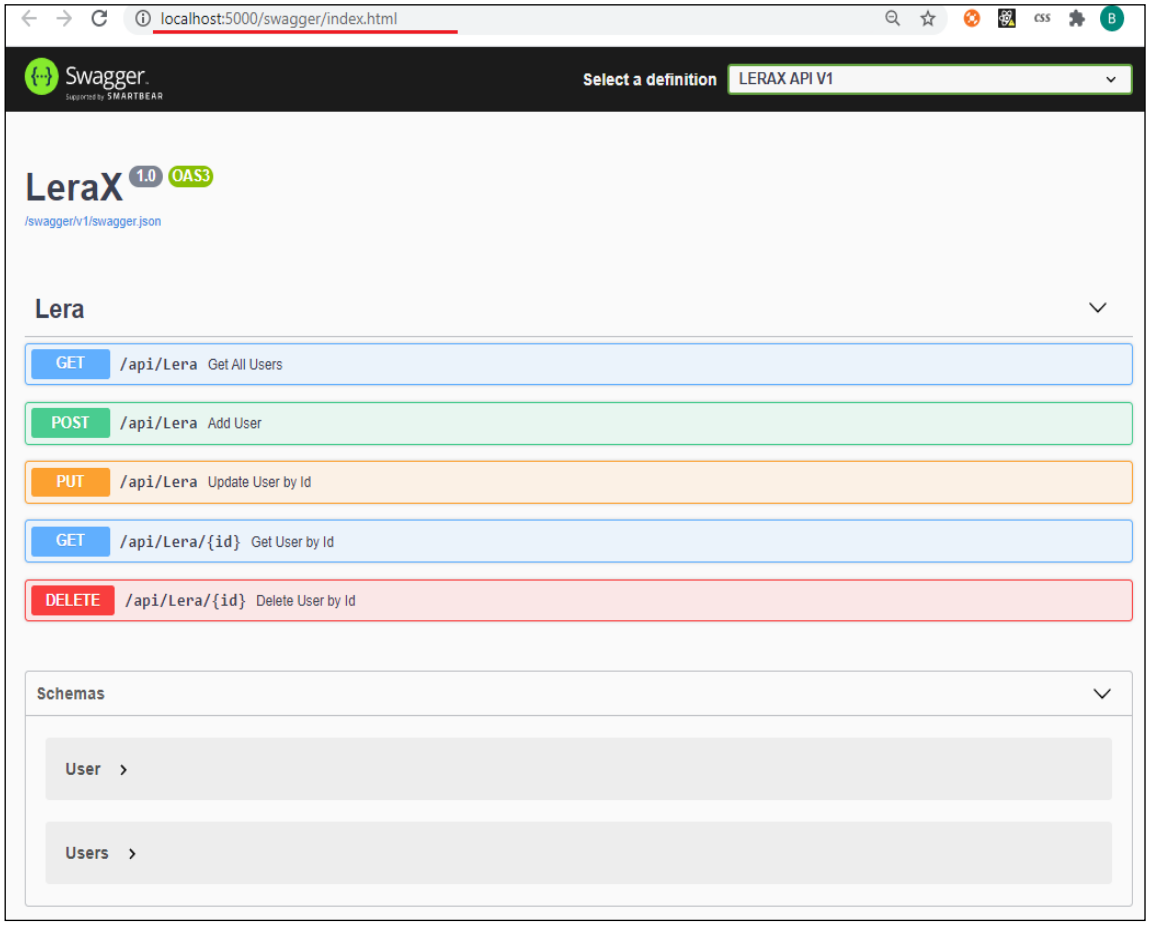
ASP.NET CORE/REST/SWAGGER

Задание 1

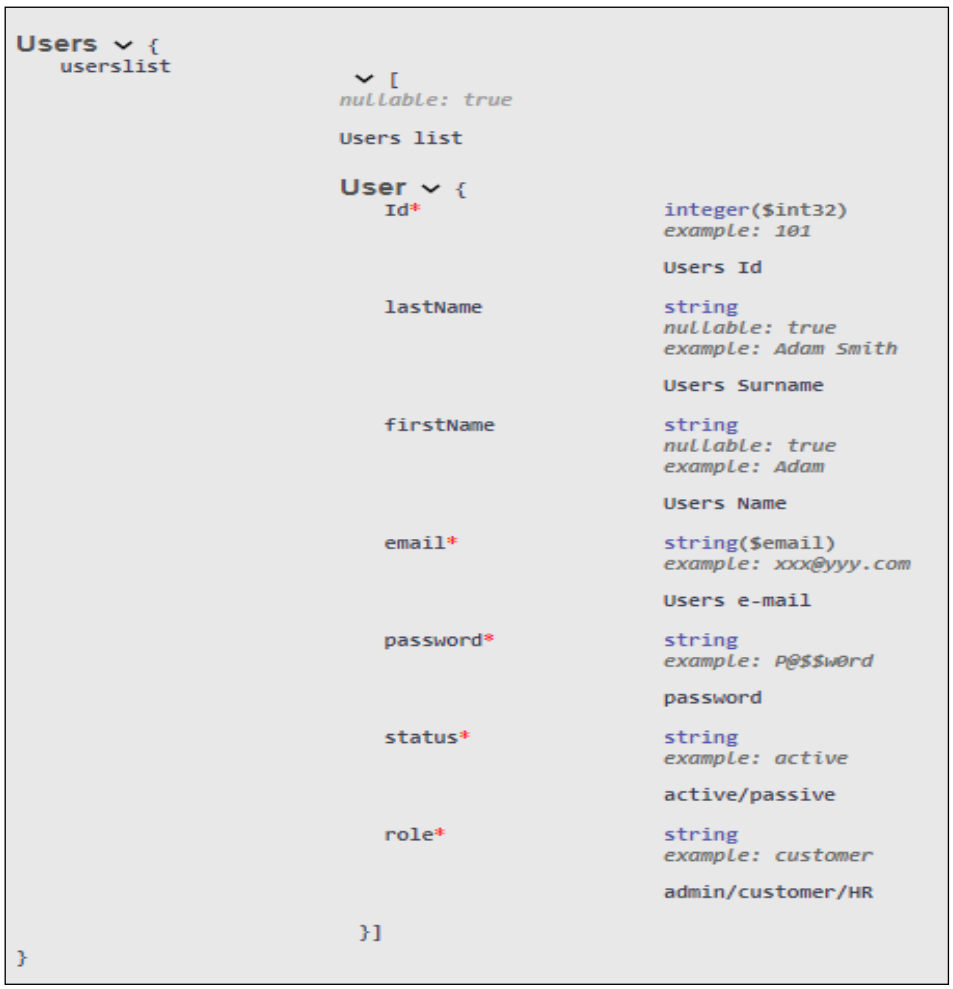
1. Разработайте ASP.NET Core MVC приложение со встроенным сервером (автономное) и реализующее следующий REST-интерфейс.



User {

Id*	integer($int32) *example: 101* Users Id
lastName	string *nullable: true* *example: Adam Smith* Users Surname
firstName	string *nullable: true* *example: Adam* Users Name
email*	string($email) *example: xxx@yyy.com* Users e-mail
password*	string *example: P@$$w0rd* password
status*	string *example: active* active/passive
role*	string *example: customer* admin/customer/HR

}



Users {

userslist [

nullable: true

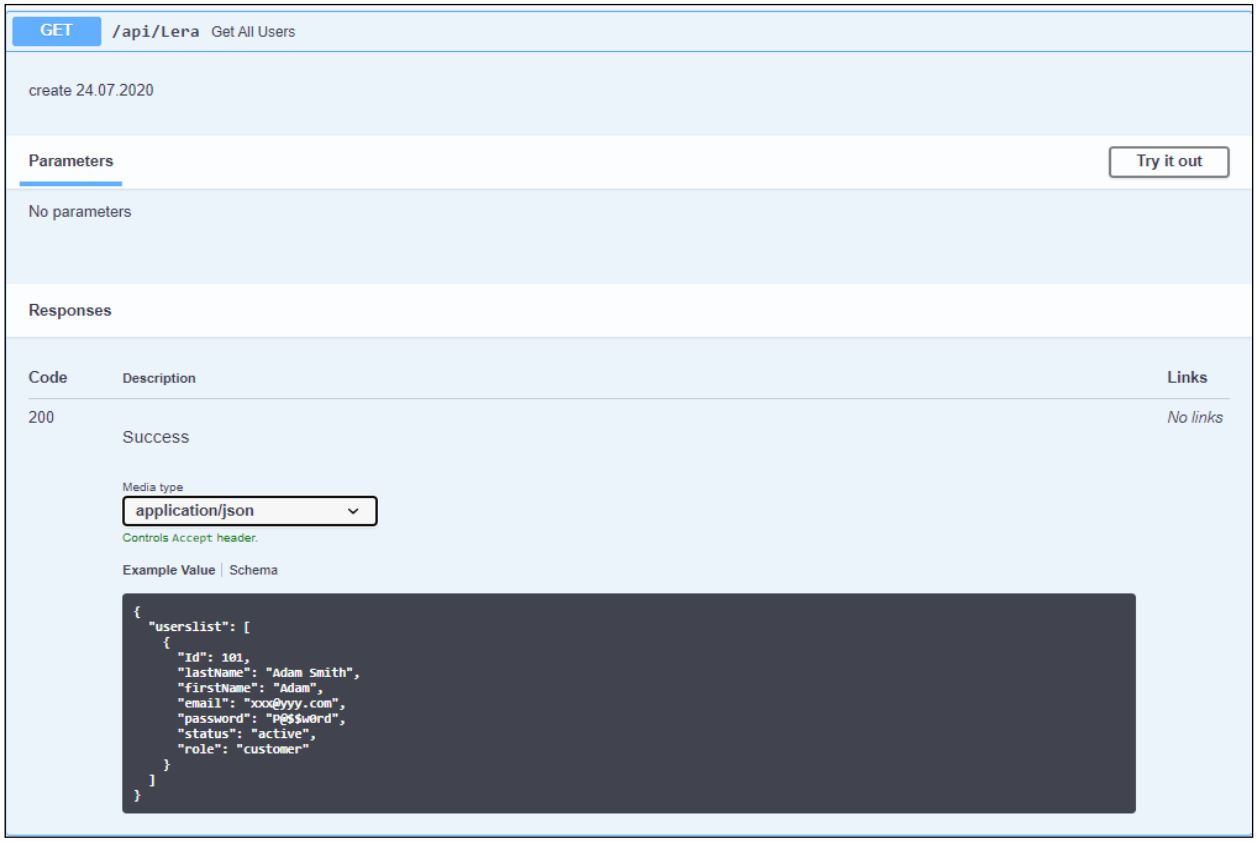
Users list

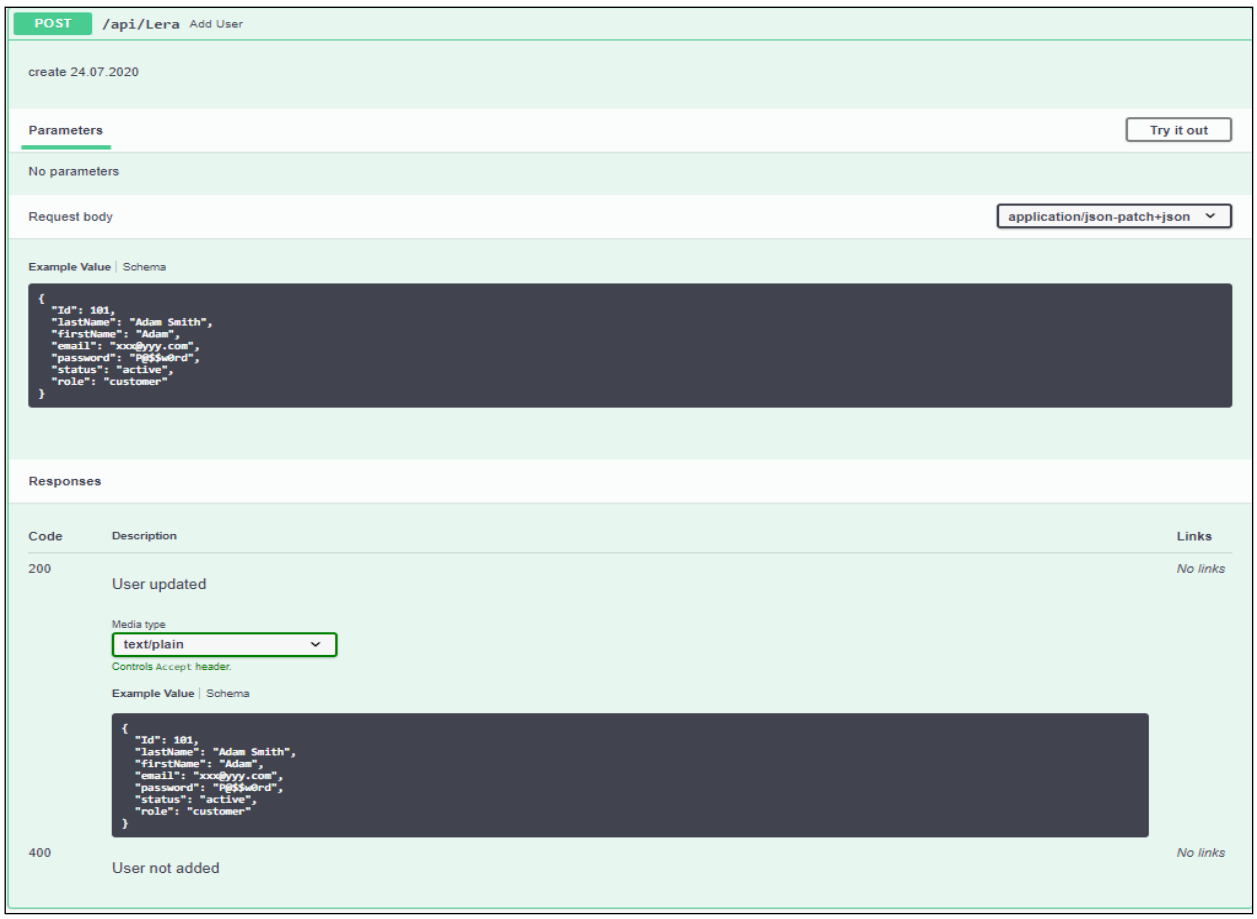
User {

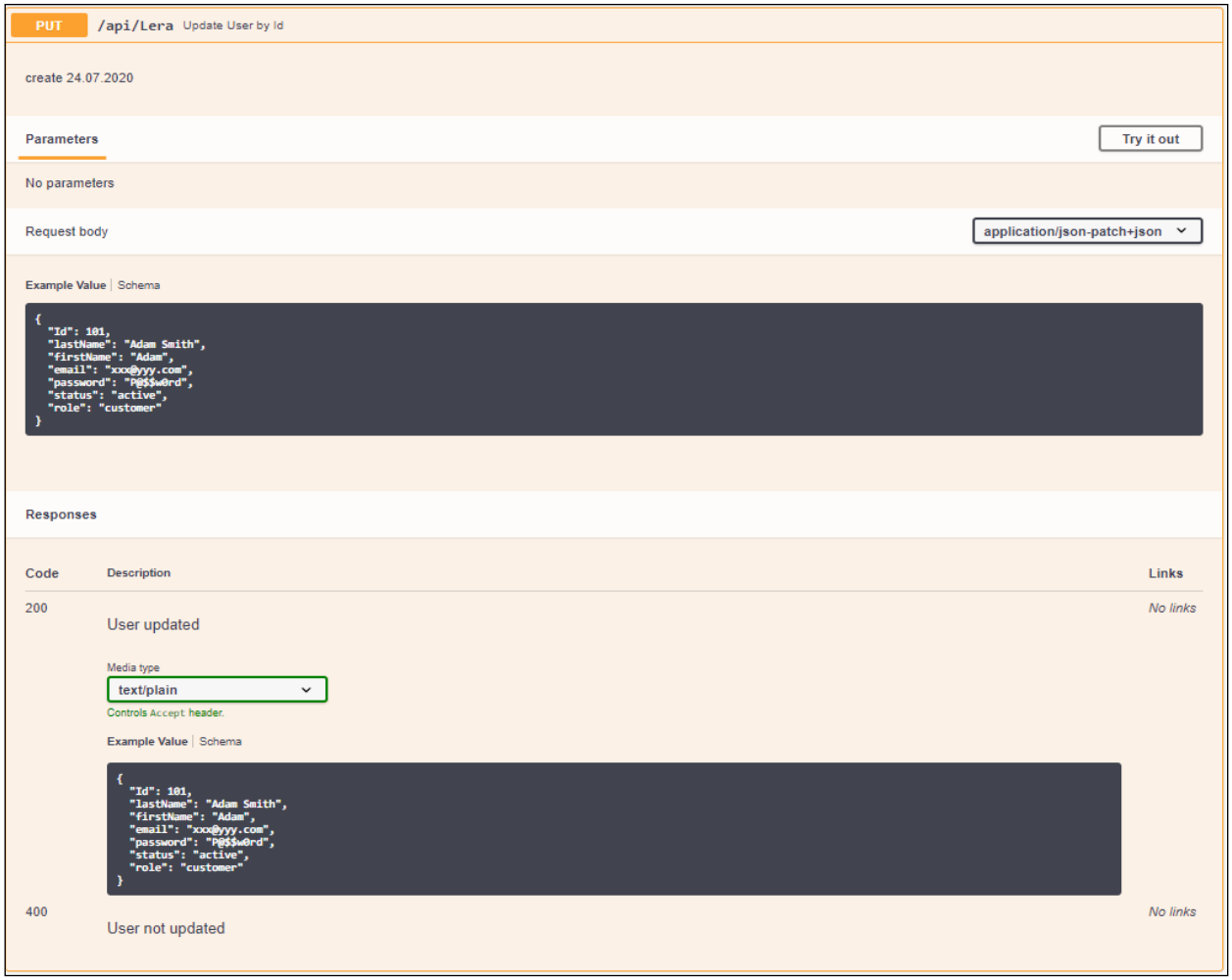
Id*	integer($int32) *example: 101* Users Id
lastName	string *nullable: true* *example: Adam Smith* Users Surname
firstName	string *nullable: true* *example: Adam* Users Name
email*	string($email) *example: xxx@yyy.com* Users e-mail
password*	string *example: P@$$w0rd* password
status*	string *example: active* active/passive
role*	string *example: customer* admin/customer/HR

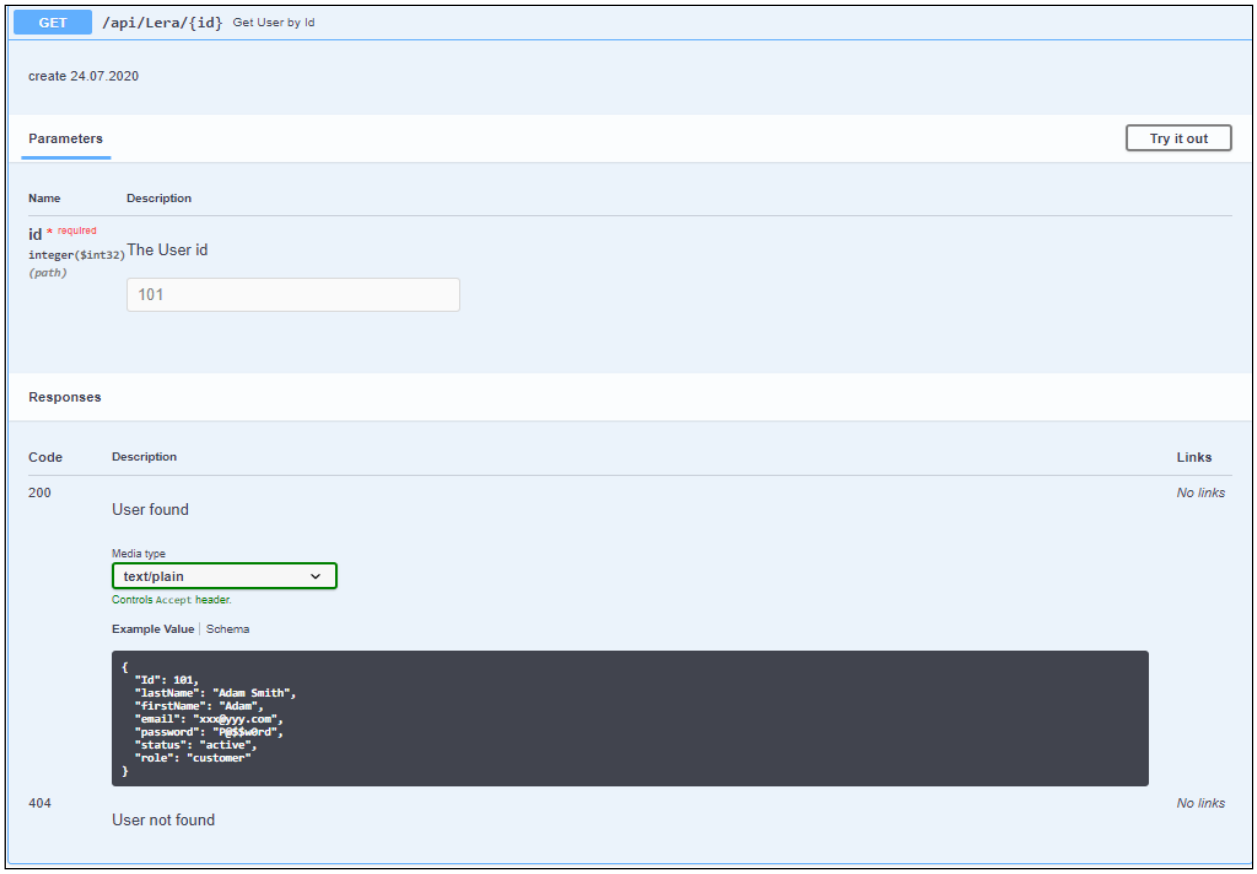
}]

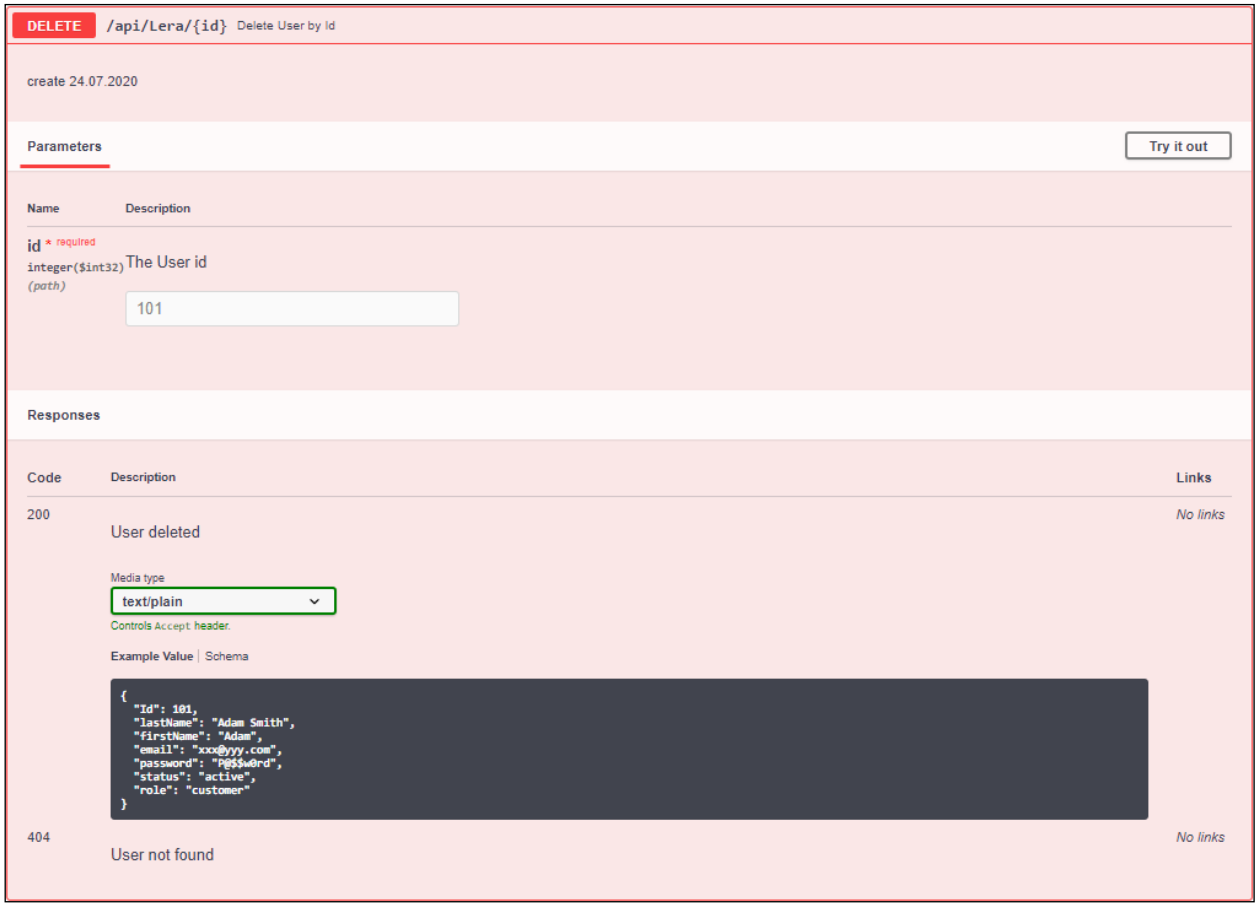
}


GET
/api/Lera
Get All Users
create 24.07.2020
Parameters
Try it out
No parameters
Responses
Code
Description
Links
200
Success
No links
Media type
application/json
Controls Accept header.
Example Value | Schema
{
"userslist": [
{
"Id": 101,
"lastName": "Adam Smith",
"firstName": "Adam",
"email": "xxx@yyy.com",
"password": "P@$$w0rd",
"status": "active",
"role": "customer"
}
]
}


POST
/api/Lera
Add User
create 24.07.2020
Parameters
Try it out
No parameters
Request body
application/json-patch+json
Example Value | Schema
{
"Id": 101,
"lastName": "Adam Smith",
"firstName": "Adam",
"email": "xxx@yyy.com",
"password": "P@$$w0rd",
"status": "active",
"role": "customer"
}
Responses
Code
Description
Links
200
User updated
No links
Media type
text/plain
Controls Accept header.
Example Value | Schema
{
"Id": 101,
"lastName": "Adam Smith",
"firstName": "Adam",
"email": "xxx@yyy.com",
"password": "P@$$w0rd",
"status": "active",
"role": "customer"
}
400
User not added
No links


PUT
/api/Lera
Update User by Id
create 24.07.2020
Parameters
Try it out
No parameters
Request body
application/json-patch+json
Example Value | Schema
{
"Id": 101,
"lastName": "Adam Smith",
"firstName": "Adam",
"email": "xxx@yyy.com",
"password": "P@$$w0rd",
"status": "active",
"role": "customer"
}
Responses
Code
Description
Links
200
User updated
No links
Media type
text/plain
Controls Accept header.
Example Value | Schema
{
"Id": 101,
"lastName": "Adam Smith",
"firstName": "Adam",
"email": "xxx@yyy.com",
"password": "P@$$w0rd",
"status": "active",
"role": "customer"
}
400
User not updated
No links


GET
/api/Lera/{id}
Get User by Id
create 24.07.2020
Parameters
Try it out
Name
Description
id * required
integer($int32)
(path)
The User id
101
Responses
Code
Description
Links
200
User found
No links
Media type
text/plain
Controls Accept header.
Example Value | Schema
{
"Id": 101,
"lastName": "Adam Smith",
"firstName": "Adam",
"email": "xxx@yyy.com",
"password": "P@$$w0rd",
"status": "active",
"role": "customer"
}
404
User not found
No links



2. Для хранения данных используйте MS SQL Server.
3. Доступ к данным реализовать с помощью ORM-репозитория.
4. Подключение ORM-репозитория осуществите с помощью встроенного механизма Inject.
5. Проверку работоспособности приложения выполните с помощью POSTMAN.

Задание 2

6. Для приложения, разработанного в задании 1 разработайте online Swagger-документацию.
7. Продемонстрируйте работоспособность web-интерфейса Swagger-документации.
8. Продемонстрируйте работоспособность функции **Try it out**.

Задание 3

9. Ответьте на следующие вопросы:
10. Что такое REST?
11. Что такое Open API, Swagger?
12. Поясните назначение xml-файла, который генерируется Swagger.