**StudentController(5):**

-getAllStudents() – returns the list of users with STUDENT role.

GET, <http://localhost:8080/final/api/std/all/>

-specialtiesOfSubject(id) – accepts id of a subject and returns the list of specialties related to given subject

GET, <http://localhost:8080/final/api/std/specs/1>

-universitiesOfSpecialty(id) – accepts id of a specialty and returns the list of university with such specialty

GET, <http://localhost:8080/final/api/std/unis/2>

-universitySubjects(id) – accepts id of an university and returns the list of subjects of this university

GET, <http://localhost:8080/final/api/std/uniSubjects/2>

-specialtySubjects(id) – accepts id of a specialty and returns the list of related subjects

GET, <http://localhost:8080/final/api/std/specSubjects/3>

AdministrationController(7):

-getAllUsers() – returns the list of all users

GET, <http://localhost:8080/final/api/admin/users/>

-getUserById(id) – accepts the id of a user and returns the user with following id

GET, <http://localhost:8080/final/api/admin/users/5>

-createUser(User user) – accepts the user in json format and inserts the row in Users table in db

POST, <http://localhost:8080/final/api/admin/createUser/>

{

    "username":"NewUser",

    "name":"NewUserName",

    "surname":"Nuga",

    "email":"Yo@mail.ru",

    "password":"123"

}

-updateUserRole(userId, roleId) – accepts the user Id and role Id and adds the role with following id to user with such id

PUT, <http://localhost:8080/final/api/admin/updRole/>

-deleteUser(id) – accepts id of User and deletes the row if such exists in Users table

DELETE, <http://localhost:8080/final/api/admin/deleteUser/7>

-authenticate(email, password) – accepts form parameters(email, password), verifies if user with following credentials exists, creates JWT token and sends back in header.

POST, <http://localhost:8080/final/api/admin/authenticate/>

-getVacancyForConsideration() – uses Job service to get jms message with Vacancy, then the vacancy is being verified and if it passes through it is created in table Vacancy(db).

GET, <http://localhost:8080/final/api/admin/considerVacancy/>

**BusinessController(4):**

-getBusinessNews() – returns all business news

GET, <http://localhost:8080/final/api/business/news>

-getNewsAuthor(id) – accepts id of news and returns the author of the news.

GET, <http://localhost:8080/final/api/business/authorOf/2>

-getNewsOfAuthor(id) – accepts id of author and returns the list of all news he/she wrote

GET, <http://localhost:8080/final/api/business/newsByAuthor/6>

-getNewsById(id) – accepts id of news and returns it

GET, <http://localhost:8080/final/api/business/newsById/1>

**TouristController(4):**

-getAllSights() – returns all tourist sights

GET, <http://localhost:8080/final/api/tourist/sights/>

-getSightById(int id) – accepts the id of a tourist sight and returns it

GET, <http://localhost:8080/final/api/tourist/sights/1>

-getSightsByPlaceId(int id) – accepts id of Place and returns Sights that located there

GET, <http://localhost:8080/final/api/tourist/sightsByPlace/2>

-getPlaceBySightId(int id) – accepts id of Sight and returns the place it’s located in

GET, <http://localhost:8080/final/api/tourist/sightPlace/2>

**JobController(3):**

-getVacancies() – returns all vacancies

GET, <http://localhost:8080/final/api/job/vacancies/>

-getVacanciesFilterdByDate(year) – accepts year and returns all vacancies which date is after the year parameter

GET, <http://localhost:8080/final/api/job/vacanciesFiltered/2020>

-sendVacancyJms(jsonVaccancy) – accepts vacancy in json and sends jms message for consideration

POST, <http://localhost:8080/final/api/job/sendVacancy/>

{

    "position":"position1",

    "postDate":"12.10.2021",

    "salary":390000

}