ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доцент, к.т.н. |  |  |  | П.А. Степанов |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

­­

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| РАЗРАБОТКА РЕСУРСА REST/JSON СЕРВИСА |
| по дисциплине: ТЕХНОЛОГИИ РАЗРАБОТКИ СЕРВЕРНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4032 |  |  |  | М.В. Мухортов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**Задание**

1. Определите перечень Rest-сервисов, выполняющих те же действия, что и в лабораторной работе 1. Внимательно отнеситесь к вопросу какой HTTP метод использует тот или иной сервис и какие коды HTTP он может возвращать.
2. Опишите Ваш API с помощью OpenAPI v3.
3. Средствами swagger сгенерируйте сервер spring
4. Замените pom файл в сгенерированном сервере на файл из моего примера
5. Поменяйте названия пакетов на уникальные
6. Реализуйте приложение.

Вариант7: Ведение списка литературы согласно последнему актуальному ГОСТу

**Описание разрабатываемого продукта**

В ходе выполнения работы необходимо разработать серверное приложение с реализацией REST контроллера средствами Spring-boot. Приложение должно иметь возможность просмотра, добавления, удаления и вставки (изменения) литературы в список.

**Описание API**

openapi: "3.0.1"  
info:  
 title: "OpenAPI definition"  
 version: "v0"  
servers:  
 - url: "http://localhost:8080/"  
 description: "Generated server url"  
paths:  
 /public/rest/literature/{id}/{source}:  
 put:  
 tags:  
 - "literature-rest-service"  
 operationId: "put"  
 parameters:  
 - name: "id"  
 in: "path"  
 required: true  
 schema:  
 type: "integer"  
 format: "int32"  
 - name: "source"  
 in: "path"  
 required: true  
 schema:  
 type: "string"  
 responses:  
 "200":  
 description: "OK"  
 content:  
 '\*/\*':  
 schema:  
 type: "object"  
 /public/rest/literature/{source}:  
 post:  
 tags:  
 - "literature-rest-service"  
 operationId: "add"  
 parameters:  
 - name: "source"  
 in: "path"  
 required: true  
 schema:  
 type: "string"  
 responses:  
 "200":  
 description: "OK"  
 content:  
 '\*/\*':  
 schema:  
 type: "object"  
 /public/rest/literature:  
 get:  
 tags:  
 - "literature-rest-service"  
 operationId: "browse"  
 responses:  
 "200":  
 description: "OK"  
 content:  
 '\*/\*':  
 schema:  
 type: "object"  
 /public/rest/literature/{id}:  
 get:  
 tags:  
 - "literature-rest-service"  
 operationId: "getOne"  
 parameters:  
 - name: "id"  
 in: "path"  
 required: true  
 schema:  
 type: "integer"  
 format: "int32"  
 responses:  
 "200":  
 description: "OK"  
 content:  
 '\*/\*':  
 schema:  
 type: "object"  
 delete:  
 tags:  
 - "literature-rest-service"  
 operationId: "delete"  
 parameters:  
 - name: "id"  
 in: "path"  
 required: true  
 schema:  
 type: "integer"  
 format: "int32"  
 responses:  
 "200":  
 description: "OK"  
components: {}

**Текст основных фрагментов кода**

@RestController  
@RequestMapping("/public/rest/literature")  
public class LiteratureRestService {  
  
 @Autowired  
 private LiteratureService literatureService;  
  
 @RequestMapping(value = "", method = RequestMethod.*GET*)  
 public ResponseEntity<Object> browse() {  
 return ResponseEntity.*ok*(literatureService.listAll());  
 }  
  
 @RequestMapping(value = "/{id}", method = RequestMethod.*DELETE*)  
 public void delete(@PathVariable("id") Integer id) {  
 literatureService.delete(id);  
 }  
  
 @RequestMapping(value = "/{id}", method = RequestMethod.*GET*)  
 public ResponseEntity<Object> getOne(@PathVariable("id") Integer id) {  
 Literature literatureById = literatureService.getById(id);  
 if (literatureById == null) {  
 return ResponseEntity.*notFound*().build();  
 } else {  
 return ResponseEntity.*ok*(literatureById);  
 }  
 }  
  
 @RequestMapping(value = "/{source}", method = RequestMethod.*POST*)  
 public ResponseEntity<Object> add(@PathVariable("source") String source) {  
 return ResponseEntity.*ok*(literatureService.add(source));  
 }  
  
 @RequestMapping(value = "/{id}/{source}", method = RequestMethod.*PUT*)  
 public ResponseEntity<Object> put(@PathVariable("id") Integer id, @PathVariable("source") String source) {  
 return ResponseEntity.*ok*(literatureService.put(id, source));  
 }  
}