

Assignment -4

1. An API to fetch all fees paid by a student
2. An API to pay fees for a student

Fee Service Controller having two services /details and /pay

```

StudentService.java  application.yml  Studentmanagemen...  studentmanagemen...  StudentController.j...  Student.java
1  /**
4  package com.eduxcel.fee.controller;
5
6  import java.util.List;
19
20 /**
21  * @author Jagadeesht
22  *
23  */
24
25 @RestController
26 @RequestMapping("/fee")
27 public class FeeController {
28     @Autowired
29     FeeService feeService;
30
31     @GetMapping("/details/{student_id}")
32     @ResponseBody
33     public List<Fees> getFeeDetails(@PathVariable("student_id") Long student_id) {
34         return feeService.getFeeDetails(student_id);
35     }
36
37     @PostMapping("/pay")
38     @ResponseBody
39     public Fees pay(@RequestBody Fees fee) {
40         System.out.println("pay fee.....");
41         return feeService.pay(fee);
42     }
43
44 }
45

```

FeeService

```

StudentServi...  application.yml  Studentmanag...  studentmanag...  StudentContr...  Student.java
1  /**
4  package com.eduxcel.fee.services;
5
6  import java.time.Instant;
16
17 /**
18  * @author Jagadeesht
19  *
20  */
21 @Service
22 public class FeeService {
23
24     @Autowired
25     private FeeRepository feeRepository;
26
27     public List<Fees> getFeeDetails(@PathVariable("student_id") Long student_id) {
28
29         return feeRepository.findByStudentId(student_id);
30     }
31
32     @Transactional
33     public Fees pay(Fees fee) {
34         fee.setDate(Instant.now().toString());
35         return feeRepository.save(fee);
36     }
37
38 }
39

```

```
39 }
40
```

FeeRepository

```
StudentServ... Studentmanag... studentmanag... StudentCont... Student.java FeeC
1 */
4 package com.eduexcel.fee.repository;
5
6 import java.util.List;
13
14 /**
15  * @author Jagadeesht
16  *
17  */
18 @Repository
19 public interface FeeRepository extends JpaRepository<Fees, Long>{
20
21     @Query("from Fees where studentId=:studentId")
22     List<Fees> findByStudentId(Long studentId);
23
24 }
25
```

Result for fee details

Feemanagement / feedetails

GET http://localhost:8010/fee/details/1

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
-----	-------

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "feeId": 1,
4     "courseId": "500",
5     "semester": "semester-1",
6     "feeDetails": 50000,
7     "studentId": 1,
8     "date": "2022-10-16T05:08:03.513929400Z"
9   },
10  {
11    "feeId": 2,
12    "courseId": "500",
13    "semester": "semester-2",
14    "feeDetails": 50000,
15    "studentId": 1,
16    "date": "2022-10-16T05:08:10.364605100Z"
17  },
18  ]
19
20 "feeId": 3,
21 "courseId": "500",
22 "semester": "semester-3",
23 "feeDetails": 50000,
24 "studentId": 1
```

```
23         "studentId": 1,  
24         "date": "2022-10-16T05:08:14.082061700Z"  
25     },  
26     {  
27         "feeId": 4,  
28         "courseId": "500",  
29         "semester": "semester-4",  
30         "feeDetails": 50000,  
31         "studentId": 1,  
32         "date": "2022-10-16T05:08:18.086052300Z"
```

/pay

POST

http://localhost:8010/fee/pay

Params

Authorization

Headers (8)

Body

Pre-request Script

none

form-data

x-www-form-urlencoded

raw

binary

```
1  {  
2    ...  
3    ... "courseId": "500",  
4    ... "semester": "semester-4",  
5    ... "feeDetails": 50000,  
6    ... "studentId": 3  
7  }
```

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1  {  
2    "feeId": 12,  
3    "courseId": "500",  
4    "semester": "semester-4",  
5    "feeDetails": 50000,  
6    "studentId": 3,  
7    "date": "2022-10-16T05:08:50.471586700Z"  
8  }
```

Updated Student API to invoke the fee management service APIs

StudentServ Studentmanag studentmanag StudentCont StudentCont StudentCont FeeControll

```

4 package com.equexcel.studentmanagement;
5
6 import java.util.List;
7
8 /**
9  * @author Jagadeesht
10  */
11 @RestController
12 public class StudentController {
13
14     @Autowired
15     private StudentService studentService;
16
17     @GetMapping("/students")
18     @ResponseBody
19     public List<Student> getStudents() {
20         return studentService.getStudents();
21     }
22
23     @GetMapping("/student/{id}")
24     @ResponseBody
25     public Student getStudents(@PathVariable("id") Long studentid) {
26         return studentService.getStudent(studentid);
27     }
28
29     @PostMapping("/student")
30     @ResponseBody
31     public Student createStudents(@RequestBody Student student) {
32         return studentService.createStudent(student);
33     }
34
35     @DeleteMapping("/student/{id}")
36     @ResponseBody
37     public String deleteStudent(@PathVariable("id") Long studentid) {
38         return studentService.deleteStudent(studentid);
39     }
40
41     @PutMapping("/student")
42     @ResponseBody
43     public Student updateStudents(@RequestBody Student student) {
44         return studentService.updateStudent(student);
45     }
46
47     @GetMapping("/feedetail/{id}")
48     @ResponseBody
49     public List<Fees> getFeeDetails(@PathVariable("id") Long studentid) {
50         return studentService.getFeeDetails(studentid);
51     }
52
53     @PostMapping("/payfee")
54     @ResponseBody
55     public Fees pay(@RequestBody Fees fee) {
56         return studentService.payFee(fee);
57     }
58 }
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75 }
76

```

Student Service has Autowired RestTemplate to call Fee management API

```

20 /**
21  * @author Jagadeesht
22  */
23 @Service
24 public class StudentService {
25
26     @Autowired
27     private StudentRepository studentRepository;
28
29     @Autowired
30     private RestTemplate restTemplate;
31
32     @Autowired
33     Environment env;
34
35     @Value("${fee.service.url}")
36     private String feeServiceUrl;
37
38
39

```

```
40 public List<Student> getStudents() {
41     return studentRepository.findAll();
42 }
43
44 public Student getStudent(Long studentId) {
45     return studentRepository.findById(studentId).get();
46 }
47
48 @Transactional
49 public Student createStudent(Student student) {
50     return studentRepository.save(student);
51 }
52
53 @Transactional
54 public String deleteStudent(Long studentId) {
55     studentRepository.deleteById(studentId);
56     return "deleted Student " + studentId;
57 }
58
59 @Transactional
60 public Student updateStudent(Student student) {
61     return studentRepository.save(student);
62 }
63
64
65 public Fees payFee(Fees fee) {
66     ResponseEntity<Fees> entity = restTemplate.postForEntity(feeServiceUrl+"/pay", fee, Fees.class);
67     return entity.getBody();
68 }
69
70 public List<Fees> getFeeDetails(Long studentid) {
71     String url = feeServiceUrl+"/details/"+studentid;
72     System.out.println("....."+studentid);
73     ResponseEntity<List> entity = restTemplate.getForEntity(url, List.class);
74     return entity.getBody();
75 }
76 }
77 }
78 }
```

Test Result

StudentManagement / feedetails

GET

http://localhost:8080/feedetail/1

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
-----	-------

Body Cookies Headers (5) Test Results

Pretty

Raw

Preview

Visualize

JSON

1

[

2

{

3

"feeId": 1,

4

"courseId": "500",

5

"semester": "semester-1",

6

"feeDetails": 50000,

7

"studentId": 1,

8

"date": "2022-10-16T05:08:03.513929400Z"

9

},

10

{

11

"feeId": 2,

12

"courseId": "500",

13

"semester": "semester-2",

14

"feeDetails": 50000,

15

"studentId": 1,

16

"date": "2022-10-16T05:08:10.364605100Z"

17

},

18

{

19

"feeId": 3,

20

"courseId": "500",

21

"semester": "semester-3"

```
21     "semester": "semester-5",
22     "feeDetails": 50000,
23     "studentId": 1,
24     "date": "2022-10-16T05:08:14.082061700Z"
25 },
26 {
27     "feeId": 4,
28     "courseId": "500",
29     "semester": "semester-4",
30     "feeDetails": 50000,
31     "studentId": 1,
32     "date": "2022-10-16T05:08:18.086052300Z"
```

/pay -result

StudentManagement / fee pay

POST

http://localhost:8080/payfee

Params

Authorization

Headers (8)

Body

Pre-request Script

none

form-data

x-www-form-urlencoded

raw

binary

1

{

2

...

3

... "courseId": "500",

4

... "semester": "semester-5",

5

... "feeDetails": 40000,

6

... "studentId": 1

7

}

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

{

2

"courseId": "500",

3

"semester": "semester-5",

4

"feeDetails": 40000,

5

"studentId": 1

6

}