

CAPSTONE PROJECT

PAYROLL MANAGEMENT SYSTEM

BATCH-6-JAVA2EE

Name: SHAIK MOHAMMED SHAFIULLAH

Email : shaikshafiullah799@gmail.com

Date: September 2,2024

Team Leader: SARAS SINGH

PROBLEM STATEMENT :

- Build an application that manages payroll for the employees of an organization.
- The system admin should be able to perform below operations
 - Add employee
 - Add/Update employee
 - Payroll processing
 - Financial Reporting
 - View payrollhistory

ADMIN SHOULD BE ABLE TO :

- Create a spring starter project and create packages for Controllers,Repository,Entity,Service,JWT security then write the test cases for service classes and add all dependencies that should need in the pom.xml with correct versions.
- Create another two spring starter projects for gate way, eureka and run these two.
- In the application properties mention all urls ports for eureka,swagger,gateway,actuator .
- Run the main application and test the cases and see while the webservices are working or not.
- Generate a token in swagger.
- Then generate a token in postman and perform the operations like post,get,put,delete,update.
- In Postman finally see payroll_history by using get operation.

- **BASIC FLOW/ SCENARIOS:**

- 1.Employee Management

- Add new employees, Update employee information,

- Retrieve employee details.

- 2.Payroll Processing:

- Automated calculation of employee salaries and deductions

- Generation of pay stubs for each pay period.

- 3.Tax calculation:

- Automatic computation of taxes based on employee income and deduction

- 4.Financial Reporting:

- Generation of financial reports, including income and tax statements.

Tax Calculation

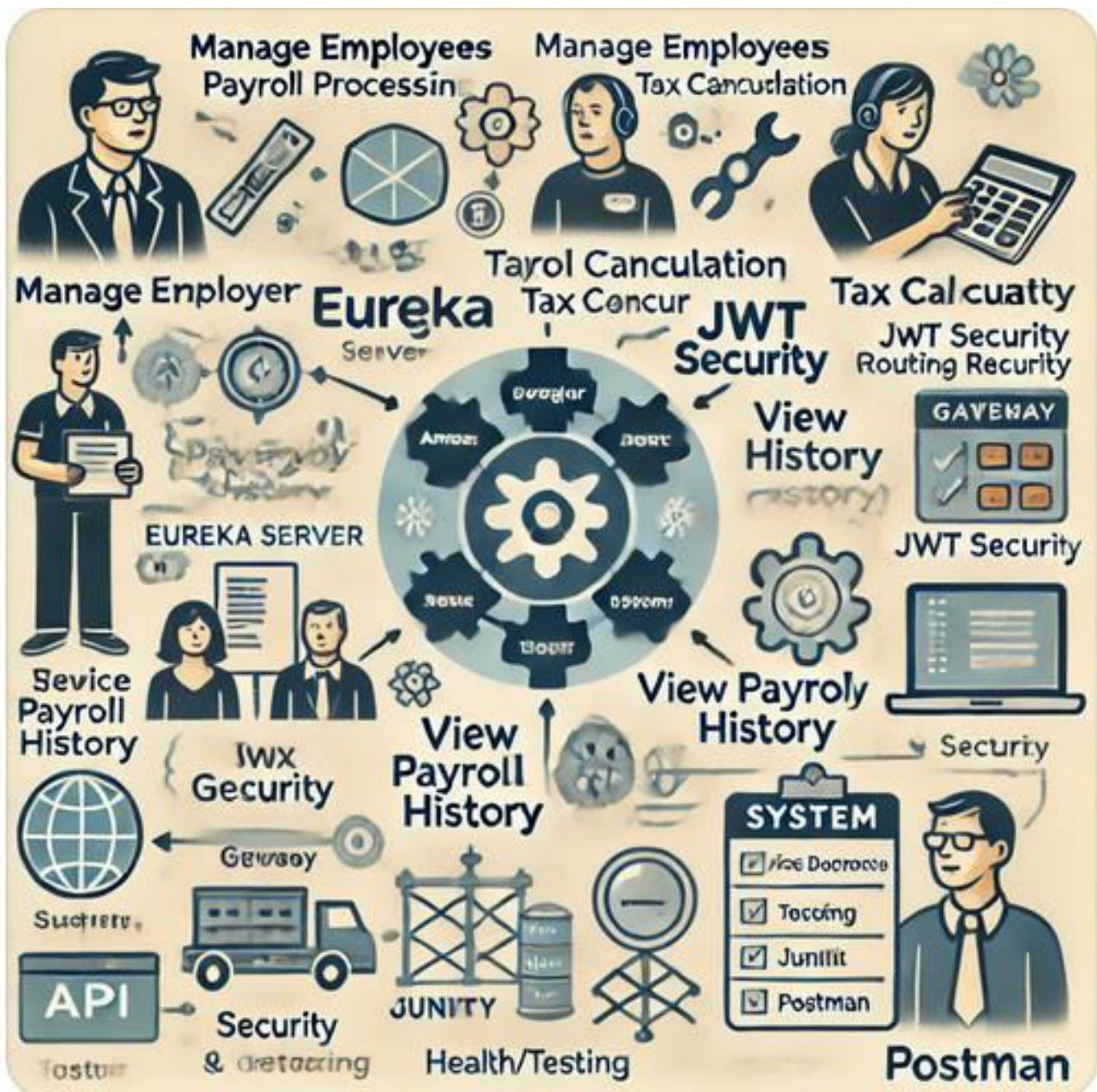
0-5 lakh no tax

5-7 lakh 10%

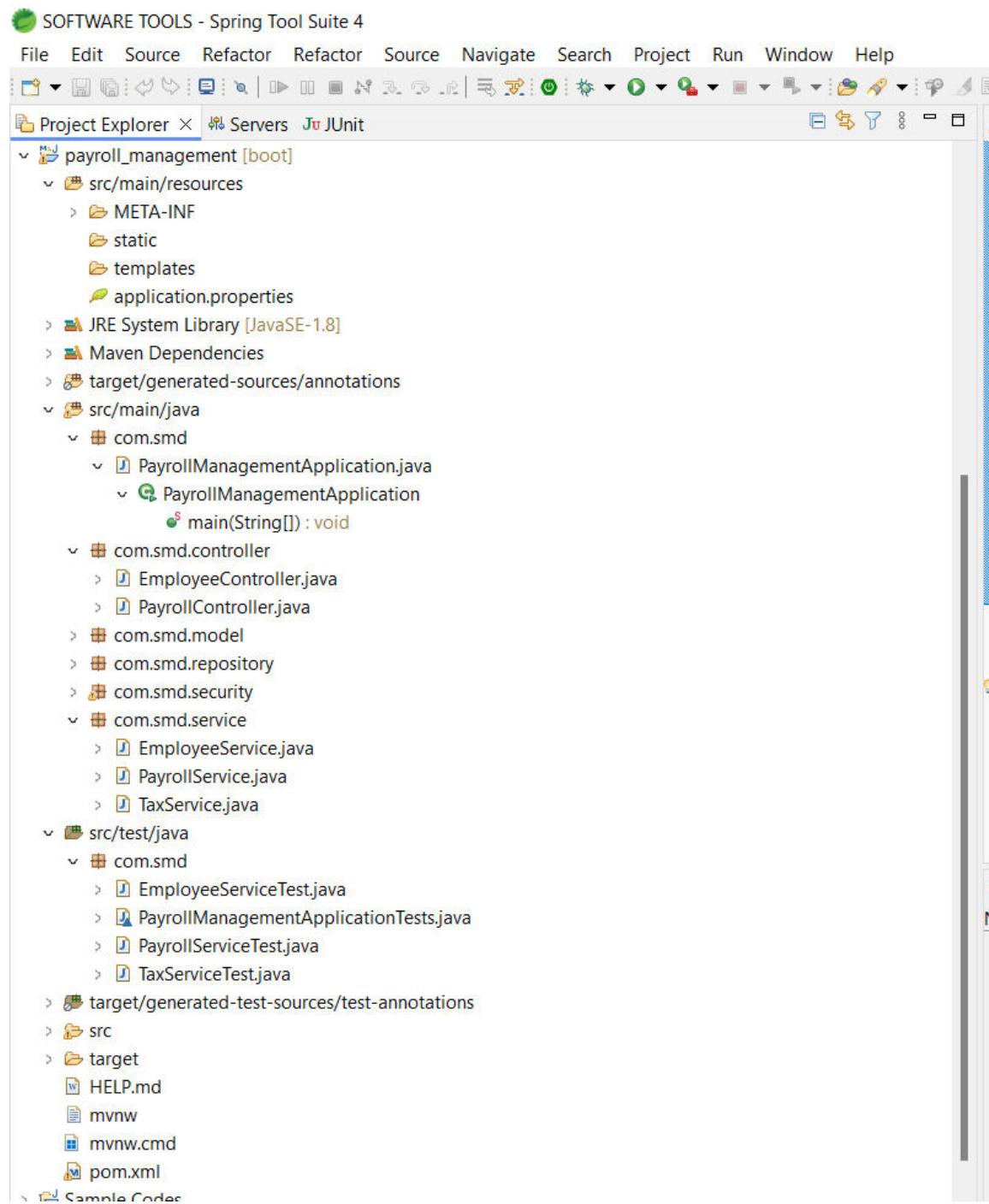
7-10 lakh 20%

10- above 30%

USECASE DIAGRAMS :



Spring -boot project format



PAYROLLMANAGEMENT MAIN APPLICATION

The screenshot shows the Spring Tool Suite (STS) interface. The central area displays the `PayrollManagementApplication.java` file:

```
1 package com.smd;
2
3 import org.springframework.boot.SpringApplication;
4
5 @SpringBootApplication
6 @EnableDiscoveryClient
7 public class PayrollManagementApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(PayrollManagementApplication.class, args);
11     }
12 }
13
14
15 }
```

The left sidebar shows the project structure under `payroll_management [boot]`, including `src/main/resources`, `src/main/java/com.smd` (containing `PayrollManagementApplication`), `src/test/java/com.smd` (containing test classes like `EmployeeServiceTest`), and `target/generated-sources/test-classes`. The bottom status bar shows the date as 02-09-2024.

CONTROLLERS - Employee,Payroll

The screenshot shows the Spring Tool Suite (STS) interface. The central area displays the `EmployeeController.java` file:

```
1 package com.smd.controller;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5 @RestController
6 @RequestMapping("/employees")
7 public class EmployeeController {
8
9     @Autowired
10    private EmployeeService employeeService;
11
12    @PostMapping
13    public ResponseEntity<Employee> addEmployee(@RequestBody Employee employee) {
14        Employee savedEmployee = employeeService.addEmployee(employee);
15        return new ResponseEntity<>(savedEmployee, HttpStatus.CREATED);
16    }
17
18    @GetMapping("/{id}")
19    public ResponseEntity<Employee> getEmployee(@PathVariable Long id) {
20        Employee employee = employeeService.getEmployee(id);
21        return new ResponseEntity<>(employee, HttpStatus.OK);
22    }
23
24    @PutMapping("/{id}")
25    public ResponseEntity<Employee> updateEmployee(@PathVariable Long id, @RequestBody Employee employee) {
26        Employee updatedEmployee = employeeService.updateEmployee(id, employee);
27        return new ResponseEntity<>(updatedEmployee, HttpStatus.OK);
28    }
29 }
```

The left sidebar shows the project structure under `payroll_management [boot]`, including `src/main/resources`, `src/main/java/com.smd` (containing `EmployeeController` and `PayrollController`), and `src/test/java/com.smd` (containing test classes like `EmployeeServiceTest`). The bottom status bar shows the date as 02-09-2024.

SOFTWARE TOOLS - payroll_management/src/main/java/com/smd/controller/EmployeeController.java - Spring Tool Suite 4

```
File Edit Source Refactor Source Navigate Project Run Window Help
Project E... × Servers JUnit × PayrollManagementApplication.java EmployeeController.java PayrollController.java
src/main/resources META-INF static templates application.properties
JRE System Library [JavaSE-1.8] Maven Dependencies target/generated-sources/annotations
src/main/java com.smd
src/main/resources META-INF static templates application.properties
src/main/java PayrollManagementApplication PayrollManagementApplicat mainString[] void
com.smd.controller EmployeeController.java PayrollController.java
com.smd.model EmployeeService.java PayrollService.java TaxService.java
src/test/java com.smd EmployeeServiceTest.java PayrollManagementApplicat PayrollServiceTest.java TaxServiceTest.java
target/generated-test-sources/test src target HELP.md mvnw mvnw.cmd
Employee savedEmployee = employeeService.addEmployee(employee);
return new ResponseEntity<>(savedEmployee, HttpStatus.CREATED);

@GetMapping("/{id}")
public ResponseEntity<Employee> getEmployee(@PathVariable Long id) {
    Employee employee = employeeService.getEmployee(id);
    return new ResponseEntity<>(employee, HttpStatus.OK);

}

@PutMapping("/{id}")
public ResponseEntity<Employee> updateEmployee(@PathVariable Long id, @RequestBody Employee employee) {
    Employee updatedEmployee = employeeService.updateEmployee(id, employee);
    return new ResponseEntity<>(updatedEmployee, HttpStatus.OK);

}

@DeleteMapping("/{id}")
public ResponseEntity<Void> deleteEmployee(@PathVariable Long id) {
    employeeService.deleteEmployee(id);
    return new ResponseEntity<>(HttpStatus.NO_CONTENT);

}

@GetMapping("/all")
public ResponseEntity<List<Employee>> getAllEmployees() {
    List<Employee> employees = employeeService.getAllEmployees();
    return new ResponseEntity<>(employees, HttpStatus.OK);

}

```

Console × Problems Debug Shell Debug Output Browser Output

No consoles to display at this time.

Writable Smart Insert 1:1:0 ENG IN 11:13 02-09-2024

SOFTWARE TOOLS - payroll_management/src/main/java/com/smd/controller/PayrollController.java - Spring Tool Suite 4

```
File Edit Source Refactor Source Navigate Project Run Window Help
Project E... × Servers JUnit × PayrollController.java
src/main/resources META-INF static templates application.properties
JRE System Library [JavaSE-1.8] Maven Dependencies target/generated-sources/annotations
src/main/java com.smd
src/main/resources META-INF static templates application.properties
src/main/java PayrollManagementApplicat PayrollManagementApplicat mainString[] void
com.smd.controller EmployeeController.java PayrollController.java
com.smd.model EmployeeService.java PayrollService.java TaxService.java
src/test/java com.smd EmployeeServiceTest.java PayrollManagementApplicat PayrollServiceTest.java TaxServiceTest.java
target/generated-test-sources/test src target HELP.md mvnw mvnw.cmd
package com.smd.controller;
import com.smd.model.Payroll;
@RestController
@RequestMapping("/payroll")
public class PayrollController {
    @Autowired
    private PayrollService payrollService;

    @PostMapping("/{employeeId}")
    public ResponseEntity<Payroll> processPayroll(@PathVariable Long employeeId) {
        Payroll payroll = payrollService.processPayroll(employeeId);
        return new ResponseEntity<>(payroll, HttpStatus.CREATED);

    }

    @GetMapping("/{id}")
    public ResponseEntity<Payroll> getPayroll(@PathVariable Long id) {
        Payroll payroll = payrollService.getPayroll(id);
        return new ResponseEntity<>(payroll, HttpStatus.OK);

    }

    @PutMapping("/{id}")
    public ResponseEntity<Payroll> updatePayroll(@PathVariable Long id, @RequestBody Payroll payrollDetails) {
        Payroll updatedPayroll = payrollService.updatePayroll(id, payrollDetails);
        return new ResponseEntity<>(updatedPayroll, HttpStatus.OK);

    }
}
```

Console × Problems Debug Shell Debug Output Browser Output

No consoles to display at this time.

Writable Smart Insert 1:1:0 ENG IN 11:17 02-09-2024

SOFTWARE TOOLS - payroll_management/src/main/java/com/smd/controller/PayrollController.java - Spring Tool Suite 4

```

File Edit Source Refactor Source Navigate Search Project Run Window Help
Project E... × Servers JUnit
payroll_management [boot]
src/main/resources
META-INF
static
templates
application.properties
I RE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotations
src/main/java
com.smd
PayrollManagementApplicat
└ PayrollManagementApplicat
└ mainString()
void
com.smd.controller
└ EmployeeController.java
└ PayrollController.java
com.smd.model
com.smd.repository
com.smd.security
com.smd.service
└ EmployeeService.java
└ PayrollService.java
└ TaxService.java
src/test/java
com.smd
└ EmployeeServiceTest.java
└ PayrollManagementApplicat
└ PayrollServiceTest.java
└ TaxServiceTest.java
target/generated-test-sources/test
src
target
HELP.md
mvnw
mvnw.cmd

```

```

27     Payroll payroll = payrollService.processPayroll(employeeId);
28     return new ResponseEntity<Payroll>(payroll, HttpStatus.CREATED);
29 }
30
31 @GetMapping("/{id}")
32 public ResponseEntity<Payroll> getPayroll(@PathVariable Long id) {
33     Payroll payroll = payrollService.getPayroll(id);
34     return new ResponseEntity<Payroll>(payroll, HttpStatus.OK);
35 }
36
37 @PutMapping("/{id}")
38 public ResponseEntity<Payroll> updatePayroll(@PathVariable Long id, @RequestBody Payroll payrollDetails) {
39     Payroll updatedPayroll = payrollService.updatePayroll(id, payrollDetails);
40     return new ResponseEntity<Payroll>(updatedPayroll, HttpStatus.OK);
41 }
42
43 @DeleteMapping("/{id}")
44 public ResponseEntity<Void> deletePayroll(@PathVariable Long id) {
45     payrollService.deletePayroll(id);
46     return new ResponseEntity<Void>(HttpStatus.NO_CONTENT);
47 }
48
49 @GetMapping("/report")
50 public ResponseEntity<List<Payroll>> getPayrollReport() {
51     List<Payroll> payrolls = payrollService.getAllPayrolls();
52     return new ResponseEntity<List<Payroll>>(payrolls, HttpStatus.OK);
53 }
54 }
```

Console × Problems Debug Shell Debug Output Browser Output

No consoles to display at this time.

Writable Smart Insert 1:1:0 ENG IN 11:17 02-09-2024

Entity Classes-Employee,Payroll

SOFTWARE TOOLS - payroll_management/src/main/java/com/smd/model/Employee.java - Spring Tool Suite 4

```

File Edit Source Refactor Source Navigate Search Project Run Window Help
Project E... × Servers JUnit
payroll_management [boot]
src/main/resources
META-INF
static
templates
application.properties
I RE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotations
src/main/java
com.smd
PayrollManagementApplicat
└ mainString()
void
com.smd.controller
└ EmployeeController.java
└ PayrollController.java
com.smd.model
└ Employee.java
└ Payroll.java
com.smd.repository
└ comsmdsecurity
└ comsmdservice
└ EmployeeService.java
└ PayrollService.java
└ TaxService.java
src/test/java
com.smd
└ EmployeeServiceTest.java
└ PayrollManagementApplicat
└ PayrollServiceTest.java
└ TaxServiceTest.java
target/generated-test-sources/test
src
target
HELP.md
mvnw
mvnw.cmd

```

```

1 package com.smd.model;
2
3 import javax.persistence.Entity;
4
5 @Entity
6 @Data
7 public class Employee {
8     @Id
9     @GeneratedValue(strategy = GenerationType.IDENTITY)
10    private Long id;
11
12    private String name;
13    private Integer age;
14    private String designation;
15    private Integer rating;
16    private Double salary;
17    public Long getId() {
18        return id;
19    }
20    public void setId(Long id) {
21        this.id = id;
22    }
23    public String getName() {
24        return name;
25    }
26    public void setName(String name) {
27        this.name = name;
28    }
29
30 }
```

Console × Problems Debug Shell Debug Output Browser Output

No consoles to display at this time.

Writable Smart Insert 1:1:0 ENG IN 11:19 02-09-2024

SOFTWARE TOOLS - payroll_management/src/main/java/com/smd/model/Employee.java - Spring Tool Suite 4

```
File Edit Source Refactor Source Navigate Project Run Window Help
Project E... × Servers JUnit EmployeeJava PayrollJava
43 public void setDesignation(String designation) {
44     this.designation = designation;
45 }
46 public Integer getRating() {
47     return rating;
48 }
49 public void setRating(Integer rating) {
50     this.rating = rating;
51 }
52 public Double getSalary() {
53     return salary;
54 }
55 public void setSalary(Double salary) {
56     this.salary = salary;
57 }
58 public Employee(Long id, String name, Integer age, String designation, Integer rating, Double salary) {
59     super();
60     this.id = id;
61     this.name = name;
62     this.age = age;
63     this.designation = designation;
64     this.rating = rating;
65     this.salary = salary;
66 }
67 public Employee() {
68     super();
69     // TODO Auto-generated constructor stub
70 }
```

Console × Problems Debug Shell Debug Output Browser Output
No consoles to display at this time.

Writable Smart Insert 1:1:0 ENG IN 11:19 02-09-2024

SOFTWARE TOOLS - payroll_management/src/main/java/com/smd/model/Payroll.java - Spring Tool Suite 4

```
File Edit Source Refactor Source Navigate Project Run Window Help
Project E... × Servers JUnit EmployeeJava PayrollJava
1 package com.smd.model;
2 import java.util.Date;
3
4 @Entity
5 @Data
6 public class Payroll {
7     @Id
8     @GeneratedValue(strategy = GenerationType.IDENTITY)
9     private Long id;
10    @ManyToOne
11    @JoinColumn(name = "employee_id")
12    private Employee employee;
13
14    private Double basicSalary;
15    private Double taxDeductions;
16    private Double netSalary;
17    private Date payDate;
18
19    public Long getId() {
20        return id;
21    }
22
23    public void setId(Long id) {
24        this.id = id;
25    }
26
27    public Employee getEmployee() {
28        return employee;
29    }
30
31    public void setEmployee(Employee employee) {
32        this.employee = employee;
33    }
34
35}
```

Console × Problems Debug Shell Debug Output Browser Output
No consoles to display at this time.

Writable Smart Insert 17:21:423 ENG IN 11:21 02-09-2024

File Edit Source Refactor Source Navigate Search Project Run Window Help

Project Employee Payroll Application

```

Employee.java
1 package com.smd.model;
2
3 public class Employee {
4     private Long id;
5     private String name;
6     private Double basicSalary;
7     private Date payDate;
8     private Double taxDeductions;
9     private Double netSalary;
10
11     // Getters and Setters
12 }

```

```

Payroll.java
1 package com.smd.model;
2
3 public class Payroll {
4     private Long id;
5     private Employee employee;
6     private Double basicSalary;
7     private Date payDate;
8     private Double taxDeductions;
9     private Double netSalary;
10
11     // Getters and Setters
12 }

```

src/main/resources

src/main/java

src/test/java

target/generated-sources/test-classes

target

HELP.mdf

Console Problems Debug Shell Debug Output Browser Output

Writable Smart Insert 17 : 21 : 423 ENG IN 02-09-2024

Repository :Employee,Payroll

File Edit Source Refactor Source Navigate Search Project Run Window Help

Project Employee Payroll Application

```

EmployeeRepository.java
1 package com.smd.repository;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4
5 public interface EmployeeRepository extends JpaRepository<Employee, Long> {
6 }

```

src/main/resources

src/main/java

src/test/java

target/generated-sources/test-classes

target

HELP.mdf

Console Problems Debug Shell Debug Output Browser Output

Writable Smart Insert 1 : 1 : 0 Building: (90%) ENG IN 11:23 02-09-2024

File Edit Source Refactor Source Navigate Search Project Run Window Help

Project Employee Payroll Application

```

PayrollRepository.java
1 package com.smd.repository;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4
5 public interface PayrollRepository extends JpaRepository<Payroll, Long> {
6 }

```

src/main/resources

src/main/java

src/test/java

target/generated-sources/test-classes

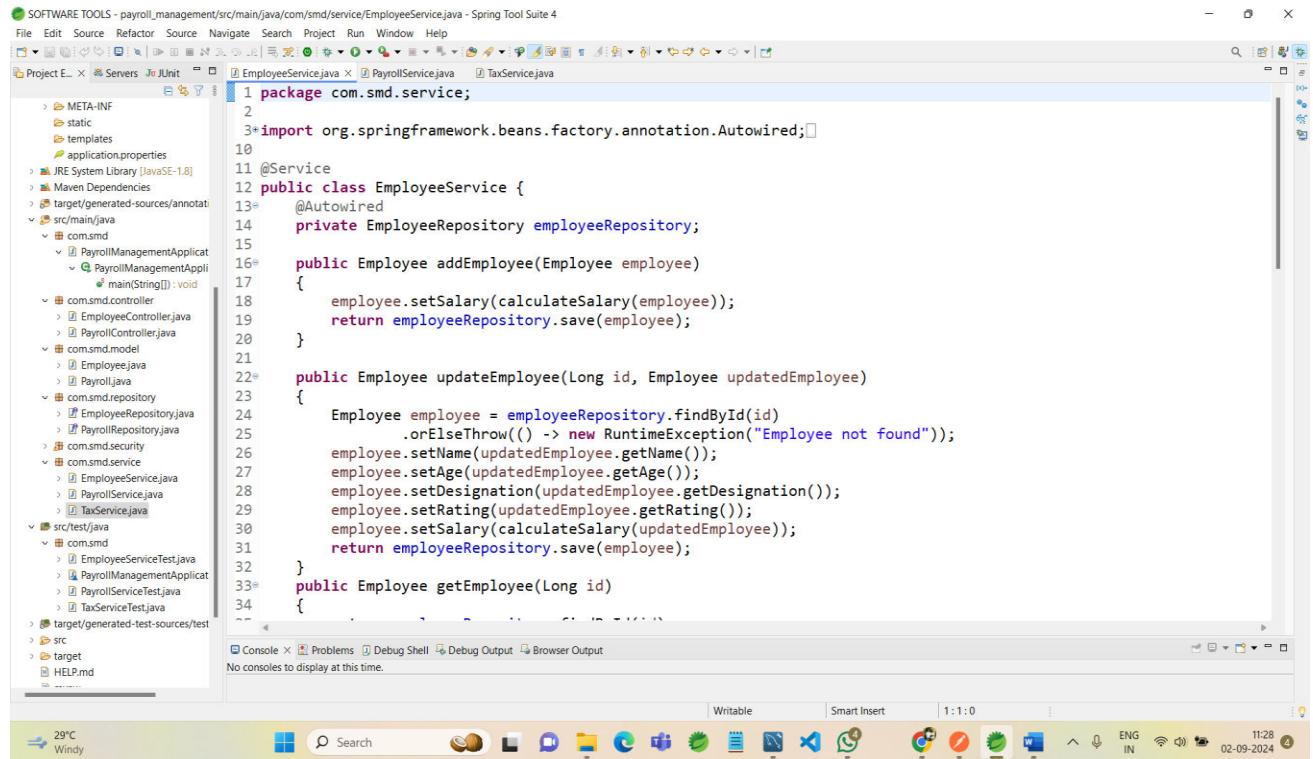
target

HELP.mdf

Console Problems Debug Shell Debug Output Browser Output

Writable Smart Insert 1 : 1 : 0 Building: (90%) ENG IN 11:23 02-09-2024

Service Classes : Employee,Payroll,Tax



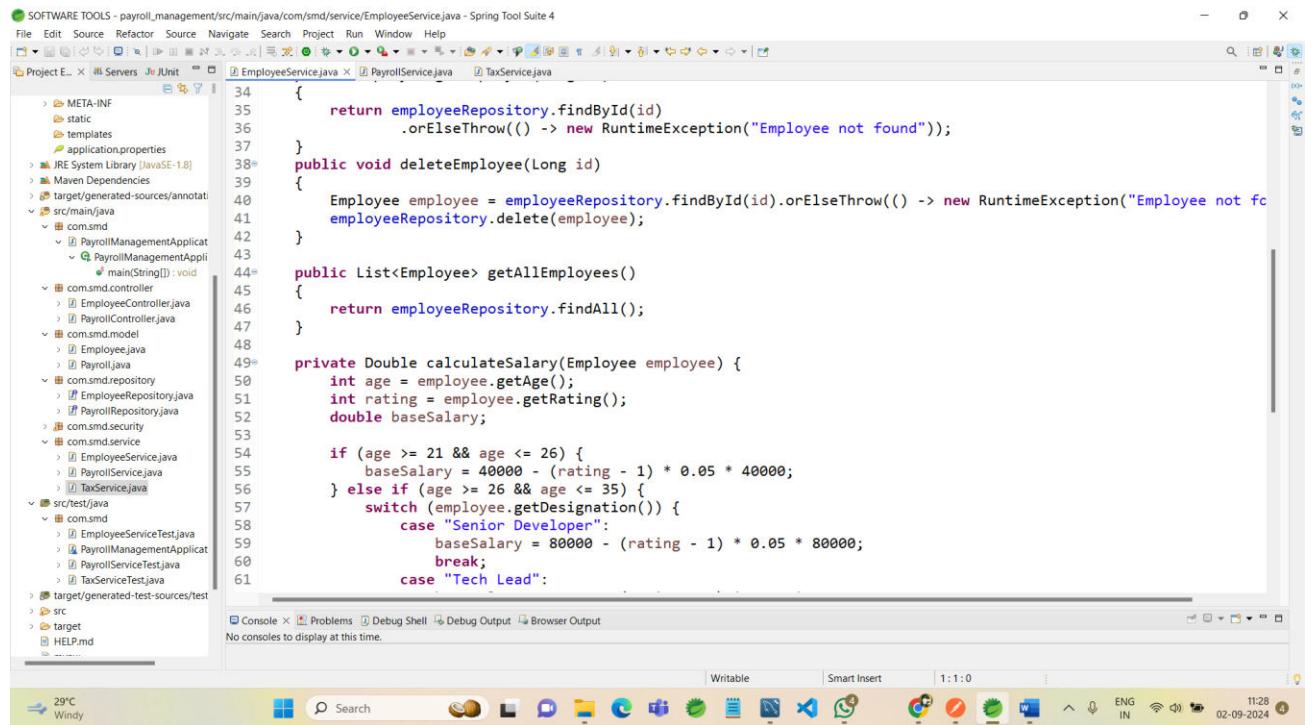
SOFTWARE TOOLS - payroll_management/src/main/java/com/smd/service/EmployeeService.java - Spring Tool Suite 4

```
1 package com.smd.service;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5 @Service
6 public class EmployeeService {
7     @Autowired
8     private EmployeeRepository employeeRepository;
9
10    public Employee addEmployee(Employee employee) {
11        employee.setSalary(calculateSalary(employee));
12        return employeeRepository.save(employee);
13    }
14
15    public Employee updateEmployee(Long id, Employee updatedEmployee) {
16        Employee employee = employeeRepository.findById(id)
17            .orElseThrow(() -> new RuntimeException("Employee not found"));
18        employee.setName(updatedEmployee.getName());
19        employee.setAge(updatedEmployee.getAge());
20        employee.setDesignation(updatedEmployee.getDesignation());
21        employee.setRating(updatedEmployee.getRating());
22        employee.setSalary(calculateSalary(updatedEmployee));
23        return employeeRepository.save(employee);
24    }
25    public Employee getEmployee(Long id) {
26
27
28
29
30
31
32
33
34 }
```

Project Explorer | Servers | JUnit | EmployeeService.java | PayrollService.java | TaxService.java

File Edit Source Refactor Source Navigate Search Project Run Window Help

29°C Windy 11:28 02-09-2024 ENG IN



SOFTWARE TOOLS - payroll_management/src/main/java/com/smd/service/EmployeeService.java - Spring Tool Suite 4

```
34    {
35        return employeeRepository.findById(id)
36            .orElseThrow(() -> new RuntimeException("Employee not found"));
37    }
38    public void deleteEmployee(Long id) {
39        Employee employee = employeeRepository.findById(id).orElseThrow(() -> new RuntimeException("Employee not found"));
40        employeeRepository.delete(employee);
41    }
42
43    public List<Employee> getAllEmployees() {
44        return employeeRepository.findAll();
45    }
46
47    private Double calculateSalary(Employee employee) {
48        int age = employee.getAge();
49        int rating = employee.getRating();
50        double baseSalary;
51
52        if (age >= 21 && age <= 26) {
53            baseSalary = 40000 - (rating - 1) * 0.05 * 40000;
54        } else if (age > 26 && age <= 35) {
55            switch (employee.getDesignation()) {
56                case "Senior Developer":
57                    baseSalary = 80000 - (rating - 1) * 0.05 * 80000;
58                    break;
59                case "Tech Lead":
```

Project Explorer | Servers | JUnit | EmployeeService.java | PayrollService.java | TaxService.java

File Edit Source Refactor Source Navigate Search Project Run Window Help

29°C Windy 11:28 02-09-2024 ENG IN

SOFTWARE TOOLS - payroll_management/src/main/java/com/smd/service/EmployeeService.java - Spring Tool Suite 4

```
File Edit Source Refactor Source Navigate Search Project Run Window Help
Project E... Servers J... JUnit EmployeeService.java PayrollService.java TaxService.java
META-INF static templates application.properties
I RE System Library [JavaSE-1.8]
Maven Dependencies target/generated-sources/annotations
src/main/java com.smd
    PayrollManagementApplication
        PayrollManagementApplicat
            mainString():void
com.smd.controller
    EmployeeController.java
    PayrollController.java
com.smd.model
    Employee.java
    Payroll.java
com.smd.repository
    EmployeeRepository.java
    PayrollRepository.java
com.smd.security
com.smd.service
    EmployeeService.java
    PayrollService.java
    TaxService.java
src/test/java com.smd
    EmployeeServiceTest.java
    PayrollManagementApplicat
        PayrollServiceTest.java
        TaxServiceTest.java
target/generated-test-sources/test
src target HELP.md
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90 }
```

Console Problems Debug Shell Debug Output Browser Output
No consoles to display at this time.

Writable Smart Insert 1:1:0 ENG IN 11:29 02-09-2024

SOFTWARE TOOLS - payroll_management/src/main/java/com/smd/service/PayrollService.java - Spring Tool Suite 4

```
File Edit Source Refactor Source Navigate Search Project Run Window Help
Project E... Servers J... JUnit EmployeeService.java PayrollService.java TaxService.java
META-INF static templates application.properties
I RE System Library [JavaSE-1.8]
Maven Dependencies target/generated-sources/annotations
src/main/java com.smd
    PayrollManagementApplication
        PayrollManagementApplicat
            mainString():void
com.smd.controller
    EmployeeController.java
    PayrollController.java
com.smd.model
    Employee.java
    Payroll.java
com.smd.repository
    EmployeeRepository.java
    PayrollRepository.java
com.smd.security
com.smd.service
    EmployeeService.java
    PayrollService.java
    TaxService.java
src/test/java com.smd
    EmployeeServiceTest.java
    PayrollManagementApplicat
        PayrollServiceTest.java
        TaxServiceTest.java
target/generated-test-sources/test
src target HELP.md
1 package com.smd.service;
2
3 import com.smd.model.Employee;
4
5 @Service
6 public class PayrollService {
7     @Autowired
8     private PayrollRepository payrollRepository;
9
10    @Autowired
11    private EmployeeService employeeService;
12
13    @Autowired
14    private TaxService taxService;
15
16    public Payroll processPayroll(Long employeeId) {
17        Employee employee = employeeService.getEmployee(employeeId);
18        Double basicSalary = employee.getSalary();
19        Double annualSalary = basicSalary * 12;
20        Double taxDeductions = taxService.calculateAnnualTax(annualSalary) / 12;
21        Double netSalary = basicSalary - taxDeductions;
22
23        Payroll payroll = new Payroll();
24        payroll.setEmployee(employee);
25        payroll.setBasicSalary(basicSalary);
26        payroll.setTaxDeductions(taxDeductions);
27        payroll.setNetSalary(netSalary);
28        payroll.setPayDate(new Date());
29
30    }
31
32
33
34
35 }
```

Console Problems Debug Shell Debug Output Browser Output
No consoles to display at this time.

Writable Smart Insert 1:1:0 ENG IN 11:31 02-09-2024

SOFTWARE TOOLS - payroll_management/src/main/java/com/smd/service/PayrollService.java - Spring Tool Suite 4

```
File Edit Source Refactor Source Navigate Search Project Run Window Help
Project E.. X Servers JUnit EmployeeService.java PayrollService.java TaxService.java
META-INF static templates application.properties
JRE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotations
src/main/java
com.smd
  PayrollManagementApplication
    PayrollManagementApplic
      main(String[] args) {
        SpringApplication.run(PayrollManagementApplication.class, args);
      }
  com.smd.controller
    EmployeeController.java
    PayrollController.java
  com.smd.model
    Employee.java
    Payroll.java
  com.smd.repository
    EmployeeRepository.java
    PayrollRepository.java
  com.smd.security
  com.smd.service
    EmployeeService.java
    PayrollService.java
    TaxService.java
src/test/java
com.smd
  EmployeeServiceTest.java
  PayrollManagementApplicat
  PayrollServiceTest.java
  TaxServiceTest.java
target/generated-test-sources/test
src
target
HELP.md
```

34 payroll.setNetSalary(netSalary);
35 payroll.setPayDate(new Date());
36
37 return payrollRepository.save(payroll);
38 }
39
40 public Payroll getPayroll(Long id) {
41 return payrollRepository.findById(id).orElseThrow(() -> new RuntimeException("Payroll not found"));
42 }
43
44 public Payroll updatePayroll(Long id, Payroll payrollDetails) {
45 Payroll payroll = payrollRepository.findById(id).orElseThrow(() -> new RuntimeException("Payroll not found"));
46 payroll.setBasicSalary(payrollDetails.getBasicSalary());
47 payroll.setTaxDeductions(payrollDetails.getTaxDeductions());
48 payroll.setNetSalary(payrollDetails.getNetSalary());
49 payroll.setPayDate(payrollDetails.getPayDate());
50 return payrollRepository.save(payroll);
51 }
52
53 public void deletePayroll(Long id) {
54 Payroll payroll = payrollRepository.findById(id).orElseThrow(() -> new RuntimeException("Payroll not found"));
55 payrollRepository.delete(payroll);
56 }
57
58 public List<Payroll> getAllPayrolls() {
59 return payrollRepository.findAll();
60 }

Console X Problems Debug Shell Debug Output Browser Output

No consoles to display at this time.

Writable Smart Insert 1:1:0 ENG IN 11:31 02-09-2024

SOFTWARE TOOLS - payroll_management/src/main/java/com/smd/service/TaxService.java - Spring Tool Suite 4

```
File Edit Source Refactor Source Navigate Search Project Run Window Help
Project E.. X Servers JUnit EmployeeService.java PayrollService.java TaxService.java
META-INF static templates application.properties
JRE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotations
src/main/java
com.smd
  PayrollManagementApplication
    PayrollManagementApplic
      main(String[] args) {
        SpringApplication.run(PayrollManagementApplication.class, args);
      }
  com.smd.controller
    EmployeeController.java
    PayrollController.java
  com.smd.model
    Employee.java
    Payroll.java
  com.smd.repository
    EmployeeRepository.java
    PayrollRepository.java
  com.smd.security
  com.smd.service
    EmployeeService.java
    PayrollService.java
    TaxService.java
src/test/java
com.smd
  EmployeeServiceTest.java
  PayrollManagementApplicat
  PayrollServiceTest.java
  TaxServiceTest.java
target/generated-test-sources/test
src
target
HELP.md
```

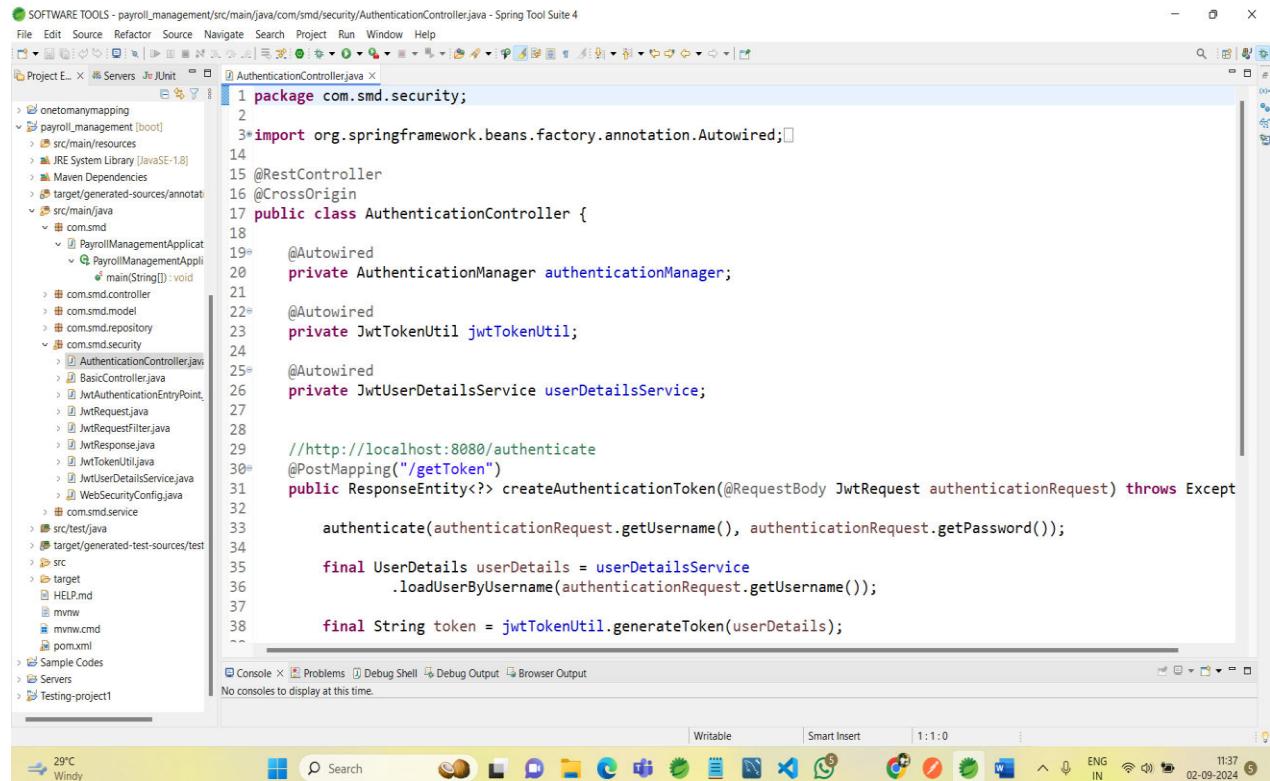
1 package com.smd.service;
2
3 import org.springframework.stereotype.Service;
4
5 @Service
6 public class TaxService {
7 public Double calculateAnnualTax(Double annualSalary) {
8 if (annualSalary <= 500000) {
9 return 0.0;
10 } else if (annualSalary <= 700000) {
11 return (annualSalary - 500000) * 0.10;
12 } else if (annualSalary <= 1000000) {
13 return (200000 * 0.10) + (annualSalary - 700000) * 0.20;
14 } else {
15 return (200000 * 0.10) + (300000 * 0.20) + (annualSalary - 1000000) * 0.30;
16 }
17 }
18 }

Console X Problems Debug Shell Debug Output Browser Output

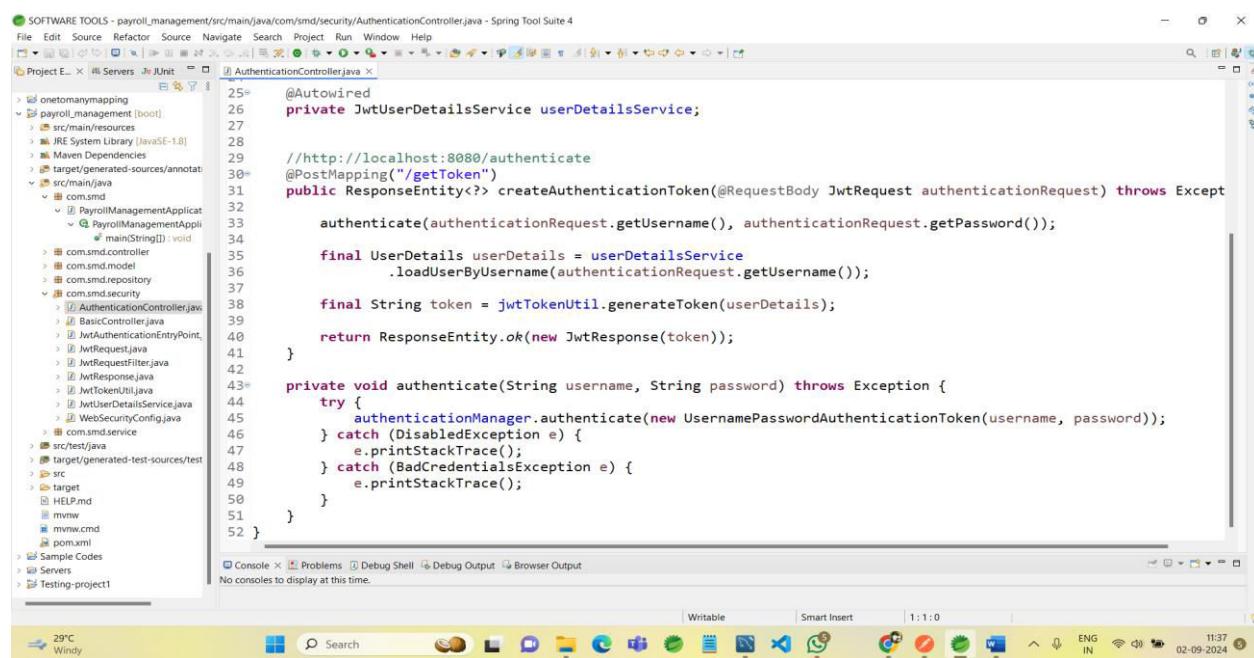
No consoles to display at this time.

Writable Smart Insert 1:1:0 ENG IN 11:34 02-09-2024

JWT SECURITY :Authentication controller, Jwt authentication entry point, Jwt request, Jwt request filter, Jwt response, Jwt util ,Jwt user details service,web security configuration.



```
1 package com.smd.security;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5 @RestController
6 @CrossOrigin
7 public class AuthenticationController {
8
9     @Autowired
10    private AuthenticationManager authenticationManager;
11
12    @Autowired
13    private JwtTokenUtil jwtTokenUtil;
14
15    @Autowired
16    private JwtUserDetailsService userDetailsService;
17
18
19    //http://localhost:8080/authenticate
20    @PostMapping("/getToken")
21    public ResponseEntity<?> createAuthenticationToken(@RequestBody JwtRequest authenticationRequest) throws Exception {
22
23        authenticate(authenticationRequest.getUsername(), authenticationRequest.getPassword());
24
25        final UserDetails userDetails = userDetailsService
26            .loadUserByUsername(authenticationRequest.getUsername());
27
28        final String token = jwtTokenUtil.generateToken(userDetails);
29
30    }
31
32
33
34
35
36
37
38 }
```



```
1 package com.smd.security;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5 @RestController
6 @CrossOrigin
7 public class AuthenticationController {
8
9     @Autowired
10    private JwtUserDetailsService userDetailsService;
11
12
13    //http://localhost:8080/authenticate
14    @PostMapping("/getToken")
15    public ResponseEntity<?> createAuthenticationToken(@RequestBody JwtRequest authenticationRequest) throws Exception {
16
17        authenticate(authenticationRequest.getUsername(), authenticationRequest.getPassword());
18
19        final UserDetails userDetails = userDetailsService
20            .loadUserByUsername(authenticationRequest.getUsername());
21
22        final String token = jwtTokenUtil.generateToken(userDetails);
23
24        return ResponseEntity.ok(new JwtResponse(token));
25    }
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52 }
```

SOFTWARE TOOLS - payroll_management/src/main/java/com/smd/security/BasicController.java - Spring Tool Suite 4

```
1 package com.smd.security;
2
3 import org.springframework.web.bind.annotation.CrossOrigin;
4
5 @RestController
6 public class BasicController {
7
8     @RequestMapping({ "/home" })
9     public String home() {
10         return "Welcome to home";
11     }
12 }
```

Project Explorer Servers JUnit BasicController.java

File Edit Source Refactor Source Navigate Search Project Run Window Help

src/main/resources IRE System Library [JavaSE-1.8] Maven Dependencies target/generated-sources/annotations src/main/java com.smd PayrollManagementApplication PayrollManagementApplication main(String[]): void com.smd.controller com.smd.model com.smd.repository com.smd.security AuthenticationController.java BasicController.java JwtAuthenticationEntryPoint.java JwtRequest.java JwtRequestFilter.java JwtResponse.java JwtTokenUtil.java JwtUserDetailsService.java WebSecurityConfig.java com.smd.service srcTest Java target/generated-test-sources/test src target HELP.md mvnw mvnw.cmd pom.xml Sample Codes Servers Testing-project1

Console Problems Debug Shell Debug Output Browser Output No consoles to display at this time.

Writable Smart Insert 1:1:0 11:37 02-09-2024 ENG IN

SOFTWARE TOOLS - payroll_management/src/main/java/com/smd/security/JwtRequest.java - Spring Tool Suite 4

```
1 package com.smd.security;
2
3 import java.io.Serializable;
4
5 public class JwtRequest implements Serializable {
6
7     private static final long serialVersionUID = 6239921560724451905L;
8
9     private String username;
10    private String password;
11
12    public JwtRequest() {
13    }
14
15    public JwtRequest(String username, String password) {
16        this.setUsername(username);
17        this.setPassword(password);
18    }
19
20    public String getUsername() {
21        return this.username;
22    }
23
24    public void setUsername(String username) {
25        this.username = username;
26    }
27 }
```

Project Explorer Servers JUnit BasicController.java JwtAuthenticationEntryPoint.java JwtRequest.java

File Edit Source Refactor Source Navigate Search Project Run Window Help

src/main/resources IRE System Library [JavaSE-1.8] Maven Dependencies target/generated-sources/annotations src/main/java com.smd PayrollManagementApplication PayrollManagementApplication main(String[]): void com.smd.controller com.smd.model com.smd.repository com.smd.security AuthenticationController.java BasicController.java JwtAuthenticationEntryPoint.java JwtRequest.java JwtRequestFilter.java JwtResponse.java JwtTokenUtil.java JwtUserDetailsService.java WebSecurityConfig.java com.smd.service srcTest Java target/generated-test-sources/test src target HELP.md mvnw mvnw.cmd pom.xml Sample Codes Servers Testing-project1

Console Problems Debug Shell Debug Output Browser Output No consoles to display at this time.

Writable Smart Insert 1:1:0 11:37 02-09-2024 ENG IN

SOFTWARE TOOLS - payroll_management/src/main/java/com/smd/security/JwtRequestFilter.java - Spring Tool Suite 4

```
File Edit Source Refactor Source Navigate Search Project Run Window Help
Project E... Servers JUnit BasicController.java JwtAuthenticationEntryPoint.java JwtRequest.java JwtRequestFilter.java
1 package com.smd.security;
2
3 import java.io.IOException;
4
5 @Component
6 public class JwtRequestFilter extends OncePerRequestFilter {
7
8     @Autowired
9     private JwtUserDetailsService jwtUserDetailsService;
10
11    @Autowired
12    private JwtTokenUtil jwtTokenUtil;
13
14    @Override
15    protected void doFilterInternal(HttpServletRequest request, HttpServletResponse response, FilterChain chain)
16        throws ServletException, IOException {
17
18        final String requestTokenHeader = request.getHeader("Authorization");
19
20        String username = null;
21        String jwtToken = null;
22        // JWT Token is passed in the form "Bearer <token>". Thus, let's remove Bearer
23        // word and get the Token
24        if (requestTokenHeader != null && requestTokenHeader.startsWith("Bearer ")) {
25            jwtToken = requestTokenHeader.substring(7);
26            try {
27                username = jwtTokenUtil.getUsernameFromToken(jwtToken);
28            } catch (IllegalArgumentException e) {
29                ...
30            }
31        }
32
33        //Validate the token
34        if (username != null && SecurityContextHolder.getContext().getAuthentication() == null) {
35
36            UserDetails userDetails = jwtUserDetailsService.loadUserByUsername(username);
37
38            // on valid token configure Spring Security to authentication
39            if (jwtTokenUtil.validateToken(jwtToken, userDetails)) {
40
41                UsernamePasswordAuthenticationToken usernamePasswordAuthenticationToken = new UsernamePasswordAuthen
42                    userDetails, null, userDetails.getAuthorities());
43                usernamePasswordAuthenticationToken
44                    .setDetails(new WebAuthenticationDetailsSource().buildDetails(request));
45
46            }
47
48            SecurityContextHolder.getContext().setAuthentication(usernamePasswordAuthenticationToken);
49        }
50
51        chain.doFilter(request, response);
52    }
53
54
55 }
```

Console Problems Debug Shell Debug Output Browser Output

No consoles to display at this time.

Writable Smart Insert 1:1:0

29°C Windy Search ENG IN 11:37 02-09-2024

SOFTWARE TOOLS - payroll_management/src/main/java/com/smd/security/JwtRequestFilter.java - Spring Tool Suite 4

```
File Edit Source Refactor Source Navigate Search Project Run Window Help
Project E... Servers JUnit BasicController.java JwtAuthenticationEntryPoint.java JwtRequest.java JwtRequestFilter.java
45 } catch (IllegalArgumentException e) {
46     System.out.println("Failed to get token");
47 } catch (ExpiredJwtException e) {
48     System.out.println("JWT Token is expired");
49 }
50 } else {
51     System.out.println("JWT Token does not begin with Bearer word");
52 }
53
54 //Validate the token
55 if (username != null && SecurityContextHolder.getContext().getAuthentication() == null) {
56
57     UserDetails userDetails = jwtUserDetailsService.loadUserByUsername(username);
58
59     // on valid token configure Spring Security to authentication
60     if (jwtTokenUtil.validateToken(jwtToken, userDetails)) {
61
62         UsernamePasswordAuthenticationToken usernamePasswordAuthenticationToken = new UsernamePasswordAuthen
63             userDetails, null, userDetails.getAuthorities());
64         usernamePasswordAuthenticationToken
65             .setDetails(new WebAuthenticationDetailsSource().buildDetails(request));
66
67         SecurityContextHolder.getContext().setAuthentication(usernamePasswordAuthenticationToken);
68     }
69 }
70 chain.doFilter(request, response);
71 }
72 }
```

Console Problems Debug Shell Debug Output Browser Output

No consoles to display at this time.

Writable Smart Insert 1:1:0

29°C Windy Search ENG IN 11:37 02-09-2024

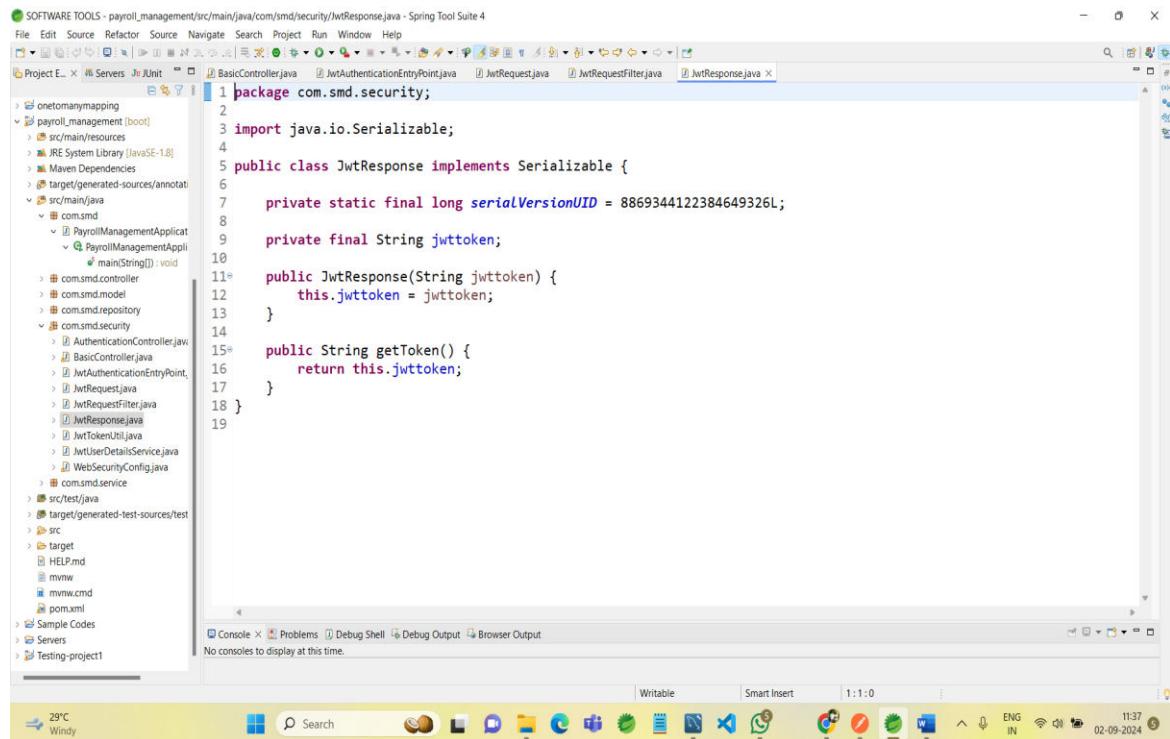
SOFTWARE TOOLS - payroll_management/src/main/java/com/smd/security/JwtResponse.java - Spring Tool Suite 4

```
File Edit Source Refactor Source Navigate Project Run Window Help
Project E... × Servers JUnit ... JwtResponse.java
1 package com.smd.security;
2
3 import java.io.Serializable;
4
5 public class JwtResponse implements Serializable {
6
7     private static final long serialVersionUID = 8869344122384649326L;
8
9     private final String jwttoken;
10
11    public JwtResponse(String jwttoken) {
12        this.jwttoken = jwttoken;
13    }
14
15    public String getToken() {
16        return this.jwttoken;
17    }
18 }
19
```

Console × Problems Debug Shell Debug Output Browser Output

No consoles to display at this time.

Writable Smart Insert 1:1:0 ENG IN 11:37 02-09-2024



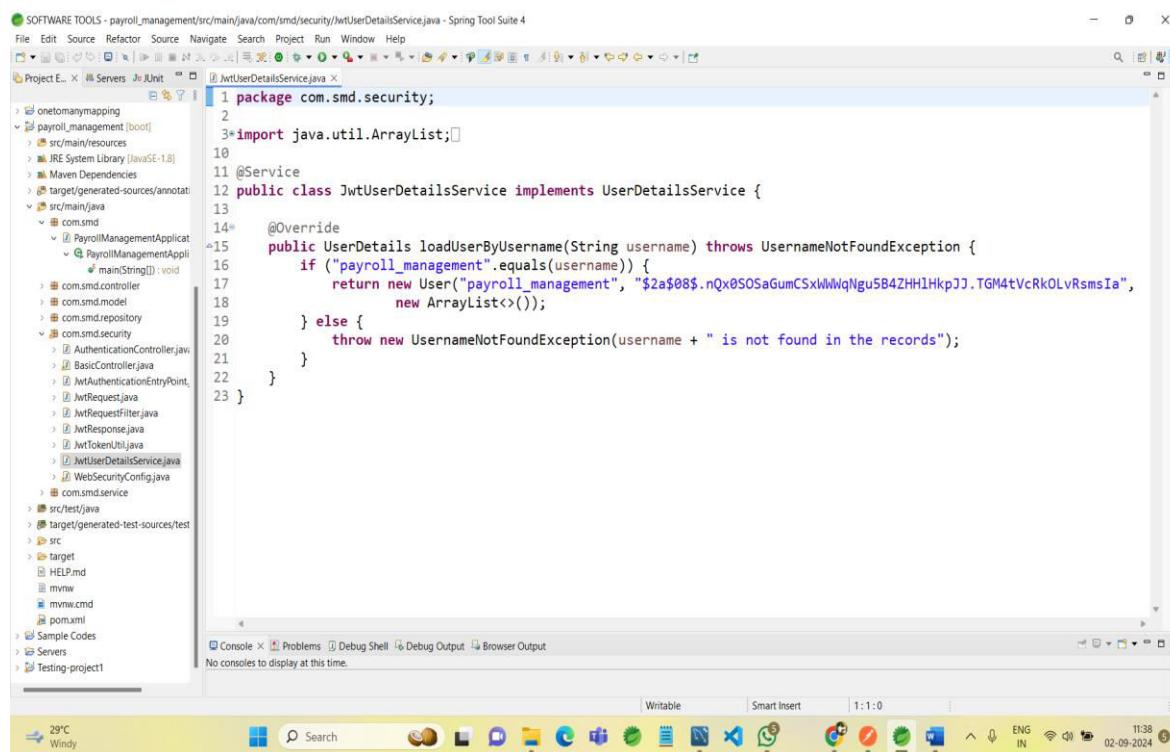
SOFTWARE TOOLS - payroll_management/src/main/java/com/smd/security/JwtUserDetailsService.java - Spring Tool Suite 4

```
File Edit Source Refactor Source Navigate Project Run Window Help
Project E... × Servers JUnit ... JwtUserDetailsService.java
1 package com.smd.security;
2
3 import java.util.ArrayList;
4
5 @Service
6 public class JwtUserDetailsService implements UserDetailsService {
7
8     @Override
9     public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
10         if ("payroll_management".equals(username)) {
11             return new User("payroll_management", "$2a$08$.nQx0SOSaGumCSxWqNgusB4ZHH1HkpJJ.TGM4tVcRkOLvRsmsIa",
12                             new ArrayList<>());
13         } else {
14             throw new UsernameNotFoundException(username + " is not found in the records");
15         }
16     }
17 }
```

Console × Problems Debug Shell Debug Output Browser Output

No consoles to display at this time.

Writable Smart Insert 1:1:0 ENG IN 11:38 02-09-2024



SOFTWARE TOOLS - payroll_management/src/main/java/com/smd/security/WebSecurityConfig.java - Spring Tool Suite 4

File Edit Source Refactor Source Navigate Search Project Run Window Help

Project Explorer Servers JUnit JUnit UserDetailsService.java WebSecurityConfig.java

```
41     "/readiness_check",
42     "/actuator/**",
43     "/liveness_check",
44     "/_ah/start",
45     "/eureka/**",
46     "/getToken",
47     "/swagger-ui/**", // Swagger UI endpoint
48     "/v3/api-docs/**", // OpenAPI 3 documentation endpoint
49     "/v2/api-docs", // Swagger 2 documentation endpoint
50     "/swagger-resources/**", // Swagger resources endpoint
51     "/configuration/ui",
52     "/configuration/security",
53     "/webjars/**" // WebJars for Swagger UI
54 ).permitAll()
55 .anyRequest().authenticated()
56 );
57
58 http.addFilterBefore(jwtRequestFilter, UsernamePasswordAuthenticationFilter.class);
59
60 return http.build();
61 }
62
63 @Bean
64 public AuthenticationManager authenticationManager(AuthenticationConfiguration authenticationConfiguration) thr
65     return authenticationConfiguration.getAuthenticationManager();
66 }
67 }
68 }
```

Console Problems Debug Shell Debug Output Browser Output

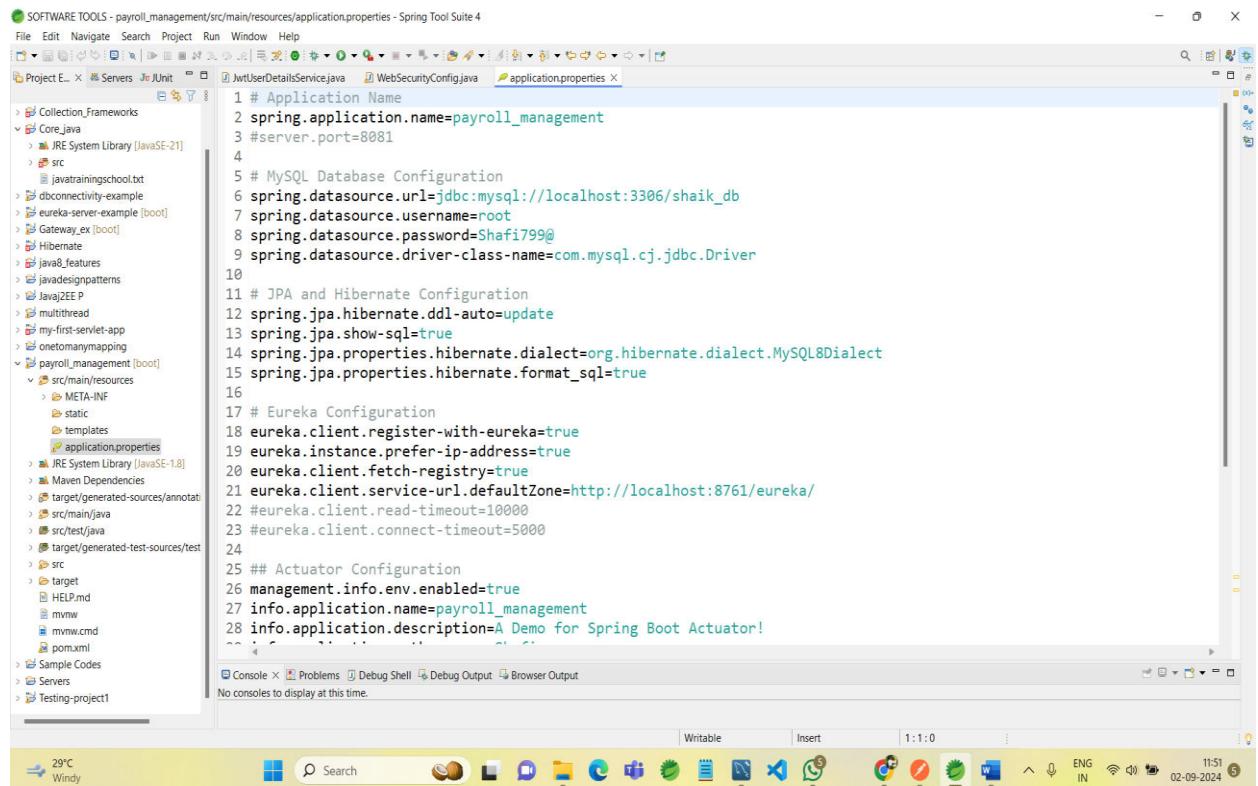
No consoles to display at this time.

Writable Smart Insert 1:1:0

29°C Windy Search

ENG IN 02-09-2024 11:38

Application.properties :



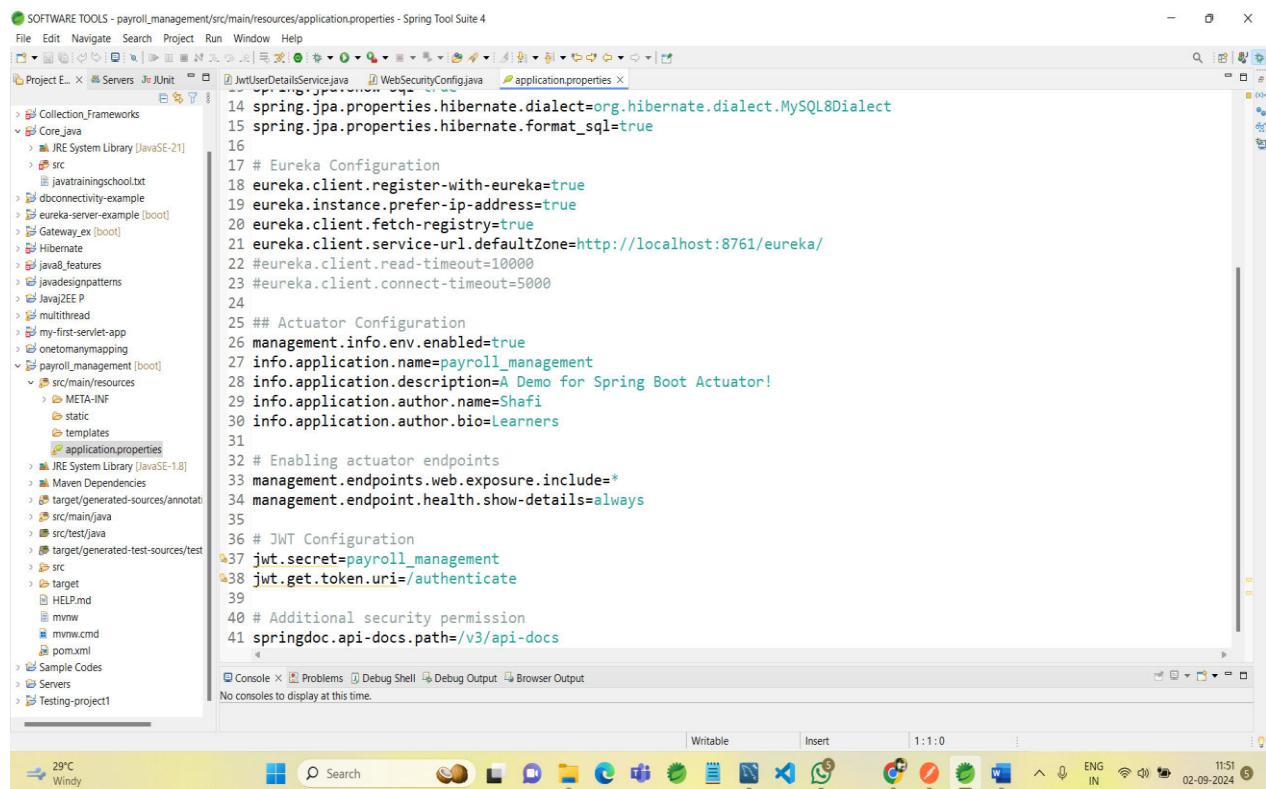
```
# Application Name
spring.application.name=payroll_management
server.port=8081

# MySQL Database Configuration
spring.datasource.url=jdbc:mysql://localhost:3306/shaik_db
spring.datasource.username=root
spring.datasource.password=Shafi799@
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

# JPA and Hibernate Configuration
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
spring.jpa.properties.hibernate.format_sql=true

# Eureka Configuration
eureka.client.register-with-eureka=true
eureka.instance.prefer-ip-address=true
eureka.client.fetch-registry=true
eureka.client.service-url.defaultZone=http://localhost:8761/eureka/
#eureka.client.read-timeout=10000
#eureka.client.connect-timeout=5000

## Actuator Configuration
management.info.enabled=true
info.application.name=payroll_management
info.application.description=A Demo for Spring Boot Actuator!
```



```
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
spring.jpa.properties.hibernate.format_sql=true

# Eureka Configuration
eureka.client.register-with-eureka=true
eureka.instance.prefer-ip-address=true
eureka.client.fetch-registry=true
eureka.client.service-url.defaultZone=http://localhost:8761/eureka/
#eureka.client.read-timeouts=1000
#eureka.client.connect-timeout=5000

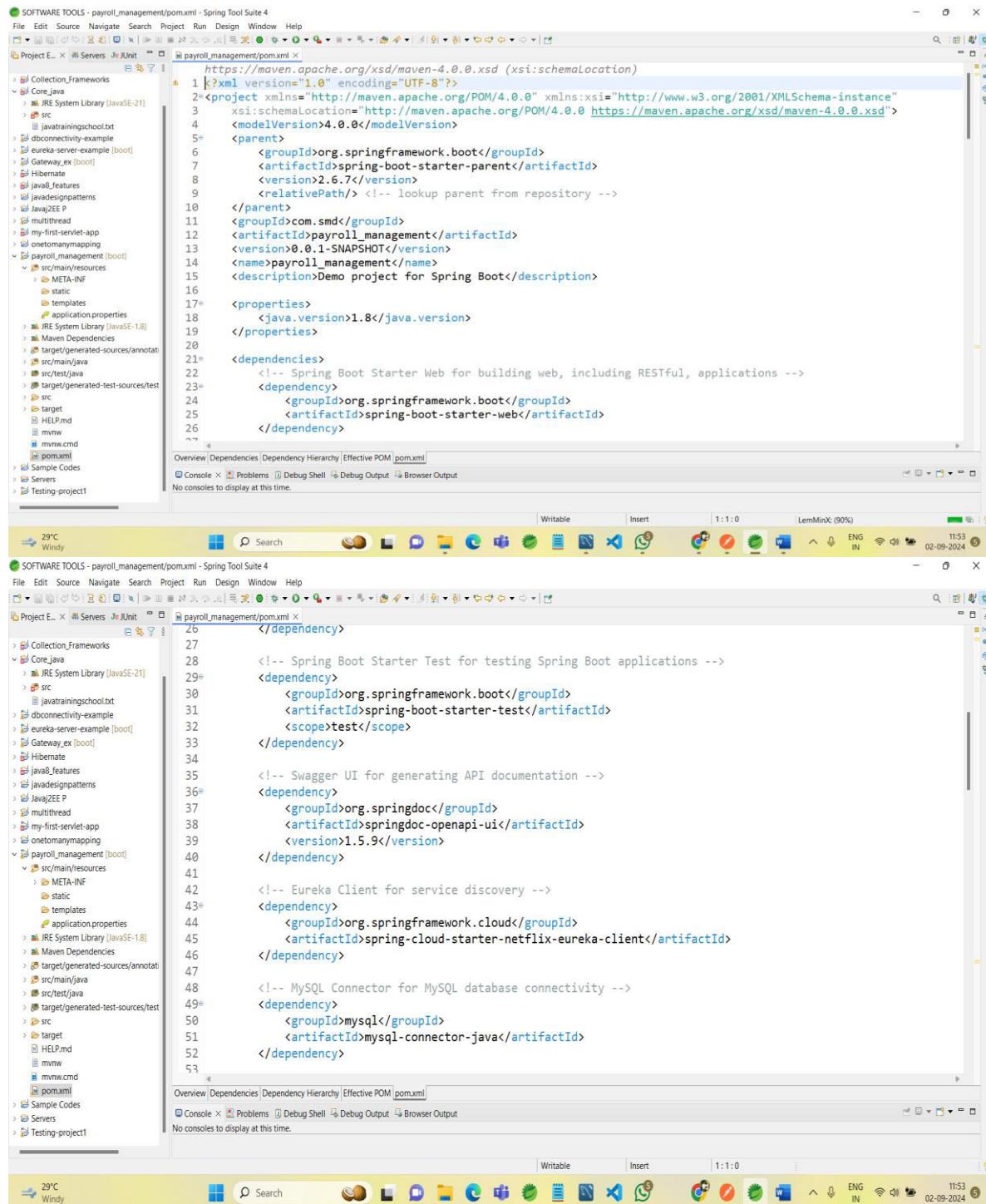
## Actuator Configuration
management.info.enabled=true
info.application.name=payroll_management
info.application.description=A Demo for Spring Boot Actuator!
info.application.author.name=Shafi
info.application.author.bio=Learners

# Enabling actuator endpoints
management.endpoints.web.exposure.include=*
management.endpoint.health.show-details=always

# JWT Configuration
jwt.secret=payroll_management
jwt.get.token.uri=/authenticate

# Additional security permission
springdoc.api-docs.path=/v3/api-docs
```

POM.XML FOR Payroll management



The screenshot shows the Spring Tool Suite (STS) interface with the 'payroll_management/pom.xml' file open in the central editor. The code is a Maven POM XML file for a Spring Boot application. The XML content includes the declaration of the parent 'spring-boot-starter-parent' with version 2.6.7, dependencies for 'spring-boot-starter-web' and 'spring-boot-starter-test', and various configuration sections like properties and scopes. The interface includes a toolbar at the top, a navigation bar, and a status bar at the bottom indicating the date and time.

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.6.7</version>
    <relativePath/> 
  </parent>
  <groupId>com.smd</groupId>
  <artifactId>payroll_management</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>payroll_management</name>
  <description>Demo project for Spring Boot</description>
  <properties>
    <java.version>1.8</java.version>
  </properties>
  <dependencies>
    <!-- Spring Boot Starter Web for building web, including RESTful, applications -->
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <!-- Spring Boot Starter Test for testing Spring Boot applications -->
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>
    <!-- Swagger UI for generating API documentation -->
    <dependency>
      <groupId>org.springdoc</groupId>
      <artifactId>springdoc-openapi-ui</artifactId>
      <version>1.5.9</version>
    </dependency>
    <!-- Eureka Client for service discovery -->
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
    </dependency>
    <!-- MySQL Connector for MySQL database connectivity -->
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
    </dependency>
  </dependencies>

```

SOFTWARE TOOLS - payroll_management/pom.xml - Spring Tool Suite 4

```

File Edit Source Navigate Search Project Run Design Window Help
Project E... < Server Jun 10 > payroll_management/pom.xml
Collection_Frameworks
Core Java
I RE System Library [JavaSE-21]
src
javatrainingschool.txt
dconnectivity-example
eureka-server-example [boot]
Gateway_ex [boot]
Hibernate
java8_features
javadesignpatterns
Java2EE_P
multithread
my-first-servlet-app
onetomanymapping
payroll_management [boot]
src/main/resources
META-INF
static
templates
application.properties
I RE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotations
src/main/java
src/test/java
target/generated-test-sources/test
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
Sample Codes
Servers
Testing-project1
54      <!-- Tomcat Embed Jasper for JSP support -->
55      <dependency>
56          <groupId>org.apache.tomcat.embed</groupId>
57          <artifactId>tomcat-embed-jasper</artifactId>
58      </dependency>
59
60      <!-- Spring Boot Starter Data JPA for database persistence -->
61      <dependency>
62          <groupId>org.springframework.boot</groupId>
63          <artifactId>spring-boot-starter-data-jpa</artifactId>
64      </dependency>
65
66      <!-- Spring Boot Starter Validation for bean validation -->
67      <dependency>
68          <groupId>org.springframework.boot</groupId>
69          <artifactId>spring-boot-starter-validation</artifactId>
70      </dependency>
71
72      <!-- Lombok for reducing boilerplate code -->
73      <dependency>
74          <groupId>org.projectlombok</groupId>
75          <artifactId>lombok</artifactId>
76          <scope>provided</scope>
77      </dependency>
78
79      <!-- Jackson Databind for JSON processing -->
80      <dependency>
81          <groupId>com.fasterxml.jackson.core</groupId>
82          <artifactId>jackson-databind</artifactId>
83      </dependency>
84
85      <!-- JSTL for JSP Standard Tag Library -->
86      <dependency>
87          <groupId>javax.servlet</groupId>
88          <artifactId>jstl</artifactId>
89      </dependency>
90
91      <!-- Spring Boot Starter Actuator for monitoring and management -->
92      <dependency>
93          <groupId>org.springframework.boot</groupId>
94          <artifactId>spring-boot-starter-actuator</artifactId>
95      </dependency>
96
97      <!-- Spring Boot Starter Security for securing the application -->
98      <dependency>
99          <groupId>org.springframework.boot</groupId>
100         <artifactId>spring-boot-starter-security</artifactId>
101     </dependency>
102
103     <!-- JWT for handling JSON Web Tokens -->
104     <dependency>
105         <groupId>io.jsonwebtoken</groupId>
106         <artifactId>jjwt</artifactId>
107     </dependency>

```

Overview | Dependencies | Dependency Hierarchy | Effective POM | pom.xml

Console × Problems × Debug Shell × Debug Output × Browser Output

No consoles to display at this time.

Writable Insert 1:1:0 ENG IN 11:54 02-09-2024

29°C Windy

SOFTWARE TOOLS - payroll_management/pom.xml - Spring Tool Suite 4

```

File Edit Source Navigate Search Project Run Design Window Help
Project E... < Server Jun 10 > payroll_management/pom.xml
Collection_Frameworks
Core Java
I RE System Library [JavaSE-21]
src
javatrainingschool.txt
dconnectivity-example
eureka-server-example [boot]
Gateway_ex [boot]
Hibernate
java8_features
javadesignpatterns
Java2EE_P
multithread
my-first-servlet-app
onetomanymapping
payroll_management [boot]
src/main/resources
META-INF
static
templates
application.properties
I RE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotations
src/main/java
src/test/java
target/generated-test-sources/test
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
Sample Codes
Servers
Testing-project1
80      <dependency>
81          <groupId>com.fasterxml.jackson.core</groupId>
82          <artifactId>jackson-databind</artifactId>
83      </dependency>
84
85      <!-- JSTL for JSP Standard Tag Library -->
86      <dependency>
87          <groupId>javax.servlet</groupId>
88          <artifactId>jstl</artifactId>
89      </dependency>
90
91      <!-- Spring Boot Starter Actuator for monitoring and management -->
92      <dependency>
93          <groupId>org.springframework.boot</groupId>
94          <artifactId>spring-boot-starter-actuator</artifactId>
95      </dependency>
96
97      <!-- Spring Boot Starter Security for securing the application -->
98      <dependency>
99          <groupId>org.springframework.boot</groupId>
100         <artifactId>spring-boot-starter-security</artifactId>
101     </dependency>
102
103     <!-- JWT for handling JSON Web Tokens -->
104     <dependency>
105         <groupId>io.jsonwebtoken</groupId>
106         <artifactId>jjwt</artifactId>
107     </dependency>

```

Overview | Dependencies | Dependency Hierarchy | Effective POM | pom.xml

Console × Problems × Debug Shell × Debug Output × Browser Output

No consoles to display at this time.

Writable Insert 1:1:0 ENG IN 11:55 02-09-2024

29°C Windy

The screenshot shows the Spring Tool Suite (STS) interface with the following details:

- Title Bar:** SOFTWARE TOOLS - payroll_management/pom.xml - Spring Tool Suite 4
- Menu Bar:** File, Edit, Source, Navigate, Search, Project, Run, Design, Window, Help
- Toolbar:** Includes icons for New, Open, Save, Cut, Copy, Paste, Find, Replace, etc.
- Project Explorer:** Shows the project structure:
 - Collection_Frameworks
 - Core Java
 - JRE System Library [JavaSE-11]
 - javatrainingschool.txt
 - dbconnectivity-example
 - eureka-server-example [boot]
 - Hibernate
 - java_features
 - javadesignpatterns
 - JavaZEE P
 - multithread
 - my-first-servlet-app
 - onetomanymapping
 - payroll_management [boot]
 - src/main/resources
 - META-INF
 - static
 - template
 - application.properties
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - target/generated-sources/annotat
 - src/main/java
 - src/test/java
 - target/generated-test-sources/test
 - src
 - target
 - HELP.mdd
 - mvnw
 - mvnw.cmd
 - pom.xml
- Editor:** The main window displays the content of the pom.xml file. The code is as follows:

```
<version>0.0.1-SNAPSHOT</version>
</dependencies>
<!-- Spring Cloud BOM -->
<dependencyManagement>
<dependencies>
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-dependencies</artifactId>
<version>2023.0.2</version> <!-- Use the latest version -->
<type>pom</type>
<scope>import</scope>
</dependency>
</dependencies>
</dependencyManagement>

<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
</project>

```

- Bottom Navigation:** Overview, Dependencies, Dependency Hierarchy, Effective POM, pom.xml
- Console:** No consoles to display at this time.
- System Tray:** Shows the date (02-09-2024), time (11:55), battery level (29%), and weather (Windy).

Gateway : Main

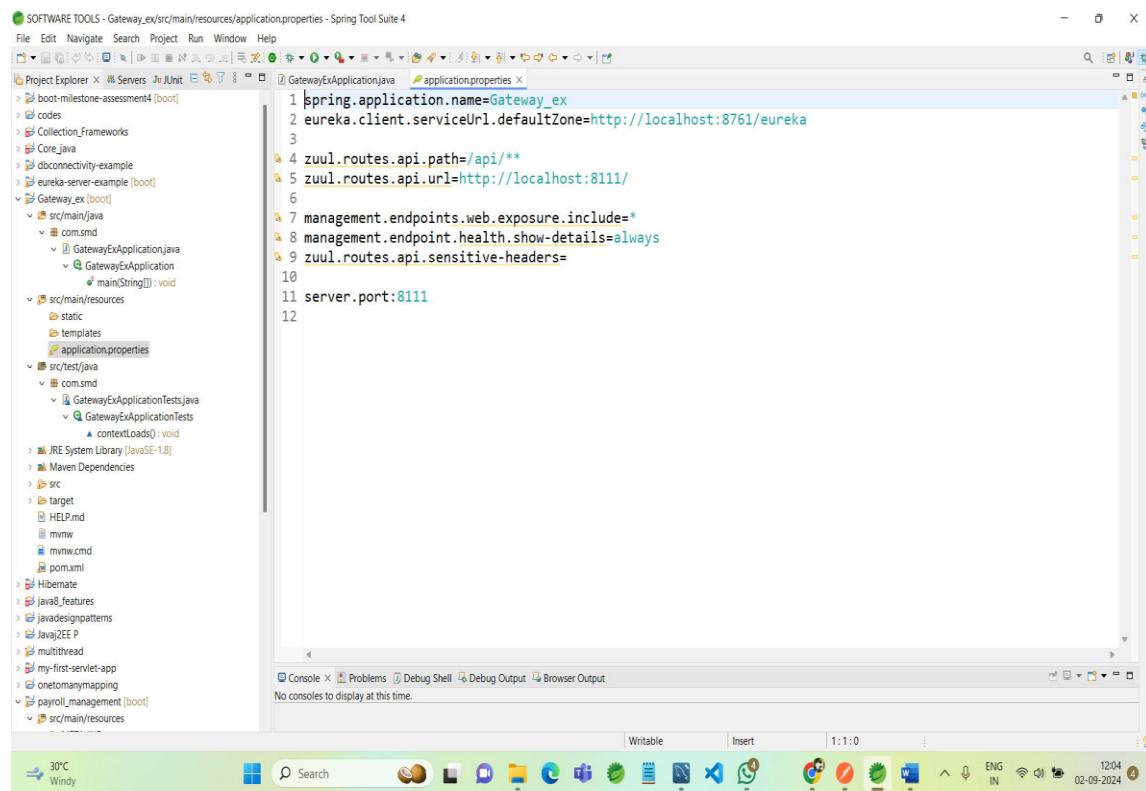
The screenshot shows the Eclipse IDE interface. The left pane, titled "Project Explorer", lists the project structure:

- boot-milestone-assessment4 [boot] (selected)
- codes
- Collection_Frameworks
- Core.java
- dbconnectivity-example
- eureka-server-example [boot]
- Gateway_ex [boot]
 - src/main/java
 - com.smd
 - GatewayExApplication.java
 - GatewayExApplication
 - main(String[]) : void
 - src/main/resources
 - static
 - templates
 - application.properties
 - src/test/java
 - com.smd
 - GatewayExApplicationTests.java
 - GatewayExApplicationTests
 - contextLoads() : void
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml

The right pane, titled "GatewayExApplication.java", displays the Java code:

```
1 package com.smd;
2
3 import org.springframework.boot.SpringApplication;
4
5 @SpringBootApplication
6 @EnableEurekaClient
7 public class GatewayExApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(GatewayExApplication.class, args);
11     }
12
13 }
14
15 }
16
```

Properties :



The screenshot shows the Spring Tool Suite 4 interface with the 'application.properties' file open in the central editor. The file contains the following configuration:

```
spring.application.name=Gateway_ex
eureka.client.serviceUrl.defaultZone=http://localhost:8761/eureka
zuul.routes.api.path=/api/**
zuul.routes.api.url=http://localhost:8111/
management.endpoints.web.exposure.include=*
management.endpoint.health.show-details=always
zuul.routes.api.sensitive-headers=
server.port:8111
```

The Project Explorer on the left shows the project structure, including 'GatewayExApplication.java' and 'application.properties'. The bottom status bar displays system information like temperature, battery level, and date/time.

POM.XML(GATEWAY):

The screenshot shows the Spring Tool Suite interface with the 'Gateway_ex/pom.xml' file open in the central editor. The code editor displays the following XML content:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.6.7</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.smd</groupId>
  <artifactId>Gateway_ex</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>Gateway_ex</name>
  <description>Demo project for Spring Boot</description>
  <url/>
  <licenses>
    <license/>
  </licenses>
  <developers>
    <developer/>
  </developers>
  <scm>
    <connection>
      <developerConnection></developerConnection>
    </connection>
    <tag/>
  </scm>
</project>
```

The Project Explorer on the left shows the project structure, including 'src/main/java' with 'GatewayExApplication.java' and 'application.properties', and 'src/test/java' with 'GatewayExApplicationTests.java'. The bottom status bar shows the date and time as 02-09-2024.

The screenshot shows the Spring Tool Suite interface with the 'Gateway_ex/pom.xml' file open in the central editor. The code editor displays the following XML content, showing additional dependencies for Spring Cloud Gateway and Eureka Client:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.6.7</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.smd</groupId>
  <artifactId>Gateway_ex</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>Gateway_ex</name>
  <description>Demo project for Spring Boot</description>
  <url/>
  <licenses>
    <license/>
  </licenses>
  <developers>
    <developer/>
  </developers>
  <scm>
    <connection>
      <developerConnection></developerConnection>
    </connection>
    <tag/>
  </scm>
  <properties>
    <java.version>1.8</java.version>
    <spring-cloud.version>2021.0.0</spring-cloud.version>
  </properties>
  <dependencies>
    <!-- Spring Cloud Gateway -->
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-gateway</artifactId>
    </dependency>
    <!-- Eureka Client -->
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
    </dependency>
    <!-- Spring WebFlux (Reactive) -->
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-webflux</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

The Project Explorer on the left shows the project structure, including 'src/main/java' with 'GatewayExApplication.java' and 'application.properties', and 'src/test/java' with 'GatewayExApplicationTests.java'. The bottom status bar shows the date and time as 02-09-2024.

SOFTWARE TOOLS - Gateway_ex/pom.xml - Spring Tool Suite 4

```

File Edit Source Navigate Project Run Design Window Help
Project Explorer Servers JUnit
Gateway_ex [boot]
src/main/java
com.smd
GatewayExApplication.java
application.properties
Gateway_ex/pom.xml
51 <groupId>org.springframework.boot</groupId>
52 <artifactId>spring-boot-starter-test</artifactId>
53 <scope>test</scope>
54 </dependency>
55 </dependencies>
56 <dependencyManagement>
57 <dependencies>
58 <dependency>
59 <groupId>org.springframework.cloud</groupId>
60 <artifactId>spring-cloud-dependencies</artifactId>
61 <version>${spring-cloud.version}</version>
62 <type>pom</type>
63 <scope>import</scope>
64 </dependency>
65 </dependencies>
66 </dependencyManagement>
67
68 <build>
69 <plugins>
70 <plugin>
71 <groupId>org.springframework.boot</groupId>
72 <artifactId>spring-boot-maven-plugin</artifactId>
73 </plugin>
74 </plugins>
75 </build>
76
77 </project>

```

Overview | Dependencies | Dependency Hierarchy | Effective POM | pom.xml

Console Problems Debug Shell Debug Output Browser Output

No consoles to display at this time.

Writable Insert 1:2 [1] ENG IN 12:07 02-09-2024

30°C Windy

Eureka : Main,properties,pom.xml

SOFTWARE TOOLS - eureka-server-example/src/main/java/com/smd/EurekaServerExampleApplication.java - Spring Tool Suite 4

```

File Edit Source Refactor Refactor Source Navigate Project Run Window Help
Project Explorer Servers JUnit
EurekaServerExampleApplication.java
1 package com.smd;
2
3 *import org.springframework.boot.SpringApplication;
4
5 @SpringBootApplication
6 @EnableEurekaServer
7 public class EurekaServerExampleApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(EurekaServerExampleApplication.class, args);
11     }
12 }
13
14
15 }
16

```

Console Problems Debug Shell Debug Output Browser Output

No consoles to display at this time.

30°C Windy

SOFTWARE TOOLS - eureka-server-example/src/main/resources/application.properties - Spring Tool Suite 4

```
spring.application.name=eureka-server-example
server.port=8761
eureka.client.register-with-eureka=true
eureka.client.fetch-registry=true
eureka.server.enable-self-preservation=true
```

Project Explorer X Servers JUnit X

src/main/java
com.smd
EurekaServerExampleApplication.java
EurekaServerExampleApplicationTests.java
application.properties
com.cmd
EurekaServerExampleApplicationTests.java
contextLoads():void

JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml

Gateway_ex [boot]
Hibernate
java8_features
javadesignpatterns
Java2EE_P
multithread
my-first-servlet-app
onetomanymapping
payroll_management [boot]

src/main/resources

Console X Problems Debug Shell Debug Output Browser Output

No consoles to display at this time.

Writable Insert 1:1:0 ENG IN 12:12 02-09-2024

SOFTWARE TOOLS - eureka-server-example/src/test/java/com/smd/EurekaServerExampleApplicationTests.java - Spring Tool Suite 4

```
package com.smd;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;
import org.springframework.boot.test.context.SpringBootTest;
@SpringBootTest
class EurekaServerExampleApplicationTests {
    @Test
    void contextLoads() {
    }
}
```

Project Explorer X Servers JUnit X

src/main/java
com.smd
EurekaServerExampleApplication.java
EurekaServerExampleApplicationTests.java
application.properties
com.cmd
EurekaServerExampleApplicationTests.java
contextLoads():void

JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml

Gateway_ex [boot]
Hibernate
java8_features
javadesignpatterns
Java2EE_P
multithread
my-first-servlet-app
onetomanymapping
payroll_management [boot]

src/main/resources

Console X Problems Debug Shell Debug Output Browser Output

No consoles to display at this time.

Writable Smart Insert 1:1:0 ENG IN 12:13 02-09-2024

SOFTWARE TOOLS - eureka-server-example/pom.xml - Spring Tool Suite 4

```
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>2.6.7</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>com.smd</groupId>
<artifactId>eureka-server-example</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>eureka-server-example</name>
<description>Demo project for Spring Boot</description>
<properties>
    <java.version>1.8</java.version>
    <spring-cloud.version>2023.0.2</spring-cloud.version>
</properties>
<dependencies>
    <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>
<dependencyManagement>
```

Overview | Dependencies | Dependency Hierarchy | Effective POM | pom.xml

Console | Problems | Debug Shell | Debug Output | Browser Output

No consoles to display at this time.

Writable Insert 1:1:0 LemMinX: (15%) ENG IN 12:14 02-09-2024

SOFTWARE TOOLS - eureka-server-example/pom.xml - Spring Tool Suite 4

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
</dependency>
</dependencies>
<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-dependencies</artifactId>
            <version>${spring-cloud.version}</version>
            <type>pom</type>
            <scope>import</scope>
        </dependency>
    </dependencies>
</dependencyManagement>
<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>
</project>
```

Overview | Dependencies | Dependency Hierarchy | Effective POM | pom.xml

Console | Problems | Debug Shell | Debug Output | Browser Output

No consoles to display at this time.

Writable Insert 1:1:0 LemMinX: (90%) ENG IN 12:14 02-09-2024

OUTPUT :

Swagger:

The screenshot shows the Swagger UI interface running in a Microsoft Edge browser. The URL is `localhost:8080/swagger-ui/index.html?configUrl=/v3/api-docs/swagger-config#`. The interface is organized into sections for different controllers:

- payroll-controller**: Contains five operations:
 - GET /payroll/{id}**
 - PUT /payroll/{id}**
 - DELETE /payroll/{id}**
 - POST /payroll/{employeeId}**
 - GET /payroll/report**
- employee-controller**: Contains one operation:
 - GET /basic/home**
- authentication-controller**: Contains one operation:
 - PUT /basic/home**
- basic-controller**: Contains four operations:
 - GET /basic/home**
 - PUT /basic/home**
 - POST /basic/home**
 - DELETE /basic/home**

Below the controller sections, there is a **Schemas** section containing the definitions for the **Employee**, **Payroll**, and **JwtRequest** models.

```
Employee v {
    id          integer($int64)
    name        string
    age         integer($int32)
    designation string
    rating      integer($int32)
    salary      number($double)
}

Payroll v {
    id          integer($int64)
    employee    Employee > {...}
    basicSalary number($double)
    taxDeductions number($double)
    netSalary   number($double)
    payDate     string($date-time)
}

JwtRequest v {
    username    string
    password    string
}
```

localhost:8080/swagger-ui/index.html?configUrl=/v3/api-docs/swagger-config#/authentication-controller/createAuthenticationToken

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/getToken' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "username": "payroll_management",
    "password": "payroll_management"
}'
```

Request URL

```
http://localhost:8080/getToken
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "token": "eyJhbGciOiJIUzIwMiJ9.eyJzdWIiOiJwVXlyb2xsX21hbmFnZW1lbnQjLCJleHAiOiE3MjUyNTMyNjIsImhdCI6MTcyNTI0OTY2Mn8.Yuf7N_3efu1PSckzIhOzt4gEbgaxDUpS2zEysJaW-m2IXwGNDs9GP8RItQebPAatA75s8d7XcskII2WKo1MIGXg" }</pre> <p>Response headers</p> <pre>cache-control: no-cache,no-store,max-age=0,must-revalidate connection: keep-alive content-type: application/json date: Mon,02 Sep 2024 04:01:02 GMT expires: 0 keep-alive: timeout=60 pragma: no-cache transfer-encoding: chunked vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers x-content-type-options: nosniff x-frame-options: DENY x-xss-protection: 1; mode=block</pre>

JUNIT Testcases

The image displays three separate screenshots of the Eclipse IDE interface, each showing the results of a JUnit test run. The screenshots are arranged vertically.

- Screenshot 1:** Shows the results of the `EmployeeServiceTest`. It indicates "Runs: 6/6", "Errors: 0", and "Failures: 0". The test completed in 0.628 seconds. The test class is highlighted in blue.
- Screenshot 2:** Shows the results of the `payrollt` test. It indicates "Runs: 6/6", "Errors: 0", and "Failures: 0". The test completed in 0.669 seconds. The test class is highlighted in blue.
- Screenshot 3:** Shows the results of the `TaxServiceTest`. It indicates "Runs: 6/6", "Errors: 0", and "Failures: 0". The test completed in 0.025 seconds. The test class is highlighted in blue.

On the right side of the interface, there is a vertical toolbar with several icons, and below it, a column of numerical values (1, 1, 1, 1, 1, 1, 1, 1) which likely correspond to the test results or progress indicators.

EUREKA :

The screenshot shows the Spring Eureka UI interface. At the top, there are tabs for Eureka, Swagger UI, and localhost:8080/actuator/health. The main content area has a header "spring Eureka" and navigation links for HOME and LAST 1000 SINCE STARTUP.

System Status

Environment	N/A	Current time	2024-08-31T10:51:23 +0530
Data center	N/A	Uptime	00:03
		Lease expiration enabled	true
		Renews threshold	5
		Renews (last min)	8

DS Replicas

localhost

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
GATEWAYEXAM	n/a (1)	(1)	UP (1) - LAPTOP-UI2CQFFD:gatewayexam:8111
PAYROLLMANAGEMENTAPPLICATION	n/a (1)	(1)	UP (1) - LAPTOP-UI2CQFFD:PayrollManagementApplication

General Info

Name	Value
version	2.9.0

ACTUATOR :

The screenshot shows the Actuator health endpoint at localhost:8080/actuator/health. The response is displayed in JSON format, showing the status of various components like db, discoveryComposite, eureka, and diskSpace.

```
status: "UP"
components:
  db:
    status: "UP"
    details:
      database: "MySQL"
      validationQuery: "isValid()"
  discoveryComposite:
    status: "UP"
    components:
      discoveryClient:
        status: "UP"
        details:
          services:
            0: "gatewayexam"
            1: "payrollmanagementapplication"
      eureka:
        description: "Remote status from Eureka server"
        status: "UP"
        details:
          applications:
            GATEWAYEXAM: 1
            PAYROLLMANAGEMENTAPPLICATION: 1
  diskSpace:
    status: "UP"
    details:
      total: 231687057408
      free: 106586386432
      threshold: 10485760
      exists: true
  ping:
    status: "UP"
  refreshScope:
    status: "UP"
```

POSTMAN :

The screenshot shows the Postman interface with a POST request to `http://localhost:8080/getToken`. The request body is a JSON object with `"username": "payroll_management"` and `"password": "payroll_management"`. The response status is 200 OK, with a token returned in the body.

```
1 {  
2   "username": "payroll_management",  
3   "password": "payroll_management"  
4 }
```

Body Cookies Headers (14) Test Results
Pretty Raw Preview Visualize JSON ↻

```
1 {  
2   "token": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJwYXlyb2xsX21hbmfNzW1lbnQiLCJleHAiOjE3MjUyNTI2NzksImhlhdCI6MTcyNTI0OTA3OX0.  
3     oqfth07k_zh1lfaiTqs0L-Itt09PMbFf0tMeTwUhkLBA0n6XAhB3qBYtjd1P5wX8aJ0oCKmAaf_4Snh_Dcz2w"  
}
```

200 OK • 2.21 s • 638 B | ↻

The screenshot shows a MySQL Workbench session with the following queries:

```
1 ● use shaik_db;  
2 ● show tables;  
3 ● select*from employee;  
4 ● select*from payroll;
```

The result grid displays the data from the `employee` table:

	id	age	designation	name	rating	salary
▶	1	24	developer	Shafi	4	34000
9	25	Data Analyst	saif	3	36000	
10	28	senior Software engineer	raina	4	41000	
11	26	Bussiness Analyst	pooja	3	36000	
12	25	devops engineer	raj	3	36000	
13	20	Accountant	Arshaf	4	30000	
*	HULL	HULL	HULL	HULL	HULL	HULL

The screenshot shows the Postman application interface. A collection named "My Workspace" is selected. A new request is being made to the URL `http://localhost:8080/payroll/1`. The request method is set to GET. In the Body section, there is a JSON payload:

```
1 {
2   "username": "payroll_management",
3   "password": "payroll_management"
4 }
```

The response status is 200 OK, with a response time of 108 ms and a size of 556 B. The response body is displayed in pretty JSON format:

```
1 {
2   "id": 1,
3   "employee": {
4     "id": 1,
5     "name": "Shafiq",
6     "age": 24,
7     "designation": "Developer",
8     "rating": 4,
9     "salary": 34000.0
10 },
11   "basicSalary": 34000.0,
12   "taxDeductions": 0.0,
13   "netSalary": 34000.0,
14   "payDate": "2024-09-01T14:28:06.789+00:00"
15 }
```

The screenshot shows the Postman application interface. A new request is being made to the URL `http://localhost:8080/payroll/2`. The request method is set to GET. In the Body section, there is a JSON payload:

```
1 {
2   "username": "payroll_management",
3   "password": "payroll_management"
4 }
```

The response status is 200 OK, with a response time of 108 ms and a size of 556 B. The response body is displayed in pretty JSON format:

```
1 {
2   "id": 2,
3   "employee": {
4     "id": 9,
5     "name": "saif",
6     "age": 25,
7     "designation": "Data Analyst",
8     "rating": 3,
9     "salary": 36000.0
10 },
11   "basicSalary": 36000.0,
12   "taxDeductions": 0.0,
13   "netSalary": 36000.0,
14   "payDate": "2024-09-01T17:35:05.993+00:00"
15 }
```

GET http://localhost:8080/payroll/3

HTTP http://localhost:8080/payroll/3

GET http://localhost:8080/payroll/3

Params Authorization Headers (9) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "username": "payroll_management",  
3   "password": "payroll_management"  
4 }
```

Body Cookies Headers (11) Test Results 200 OK

Pretty Raw Preview Visualize JSON

```
1 {  
2   "id": 3,  
3   "employee": {  
4     "id": 10,  
5     "name": "raina",  
6     "age": 28,  
7     "designation": "senior Software engineer",  
8     "rating": 4,  
9     "salary": 41000.0  
10   },  
11   "basicSalary": 38000.0,  
12   "taxDeductions": 0.0,  
13   "netSalary": 38000.0,  
14   "payDate": "2024-09-01T17:35:21.717+00:00"  
15 }
```

GET http://localhost:8080/payroll/4

HTTP http://localhost:8080/payroll/4

GET http://localhost:8080/payroll/4

Params Authorization Headers (9) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "username": "payroll_management",  
3   "password": "payroll_management"  
4 }
```

Body Cookies Headers (11) Test Results 200 OK

Pretty Raw Preview Visualize JSON

```
1 {  
2   "id": 4,  
3   "employee": {  
4     "id": 11,  
5     "name": "pooja",  
6     "age": 26,  
7     "designation": "Bussiness Analyst",  
8     "rating": 3,  
9     "salary": 36000.0  
10   },  
11   "basicSalary": 36000.0,  
12   "taxDeductions": 0.0,  
13   "netSalary": 36000.0,  
14   "payDate": "2024-09-01T17:35:40.519+00:00"  
15 }
```

GET http://localhost:8080/payroll/5

http://localhost:8080/payroll/5

GET http://localhost:8080/payroll/5

Params Authorization Headers (9) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "username": "payroll_management",  
3   "password": "payroll_management"  
4 }
```

Body Cookies Headers (11) Test Results 200 OK 36

Pretty Raw Preview Visualize JSON

```
1 {  
2   "id": 5,  
3   "employee": {  
4     "id": 12,  
5     "name": "raj",  
6     "age": 25,  
7     "designation": "devops engineer",  
8     "rating": 3,  
9     "salary": 36000.0  
10    },  
11   "basicSalary": 36000.0,  
12   "taxDeductions": 0.0,  
13   "netSalary": 36000.0,  
14   "payDate": "2024-09-01T17:35:46.896+00:00"  
15 }
```

Query 1

```
1 • use shaik_db;  
2 • show tables;  
3 • select*from employee;  
4 • select*from payroll;
```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content: Result Grid Form Editor Field Types

	id	basic_salary	net_salary	pay_date	tax_deductions	employee_id
▶	1	34000	34000	2024-09-01 19:58:06.789000	0	1
	2	36000	36000	2024-09-01 23:05:05.993000	0	9
	3	38000	38000	2024-09-01 23:05:21.717000	0	10
	4	36000	36000	2024-09-01 23:05:40.519000	0	11
	5	36000	36000	2024-09-01 23:05:46.896000	0	12
*	6	30000	30000	2024-09-02 10:42:54.373000	0	13
	HULL	HULL	HULL	HULL	HULL	HULL

