

PDF report in SpringBoot - build.gradle (PDF report in SpringBoot)

PDF report in SpringBoot > build.gradle

```
1 plugins {
2     id 'java'
3     id 'org.springframework.boot' version '3.2.0'
4     id 'io.spring.dependency-management' version '1.1.4'
5 }
6
7 group = 'com.zetcode'
8 version = '0.0.1-SNAPSHOT'
9
10 java {
11     sourceCompatibility = '17'
12 }
13
14 repositories {
15     mavenCentral()
16 }
17
18 dependencies {
19     implementation 'org.springframework.boot:spring-boot-starter-web'
20     implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
21     implementation 'com.github.librepdf:openpdf:1.3.30'
22     runtimeOnly 'com.h2database:h2'
23 }
```

PDF report in SpringBoot - City.java [PDF_report_in_SpringBoot.main]

PDF report in SpringBoot > src > main > java > com > zetcode > model > City > id

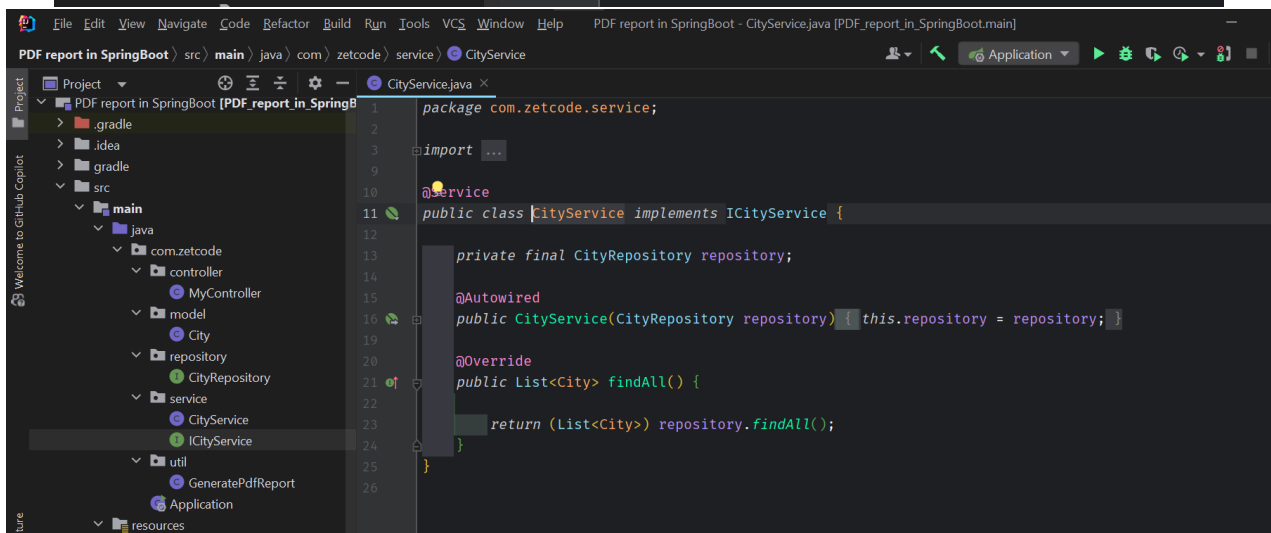
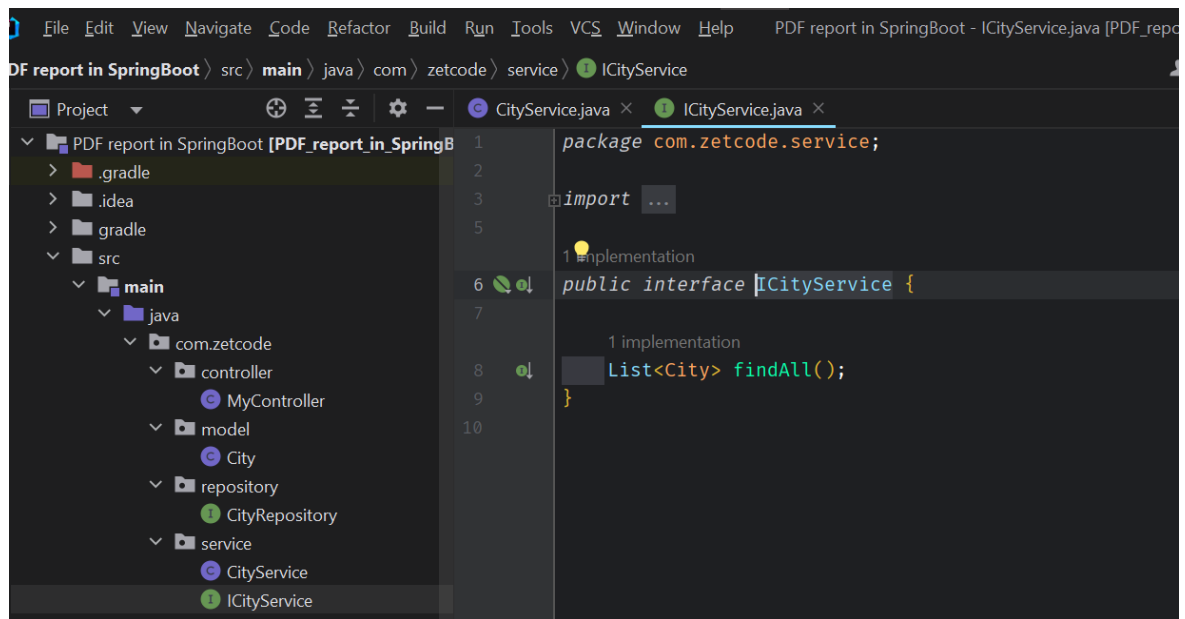
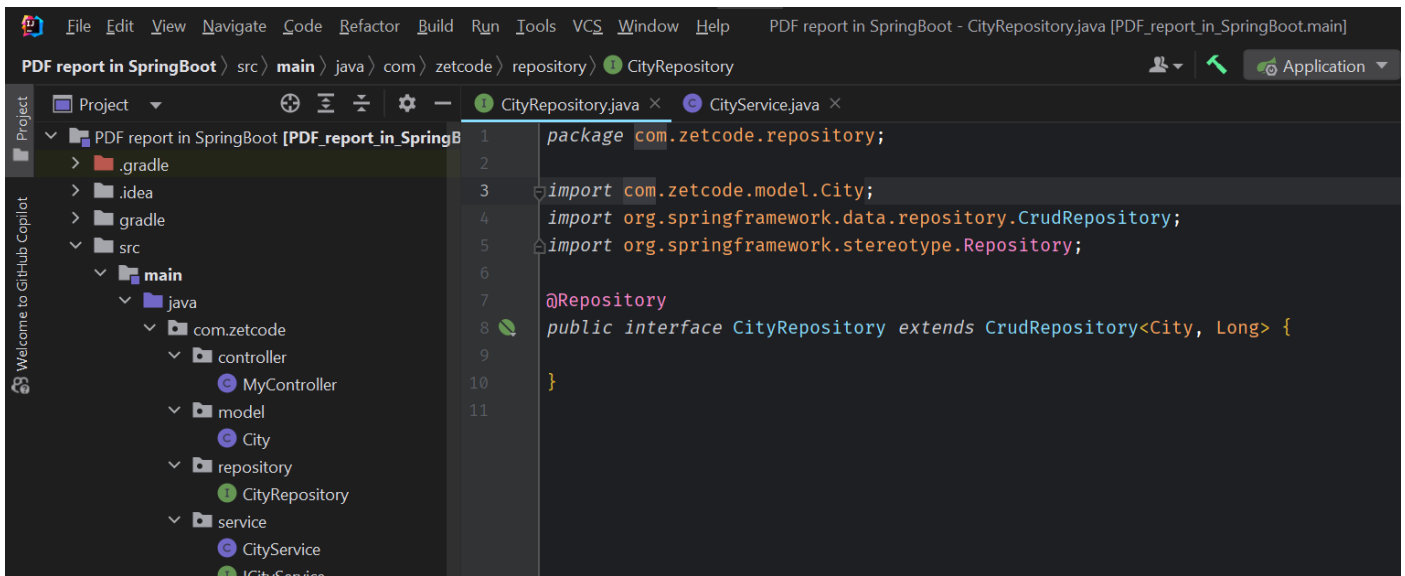
```
10
11
12 @Entity
13 @Table(name = "cities")
14 public class City {
15
16     @Id
17     @GeneratedValue(strategy = GenerationType.IDENTITY)
18     private Long id;
19
20     private String name;
21     private int population;
22
23     public City() {
24     }
25
26     public City(String name, int population) {
27         this.name = name;
28         this.population = population;
29     }
30
31     public Long getId() { return id; }
32
33     public void setId(Long id) { this.id = id; }
34
35     public String getName() { return name; }
36
37     public void setName(String name) { this.name = name; }
38
39 }
```

```
74 }
75
76 final City other = (City) obj;
77 if (this.population != other.population) {
78     return false;
79 }
80 if (!Objects.equals(this.name, other.name)) {
81     return false;
82 }
83 return Objects.equals(this.id, other.id);
84 }
85
86 @Override
87 public String toString() {
88     var builder = new StringBuilder();
89     builder.append("City{id=").append(id).append(", name=")
90         .append(name).append(", population=")
91         .append(population).append("}");
92
93     return builder.toString();
94 }
95 }
96 }
```

```
{(<)} Rainbowify tag name for YAML can be enabled
1 spring:
2   main:
3     banner-mode: "off"
4   sql:
5     init:
6       platform: h2
7
8 logging:
9   level:
10    org:
11      springframework: ERROR
12 }
```

Importamos datos para la base de datos SQL

```
No data sources are configured to run this SQL and provide advanced code assistance.
SQL dialect is not configured.
1 INSERT INTO cities(name, population) VALUES('Bratislava', 432000);
2 INSERT INTO cities(name, population) VALUES('Budapest', 1759000);
3 INSERT INTO cities(name, population) VALUES('Prague', 1280000);
4 INSERT INTO cities(name, population) VALUES('Warsaw', 1748000);
5 INSERT INTO cities(name, population) VALUES('Los Angeles', 3971000);
6 INSERT INTO cities(name, population) VALUES('New York', 8550000);
7 INSERT INTO cities(name, population) VALUES('Edinburgh', 464000);
8 INSERT INTO cities(name, population) VALUES('Suzhou', 4327066);
9 INSERT INTO cities(name, population) VALUES('Zhengzhou', 4122087);
10 INSERT INTO cities(name, population) VALUES('Berlin', 3671000);
11 INSERT INTO cities(name, population) VALUES('Brest', 139163);
12 INSERT INTO cities(name, population) VALUES('Bucharest', 1836000);
13
```



```
PDF report in SpringBoot - MyController.java [PDF_report_in_SpringBoot.main]
PDF report in SpringBoot > src > main > java > com > zetcode > controller > MyController

Project
  PDF report in SpringBoot [PDF_report_in_SpringB
    .gradle
    .idea
    gradle
    src
      main
        java
          com.zetcode
            controller
              MyController
            model
              City
            repository
              CityRepository
            service
              CityService
              ICityService
            util
              GeneratePdfReport
            Application
          resources
            application.yml
            import.sql
        build.gradle
        gradlew
        gradlew.bat
    External Libraries

Welcome to GitHub Copilot

Structure
Bookmarks

17
18 @Controller
19 public class MyController {
20
21     private final ICityService cityService;
22
23     @Autowired
24     public MyController(ICityService cityService) { this.cityService = cityService; }
25
26     @RequestMapping(value = "/pdfreport", method = RequestMethod.GET,
27                     produces = MediaType.APPLICATION_PDF_VALUE)
28     public ResponseEntity<InputStreamResource> citiesReport() {
29
30         var cities = (List<City>) cityService.findAll();
31
32         ByteArrayInputStream bis = GeneratePdfReport.citiesReport(cities);
33
34         var headers = new HttpHeaders();
35         headers.add("Content-Disposition", "inline; filename=citiesreport.pdf");
36
37         return ResponseEntity
38             .ok()
39             .headers(headers)
40             .contentType(MediaType.APPLICATION_PDF)
41             .body(new InputStreamResource(bis));
42     }
43
44 }
45
```

```
PDF report in SpringBoot - GeneratePdfReport.java [PDF_report_in_SpringBoot.main]
PDF report in SpringBoot > src > main > java > com > zetcode > util > GeneratePdfReport

Project
  PDF report in SpringBoot [PDF_report_in_SpringB
    .gradle
    .idea
    gradle
    src
      main
        java
          com.zetcode
            controller
              MyController
            model
              City
            repository
              CityRepository
            service
              CityService
              ICityService
            util
              GeneratePdfReport
            Application
          resources
            application.yml
            import.sql
        build.gradle
        gradlew
        gradlew.bat
    External Libraries

Welcome to GitHub Copilot

Structure
Bookmarks

19
20 public class GeneratePdfReport {
21
22     private static final Logger logger = LoggerFactory.getLogger(GeneratePdfReport.class);
23
24     @
25     public static ByteArrayInputStream citiesReport(List<City> cities) {
26
27         Document document = new Document();
28         ByteArrayOutputStream out = new ByteArrayOutputStream();
29
30         try {
31
32             PdfPTable table = new PdfPTable(numColumns: 3);
33             table.setWidthPercentage(60);
34             table.setWidths(new int[]{1, 3, 3});
35
36             Font headFont = FontFactory.getFont(FontFactory.HELVETICA_BOLD);
37
38             PdfPCell hcell;
39             hcell = new PdfPCell(new Phrase("Id", headFont));
40             hcell.setHorizontalAlignment(Element.ALIGN_CENTER);
41             table.addCell(hcell);
42
43             hcell = new PdfPCell(new Phrase("Name", headFont));
44             hcell.setHorizontalAlignment(Element.ALIGN_CENTER);
45             table.addCell(hcell);
46
47         } catch (Exception e) {
48             logger.error("Error generating PDF report: " + e.getMessage());
49         }
50
51         return out.toByteArray();
52     }
53 }
54
```

