* Ca REST API sa returneze un pdf, va fi necesar sa returnam un byte[]
* Deci, raportul trebuie exportat nu ca File, ci ca byte[]
* Vom avea nevoie de un Bean, si de ex sa fie de tip JasperPrint, care e raportul compilat cu date, doar ca inca netransformat in vreun file sau ceva de genu

@SpringBootApplication  
public class Main {  
 public static void main(String[] args) throws JRException {  
 SpringApplication.*run*(Main.class);  
  
 }  
 @Bean  
 public JasperPrint Report() throws JRException {  
 String path = "C:\\Users\\user\\Desktop\\demo\\src\\main\\resources\\templates\\";  
 Map<String, Object> reportParam = new HashMap<>();  
 Map<String, Object> subreportParam = new HashMap<>();  
  
 JasperReport jasperReport = JasperCompileManager.*compileReport*(path+"clients.jrxml");  
 JasperReport subreport = JasperCompileManager.*compileReport*(path+"subreport.jrxml");  
  
 reportParam.put("subreport", subreport);  
 subreportParam.put("data", new JRBeanCollectionDataSource(List.*of*(  
 new Customer("Mititiuc","Eduard",21,20000.0),  
 new Customer("Goncear","Alex",25,1000.0),  
 new Customer("Petru","Turcan",20,500.0)  
  
 )));  
 reportParam.put("subreportParameters", subreportParam);  
  
 return JasperFillManager.*fillReport*(jasperReport, reportParam, new JREmptyDataSource());  
 }  
}

* Acum, cream REST API
* @RestController  
  @RequiredArgsConstructor  
  public class REST {  
   private final JasperPrint jasperPrint;  
   @GetMapping  
   public ResponseEntity<byte[]> getReport() throws JRException {  
   HttpHeaders httpHeaders = new HttpHeaders();  
   httpHeaders.setContentType(MediaType.*APPLICATION\_PDF*);  
    
   return new ResponseEntity<>(JasperExportManager.*exportReportToPdf*(jasperPrint), httpHeaders, HttpStatus.*OK*);  
   }  
  }

- trebuie sa specificam in httpHeaders ca ce vom returna nu va fi un JSON, ci un PDF fisier

- exportReportToPdf returneaza un byte[] al raportului, pe can exportReportToPdfFile un File

**Metoda mai grea**

@RestController  
@RequiredArgsConstructor  
public class REST {  
 private final JasperPrint jasperPrint;  
 @GetMapping  
 public ResponseEntity<byte[]> getReport() throws JRException {  
 HttpHeaders httpHeaders = new HttpHeaders();  
 httpHeaders.setContentType(MediaType.*APPLICATION\_PDF*);  
  
 ByteArrayOutputStream byteArray = new ByteArrayOutputStream();  
 JRPdfExporter jrPdfExporter = new JRPdfExporter();  
 SimplePdfExporterConfiguration simplePdfExporterConfiguration = new SimplePdfExporterConfiguration();  
 simplePdfExporterConfiguration.setCompressed(true);  
 jrPdfExporter.setConfiguration(simplePdfExporterConfiguration);  
 jrPdfExporter.setExporterInput(new SimpleExporterInput(jasperPrint));  
 jrPdfExporter.setExporterOutput(new SimpleOutputStreamExporterOutput(byteArray));  
 jrPdfExporter.exportReport();  
  
 return new ResponseEntity<>(byteArray.toByteArray(), httpHeaders, HttpStatus.*OK*);  
 }  
}