

Roteiro do minicurso

Desenvolvimento de APIs Rest com Java, Spring e Solid Pattern

Passo a passo

1. Mostrar o uso do Spring Initializr.
2. Explicar o corpo do projeto, o application properties e o pom.xml.
3. Explicar cada configuração do application.properties.

```
#Spring Boot server configuration
spring.datasource.url=jdbc:postgresql://localhost:5432/minicurso
spring.datasource.username=postgres
spring.datasource.password=postgres
spring.jpa.hibernate.ddl-auto=update
// Mostra todos os comandos SQL executados.
spring.jpa.show-sql=true
```

4. Mostrar e explicar a modelagem do banco de dados.
5. Criar os Models, explicando as @annotation's do Jakarta Persistence.
 - O que é o Jakarta Persistence?
 - R: API de gerenciamento de persistência e mapeamento objeto/relacional.
 - Explicar o uso do Lombok.
 - Deixar propositalmente o erro de referência circular e/ou resposta gigantesca na API (Não uso do @JsonIgnoreProperties).
6. Criar e explicar os Repositories, explicar porque não usaremos ele diretamente no controller.
 - Explicar que o uso de @Repository é opcional quando extendemos JpaRepository.
7. Criar e explicar DTOs, explicar o porque de usar o Record do Java 16.
8. Criar e explicar os Services, e também explicar que é um padrão/boa prática utilizar a Interface de Service ao invés da implementação, para que seja de fácil manutenção nos códigos.
9. Criar os Controllers, explicar como funcionam as annotations @RestController e @RequestMapping, @GetMapping, @PostMapping, @PutMapping, @DeleteMapping e comentar sobre o @PatchMapping.
10. Explicar como funcionam as annotations @PathVariable, @RequestBody e @RequestParam.
11. Explicar ResponseEntity.
12. Explicar diferença entre @PathVariable e @RequestParam.
13. Mostrar a segunda forma de fazer o método de adicionar aluno.

```
@PostMapping("/adicionar-aluno")
public ResponseEntity<Turma> addAluno(@RequestParam int turma_id, @RequestParam
int aluno_id) throws Exception {
    Turma turma = turmaService.addAluno(turma_id, aluno_id);
    return ResponseEntity.ok().body(turma);
}
```

14. Criar e explicar o Controller de tratamento de exceções
"ExceptionHandlerController", explicar @RestControllerAdvice.

Conteúdo extra, caso dê tempo:

- Para alterar os ícones das pastas, usando o Material Icons Theme: // Nome dos ícones é encontrado na documentação do Material Icons Theme, pode ser visto dentro do próprio VS Code.

```
"material-icon-theme.folders.associations": {  
  "nomePasta": "nomeIcon"  
}
```