@RestController  
@RequestMapping(value = "/livros")  
public class LivroResources {  
  
 @Autowired  
 private LivroService livroService;  
  
 @GetMapping(value = "/{id}")  
 public ResponseEntity<Livro> findById(@PathVariable Integer id){  
 Livro obj = livroService.findById(id);  
 return ResponseEntity.*ok*().body(obj);  
 }  
  
 @GetMapping  
 public ResponseEntity<List<LivroDTO>> findAll(@RequestParam(value = "categoria",defaultValue = "0") Integer id\_cat) {  
 List<Livro> list = livroService.findAll(id\_cat);  
 List<LivroDTO> listDTO = list.stream().map(obj -> new LivroDTO(obj)).collect(Collectors.*toList*());  
 return ResponseEntity.*ok*().body(listDTO);  
 }  
  
  
 @PutMapping(value = "/{id}")  
 public ResponseEntity<Livro> update(@PathVariable Integer id, @RequestBody Livro obj){  
 Livro newObj = livroService.update(id, obj);  
 return ResponseEntity.*ok*().body(newObj);  
 }  
  
 @PatchMapping(value = "/{id}")  
 public ResponseEntity<Livro> updatePatch(@PathVariable Integer id, @RequestBody Livro obj){  
 Livro newObj = livroService.update(id, obj);  
 return ResponseEntity.*ok*().body(newObj);  
 }  
  
  
 @PostMapping  
 public ResponseEntity<Livro> create (@RequestParam(value = "categoria",defaultValue = "0") Integer id\_cat, @RequestBody Livro obj){  
 Livro newObj = livroService.create(id\_cat,obj);  
 URI uri = ServletUriComponentsBuilder.*fromCurrentRequest*().path("/livros{id}").buildAndExpand(newObj.getId()).toUri();  
 return ResponseEntity.*created*(uri).build();  
 }  
  
  
  
  
}

@Service  
public class LivroService {  
  
 @Autowired  
 private LivroRepository livroRepository;  
  
 @Autowired  
 private CategoriaService categoriaService;  
  
  
 public Livro findById (Integer id){  
 Optional<Livro> obj = livroRepository.findById(id);  
 return obj.orElseThrow(() -> new ObjectNotFoundException(  
 "Livro não encontrado! ID: " + id + ", TIPO: " + Livro.class.getName() ));  
  
 }  
  
 public List<Livro> findAll(Integer id\_cat){  
 categoriaService.findById(id\_cat);  
 return livroRepository.findAllByCategoria(id\_cat);  
 }  
  
  
 public Livro update(Integer id, Livro obj){  
 Livro newObj = findById(id);  
 updateData(newObj,obj);  
 return livroRepository.save(newObj);  
  
 }  
  
 public void updateData(Livro newObj, Livro obj){  
 newObj.setNome\_autor(obj.getNome\_autor());  
 newObj.setTexto(obj.getTexto());  
 newObj.setTitulo(obj.getTitulo());  
  
 }  
  
  
 public Livro create(Integer id\_cat, Livro obj) {  
 obj.setId(null);  
 Categoria cat = categoriaService.findById(id\_cat);  
 obj.setCategoria(cat);  
 return livroRepository.save(obj);  
 }  
  
  
  
}

@Repository  
public interface LivroRepository extends JpaRepository<Livro,Integer> {  
  
 @Query("SELECT obj FROM Livro WHERE obj.categoria.id = :id\_cat ORDER BY titulo")  
 List<Livro> findAllByCategoria(@Param(value = "id\_cat") Integer id\_cat);  
  
  
  
}