// É uma classe que o spring vai gerenciar  
// É uma classe que pertence a logica do serviço  
// O Controller vai acessar ela para retirar as informações e mandar para o front  
// A classe Service geralmente ela que faz a transação dos dados  
// Ela que vai acessar ao banco e devolver os dados para o controller  
@Service  
public class SerieService {  
  
 @Autowired  
 private SerieRepository repositorio;  
  
 public List<SerieDTO> obterTodasAsSeries() {  
 return repositorio.findAll()  
 .stream()  
 .map(s -> new SerieDTO(s.getId(), s.getTitulo(), s.getTotalTemporadas(), s.getAvaliacao(), s.getGenero(), s.getAtores(), s.getSinopse(), s.getPoster()))  
 .collect(Collectors.*toList*());  
 }  
  
 **public List<SerieDTO> obterTop5Series() {  
 return repositorio.findTop5ByOrderByAvaliacaoDesc()  
 .stream()  
 .map(s -> new SerieDTO(s.getId(), s.getTitulo(), s.getTotalTemporadas(), s.getAvaliacao(), s.getGenero(), s.getAtores(), s.getSinopse(), s.getPoster()))  
 .collect(Collectors.*toList*());  
 }**}

Repare que criamos um novo metodo para retornar o top 5

Onde pegamos o reposito criamos uma nova coleção

Transformando os dados MAP em nosso RECORD e passando os parâmetros que queremos, e ai depois criando uma nova coleção de list.

E no nosso controller estamos chamando ele.

@RestController // Diz para retornar um objeto serializado em JSON ou XML como reposta.  
public class SerieController {  
  
  
 @Autowired  
 private SerieService serieService;  
  
 @GetMapping("/series")  
 public List<SerieDTO> obterSeries(){  
 return serieService.obterTodasAsSeries();  
 }  
  
  
 @GetMapping("/series/top5")  
 public List<SerieDTO> obterTop5Series(){  
 return serieService.obterTop5Series();  
}

Mas repara que estamos repetindo os códigos, então vamos criar um método privado para resolver esse problema

Repare agora como ficou

public List<SerieDTO> obterTodasAsSeries() {  
 return converteDados(repositorio.findAll());  
   
 }  
  
 public List<SerieDTO> obterTop5Series() {  
 return converteDados(repositorio.findTop5ByOrderByAvaliacaoDesc());  
  
 }  
  
 private List<SerieDTO> converteDados(List<Serie> series) {  
 return series.stream()  
 .map(s -> new SerieDTO(s.getId(), s.getTitulo(), s.getTotalTemporadas(), s.getAvaliacao(), s.getGenero(), s.getAtores(), s.getSinopse(), s.getPoster()))  
 .collect(Collectors.*toList*());  
 }  
}