Agora vamos acerta nosso controller para poder enviar os dados reais para o navegador.

Agora vamos passar uma lista de SERIES

@RestController  
public class SerieController {  
  
 @Autowired  
 private SerieRepository repositorio;  
   
 @GetMapping("/series")  
 **public List<Serie>** obterSeries(){  
 return repositorio.findAll();  
 }  
  
}

Vamos inicializar nosso banco na classe CONTROLER

Vamos modificar o método para uma LIST

E vamos retornar todos os dados de Serie.

Lembrar que aqui está rodando uma operação circular, onde Serie tem episódios e Episódios tem Serie.

## DTO

Agora não vamos mais devolver uma serie vamos devolver outro tipo de objeto = DTO – Date Transfer Object – (Objeto de Transferencia de dados)

Vamos utilizar um record para devolver esses dados.

package br.com.alura.screenmatch.dto;  
  
import br.com.alura.screenmatch.model.Categoria;  
  
public record SerieDTO(Long id,  
 String titulo,  
 Integer totalTemporadas,  
 Double avaliacao,  
 Categoria genero,  
 String atores,  
 String sinopse,  
 String poster) {  
}

Agora vamos transformar esses dados de serie em outro usando o MAP

Ficando dessa forma.

@RestController  
public class SerieController {  
  
 @Autowired  
 private SerieRepository repositorio;  
  
 @GetMapping("/series")  
 public List<SerieDTO> obterSeries(){  
 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*());  
 }  
  
}

Agora sim o DTO ta convertendo os dados e trazendo para a gente.

Usando o mesmo modelo de record para fazer uma nova lista

Usando o MAP o map serve para transformar os dados de um para outro.

Ele é uma operação intermediaria.