Classe CONTROLLER

MINHA ROTA

@RestController  
public class FrasesController {  
  
1 @Autowired  
2 private ServiceFrases servico;  
3  
4 @GetMapping("/series/frases")  
5 public FrasesDTO obterFrase(){  
6 return servico.obterFrase();  
  
 }

1 – Dando poder para ele acessar essa classe

4 – Dizendo qual vai ser o link desse método

5 – Dizendo qual tipo de dado deve retornar no caso FraseDTO

6 – Obtendo o retorno da classe service no método .obterFrase

CLASSE SERVICE --------

@Service  
public class ServiceFrases {  
 @Autowired  
 private FraseRepository repositorio;  
  
 private List<FrasesDTO> converterDados(List<Frases> frases){  
 return frases.stream()  
 .map(f -> new FrasesDTO(f.getId(), f.getFrase(), f.getPersonagem(), f.getTitulo(), f.getPoster()))  
 .collect(Collectors.*toList*());  
 }  
  
  
**1 public FrasesDTO obterFrase() {  
2 Optional<Frases> frases = repositorio.findRandom();  
3 if (frases.isPresent()) {  
4 Frases f = frases.get();  
5 return new FrasesDTO(f.getId(), f.getFrase(), f.getPersonagem(), f.getTitulo(), f.getPoster());  
 }  
6 return null;**  
 }  
}

1 - Nome do método e qual tipo de dado ele retorna

2 – Optional pq pode ter ou não esse dado dentro dele

2 – Ele deve acessar o repositório no método findRandom...

3 – Se tiver algo dentro do frases ele acessa.

4 – Pegamos os dados que acessamos do banco e passamos para o objeto f

5 – Retornamos um novo RECORD passando os dados do objeto

Fazendo isso agora vamos retornar o FraseDTO com os dados dentro já serializado para poder usar como JSON

Aqui esta nossa interface JPA

@Query("SELECT f FROM Frases f order by function('RANDOM') LIMIT 1")  
Optional<Frases> findRandom();