Após fazer as anotações, vamos então salvar.

Criamos um modelo chamado REPOSITORY

package br.com.alura.screenmatch.repository;  
  
import br.com.alura.screenmatch.model.Serie;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface SerieRepository extends JpaRepository<Serie, Long> {  
 // primeiro parametro quem é a entidade que estamos persistindo  
 // segundo parametro o tipo do ID que nesse caso definimos como LONG  
}

Só de criamos isso daqui, já vamos poder salvar e recuperar no banco.

Injeção de dependência – É onde eu delego para o Spring instanciar uma classe que vamos precisar utilizar.

Devemos instanciar na classe onde o Spring “Manda” porque ele que vai fazer a magica  
@SpringBootApplication  
public class ScreenmatchApplication implements CommandLineRunner {  
   
 @Autowired  
 private SerieRepository repository;

E ai passamos ele como parâmetro na classe principal

@Override  
 public void run(String... args) throws Exception {  
 Principal principal = new Principal(repository);  
 principal.exibeMenu();  
}  
}

E ai na nossa classe principal precisamos criar o construtor  
private SerieRepository repository;  
  
public Principal(SerieRepository repository) {  
 this.repository = repository;  
}

Agora sim podemos salvar com os dados que estão chegando

private void buscarSerieWeb() {  
 DadosSerie dados = getDadosSerie();  
 Serie serie = new Serie(dados);  
 repository.save(serie);

Lembrar que estamos usando a interface com o nome Serie, então vamos salvar somente as informações da serie

Atençao lembrar da estrutura dos pacotes, tudo deve esta dentro do spring como   
com.allephnogueira.vendas

├── VendasPostegreSqlApplication.java

├── model

│ └── MinhasVendas.java

└── repository

└── MinhasVendasRepository.java

Texto

Descrição gerada automaticamente