Para criação do projeto, na aba: Criar novo projeto:

Spring Starter Project

Spring version (2.3.3 mesmo)

Procurar por dependência: Spring Web (pois não tem mais o Spring Web Starter)

No teste, iniciamos o projeto, já tem o Tomcat embutido no projeto automaticamente.

1 commit:

Entidade users e criado nosso resource, onde teremos no controle REST

2 commit:

Instalação do BD H2, ele é instalado em memória. Criando um perfil de teste.

Fazendo a instalação do H2 pelo pom.xml já temos junto o JPA automaticamente.

Entendendo as configurações do application properties:

spring.datasource.url=jdbc:h2:mem:testdb String de conexão com o banco, sendo em memória (mem)

spring.datasource.username=sa

spring.datasource.password=

spring.h2.console.enabled=true console está habilitado

spring.h2.console.path=/h2-console acesso ao adm do H2 é este

spring.jpa.show-sql=true mostrar no log da aplicação para mostrar o comando

spring.jpa.properties.hibernate.format\_sql=true formatar o comando SQL

3 commit:

Criando JPA repository, um “sub” framework do Spring, inserção

JPA Repository é uma interface.

TestConfig é uma Classe auxiliar – não e Controlador, Service ou Respository.

Database seeding acontece no TestConfig, quando instanciamos no método run.

4 commit:

Neste momento estamos começando a mexer na nossa camada de serviço, retirando o mockup que colocamos no UserResource da Maria, e chamando o findAll com uma Lista. Para isso, fazemos uma autoinstanciação do UserService para podemos aproveitar a camada de serviço.

A partir daí retornaremos uma Lista de Usuários, portanto ResponseEntity<List<User>>

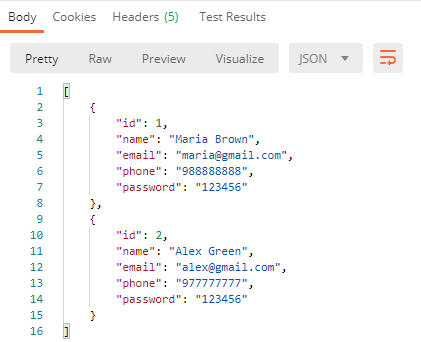
Aqui também registramos pelo framework que o UserService é um @Service, mas no UserRepository não fizemos, pois se ver no código verá que ele herda do próprio JPARepository

public interface UserRepository extends JpaRepository<User, Long>

Portanto é opcional o uso do @.

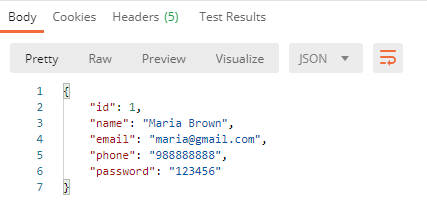
A pilha de camadas funciona aqui: resource chama o service que chama o repositor





Depois disso criamos nosso endpont para recebermos o id no get





Feito na camada de Resource, pois acessa o BD.

Entidade pedido: sem o orderStatus e +total() por enquanto

Para se criar essa entidade, temos no documento o checklist.

Instanciamos a coleção ArrayList de orders....

OBS: set list não tem porque em nenhum momento iremos trocar a lista utilizando o Set, apenas acrescentar e remover elementos.

Em Order, tivemos que fazer esta anotação (aproveitamos e fizemos pro User também):

*@Table*(name = "tb\_order") //tivemos que criar desta forma pois deu erro no build, deu conflito com o Banco de Dados, é uma palavra reservada do SQL

*@Table*(name = "tb\_user")