Oieee migo espero que ajude ^^

Usando rotas no Postman(vou exportar as rotas do Postman para você)

\*\* Pode-se alterar o que será inserido

Métodos HTTP

GET = Consulta

POST= INSERI

PUT= ALTERA

DELETE = Deleta

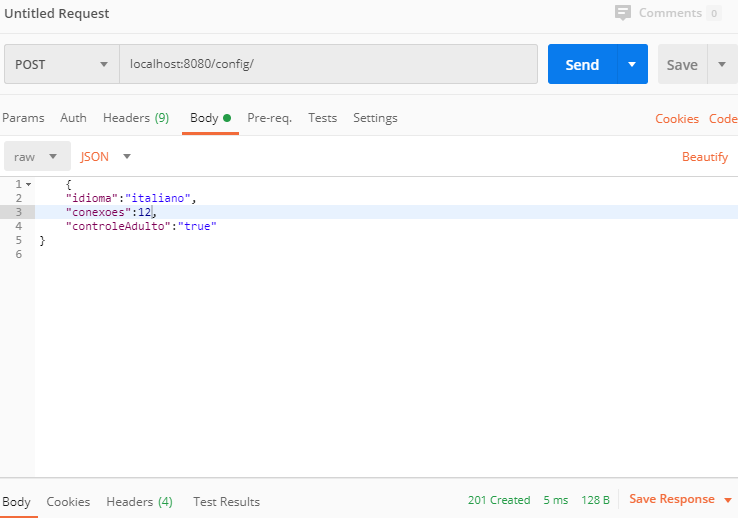
Obs. Você pode usar n vezes esses protocolos http \*\*\* E dentro deles você insere a URI

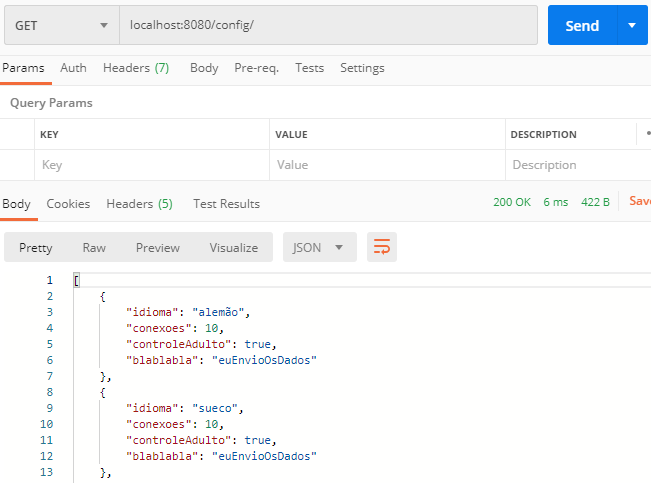
Sua pergunta sobre Headers são cabeçalhos de respostas enviados ao navegador ou cliente

\*\* O content-type é o conteúdo, ele diz ao navegador o tipo/conteúdo do arquivo(para o navegador saber o que fará com o arquivo recebido)

Content-lenght- exibe o tamanho do arquivo para download (tamanho da resposta)

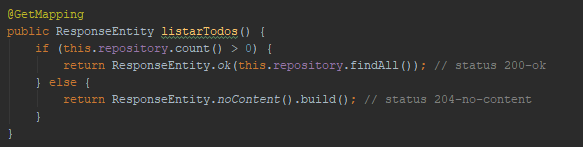
Existem muitos tipos de headers

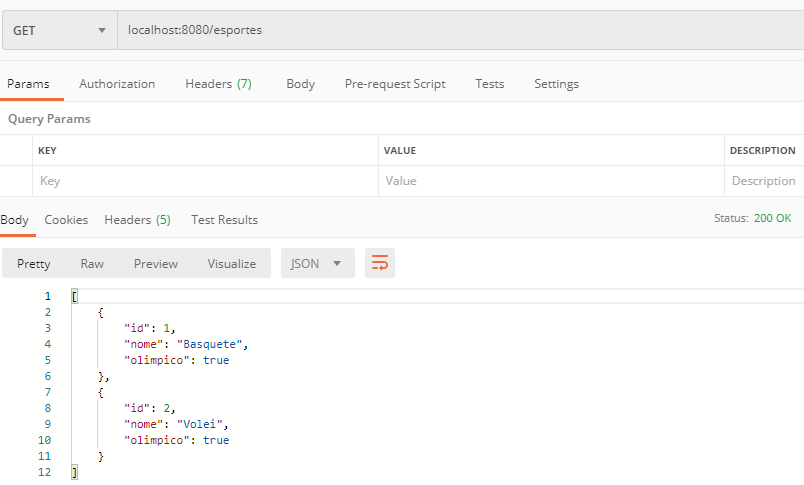


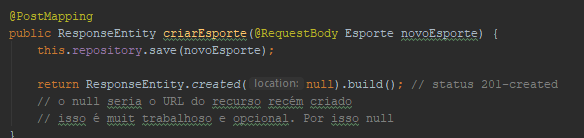


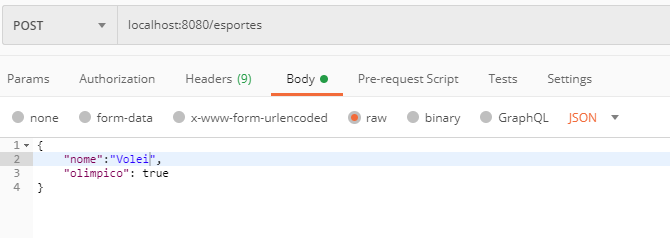
Essa URI será utilizada para todos o que muda é o verbo Https usado get, post etc.

Você só acrescentará algo na uri se for passado por parâmetro .

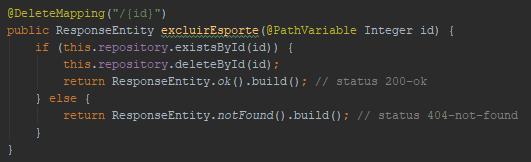


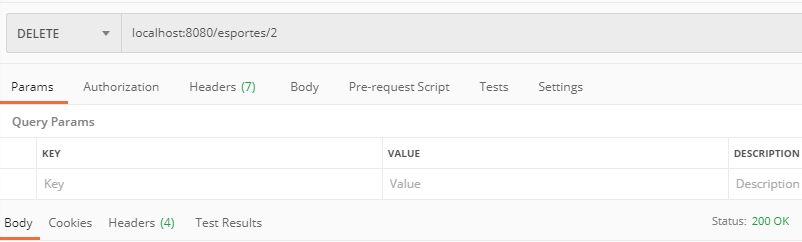




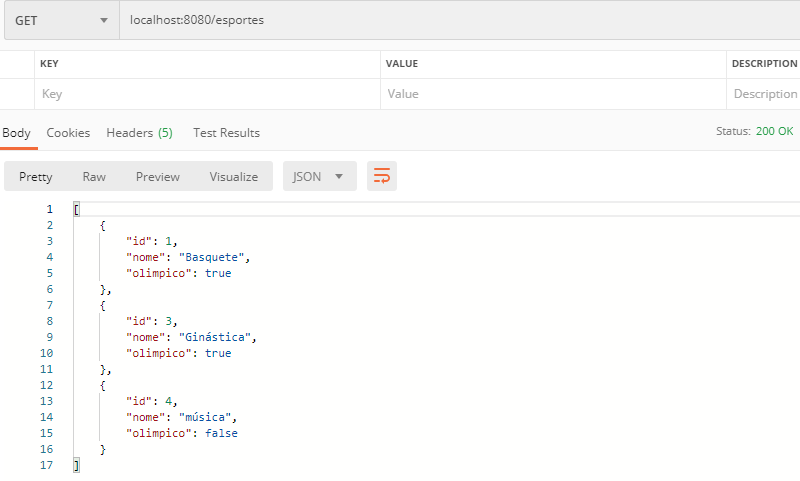


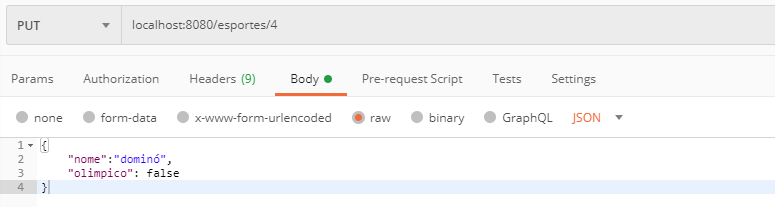
Será incluso na rota o ID

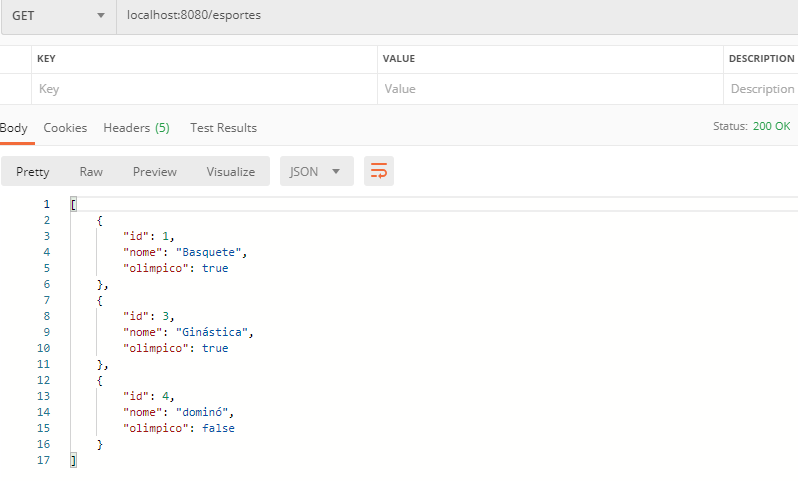




Testando o PUT



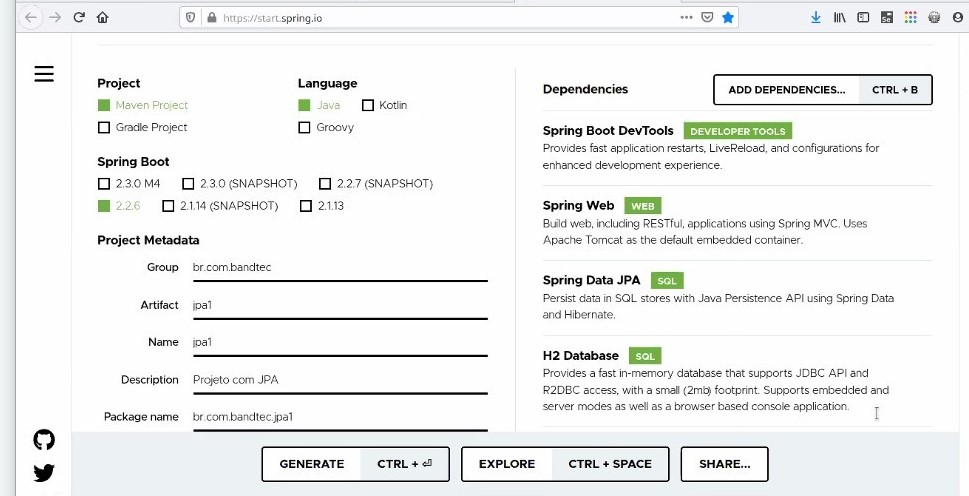




#############################################################################

Pacotes que devem ser baixados

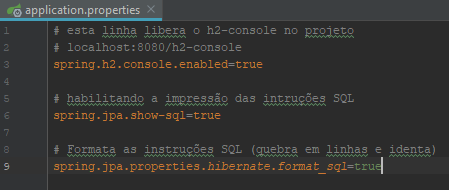
startspring.io



Após salvar o projeto, antes de abrir no IntelliJ, abra o pom com notepad++ ou bloco de notas e mude a versão para 2.2.4

**Configurações Spring**

Para poder usar o h2 é preciso inserir no Application properties no diretório resources



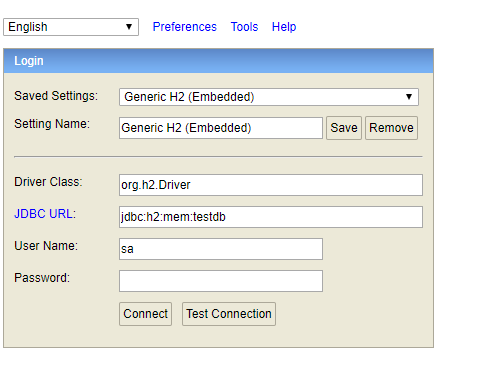
spring.h2.console.enabled=true

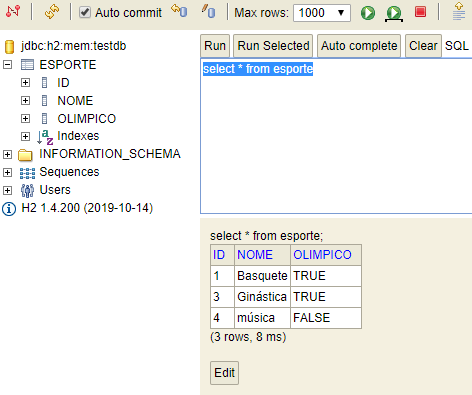
spring.jpa.show-sql=true

spring.jpa.properties.hibernate.format\_sql=true

Endereço da h2 para testes no navegador

<http://localhost:8080/h2-console/>

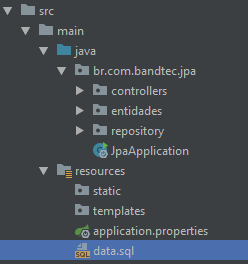




Se quiser inserir um script ao seu banco assim que iniciar a aplicação

Cria um arquivo com o data.sql no diretório Resources

* A estrutura ficará assim



Exemplo de arquivo data.sql

Insert into carro (id, potencia) values (1,2.0);

Select\*from carro;

insert into carro (id, marca, modelo, preco)values

(1, 'BMW', 'BM001', 300000),

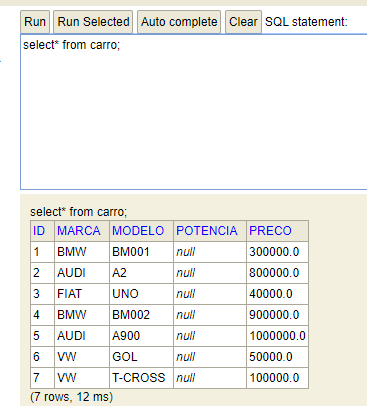
(2, 'AUDI', 'A2', 800000),

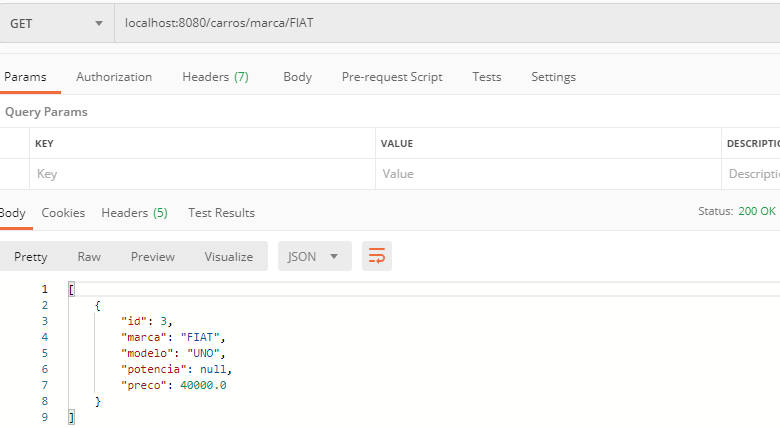
(3, 'FIAT', 'UNO', 40000),

(4, 'BMW', 'BM002', 900000),

(5, 'AUDI', 'A900', 1000000),

(6, 'VW', 'GOL', 50000);





\*\* turmas

insert into turma (id, nome) values

(1, '3ADSA'),

(2, '3ADSB'),

(3, '1CCO');

insert into aluno (id, nome, nascimento, classe\_social, bairro, turma\_id) values

(1, 'Zé Ruela', '1995-01-01', 'A', 'Tatuape', 1),

(2, 'Zé Buduia', '1996-01-01', 'C', 'Se', 1),

(3, 'Maria Bigodenha', '1997-01-01', 'B', 'Braz', 1),

(4, 'Maria Chica', '1998-01-01', 'D', 'Mooca', 2),

(5, 'Lady Gaga', '1999-01-01', 'A', 'Braz', 2),

(6, 'Jojo Todinho', '2000-01-01', 'C', 'Mooca', 3),

(7, 'Zeca Tatú', null, null, null, 3),

(8, 'Maria Capivara', null, null, null, 3),

(9, 'João Ninguém', null, null, null, null);

\*\* Ele pode ser usado tanto no H2 como inserido diretamente em resources, a diferença entre eles é que o h2 não armazena essas informações, então toda vez que entrar deverá inserir.

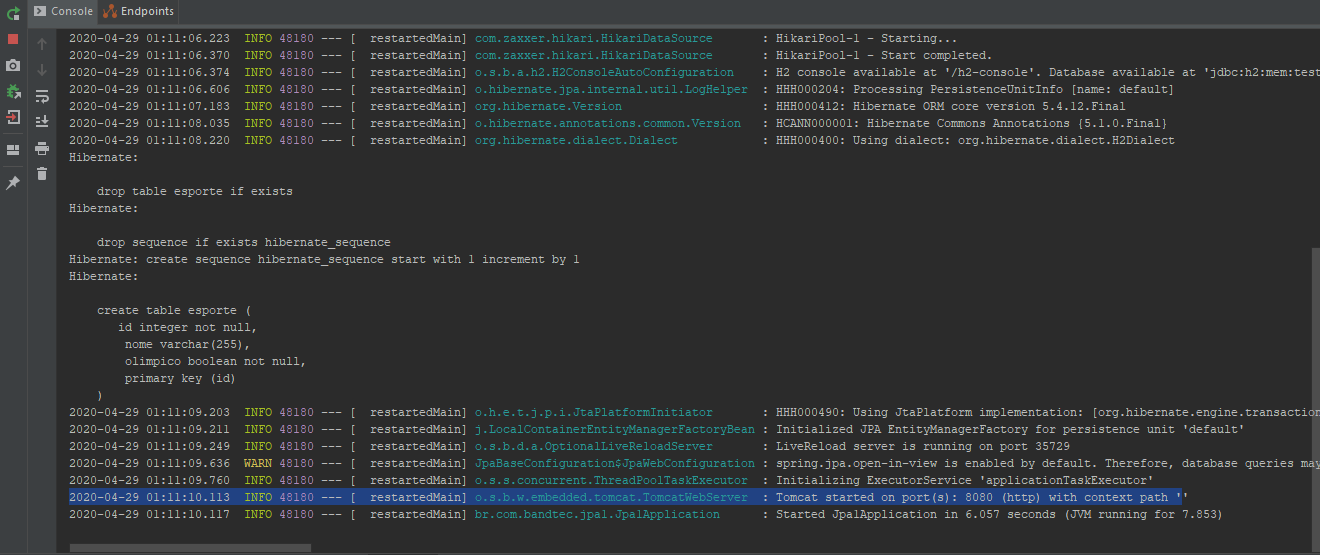
E no data.sql toda vez que a aplicação iniciar esses dados já estarão lá no h2 basta usar o

select \* from nomeDaTabela para consultar.

\*\*E ao usar o Postman e inserir dados, alterar e deletar isso também é feito no H2, pode se conferir dando um select.

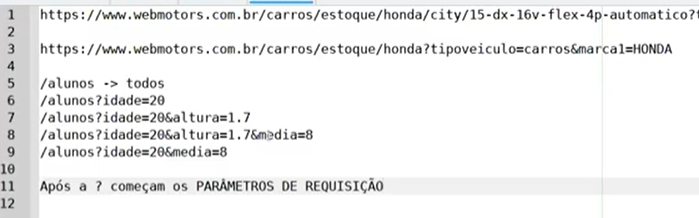
Identificando porta disponível

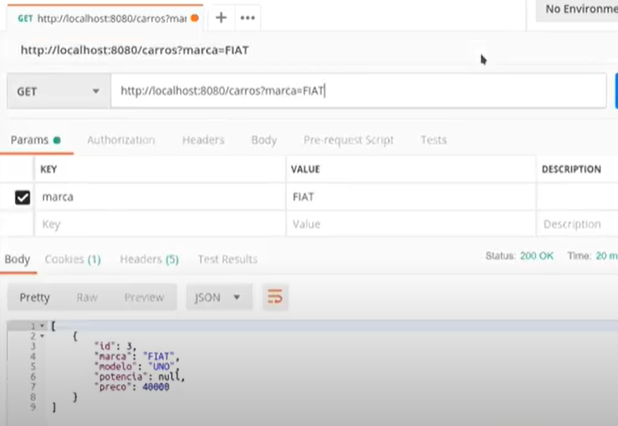
* Esse tomcat started indica a porta, ou seja, qual número vai inserir no localhost no meu caso é 8080

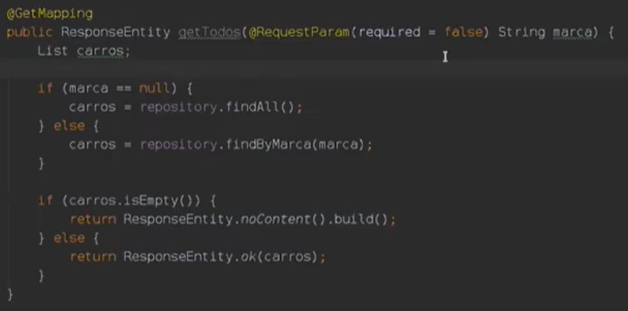


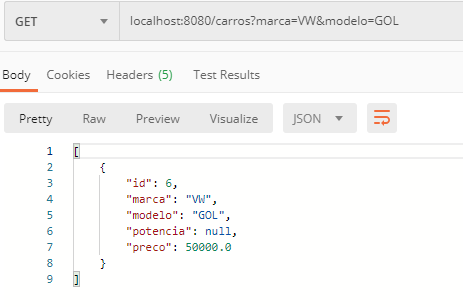
Lembrete: Os getters são usados para exibir e os setters são usados para enviar dados no Json.

Pesquisas por parâmetros usam o ? após a uri passada no mapping e o & para passar outros variáveis como na imagem abaixo.









Usando dois métodos no mapping

