



Cento Universitário UNA

Sistemas de Informação

Tecnologias Emergentes

Práticas de Laboratório

Wesley Dias Maciel

2019/02



Centro Universitário UNA
Sistemas de Informação
Tecnologias Emergentes
Prática de Laboratório
Wesley Dias Maciel
2019/02

Spring Boot, Hibernate e JPA



Prática 13

Você está recebendo, juntamente com esta prática, o projeto da aplicação. Nessa versão, o projeto apresenta algumas observações sobre interação com banco de dados.

Paginação

- 1) Quando realizamos uma consulta a um banco de dados, podemos obter uma resposta com muitos registros. Para reduzir o consumo de memória, largura de banda e tempo de processamento, é apropriado dividir essa resposta em grupos menores, páginas. A implementação desta versão do projeto apresenta como dividir a resposta de uma consulta em páginas e como ler essas páginas. Foram realizadas as seguintes alterações:

Criação do método abaixo na interface `LinkRepository`:

```
@Query (value = "SELECT * FROM link", nativeQuery = true)  
public Slice<Link> getPage (Pageable pageable);
```

Criação do método abaixo na classe `ColetorService`:

```
public String buscarPagina () {  
    Slice<Link> pagina = null;  
    Pageable pageable = PageRequest.of (0, 3, Sort.by (Sort.Direction.DESC, "url"));  
  
    while (true) {  
        pagina = lr.getPage (pageable);  
        int numeroDaPagina = pagina.getNumber ();  
        int numeroDeElementosNaPagina = pagina.getNumberOfElements ();  
        int tamanhoDaPagina = pagina.getSize ();  
        System.out.println ("\n\nPágina: " + numeroDaPagina + " Número de Elementos: " +  
numeroDeElementosNaPagina + " Tamaho da Página: " + tamanhoDaPagina);  
        List<Link> links = pagina.getContent ();  
        links.forEach (System.out::println);  
        if (!pagina.hasNext ()) {  
            break;  
        }  
    }  
}
```



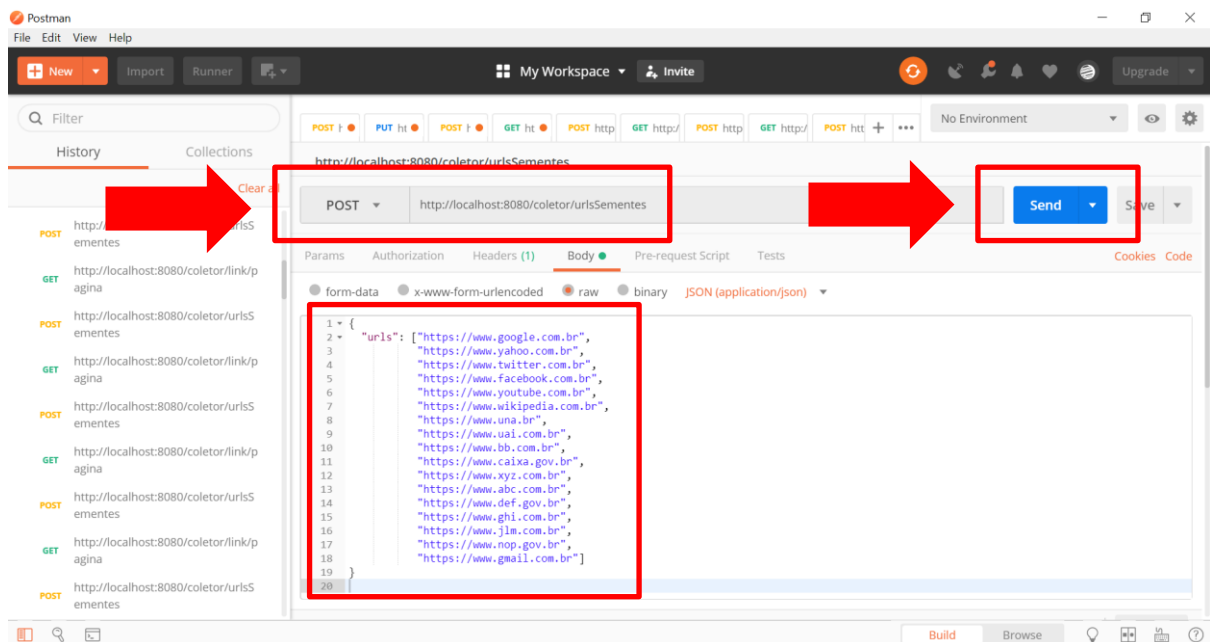
```
pageable = pagina.nextPageable ();  
}  
return "{\"resposta\": \"Ok\"}";  
}
```

Criação do método abaixo na classe Coletor:

```
// Request for: http://localhost:8080/coletor/link/pagina  
@GetMapping (value = "/link/pagina",  
             produces = MediaType.APPLICATION_JSON_UTF8_VALUE)  
public ResponseEntity listarPagina () {  
    return new ResponseEntity (cs.buscarPagina (), HttpStatus.OK);  
}
```

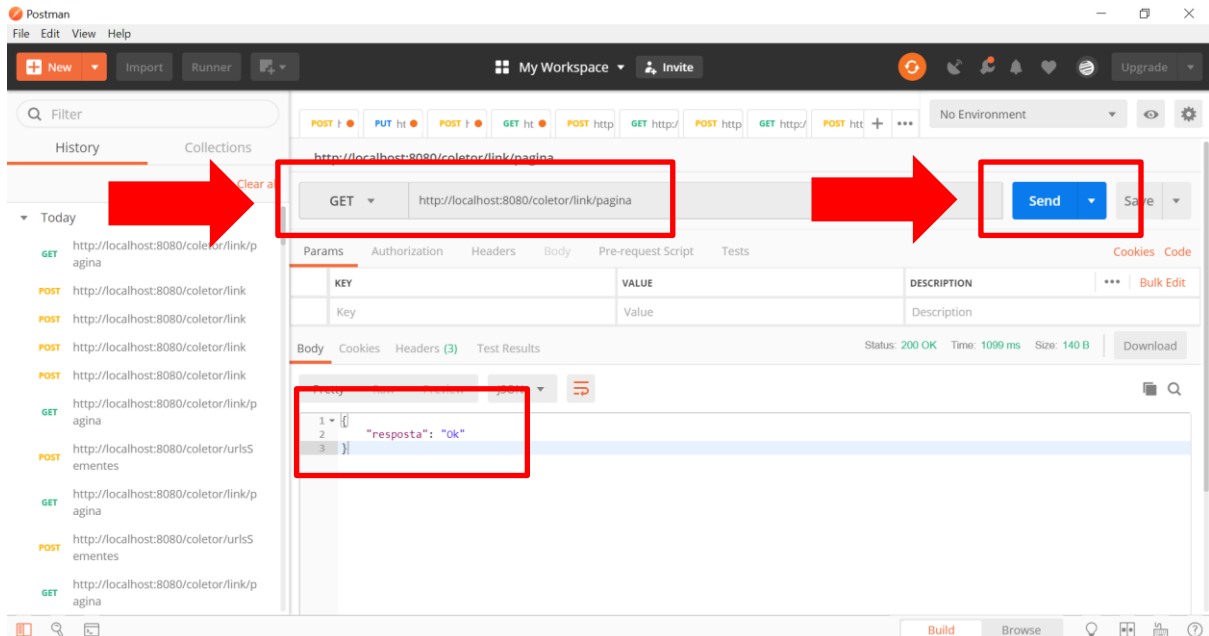
Execute o projeto e observe os resultados retornados.

Inserção de URLs sementes:

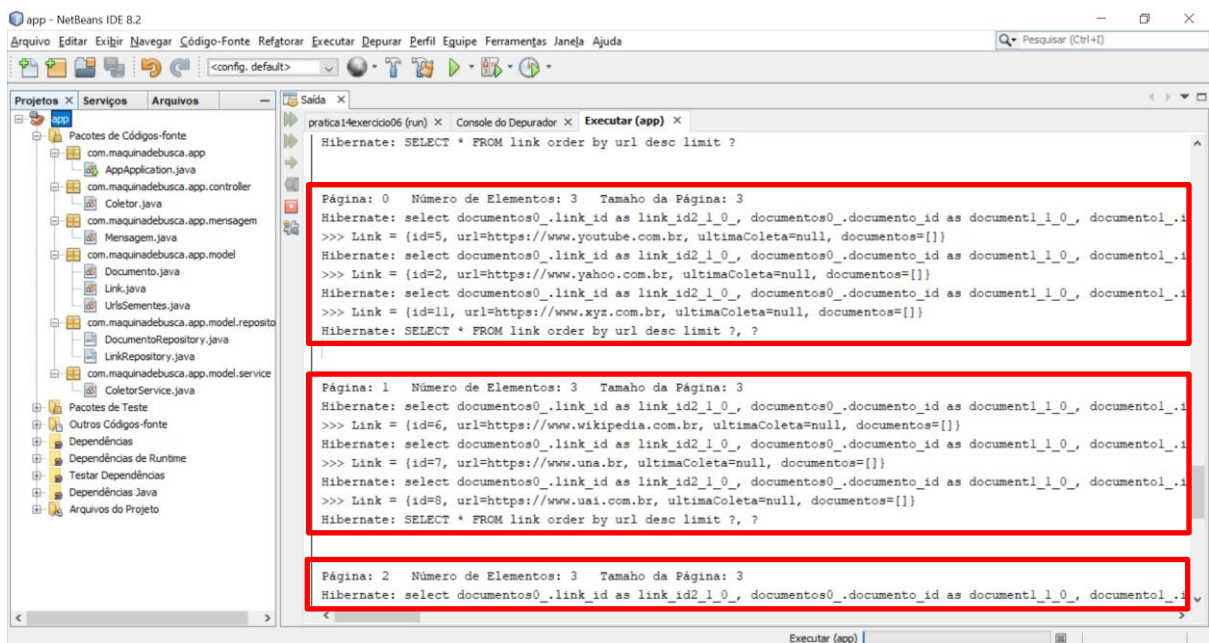




Geração e leitura das páginas:



Na versão corrente, as páginas são apresentadas no log do servidor:



- 2) Altere o projeto para que ele retorne as páginas para a aplicação cliente, neste exemplo, o Postman.



- 3) Crie uma API que permita à aplicação cliente especificar qual página deve ser retornada. O número da página a ser retornada deve ser informado na URL de requisição.
- 4) Realize a validação de parâmetros recebidos.
- 5) Analise a API do seu projeto. Sempre que necessário, faça alterações para melhoria da API, adequando-a ao padrão arquitetural REST (Representational State Transfer).
- 6) Para todos os métodos que interagem com aplicações cliente, retorne respostas com mensagens significativas para os clientes da aplicação, obedecendo os códigos adequados do protocolo HTTP.

Lista de códigos de status HTTP:

1xx Informativa

- 100 Continuar
- 101 Mudando protocolos
- 102 Processamento (WebDAV) (RFC 2518)
- 122 Pedido-URI muito longo

2xx Sucesso

- 200 OK
- 201 Criado
- 202 Aceito
- 203 não-autorizado (desde HTTP/1.1)
- 204 Nenhum conteúdo
- 205 Reset
- 206 Conteúdo parcial
- 207-Status Multi (WebDAV) (RFC 4918)

3xx Redirecionamento

- 300 Múltipla escolha
- 301 Movido
- 302 Encontrado
- 303 Consulte Outros
- 304 Não modificado
- 305 Use Proxy (desde HTTP/1.1)
- 306 Proxy Switch
- 307 Redirecionamento temporário (desde HTTP/1.1)
- 308 Redirecionamento permanente (RFC 7538[2])

4xx Erro de cliente

- 400 Requisição inválida
- 401 Não autorizado
- 402 Pagamento necessário
- 403 Proibido
- 404 Não encontrado
- 405 Método não permitido
- 406 Não Aceitável
- 407 Autenticação de proxy necessária
- 408 Tempo de requisição esgotou (Timeout)
- 409 Conflito
- 410 Gone
- 411 comprimento necessário



412 Pré-condição falhou
413 Entidade de solicitação muito grande
414 Pedido-URI Too Long
415 Tipo de mídia não suportado
416 Solicitada de Faixa Não Satisfatória
417 Falha na expectativa
418 Eu sou um bule de chá
422 Entidade improcessável (WebDAV) (RFC 4918)
423 Fechado (WebDAV) (RFC 4918)
424 Falha de Dependência (WebDAV) (RFC 4918)
425 coleção não ordenada (RFC 3648)
426 Upgrade Obrigatório (RFC 2817)
450 bloqueados pelo Controle de Pais do Windows
499 cliente fechou Pedido (utilizado em ERPs/VPsA)

5xx outros erros (erro de servidor)

500 Erro interno do servidor (Internal Server Error)
501 Não implementado (Not implemented)
502 Bad Gateway
503 Serviço indisponível (Service Unavailable)
504 Gateway Time-Out
505 HTTP Version not supported