



Centro 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 15

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

Transações com a anotação **@Transactional**

Uma transação garante que um processo seja executado totalmente com êxito ou que ele seja desfeito, caso alguma tarefa não possa ser executada. Assim, uma transação garante o princípio da atomicidade: ou todas as tarefas da transação são processadas, ou a transação é abortada (tudo ou nada). Na JPA, uma transação pode ser especificada através da anotação **@Transactional**.

Atualizações com a anotação **@Modifying**

Podemos usar a anotação **@Query** para modificar o estado do banco de dados. O método do repositório que realiza a modificação deve receber a anotação **@Modifying**. Pode ser que a modificação exija a anotação **@Transactional**, para assegurar a atomicidade na modificação.

Atualização da Data de Última Coleta de uma URL Semente

1) Foram feitas as seguintes alterações no projeto:

Criação da assinatura, protótipo, de método abaixo na interface **LinkRepository:**

@Transactional

@Modifying

@Query (value = "UPDATE link l SET l.ultimaColeta = :data WHERE l.url LIKE CONCAT ('%':host,'%')", nativeQuery = true)

int updateLastCrawlingDate (@Param ("data") LocalDateTime ultimaColeta, @Param ("host") String nomeHost);



Criação do método abaixo na classe **ColetorService**:

```
public int atualizarDataUltimaColeta (String host, LocalDateTime dataUltimaColeta) {  
    return lr.updateLastCrawlingDate (dataUltimaColeta, host);  
}
```

Criação do método abaixo na classe **Coletor**:

```
// Request for: http://localhost:8080/coletor/link/ultima/coleta/{host}/{data}  
@PutMapping (value = "/link/ultima/coleta/{host}/{data}", produces =  
MediaType.APPLICATION_JSON_UTF8_VALUE)  
public ResponseEntity atualizarUltimaColeta (@PathVariable (value = "host") String host,  
@PathVariable (value = "data") @DateTimeFormat (iso = DateTimeFormat.ISO.DATE_TIME)  
LocalDateTime data) {  
    int n = cs.atualizarDataUltimaColeta (host, data);  
    ResponseEntity resposta = new ResponseEntity (new Mensagem ("sucesso", "número de  
registros atualizados: " + n), HttpStatus.OK);  
    return resposta;  
}
```

No método acima, observe a utilização da anotação:

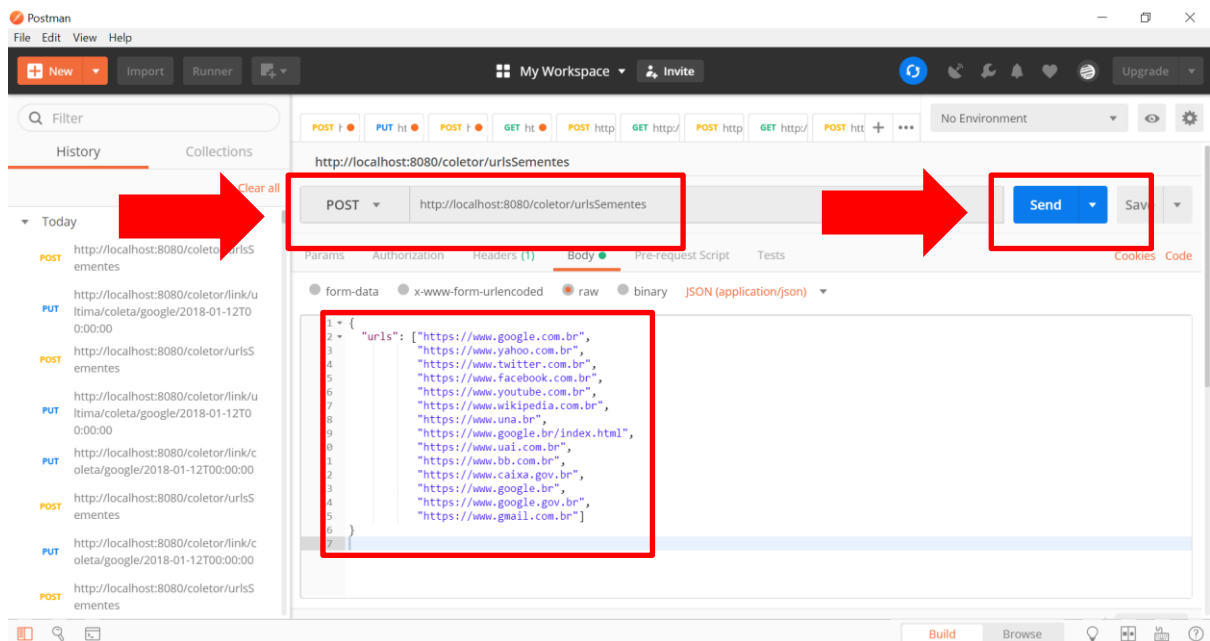
```
@DateTimeFormat (iso = DateTimeFormat.ISO.DATE_TIME)
```

Essa anotação permite realizar a atualização de uma data no formato LocalDateTime que é passada como parâmetro para o método.

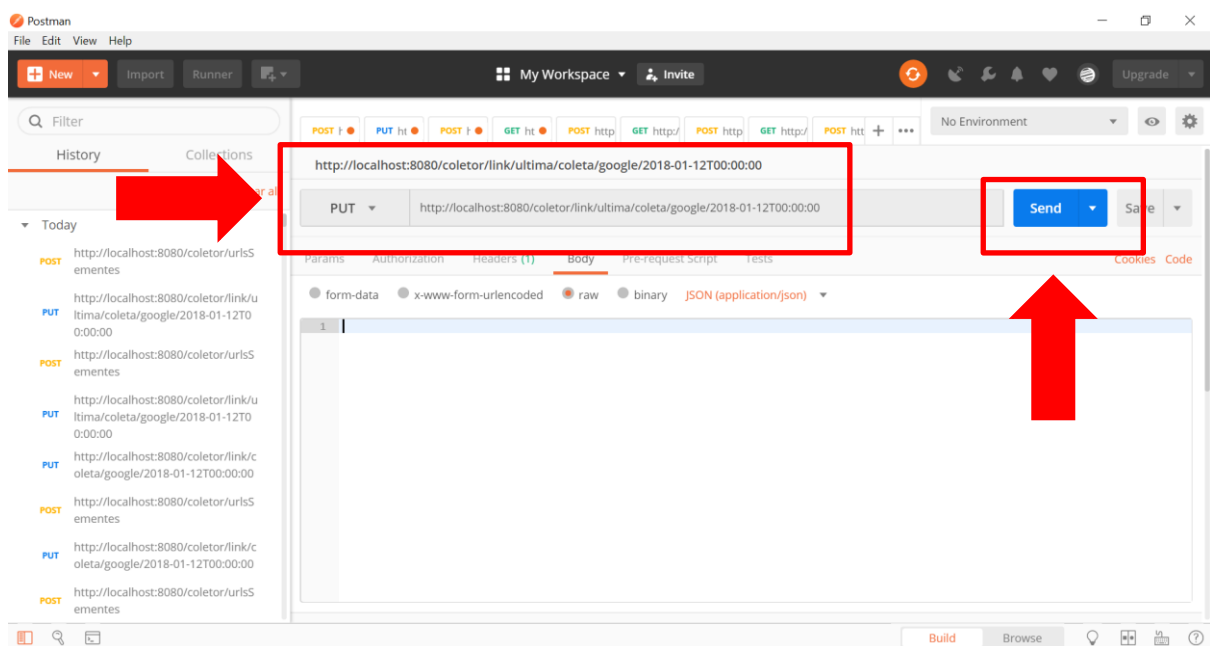


Execute o projeto e observe os resultados retornados.

Inserção de URLs sementes:

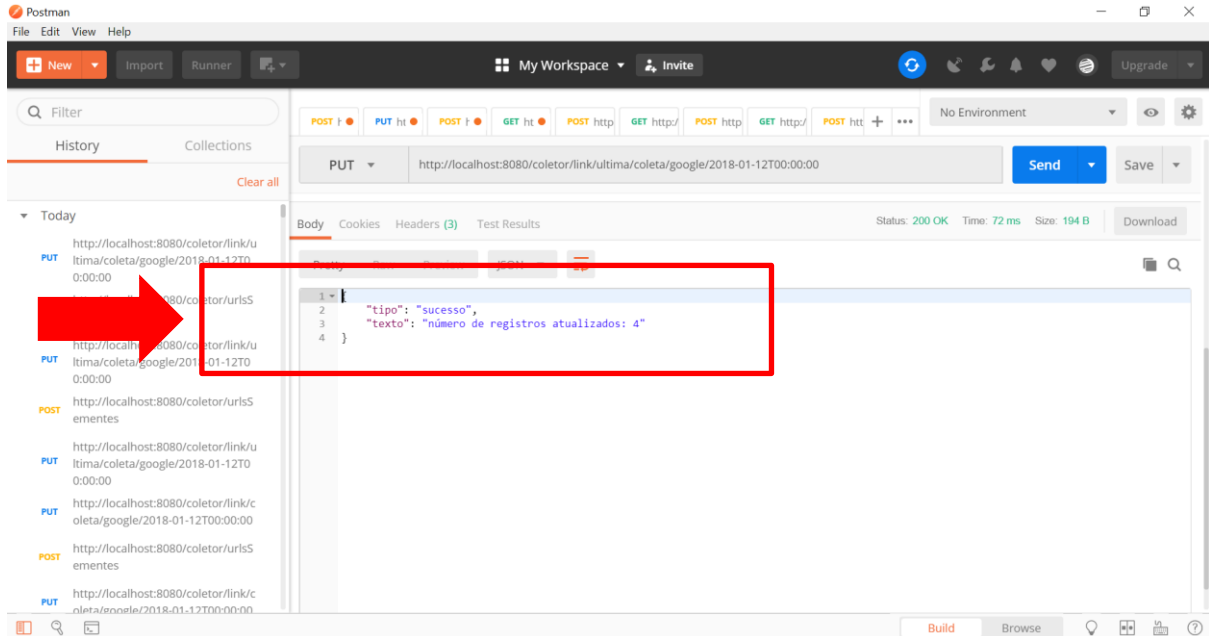


Atualização da data de última coleta:





Resposta:



- 2) Usando as anotações @Query, @Modifying, @Transactional e @DateTimeFormat, crie uma API que permita a usuários e administradores realizar atualizações no sistema. Exemplo: atualização de dados pessoais de cadastro, atualização de senhas, etc.
- 3) Analise a API do seu projeto. Sempre que necessário, faça alterações para melhoria da API, adequando-a ao padrão arquitetural REST (Representational State Transfer).
- 4) 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/VPAs)

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