



Cento Universitário UNA

Sistemas de Informação

Recuperação de Informação

Práticas de Laboratório

Wesley Dias Maciel

2019/02



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

Spring Boot, Hibernate e JPA

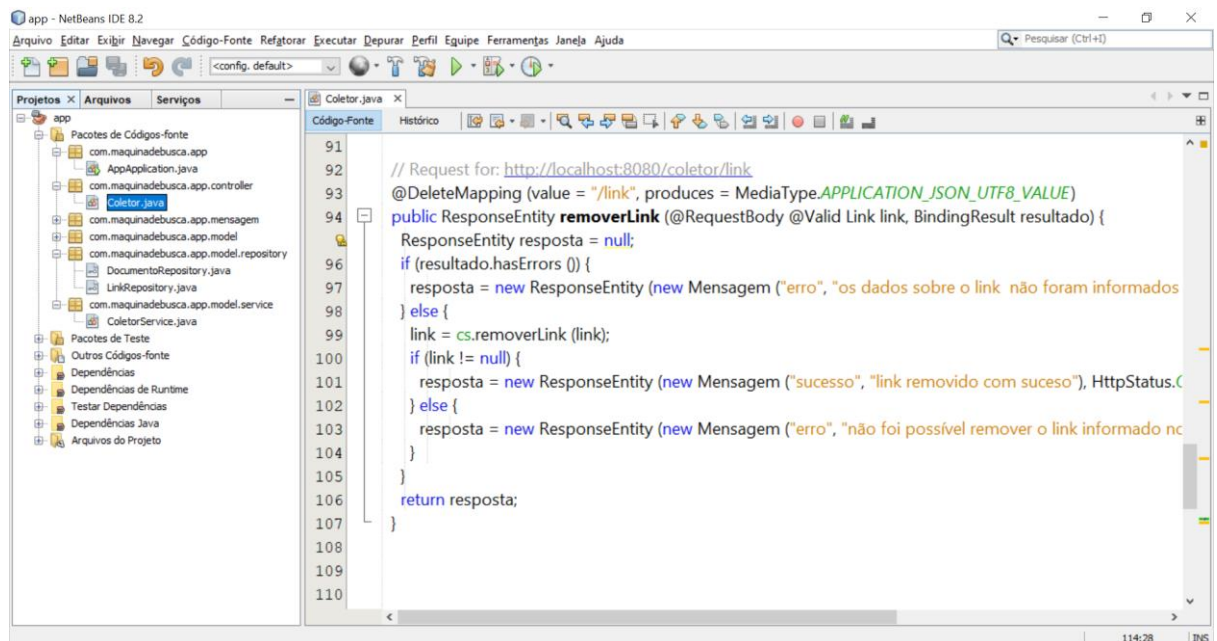


Prática 09

- 1) Você está recebendo, juntamente com esta prática, o projeto da aplicação. Nessa versão, o projeto realiza a remoção de uma URL semente informada manualmente pelo administrador e previamente inserida no banco de dados. A remoção é realizada através do método DELETE do protocolo HTTP. A remoção é realizada de duas formas: (i) informando o objeto a ser removido no corpo da requisição ou (ii) informando o identificador do objeto a ser removido na URL da requisição.

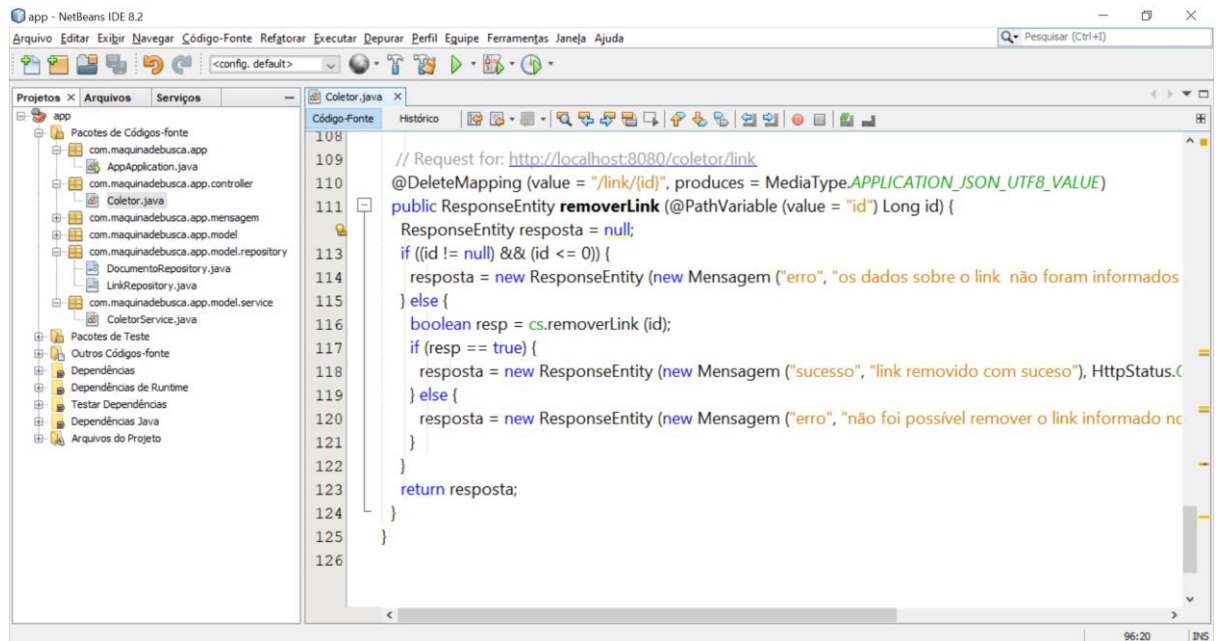
Remoção através do objeto informado:

```
// Request for: http://localhost:8080/coleitor/link
@DeleteMapping (value = "/link", produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
public ResponseEntity removerLink (@RequestBody @Valid Link link, BindingResult resultado) {
    ResponseEntity resposta = null;
    if (resultado.hasErrors ()) {
        resposta = new ResponseEntity (new Mensagem ("erro", "os dados sobre o link não foram
informados corretamente"), HttpStatus.BAD_REQUEST);
    } else {
        link = cs.removerLink (link);
        if (link != null) {
            resposta = new ResponseEntity (new Mensagem ("sucesso", "link removido com sucesso"),
HttpStatus.OK);
        } else {
            resposta = new ResponseEntity (new Mensagem ("erro", "não foi possível remover o link
informado no banco de dados"), HttpStatus.NOT_ACCEPTABLE);
        }
    }
    return resposta;
}
```



Remoção através do identificador informado:

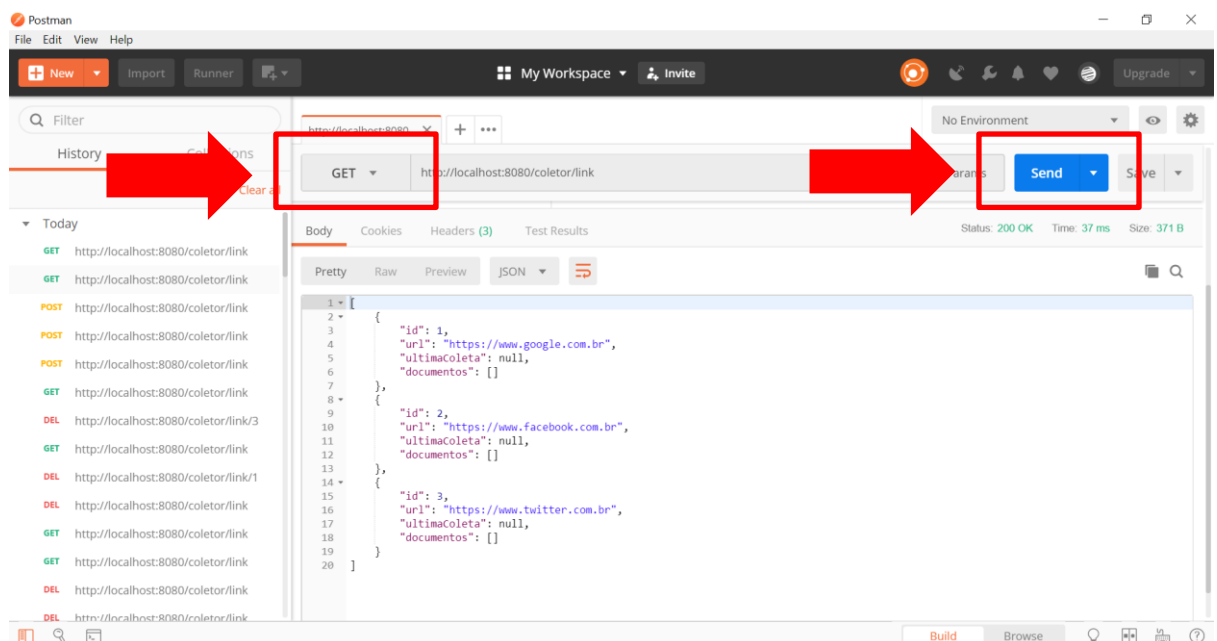
```
// Request for: http://localhost:8080/coletor/link
@DeleteMapping (value = \"/link/{id}", produces =
MediaType.APPLICATION_JSON_UTF8_VALUE)
public ResponseEntity removerLink (@PathVariable (value = "id") Long id) {
    ResponseEntity resposta = null;
    if ((id != null) && (id <= 0)) {
        resposta = new ResponseEntity (new Mensagem ("erro", "os dados sobre o link não foram
informados corretamente"), HttpStatus.BAD_REQUEST);
    } else {
        boolean resp = cs.removerLink (id);
        if (resp == true) {
            resposta = new ResponseEntity (new Mensagem ("sucesso", "link removido com sucesso"),
HttpStatus.OK);
        } else {
            resposta = new ResponseEntity (new Mensagem ("erro", "não foi possível remover o link
informado no banco de dados"), HttpStatus.NOT_ACCEPTABLE);
        }
    }
    return resposta;
}
```



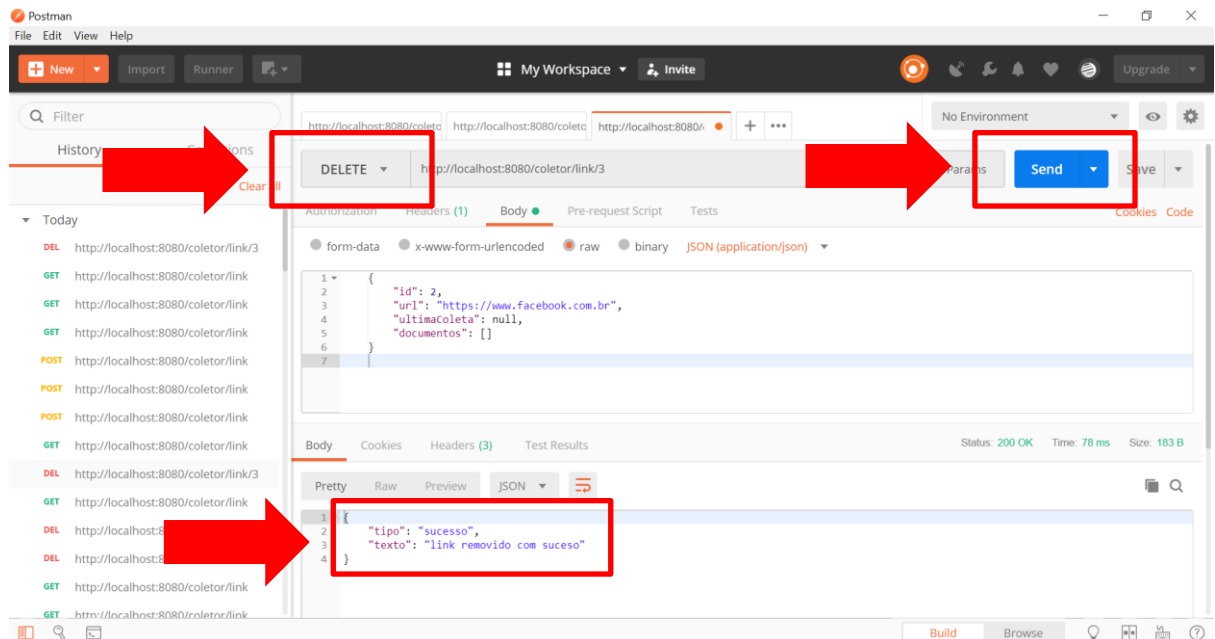
Para testar a API de remoção das URLs, use um programa para teste de APIs REST, como o Postman (<https://www.getpostman.com/>).

Execute o projeto e observe as respostas geradas pelo servidor no Postman.

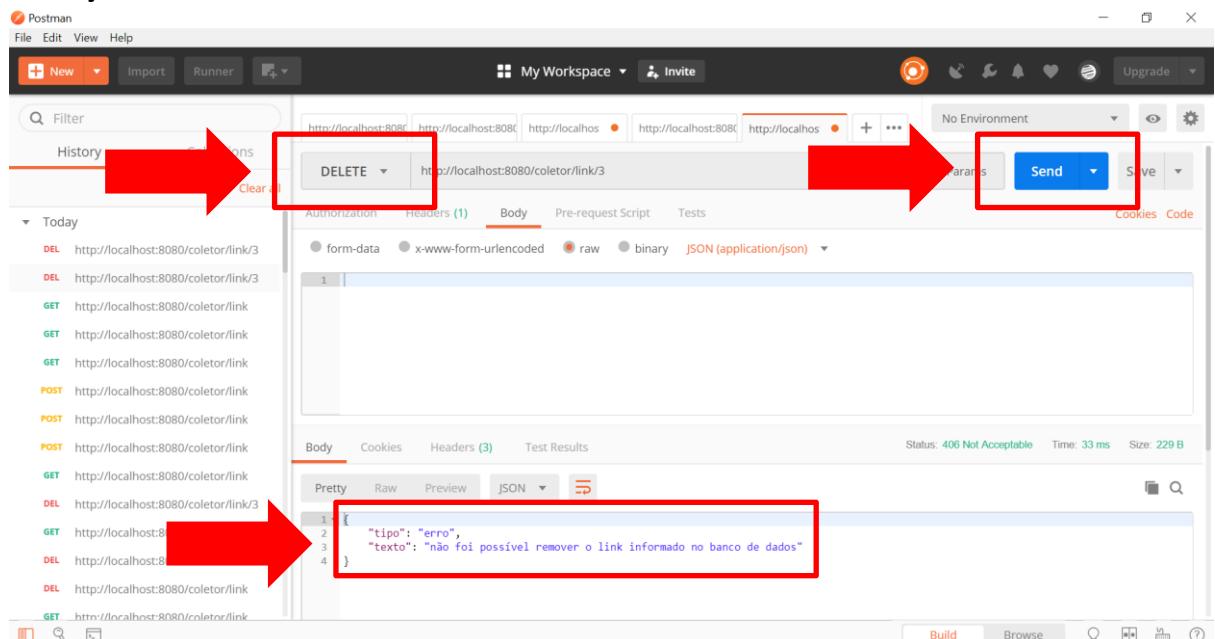
URLs sementes inseridas no banco de dados:



Remoção de uma nova URL semente através do objeto informado:



Remoção de uma nova URL semente através do identificador informado:



- 2) Em seu projeto, os administradores devem ser capazes de realizar remoções, como remover usuários, links e documentos. Por outro lado, os usuários “logados” devem ser capazes de remover suas próprias contas, cadastros. Crie a API adequada para essas operações de remoção. Realize a validação de parâmetros recebidos.



- 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



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

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