



# Cento Universitário UNA

Sistemas de Informação

Recuperação de Informação

Práticas de Laboratório

Wesley Dias Maciel

2019/01



Centro Universitário UNA  
Sistemas de Informação  
Recuperação de Informação  
Prática de Laboratório  
Wesley Dias Maciel  
2019/01

# Spring Boot, Hibernate e JPA



## Prática 08

- 1) Você está recebendo, juntamente com esta prática, o projeto da aplicação. Nessa versão, o projeto realiza a atualização de uma URL semente informada manualmente pelo administrador e previamente inserida no banco de dados. A atualização é realizada através o método PUT do protocolo HTTP. Assim como para inserir uma nova entidade no banco de dados, a atualização também é realizada através do método **save ()** do repositório JPA. Quando um objeto é recebido no servidor sem um ID, o método **save ()** insere o objeto no banco de dados. Nesse caso, o método **save ()** entende que é uma nova entidade para o banco de dados. Por outro lado, quando um objeto é recebido no servidor com um ID, o método **save ()** atualiza o objeto no banco de dados. Nesse caso, o método **save ()** entende que é uma atualização de uma entidade já existente no banco de dados. Quando o administrador atualiza uma entidade Link para que ela tenha o campo URL igual ao de uma outra já inserida no banco de dados, o servidor retorna uma mensagem de erro. Quando o administrador requisita uma atualização de uma entidade Link, mas não informa o campo URL dessa entidade, o servidor também retorna uma mensagem de erro.

```
// Request for: http://localhost:8080/coletor/link
@PutMapping (value = "/link", produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
public ResponseEntity atualizarLink (@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.atualizarLink (link);
        if ((link != null) && (link.getId () > 0)) {
            resposta = new ResponseEntity (link, HttpStatus.OK);
        } else {
            resposta = new ResponseEntity (new Mensagem ("erro", "não foi possível atualizar o link
informado no banco de dados"), HttpStatus.NOT_ACCEPTABLE);
        }
    }
    return resposta;
}
```

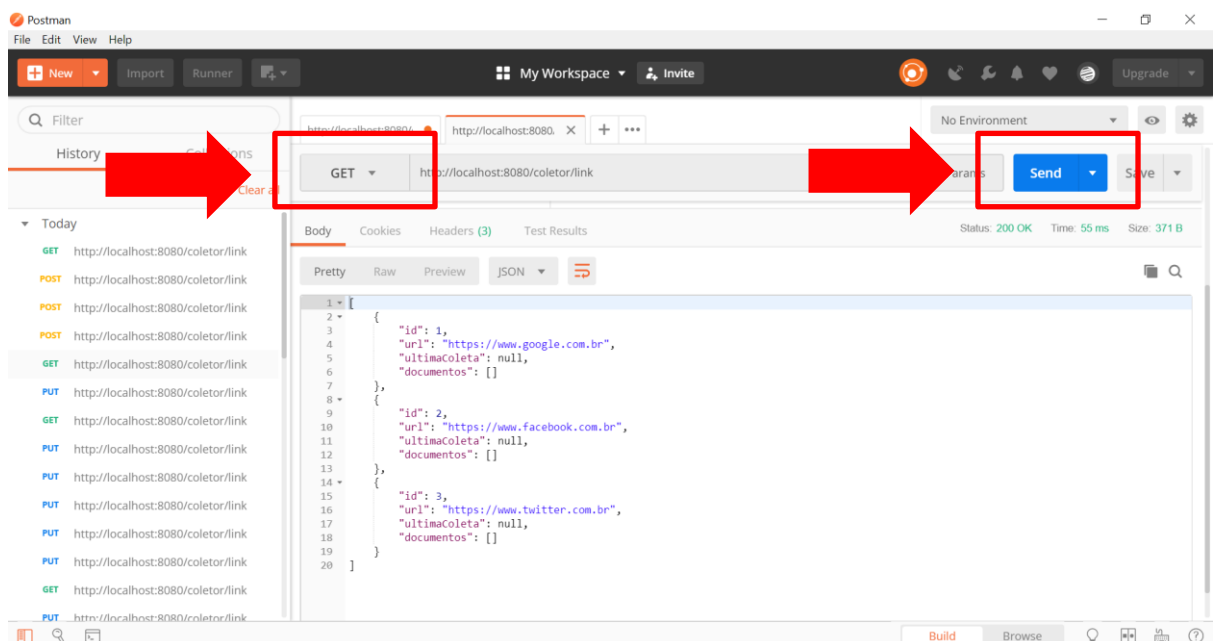


```
73  
74 // Request for: http://localhost:8080/coleto/link  
75 @PutMapping(value = "/link", produces = MediaType.APPLICATION_JSON_UTF8_VALUE)  
76 public ResponseEntity atualizarLink (@RequestBody @Valid Link link, BindingResult resultado) {  
77     ResponseEntity resposta = null;  
78     if (resultado.hasErrors()) {  
79         resposta = new ResponseEntity (new Mensagem ("erro", "os dados sobre o link não foram informados c  
80     } else {  
81         link = cs.atualizarLink (link);  
82         if ((link != null) && (link.getId () > 0)) {  
83             resposta = new ResponseEntity (link, HttpStatus.OK);  
84         } else {  
85             resposta = new ResponseEntity (new Mensagem ("erro", "não foi possível atualizar o link informado no  
86         }  
87     }  
88     return resposta;  
89 }  
90 }  
91 }
```

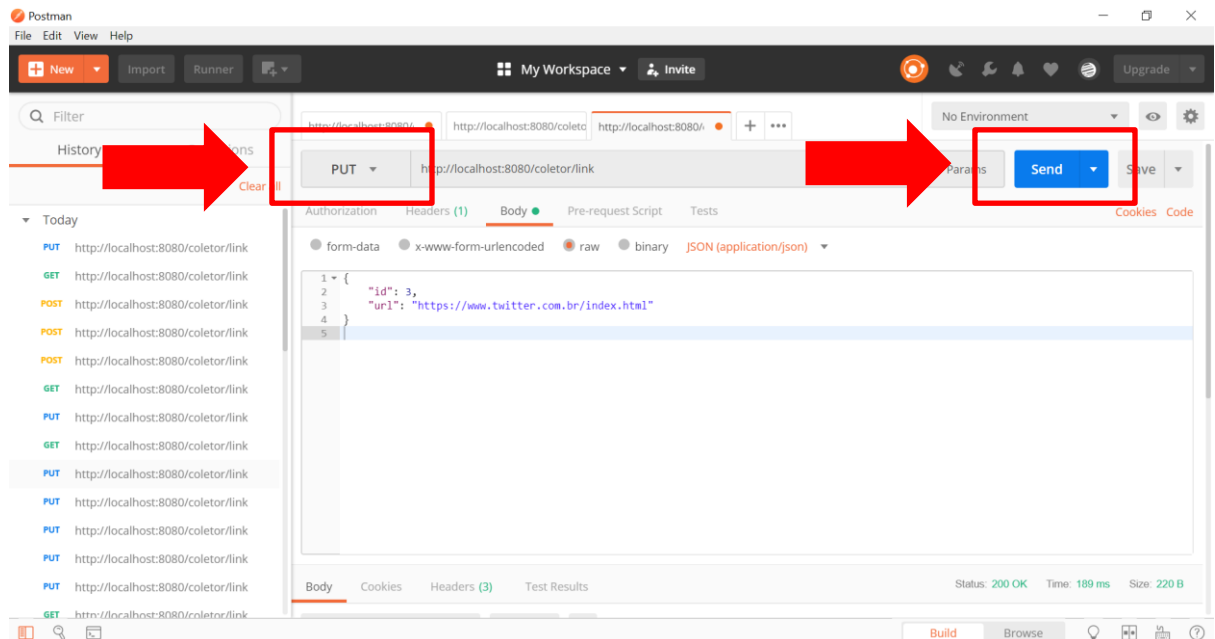
Para testar a API de atualização das URLs sementes com validação, use um programa para teste de APIs REST, como o Postman (<https://www.getpostman.com/>).

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

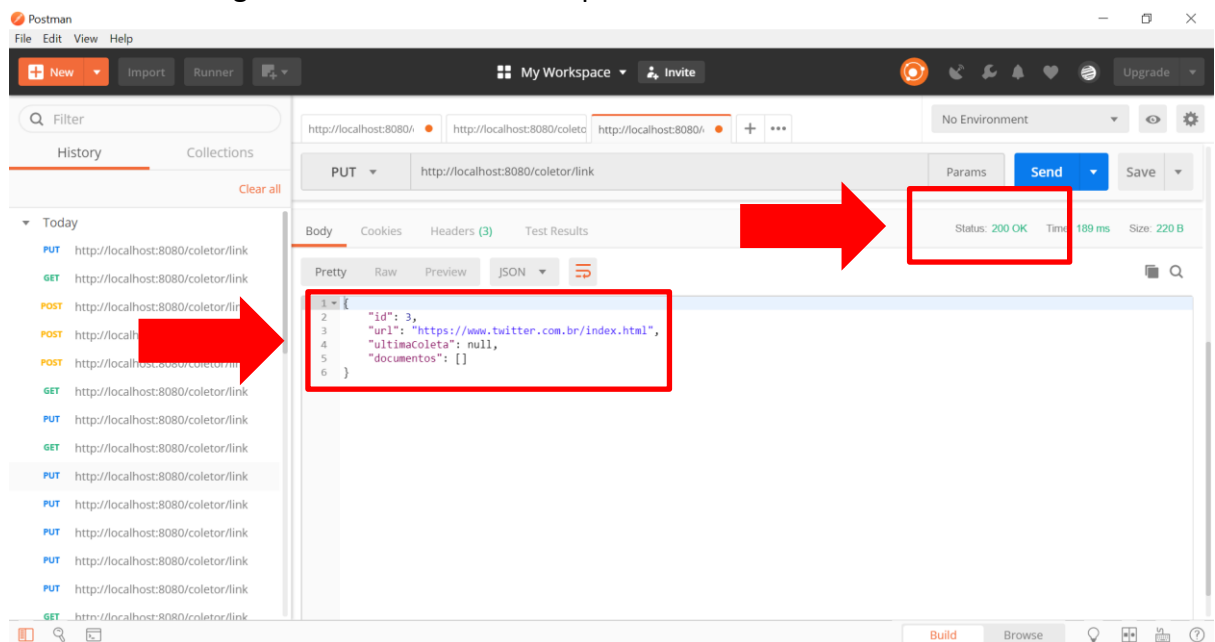
URLs sementes inseridas no banco de dados:



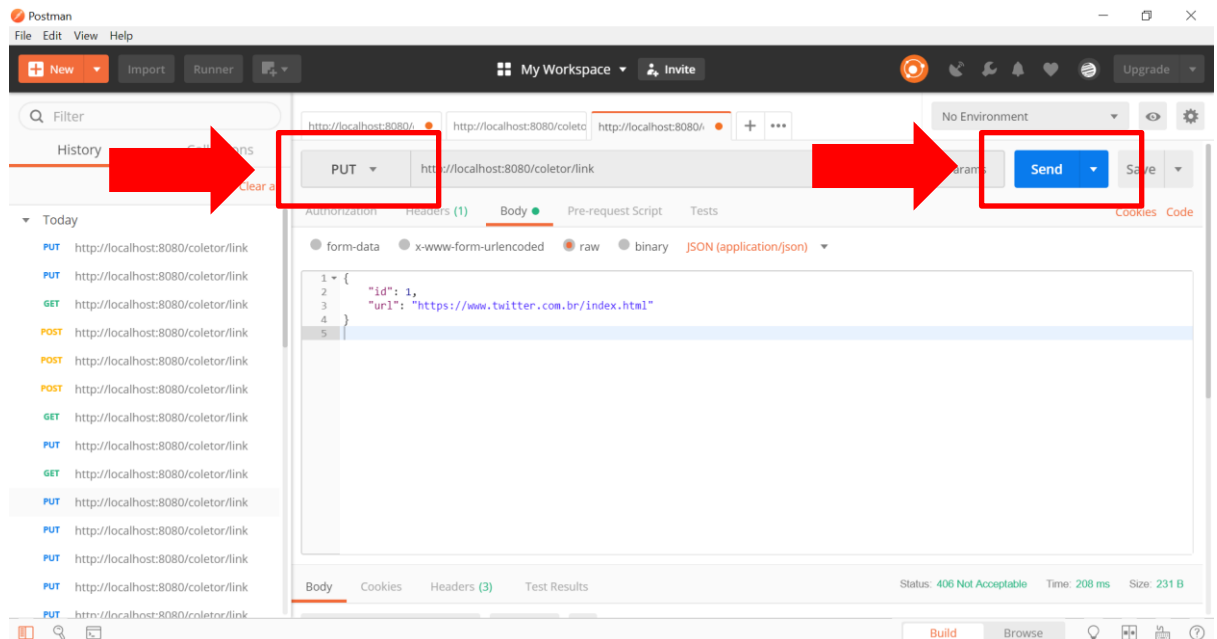
Atualização de uma nova URL semente:



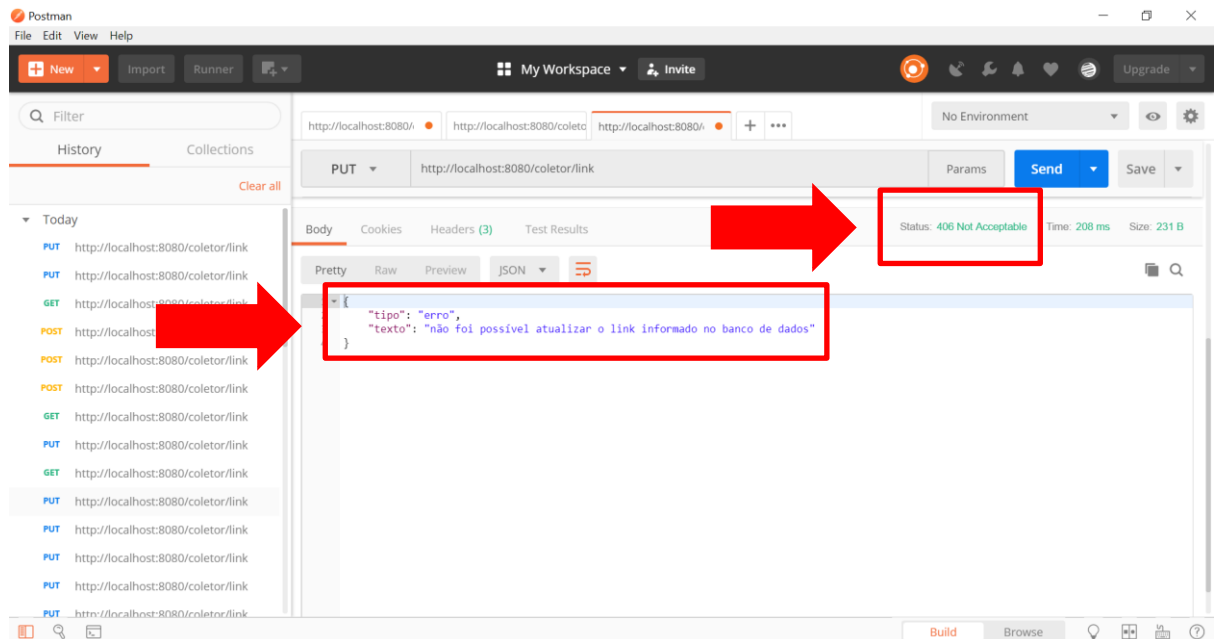
Observe a mensagem de sucesso retornada pelo servidor:



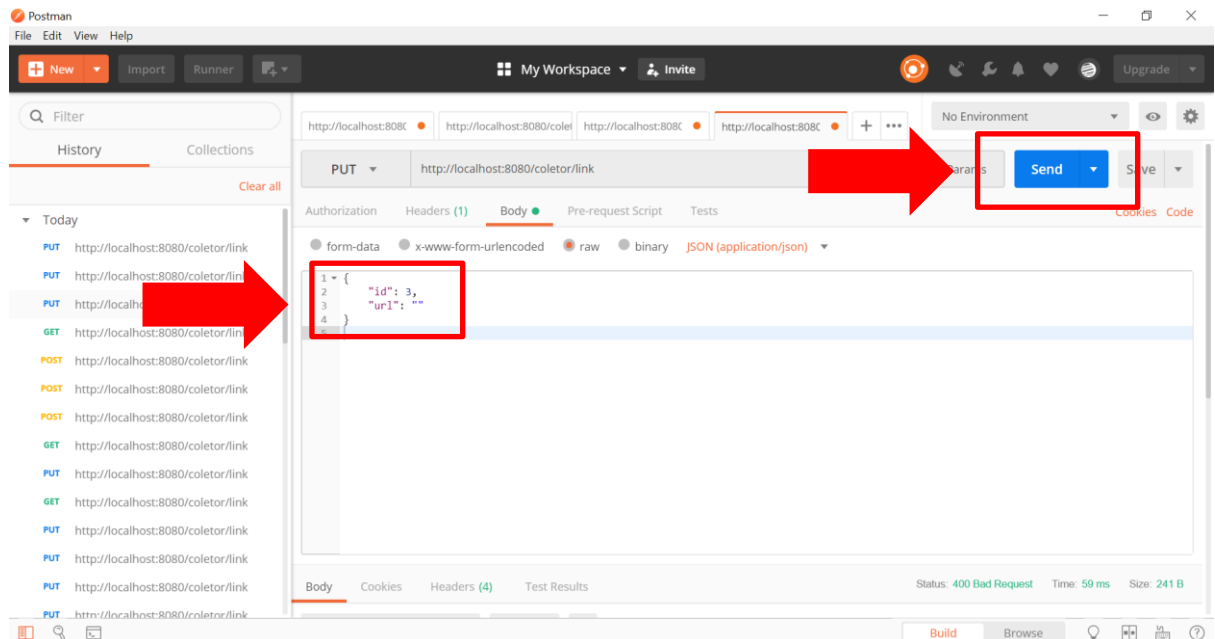
Atualização de uma URL semente, fazendo com que ela tenha o mesmo campo URL de uma outra já inserida no banco de dados:



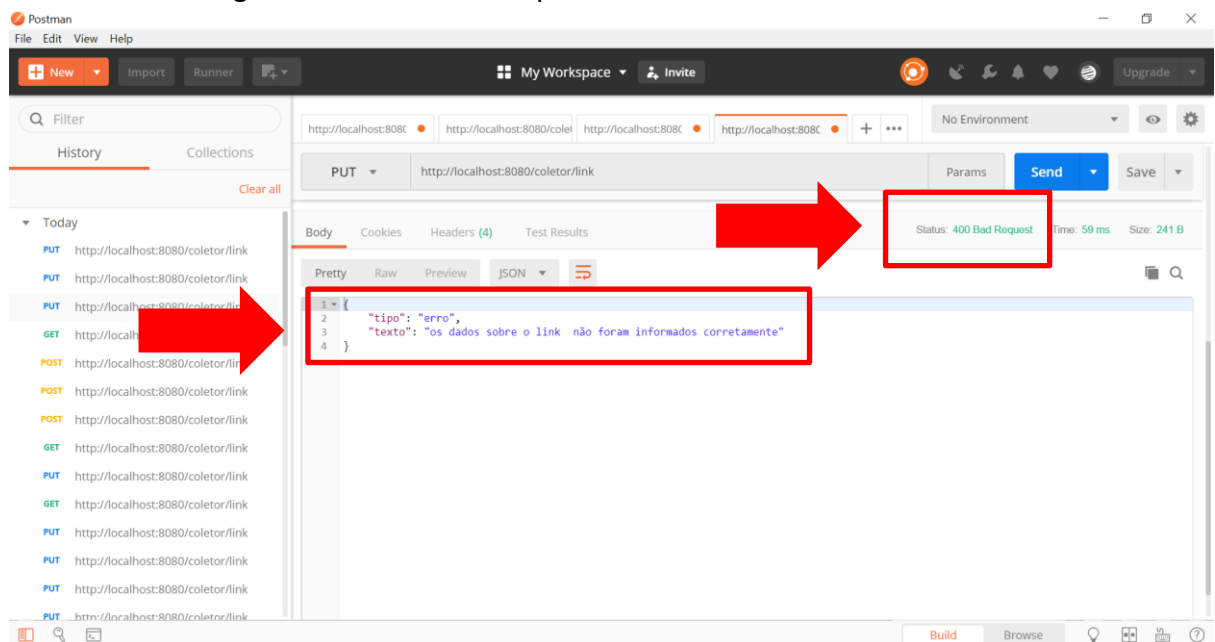
Observe a mensagem de erro retornada pelo servidor:



Atualização de uma URL semente sem informar o campo URL:



Observe a mensagem de erro retornada pelo servidor:



- 2) Em seu projeto, crie uma API para cadastro dos usuários e administradores da máquina de busca. Conceda os privilégios adequados para uso da máquina de busca a cada um desses tipos de clientes. Permita que sejam realizadas inserções e atualizações, conforme o nível de privilégio dado ao tipo de cliente. Realize a validação de parâmetros recebidos. Para todos os métodos que interagem com os clientes, retorne respostas com mensagens significativas, 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/VPSS)

### 5xx outros erros (erro de servidor)

- 500 Erro interno do servidor (Internal Server Error)
- 501 Não implementado (Not implemented)





Centro Universitário UNA  
Sistemas de Informação  
Recuperação de Informação  
Prática de Laboratório  
Wesley Dias Maciel  
2019/01

502 Bad Gateway  
503 Serviço indisponível (Service Unavailable)  
504 Gateway Time-Out  
505 HTTP Version not supported