



# Centro Universitário UNA

Sistemas de Informação

Tecnologias Emergentes

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

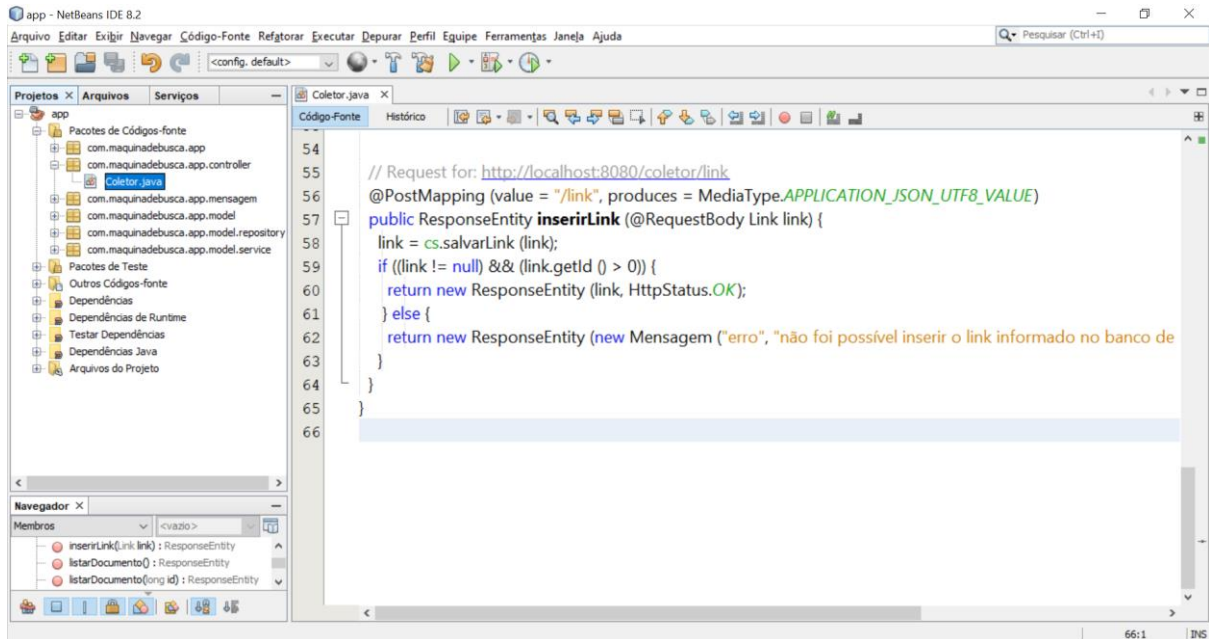
# Spring Boot, Hibernate e JPA



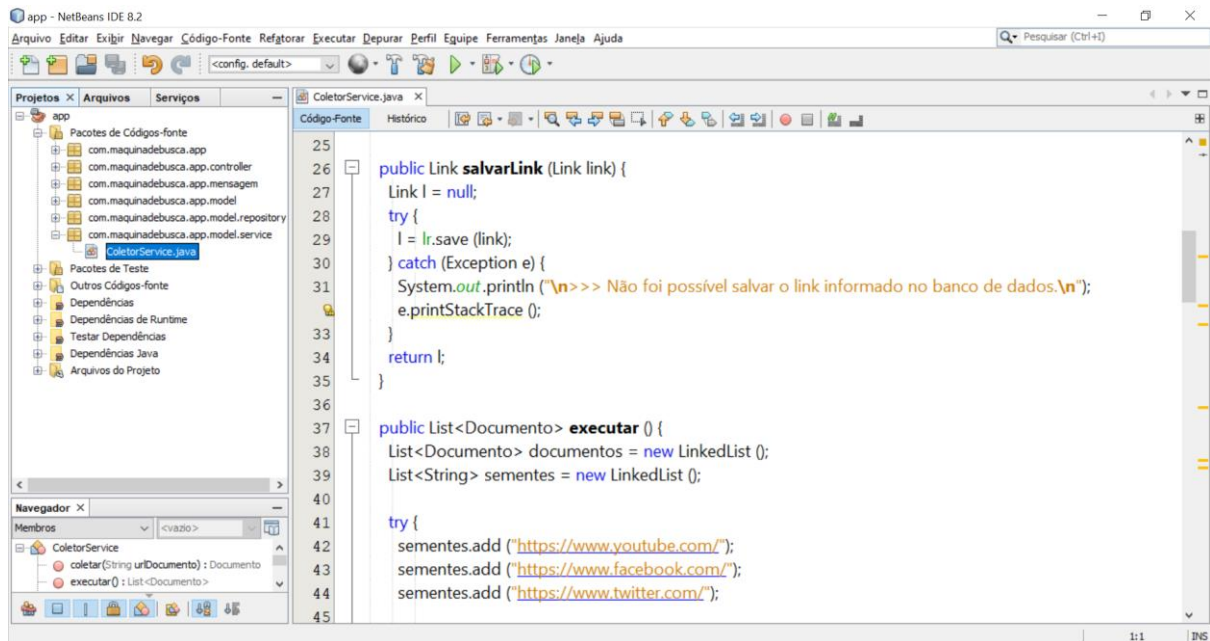
## Prática 06

- 1) Você está recebendo, juntamente com esta prática, o projeto da aplicação. Nesta versão, o projeto permite que o administrador do sistema informe manualmente cada URL semente a ser coletada. As URL sementes devem ser informadas para o sistema no formato JSON. As URLs informadas devem ser distintas. Caso uma URL repetida seja informada pelo administrador, uma mensagem de erro no formato JSON é disparada pelo sistema. Para testar a API de cadastro das URLs sementes, use um programa para teste de APIs REST, como o Postman (<https://www.getpostman.com/>).

Abra o projeto no IDE Netbeans. Observe que o método `inserirLink ()` foi adicionado à classe `Coletor`.

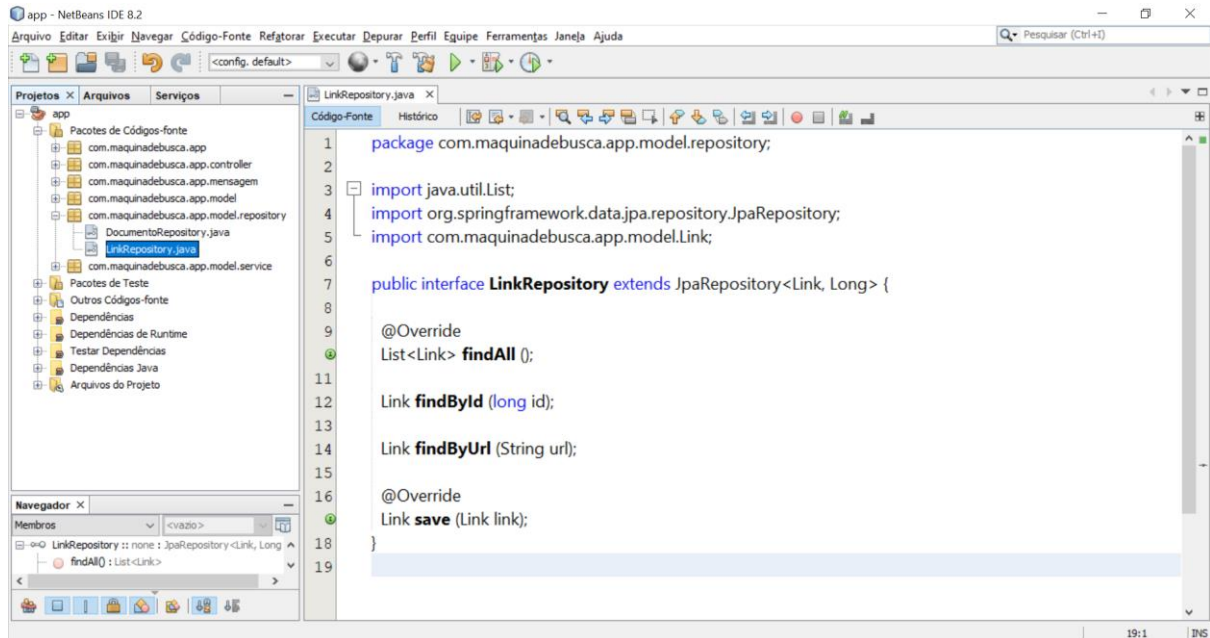


Observe também que o método `salvarLink ()` foi acrescentado à classe `ColetorService`.



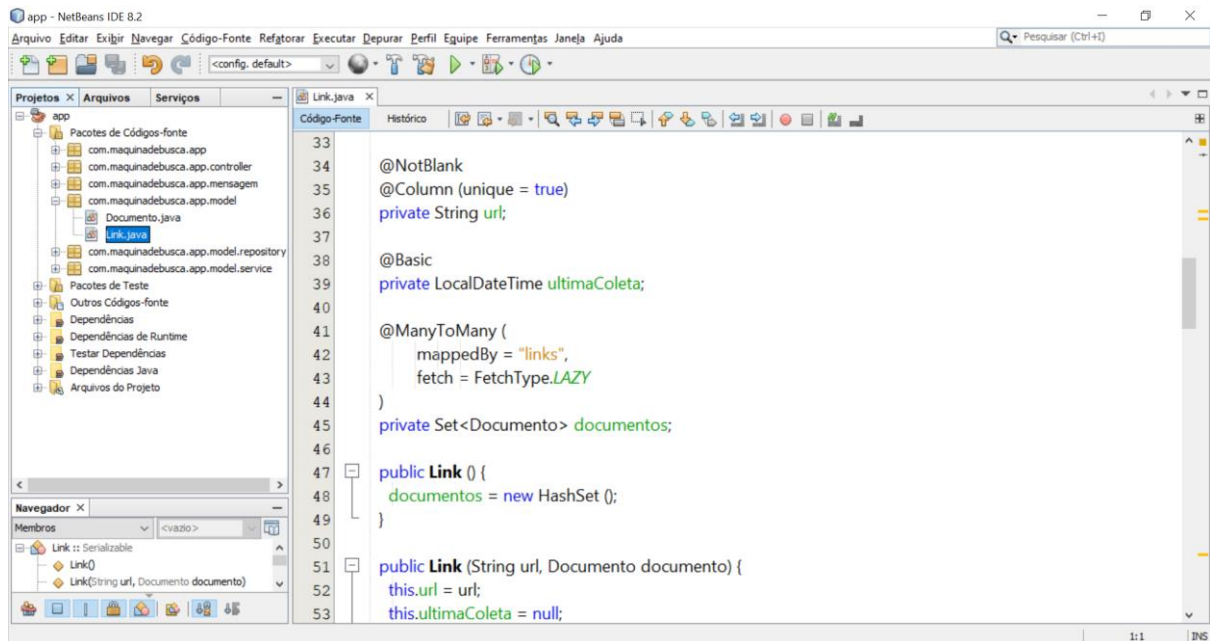
```
25  
26 public Link salvarLink (Link link) {  
27     Link l = null;  
28     try {  
29         l = lr.save (link);  
30     } catch (Exception e) {  
31         System.out.println ("\n>>> Não foi possível salvar o link informado no banco de dados.\n");  
32         e.printStackTrace ();  
33     }  
34     return l;  
35 }  
36  
37 public List<Documento> executar () {  
38     List<Documento> documentos = new LinkedList ();  
39     List<String> sementes = new LinkedList ();  
40  
41     try {  
42         sementes.add ("https://www.youtube.com/");  
43         sementes.add ("https://www.facebook.com/");  
44         sementes.add ("https://www.twitter.com/");  
45     }
```

O método save (Link link) foi acrescentado à classe LinkRepository.

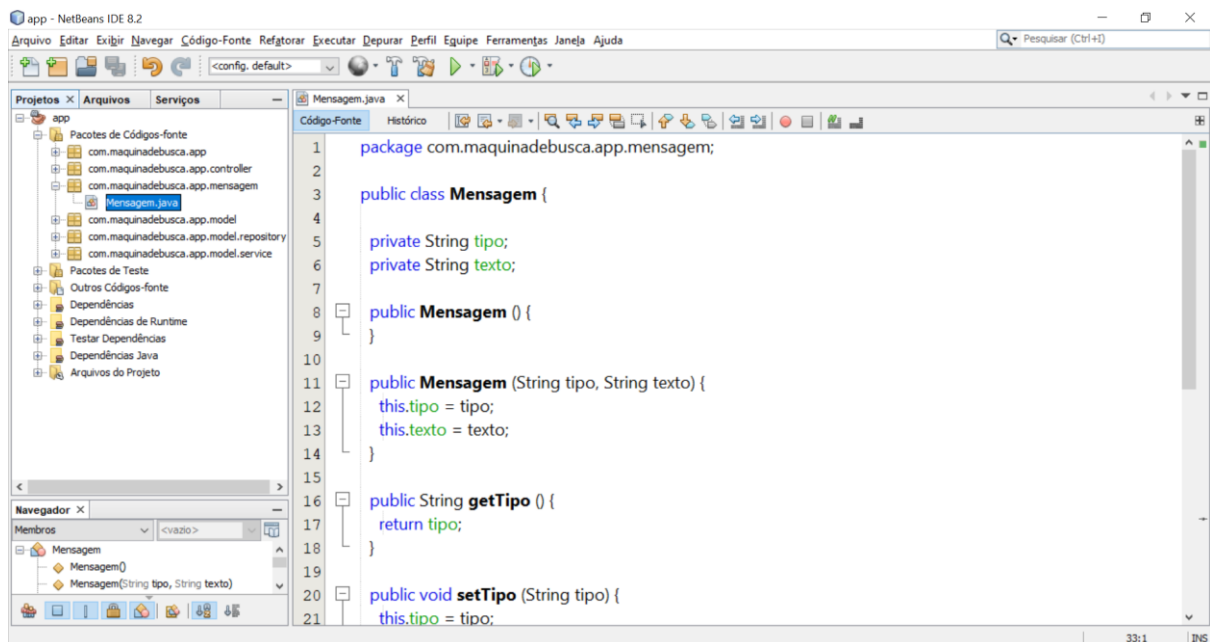


```
1 package com.maquinadebusca.app.model.repository;  
2  
3 import java.util.List;  
4 import org.springframework.data.jpa.repository.JpaRepository;  
5 import com.maquinadebusca.app.model.Link;  
6  
7 public interface LinkRepository extends JpaRepository<Link, Long> {  
8  
9     @Override  
10     List<Link> findAll ();  
11  
12     Link findById (long id);  
13  
14     Link findByUrl (String url);  
15  
16     @Override  
17     Link save (Link link);  
18  
19 }
```

O atributo url da classe Link foi marcado com a anotação @Column (unique = true). Isso faz com que esse campo da tabela no banco de dados aceite apenas valores distintos.



Também foram criados o pacote mensagem e a classe Mensagem, para envio de mensagens personalizadas para o administrador do sistema.

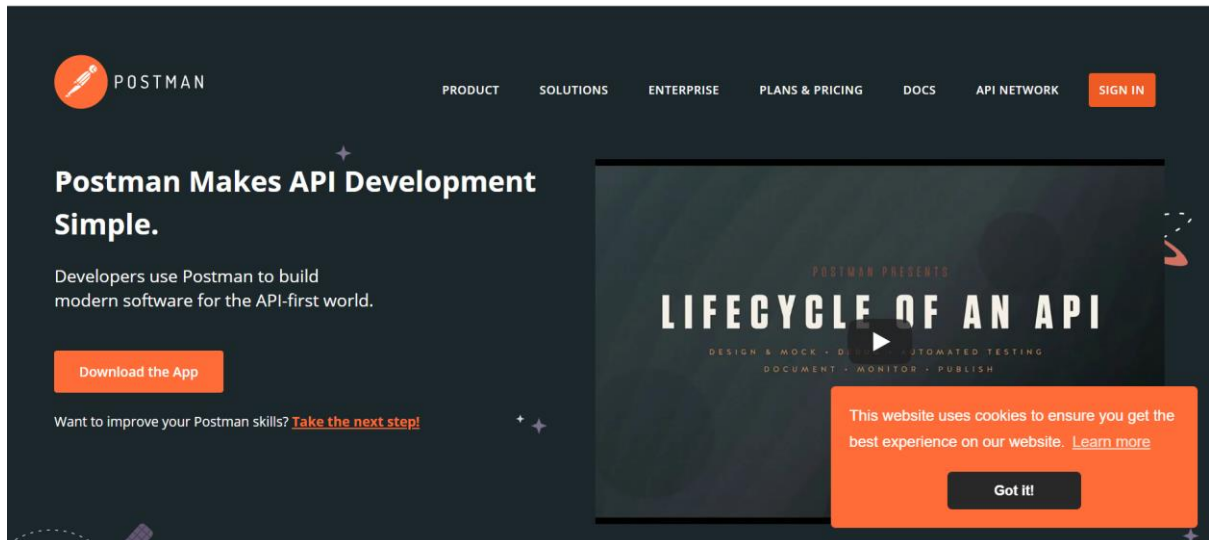


Execute o servidor, para que ele possa receber requisições de clientes. Crie sua conta no Postman. Baixe o aplicativo e instale-o em seu computador, ou instale a extensão para o navegador Chrome.

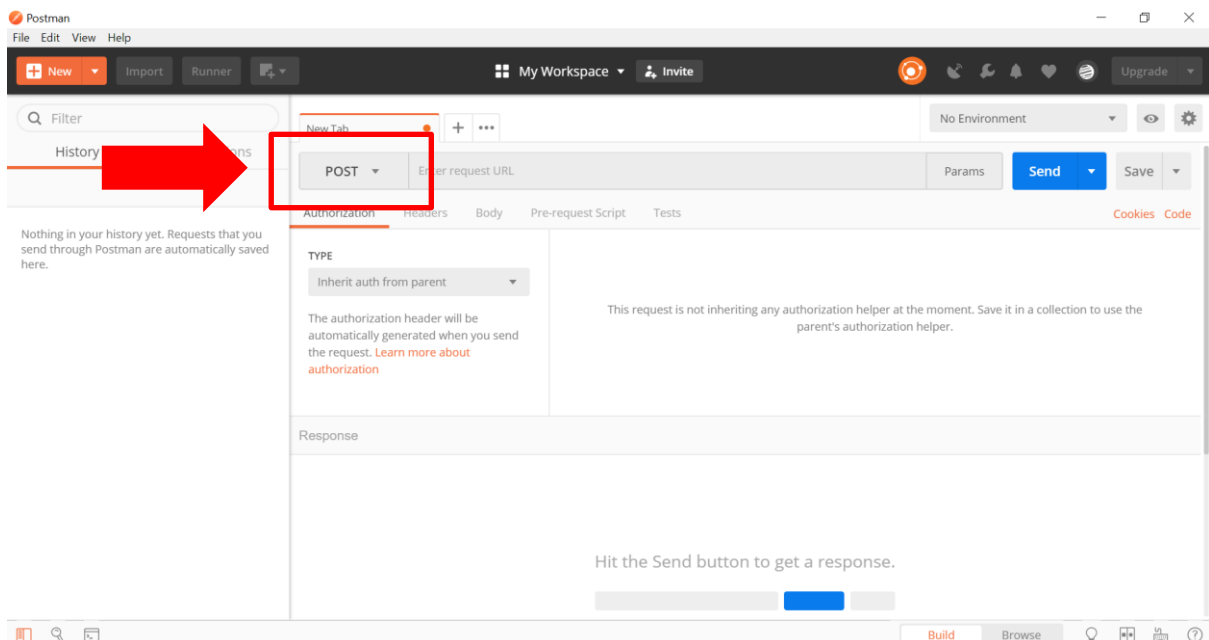


Centro Universitário UNA  
Sistemas de Informação  
Recuperação de Informação  
Prática de Laboratório  
Wesley Dias Maciel  
2019/02

https://www.getpostman.com



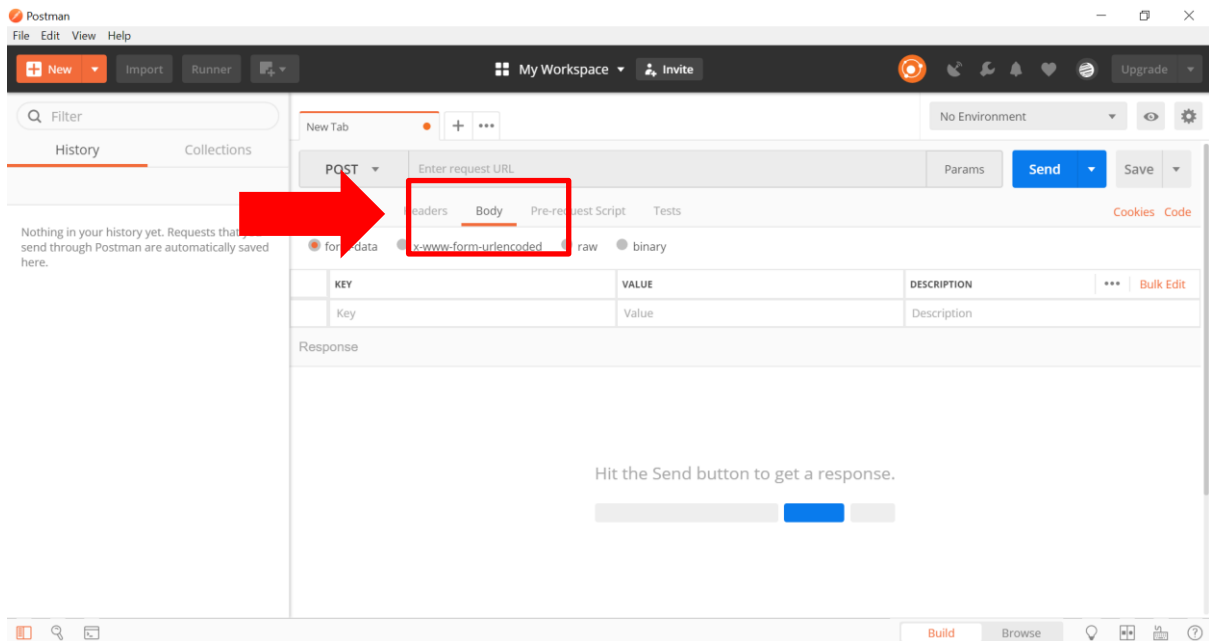
No Postman, selecione o método POST do protocolo HTTP.



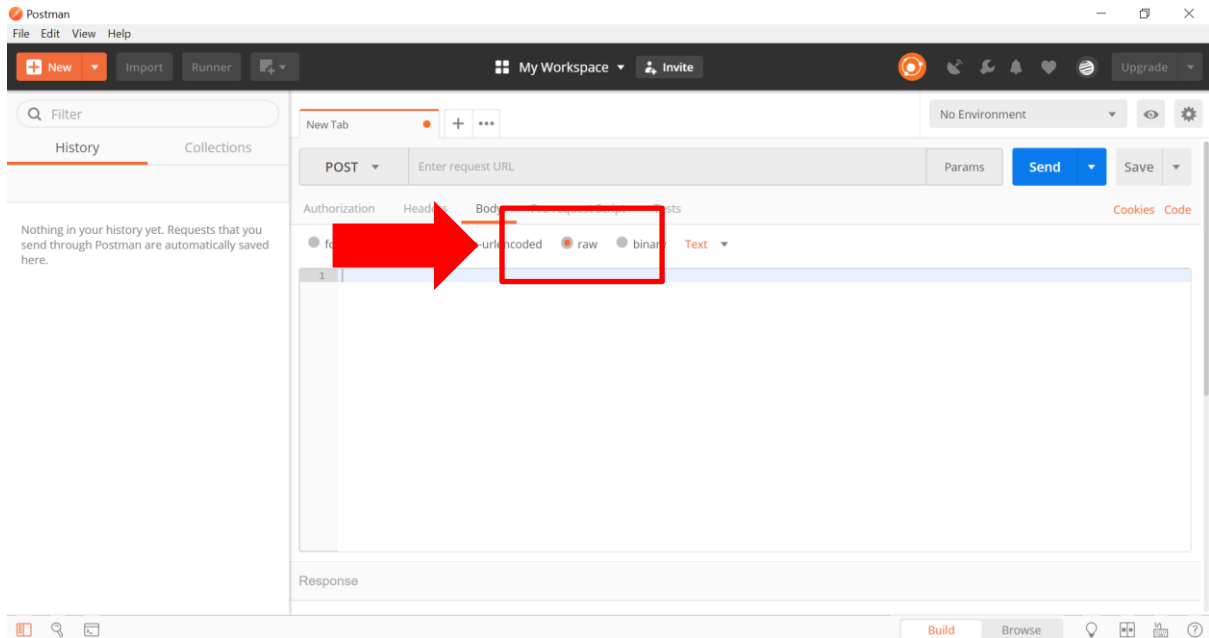
Em seguida, clique em Body.



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



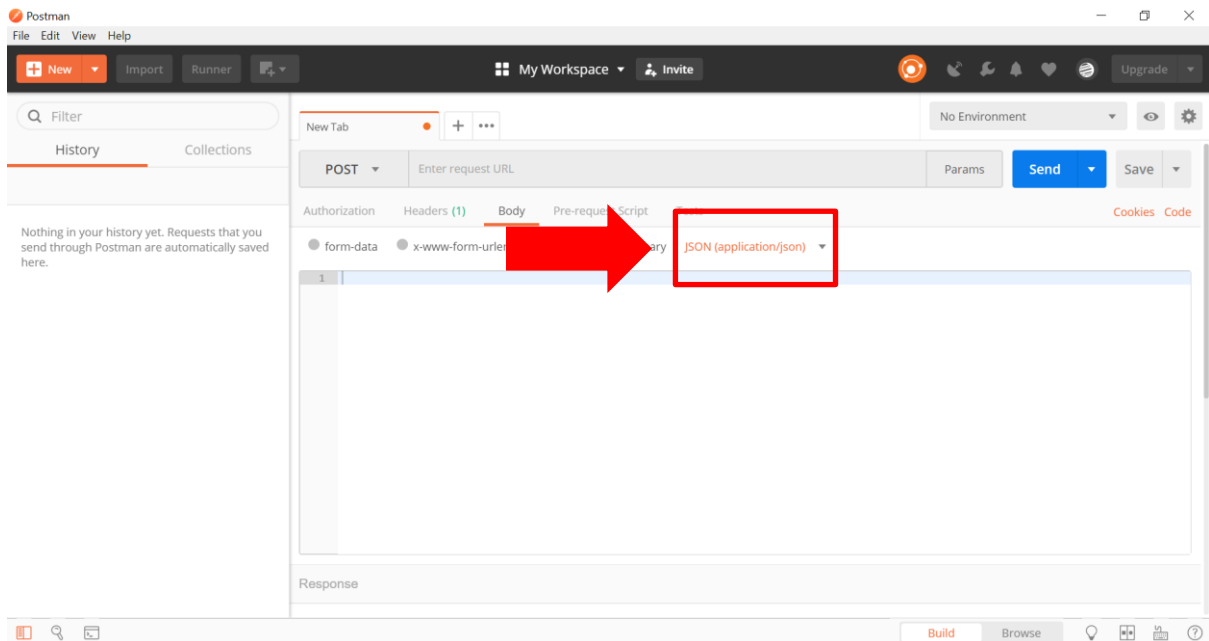
Depois, clique em raw.



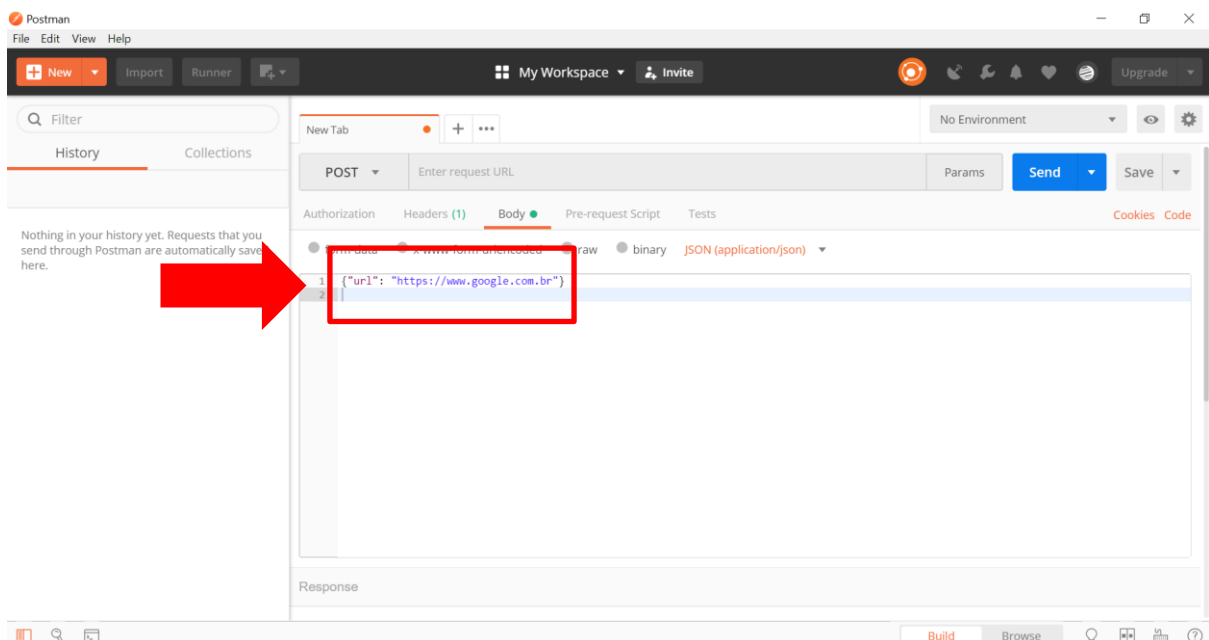
Em seguida, escolha o format JSON.



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



Na área para escrita do corpo da mensagem, informe: {"url": "https://www.google.com.br"}.

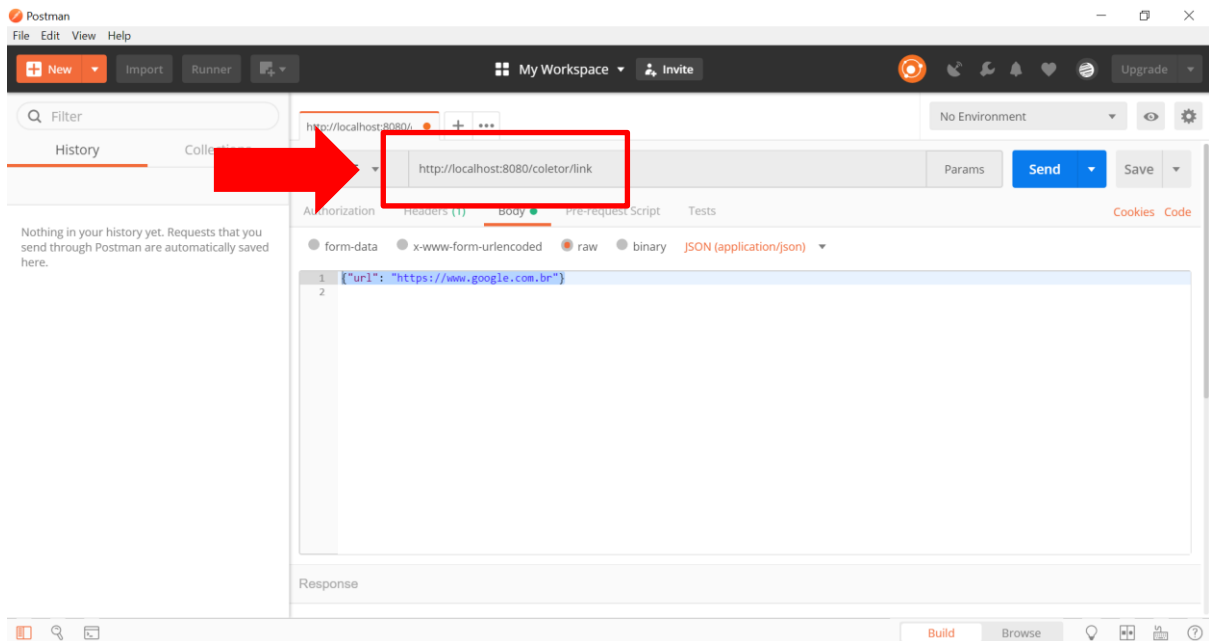


No campo para URL, informe: http://localhost:8080/coletor/link.

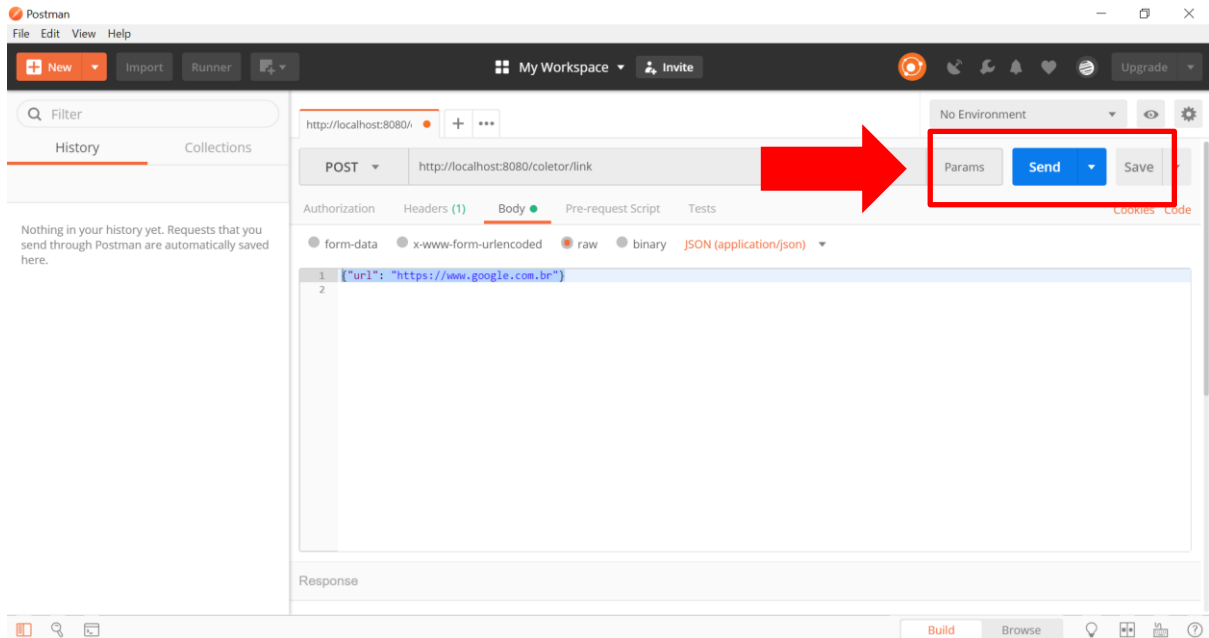




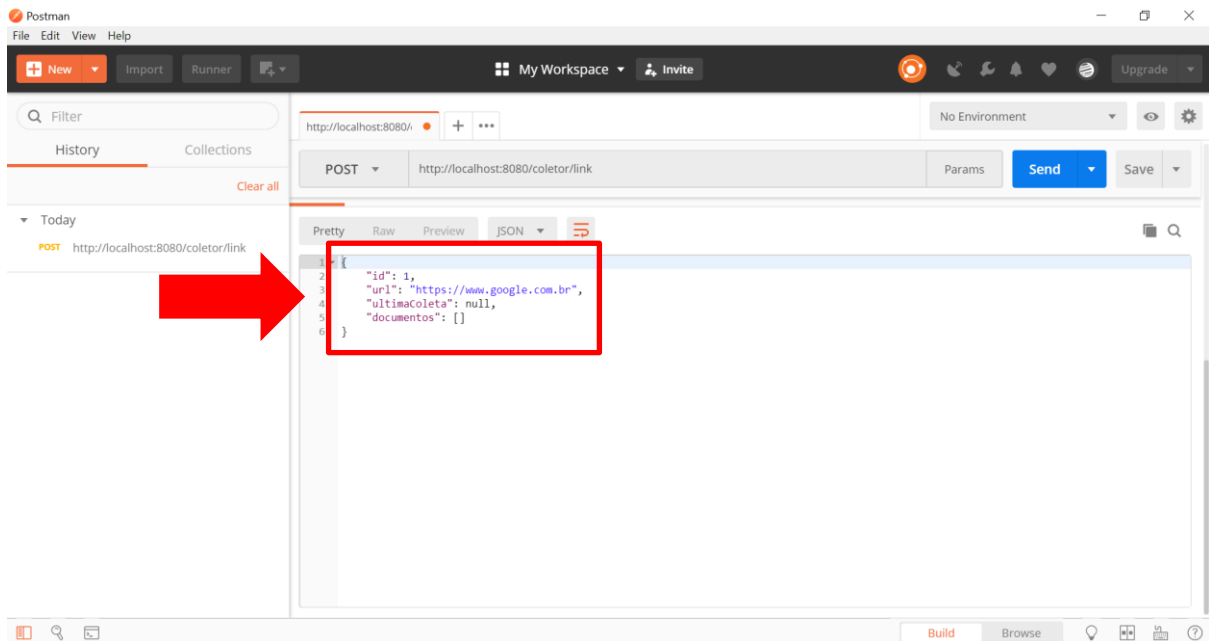
Centro Universitário UNA  
Sistemas de Informação  
Recuperação de Informação  
Prática de Laboratório  
Wesley Dias Maciel  
2019/02



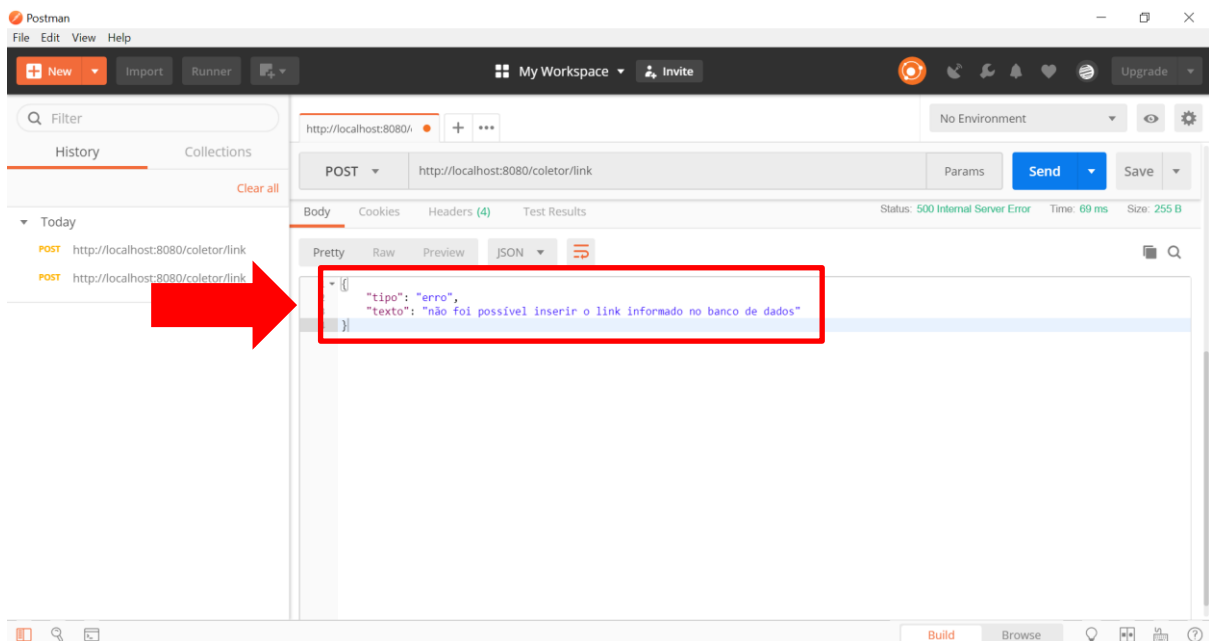
Clique no botão Send.



Observe a resposta no format JSON retornada pelo servidor.



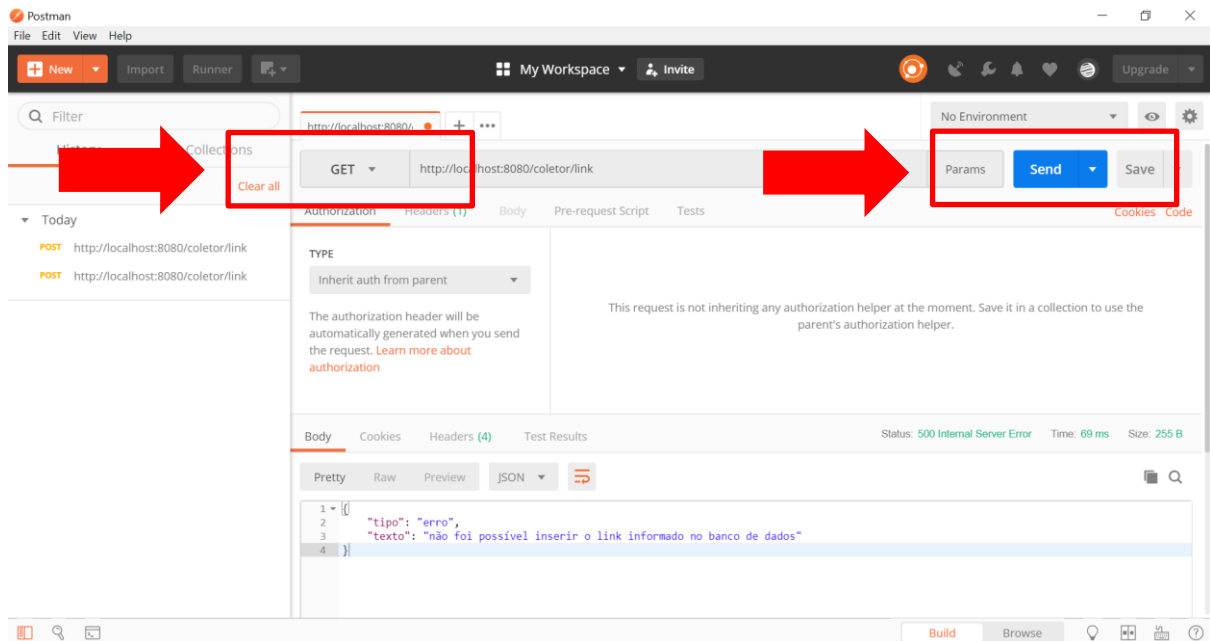
Usando a mesma URL semente informada anteriormente, clique novamente no botão Send. Observe a mensagem de erro retornada pelo servidor. Essa mensagem de erro é retornada, porque o servidor está configurado para não aceitar URLs sementes repetidas.



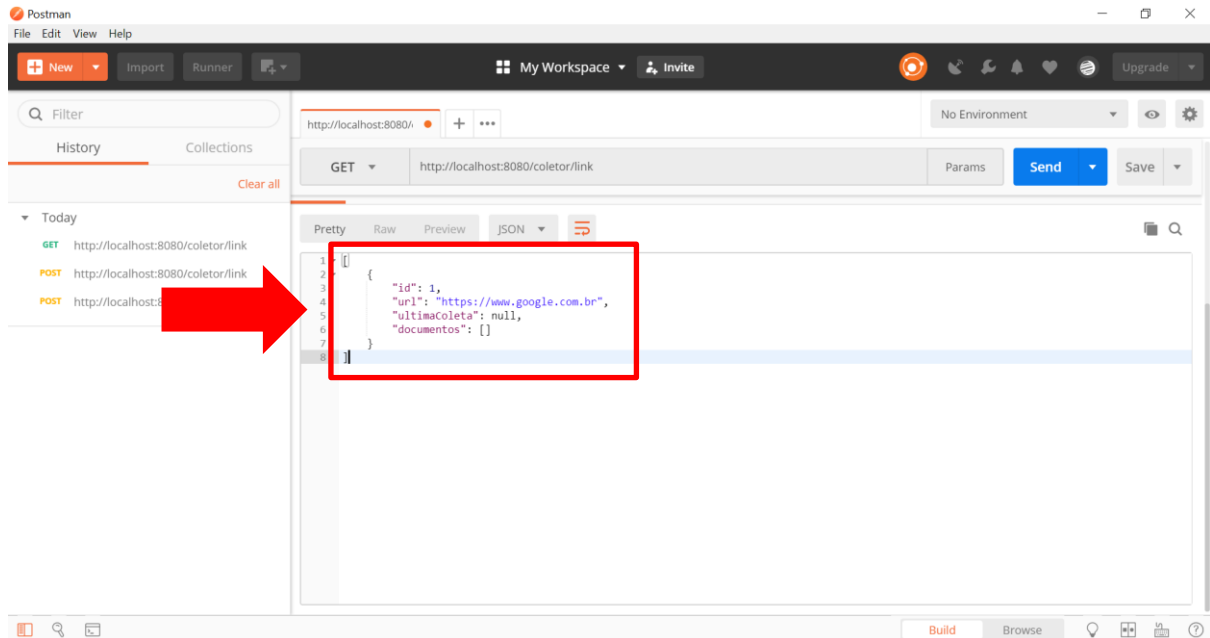
Troque o método do protocolo HTTP para GET. Em seguida, clique no botão Send.



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

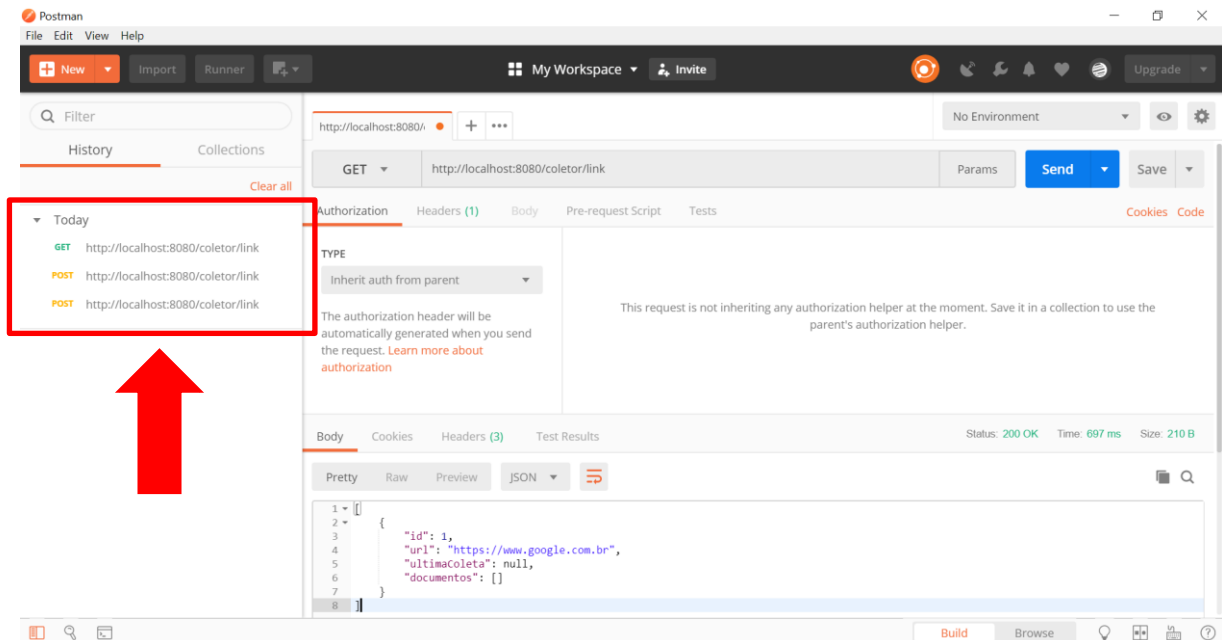


Observe a resposta retornada pelo servidor.





Observe que o Postman armazena em um log as requisições realizadas, para que possam ser reaproveitadas futuramente.



- 2) Altere o projeto, para que o administrador do sistema possa informar várias URLs sementes em uma única requisição POST.
- 3) Modifique o projeto, para que ele atenda os requisitos especificados nas práticas anteriores.
- 4) Analise a API do seu projeto. Sempre que necessário, faça alterações para melhoria da API, adequando-a ao padrão arquitetural REST (Representational State Transfer).
- 5) No projeto, sempre retornar respostas que obedeçam aos 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/VPs)

### **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



**OBS:**

- 1) RFC (Request for Comments - "pedido de comentários"): são documentos técnicos desenvolvidos e mantidos pelo IETF (Internet Engineering Task Force), uma instituição que especifica os padrões que serão implementados e utilizados em toda a Internet.
- 2) WebDAV (Web-based Distributed Authoring and Versioning): é uma extensão do protocolo HTTP para transferência de arquivos; suporta bloqueio de recursos. Quando uma pessoa está editando um arquivo, ele fica bloqueado, impedindo que outras pessoas façam alterações ao mesmo tempo.