



## Manual – Rest Services

## Clima Tempo

Versão 1

### Histórico de Alterações

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>
28/01/2019	1.0	Versão Inicial	Géssica Souza

## 1. FINALIDADE

Este documento tem a finalidade de descrever métodos Rest a serem utilizados, listando informações a serem enviadas e recebidas.

## 2. SERVIÇOS DISPONÍVEIS

A integração é realizada através de serviços disponibilizados como Web Services, URLs (endpoint). Essas URLs receberão as mensagens HTTP através dos métodos POST, GET, DELETE ou PUT.

POST – criação de uma transação. Cliente.

GET – é utilizado para consultas de um cliente passando um identificador, ou todos.

PUT – atualizar um cliente existente.

DELETE – remover um cliente.

	MÉTODOS	ENDPOINTS
Requisição de transação	POST/PUT/DELETE	<a href="http://localhost:8080/clima/id">http://localhost:8080/clima/id</a>
Consultas	GET	<a href="http://localhost:8080/clima">http://localhost:8080/clima</a> ou <a href="http://localhost:8080/clima">http://localhost:8080/clima</a>

## 3. INFORMAÇÕES TÉCNICAS

### 3.1 Tecnologias

Ferramentas utilizadas: Eclipse IDE, Postman para testar os endpoints.

Tecnologias do projeto: SpringBoot, SpringMVC, Maven, H2 (banco de dados em memória, criado quando a aplicação é iniciada), Gson (Utilizada para converter Json), Java 8.

Para executar: springBoot indicado no enunciado do teste

Empacotar: Maven

### 3.2 Passos para rodar o projeto:

Instalar eclipse (utilizado foi neon 3), necessário java 8.

Importar projeto com maven, baixar dependências do maven a partir da versão 3.6.0.

Empacotar projeto utilizando maven install.

Iniciar aplicação na classe Application- método main >> Run SpringApplication

Acessar através de navegador ou postman a url anteriormente informando para a requisição desejada.

### 3.3 Api Externa

Foi utilizada a Api externa: <https://www.ipvigilante.com/> para localizar o ip da pessoa que realizou um post através do serviço.

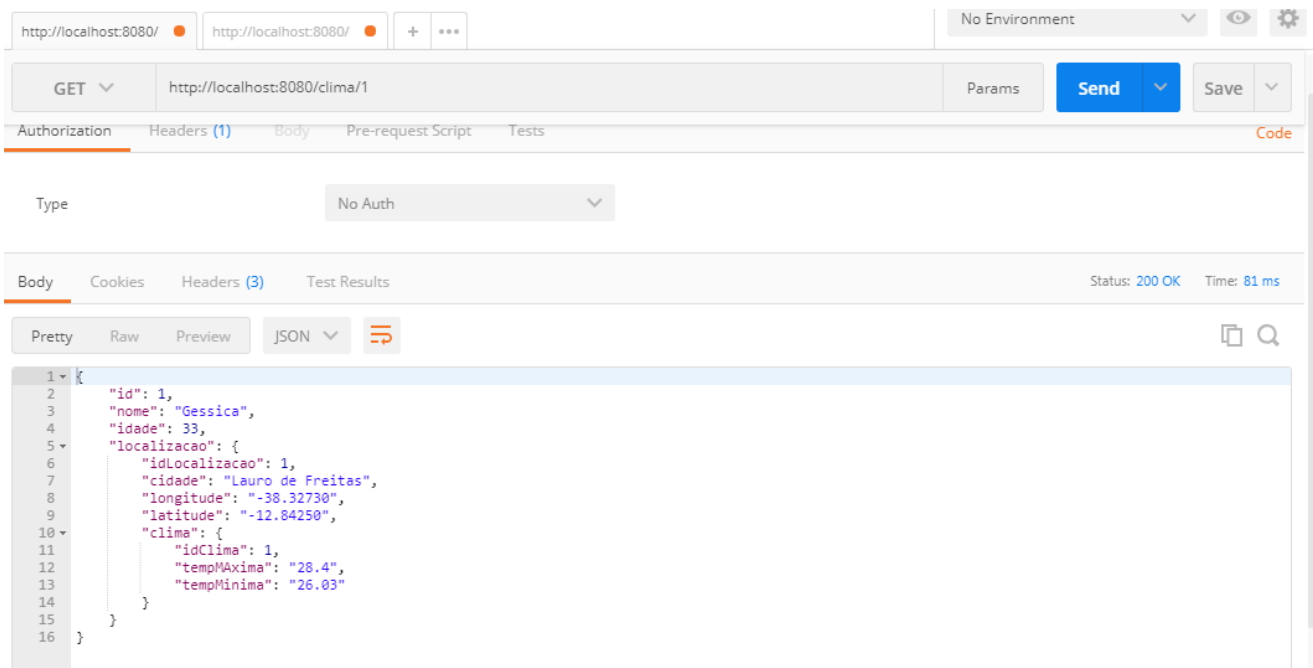
Para buscar a temperatura através da localização encontrada na api anterior, foi utilizado: <https://www.metaweather.com/api/>. Primeiro buscando pela cidade para encontrar o woeid e então encontrar a temperatura para a localização.

## 4. MÉTODOS PUBLICADOS

### 4.1 Consultar

Método responsável por retornar um registro da base.

**Operação: GET**



Parâmetros passados para o método.

Parâmetros		
Atributos	Obrig.	Descrição
id	Sim	Id

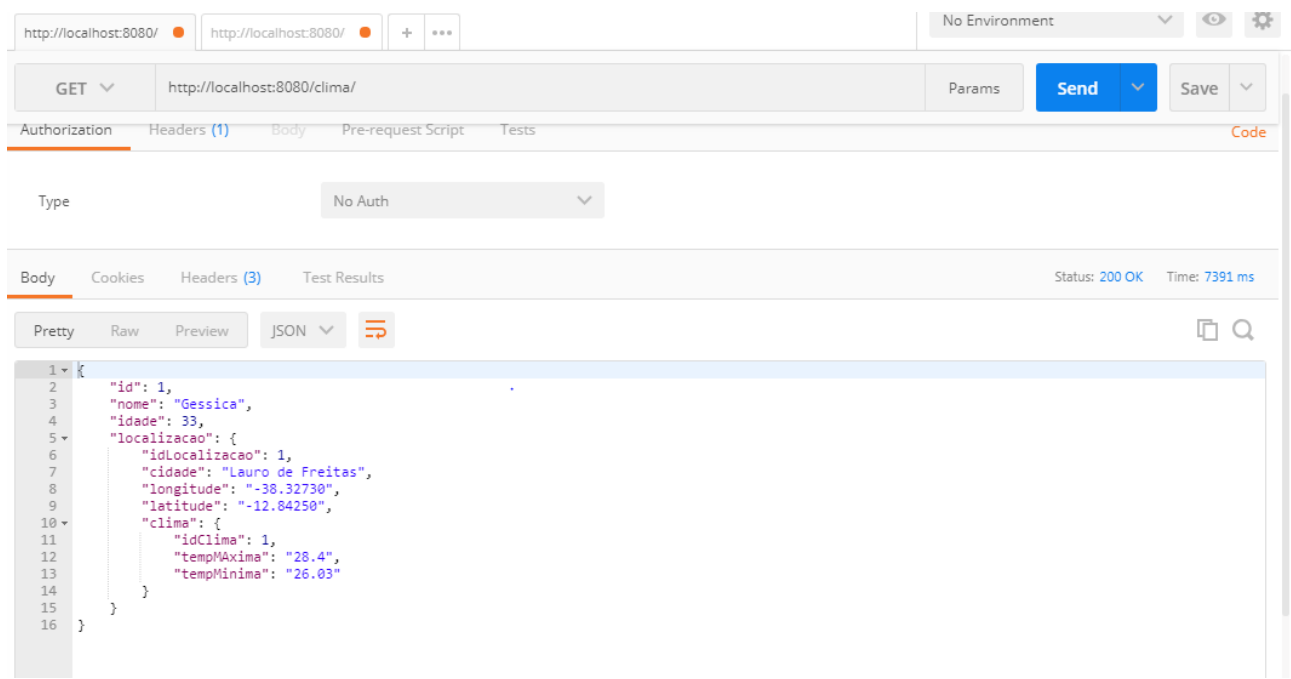
Retorno

Dados de retorno	
Atributos	
nome	
Idade	
localizacao	
clima	

## 4.2 Listar

Método responsável por retornar todos os registros da base.

### Operação: GET



Parâmetros passados para o método.

Parâmetros		
Atributos	Obrig.	Descrição

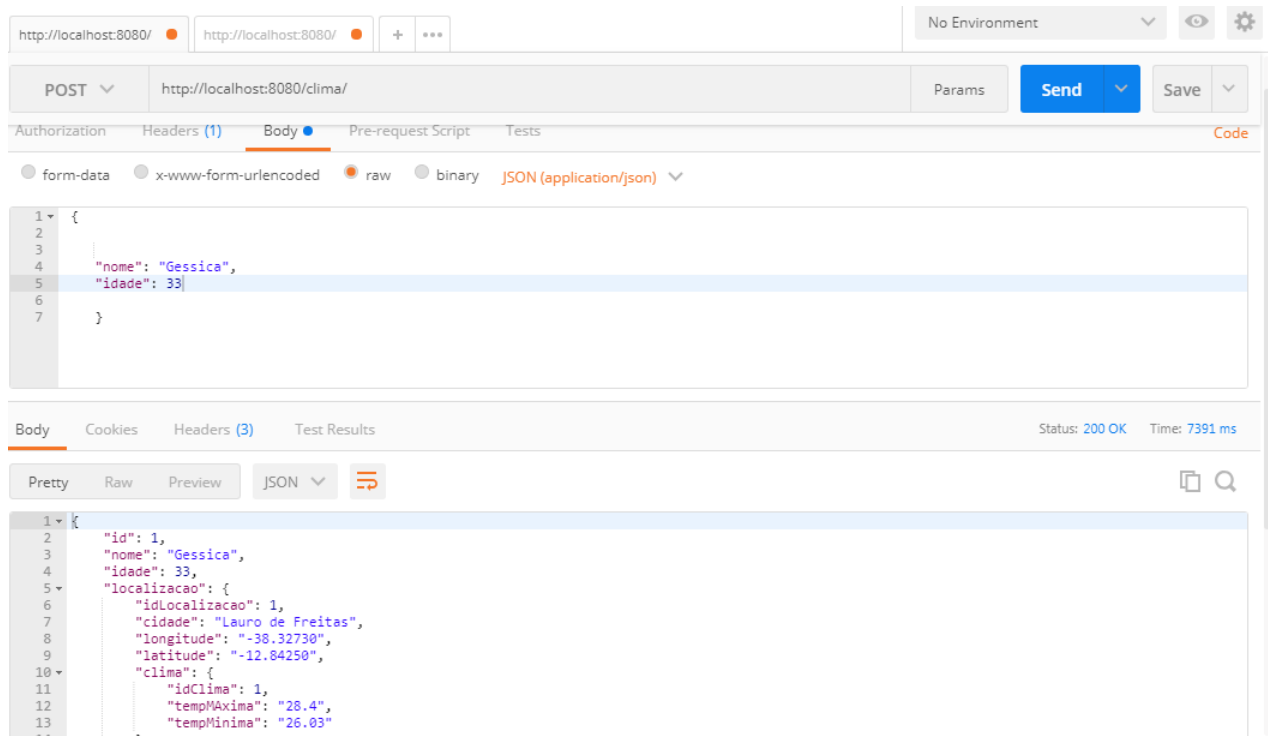
### Retorno

Dados de retorno	
Atributos	
nome	
Idade	
localizacao	
clima	

### 4.3 Adicionar

Método responsável por inserir um registro da base.

#### Operação: POST



Parâmetros passados para o método.

Parâmetros		
Atributos	Obrig.	Descrição
id	Sim	Id

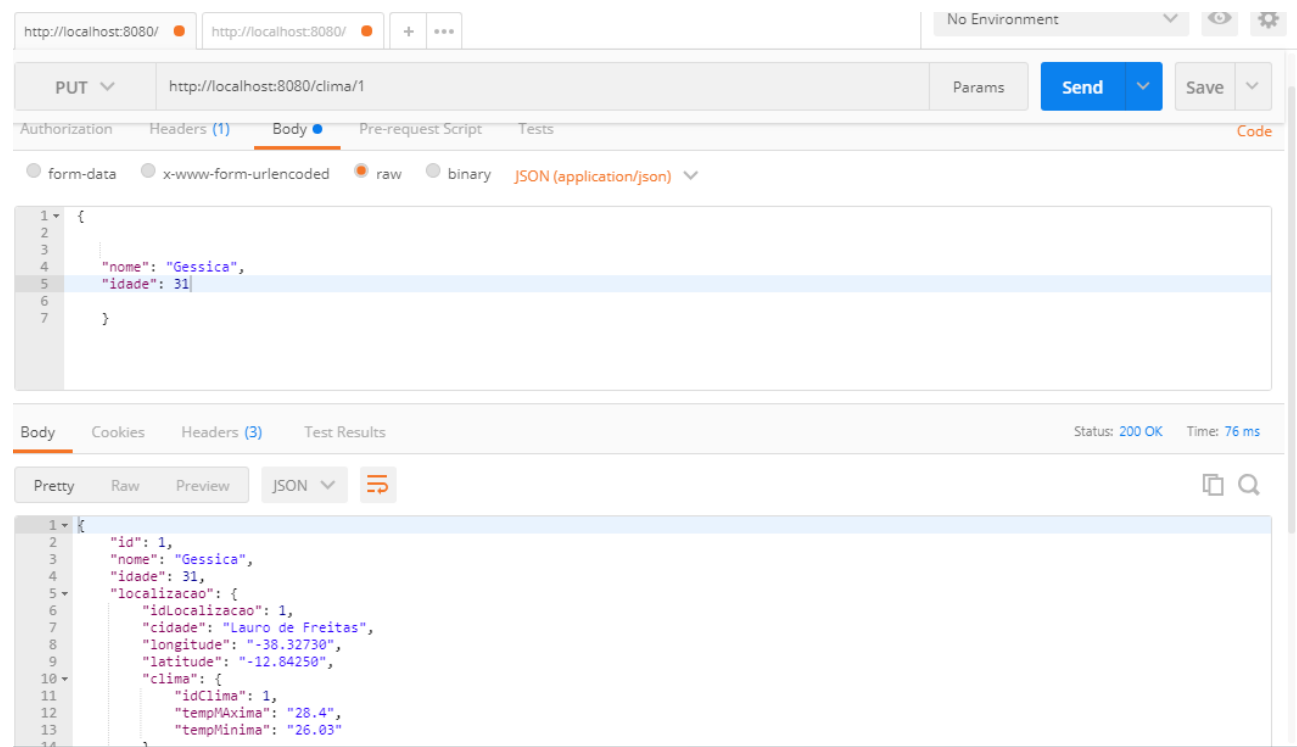
Retorno

Dados de retorno	
Atributos	
nome	
Idade	
localizacao	
clima	

## 4.4 Alterar

Método responsável por alterar um registro da base.

### Operação: PUT



Parâmetros passados para o método.

Parâmetros		
Atributos	Obrig.	Descrição
id	Sim	Id

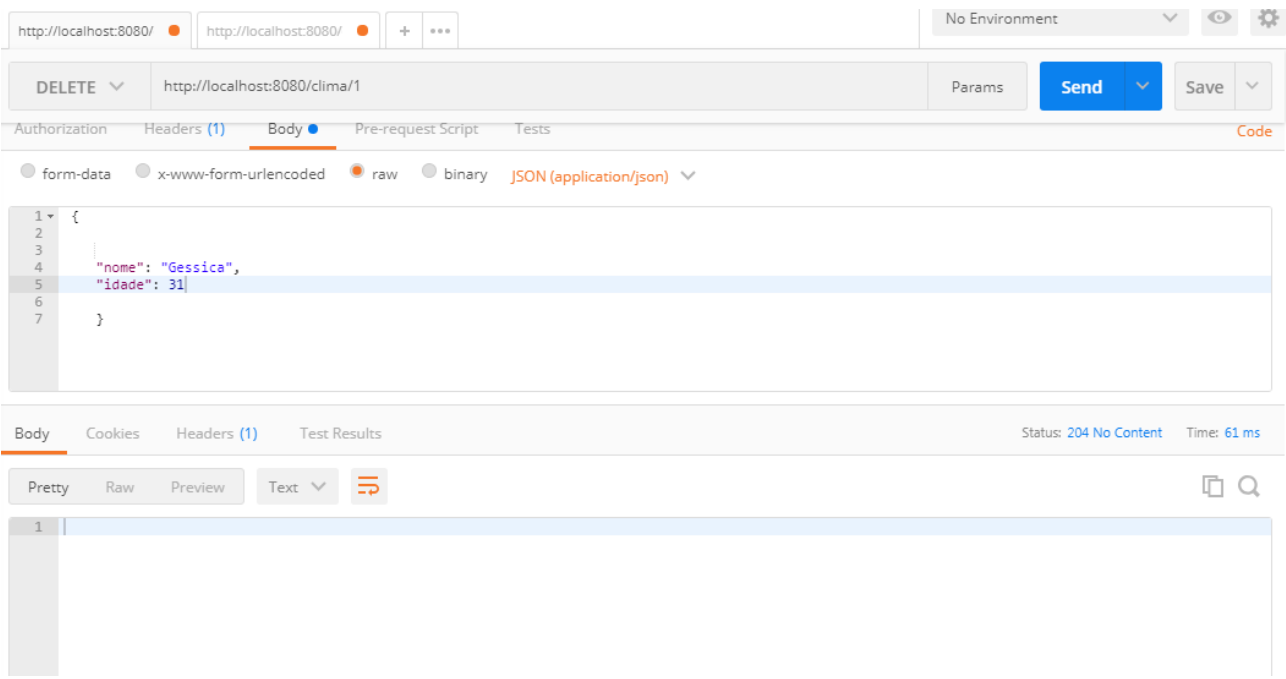
Retorno

Dados de retorno	
Atributos	
nome	
Idade	
localizacao	
clima	

## 4.5 Remover

Método responsável por remover um registro da base.

### Operação: DELETE



Parâmetros passados para o método.

Parâmetros		
Atributos	Obrig.	Descrição
id	Sim	Id