

31. Inserindo uma nova Categoria com POST

No padrão **REST** quando inserimos um novo recurso usamos o método **POST** do protocolo **HTTP**.

- Em **CategoriaResource** criar um método com assinatura **insert(Categoria obj)** para receber uma Categoria no formato JSON
- Utilizar a notação **@RequestMapping(method = RequestMethod.POST)** acima do método
- O método recebe uma Categoria como parametro e utiliza a notação **@RequestBody** nele
- A categoria é recebida no formato JSON e inserida no BD através de um método de serviço
- O método de serviço está na classe **CategoriaService** → **Categoria insert(Categoria obj)**, ele salva e retorna a categoria para quem invocou
- O método retorna uma **ResponseEntity<Void>** com uma URI especifica para a categoria salva

CategoriaResource.java

```
@RequestMapping(method = RequestMethod.POST)
public ResponseEntity<Void> insert(@RequestBody Categoria obj) {
    obj = categoriaService.insert(obj);
    URI uri = ServletUriComponentsBuilder.fromCurrentRequest().path("/{id}").buildAndExpand(obj.getId()).toUri();
    return ResponseEntity.created(uri).build();
}
```

Quando uma requisição **HTTP** ocorre com **sucesso** ela é do tipo **2xx**. Ao inserir um novo recurso, o código de sucesso é **201**. Além do código HTTP devemos **retornar** a **URI** do recurso **criado**:

ServletUriComponentsBuilder.

fromCurrentRequest().path("/{id}").

buildAndExpand(obj.getId()).toUri();

FromCurrentRequest() concatena URI localhost:8080/categorias com o id da categoria criada, isto é, ao final teremos localhost:8080/categorias/id

- Em **CategoriaService** criar um método com assinatura **Categoria insert(Categoria obj)** para receber uma Categoria
- O método deve salvar a Categoria no BD e retorna-la
- **obj.setId(null)** garante que uma categoria nova está sendo inserida
- **categoriaRepository.save(obj)** salva a categoria no BD, se o ID já estiver cadastrado esta operação é vista como uma atualização

CategoriaService.java

```
public Categoria insert(Categoria obj) {
    obj.setId(null);
    return categoriaRepository.save(obj);
}
```

Para testar o POST vamos usar o Postman:

- Acessar localhost:8080/categorias e colocar o método ao lado da URI como POST.
- Abaixo do address selecionar BODY. Dentro do BODY selecionar RAW e á direita de RAW selecionar JSON(application/json) e então escrever a nova categoria no formato JSON:

```
{ "nome": "cama mesa e banho" }
```

- Abaixo do campo onde a categoria em formato JSON foi escrita clique em Headers e você verá o URI e o HTTP status code.

Complemento:**@RequestBody:**

A anotação mapeia o corpo de uma requisição HTTP para um DTO ou objeto do dominio, permitindo a desserialização automática do corpo da requisição HTTP em um objeto java.

O Spring desserializa JSON em objeto Java automaticamente assumindo que um tipo apropriado seja especificado. Por padrão o tipo que nós anotamos com @RequestBody corresponde ao JSON enviado ao controller pelo cliente.

Fonte: <https://www.baeldung.com/spring-request-response-body>