

Insomnia - Security

Quando a Spring Security está habilitada na nossa api, a forma como o Insomnia irá interagir com a aplicação Spring será um pouco diferente. Para efetuarmos as nossas requisições, será necessário seguir alguns passos:

1. A API precisa disponibilizar pelo menos 2 endpoints livres de autenticação (Cadastrar Usuário e Login), para acessarmos a aplicação, ou seja, que não necessitem de usuário, senha e token para serem acessados.
2. Precisamos efetuar o cadastro de pelo menos um usuário
3. Efetuar o login (Autenticação) com o usuário e a senha cadastrada para obtermos o Token de segurança
4. Alterar os cabeçalhos (Header) de todas as requisições acrescentando o Token como parâmetro para liberar o acesso ao endpoint protegido.

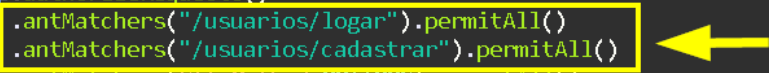
No projeto **Blog Pessoal**, os 2 endpoints liberados são:

1. `/usuarios/logar`
2. `/usuarios/cadastrar`

Estes endpoints foram liberados na Classe **BasicSecurityConfig**, no método **configure(HttpSecurity http)**, como mostra a figura abaixo:


```
@Override
protected void configure(HttpSecurity http) throws Exception {

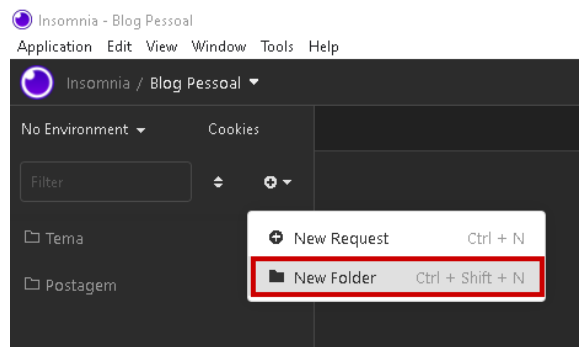
    http.authorizeRequests()
        .antMatchers("/usuarios/logar").permitAll()
        .antMatchers("/usuarios/cadastrar").permitAll()
        .antMatchers(HttpMethod.OPTIONS).permitAll()
        .anyRequest().authenticated()
        .and().httpBasic()
        .and().sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS)
        .and().cors()
        .and().csrf().disable();
}
```



Passo 01 - Criar a Pasta Usuario

Vamos criar dentro da **Collection Blog Pessoal** a **Pasta Usuario**, que guardará todas as requisições do **Recurso Usuario**.

1. Na **Collection Blog Pessoal**, clique no botão . No menu que será aberto, clique na opção **New Folder**.



2. Na janela que será aberta, informe o nome da pasta (**Usuario**) e clique no botão **Create** para concluir.



Passo 02 - Criar a Requisição Cadastrar Usuario

Vamos começar pela requisição Cadastrar Usuario porque sem um usuário cadastrado não será possível autenticar (logar) no sistema e acessar os demais endpoints.

1. Clique com o botão direito do mouse sobre a **Pasta Usuario** para abrir o menu e clique na opção **New Request**.
2. Na janela que será aberta, informe o nome da requisição e o Método HTTP que será utilizado (**POST**). Depois de selecionar o Método **POST**, observe que será necessário informar o formato em que os dados serão enviados através do Corpo da Requisição. Selecione o formato **JSON** e clique no botão **Create** para concluir.

New Request

Name (defaults to your request URL if left empty)

Cadastrar Usuario

POST

JSON

* Tip: paste Curl command into URL afterwards to import it

Create

3. Configure a requisição conforme a imagem abaixo:

POST

http://localhost:8080/usuarios/cadastrar

Send

Other

Auth

Query

Header 1

Docs

```
1 {
2   "nome": "Administrador",
3   "usuario": "admin@email.com.br",
4   "senha": "admin123",
5   "foto": "https://i.imgur.com/rNUV85x.png"
6 }
```

4. Observe que na requisição do tipo Post o Corpo da requisição (Request Body), deve ser preenchido com um JSON contendo o nome, o usuario(e-mail), a senha e a foto(link), que vc deseja persistir no Banco de dados.
5. Observe que depois de persistir os dados do usuario, a aplicação retornará a senha criptografada.

201 Created

797 ms

190 B

Just Now

Preview

Header 12

Cookie

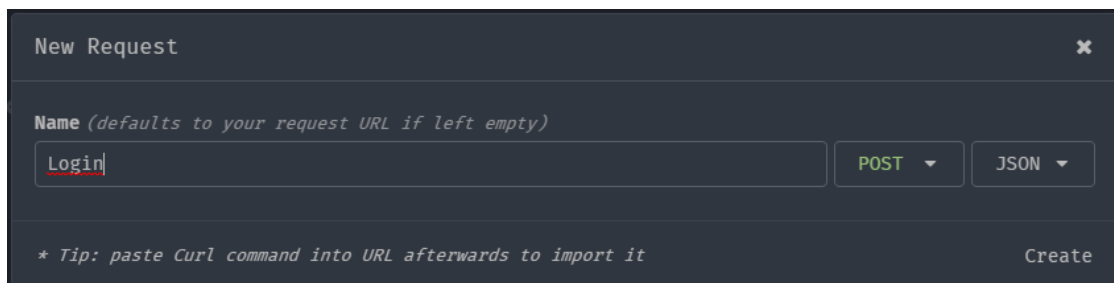
Timeline

```
1 {
2   "id": 1,
3   "nome": "Administrador",
4   "foto": "https://i.imgur.com/rNUV85x.png",
5   "usuario": "admin@email.com.br",
6   "senha": "$2a$10$UFjh2/KitNH03HFa1I8fWuiwPdBTWAOn8wPE.st4euznkv/bCJziq",
7   "postagem": null
8 }
```

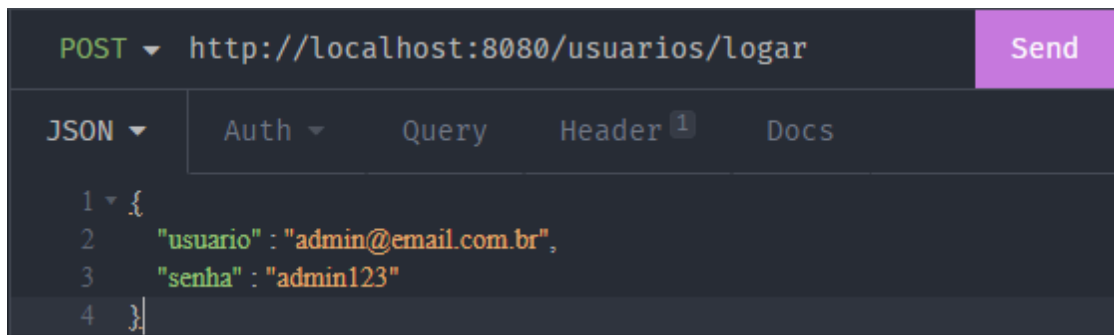
Passo 03 - Criar a Requisição Autenticar Usuario (Login)

Usuário persistido, agora vamos efetuar o login no sistema, que nos retornará o **Token de Autorização**, que nos permitirá consumir os demais endpoints e recursos da aplicação.

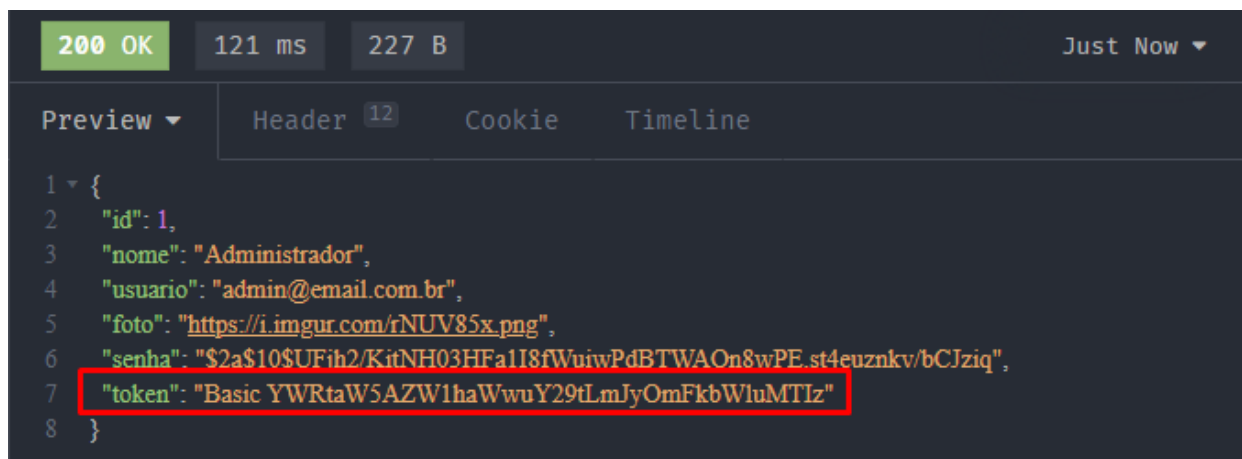
1. Clique com o botão direito do mouse sobre a **Pasta Usuario** para abrir o menu e clique na opção **New Request**.
2. Na janela que será aberta, informe o nome da requisição e o Método HTTP que será utilizado (**POST**). Depois de selecionar o Método **POST**, observe que será necessário informar o formato em que os dados serão enviados através do Corpo da Requisição. Selecione o formato **JSON** e clique no botão **Create** para concluir.



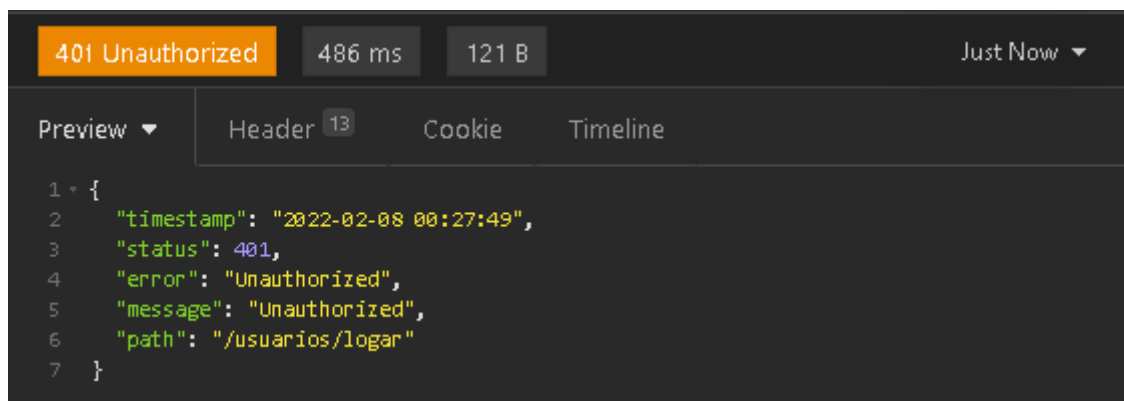
3. Configure a requisição conforme a imagem abaixo:



4. Observe que no JSON estamos passando apenas o usuário e a senha, porquê são as únicas informações que utilizaremos no login. A Classe UsuarioLogin possui outros atributos que serão preenchidos pela aplicação (se o login for efetuado com sucesso) e retornados no JSON da resposta, como vemos na figura abaixo:

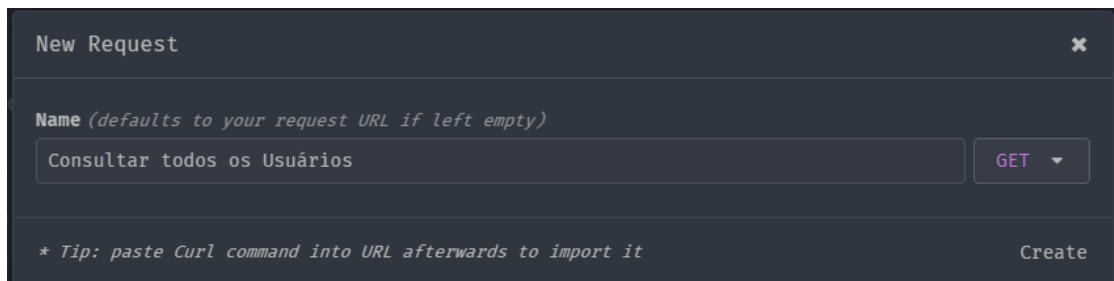


5. Observe que o Token foi gerado e enviado na resposta. Copie o Token da resposta porque vamos precisar dele nas próximas requisições.
6. Caso o login falhe, você receberá o **status 401 (Unauthorized)**, como mostra a figura abaixo:

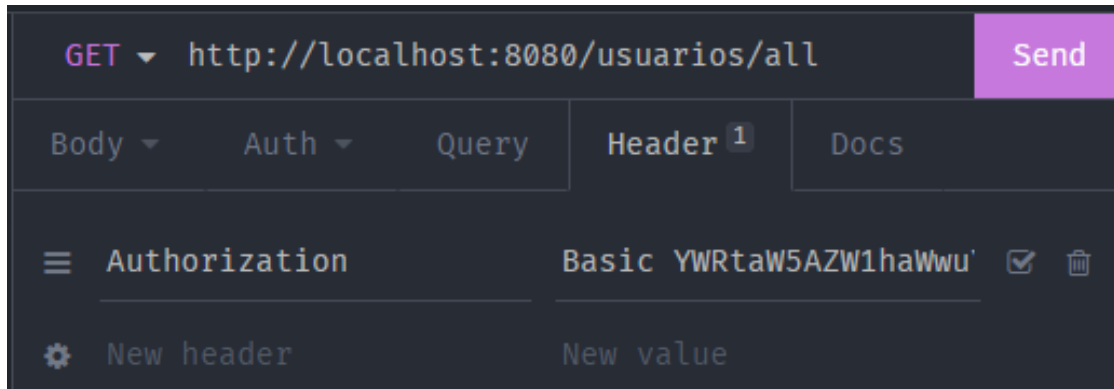


Passo 04 - Criar a Requisição Consultar todos os Usuarios - findAll()

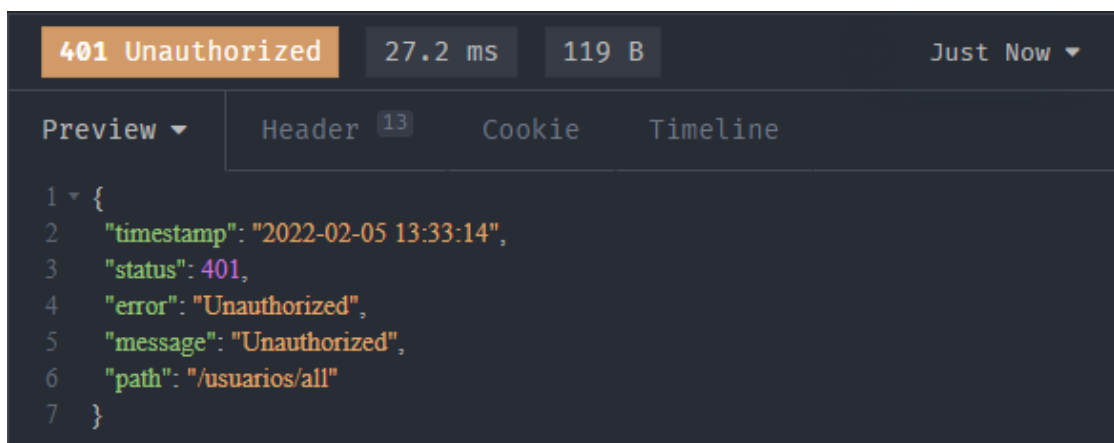
1. Clique com o botão direito do mouse sobre a **Pasta Usuario** para abrir o menu e clique na opção **New Request**.
2. Na janela que será aberta, informe o nome da requisição e o Método HTTP que será utilizado (**GET**). Clique no botão **Create** para concluir.



3. Configure a requisição conforme a imagem abaixo:



4. Como a segurança foi habilitada, caso você envie a requisição do jeito que está, você receberá o **status 401 (Unauthorized)**, como mostra a figura abaixo:



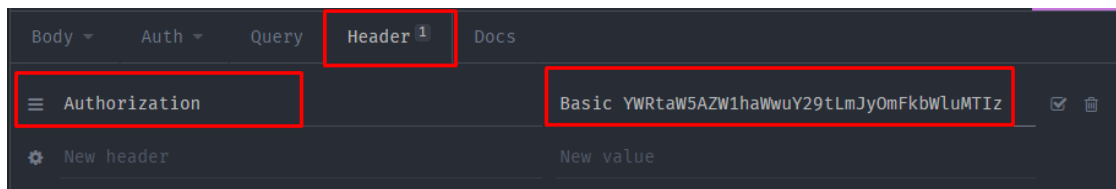
5. A explicação é simples: **Você precisa passar no cabeçalho da requisição o Token recebido no login**, caso contrário não será possível efetuar nenhuma operação dentro da API.



ATENÇÃO: No Frontend, independente de ser implementado no Angular ou no React, a passagem do Token será automatizada.

6. Clique na Guia **Header** e adicione uma nova linha de cabeçalho.

7. Na coluna **New header** digite **Authorization**. Na coluna **New value** da linha insira o **Token** que você recebeu no Login, como mostra a figura abaixo:



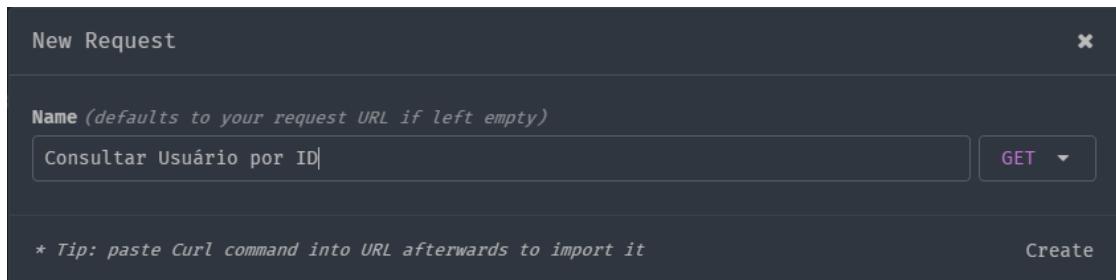
8. Agora a sua requisição funcionará!



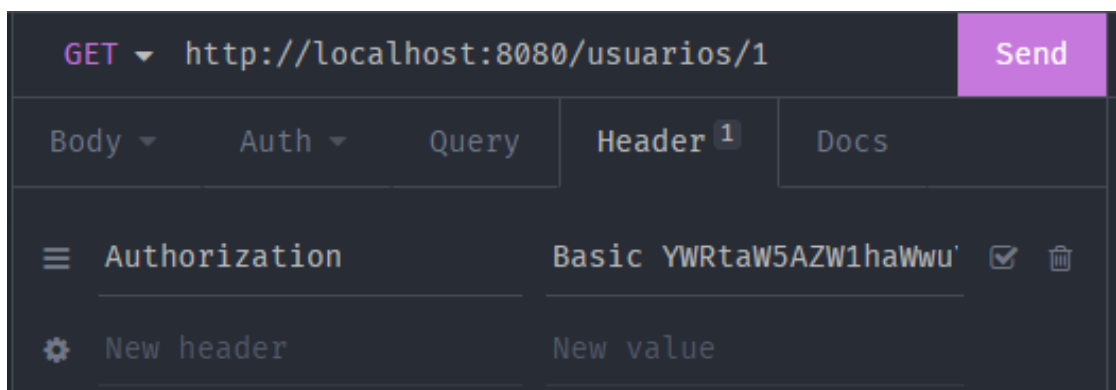
ALERTA DE BSM: Mantenha a Atenção aos Detalhes! Você deverá inserir o token no cabeçalho de todas requisições dos Recursos Postagem, Tema e Usuario (Exceto Cadastrar e Logar), caso contrário você receberá o status 401 (Unauthorized) em todas elas.

Passo 05 - Criar a Requisição Consultar Usuario por ID - findById(id)

1. Clique com o botão direito do mouse sobre a **Pasta Usuario** para abrir o menu e clique na opção **New Request**.
2. Na janela que será aberta, informe o nome da requisição e o Método HTTP que será utilizado (**GET**). Clique no botão **Create** para concluir.

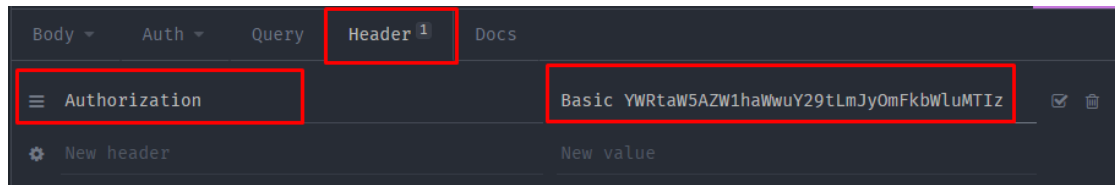


3. Configure a requisição conforme a imagem abaixo:



4. Clique na Guia **Header** e adicione uma nova linha de cabeçalho.

5. Na coluna **New header** digite **Authorization**. Na coluna **New value** da linha insira o **Token** que você recebeu no Login, como mostra a figura abaixo:

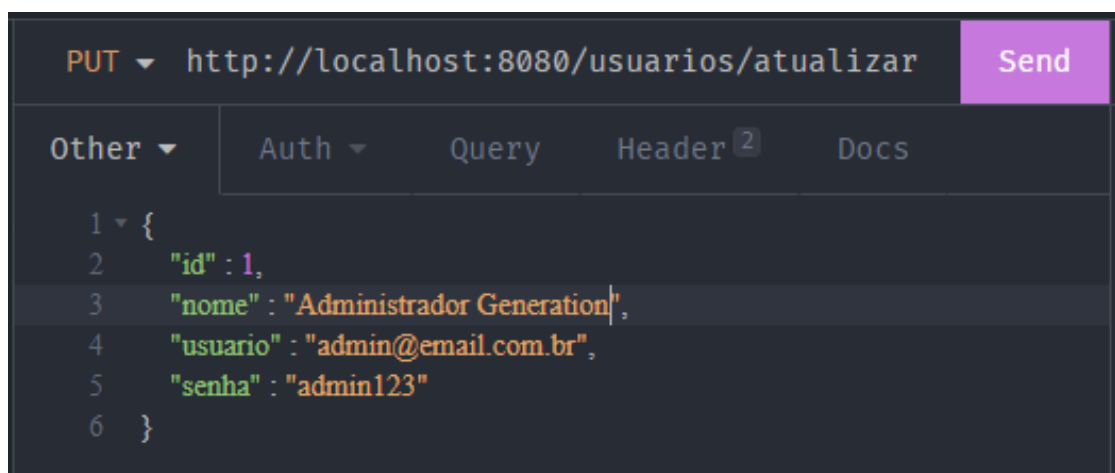


Passo 06 - Criar a Requisição Atualizar Usuário

1. Clique com o botão direito do mouse sobre a **Pasta Usuario** para abrir o menu e clique na opção **New Request**.
2. Na janela que será aberta, informe o nome da requisição e o Método HTTP que será utilizado (**PUT**). Depois de selecionar o Método **PUT**, observe que será necessário informar o formato em que os dados serão enviados através do Corpo da Requisição. Selecione o formato **JSON** e clique no botão **Create** para concluir.



3. Configure a requisição conforme a imagem abaixo:

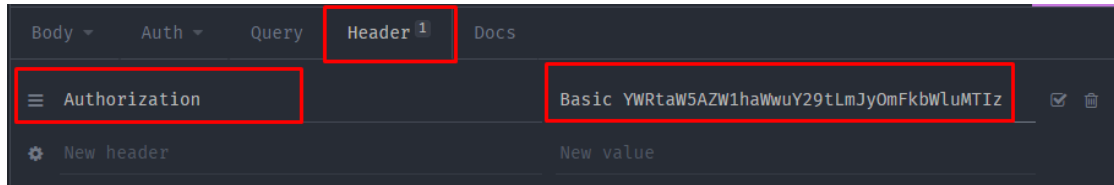


4. Observe que na requisição do tipo PUT o Corpo da requisição (Request Body), deve ser preenchido com um JSON contendo o Id, o nome, o usuario(e-mail), a

senha e a foto(link) que vc deseja atualizar no Banco de dados.

5. Clique na Guia **Header** e adicione uma nova linha de cabeçalho.

6. Na coluna **New header** digite **Authorization**. Na coluna **New value** insira o **Token** que você recebeu no Login, como mostra a figura abaixo:



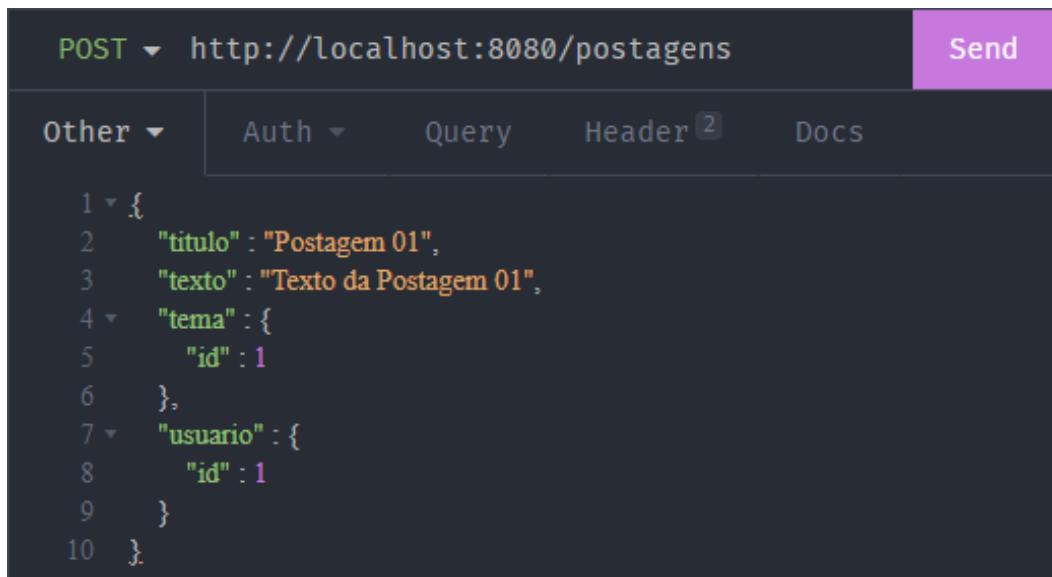
Passo 07 - Atualizar as Requisições Cadastrar e Atualizar Postagem no Insomnia

Como habilitamos o Relacionamento entre as Classes Postagem e Usuario, para Cadastrarmos e Alterarmos as Postagens vamos precisar atender alguns requisitos:

- O Tema e o Usuario devem ser persistidos antes de criar a nova Postagem.
- Na requisição Cadastrar e Atualizar Postagem, o JSON enviado no Corpo da Requisição deve conter um Objeto da Classe Tema identificado apenas pelo **Atributo id** e um Objeto da Classe Usuario identificado apenas pelo **Atributo id**.

7.1. Atualizar a Requisição Cadastrar Postagem

Vamos alterar o Corpo da requisição (Body), conforme a imagem abaixo:



Observe que está sendo passado dentro do JSON um **Objeto da Classe Usuario** chamado **usuario**, identificado apenas pelo **Atributo id**.

Não esqueça de inserir o token no cabeçalho da requisição.



ATENÇÃO: O Objeto Usuario deve ser persistido no Banco de dados antes de ser inserido no JSON da requisição Cadastrar Postagem.

7.2. Atualizar a Requisição Atualizar Postagem

Vamos alterar o Corpo da requisição (Body), conforme a imagem abaixo:

```
PUT http://localhost:8080/postagens/

1 {
2   "id" : "1",
3   "titulo" : "Minha Postagem 12 - Atualizada!",
4   "texto" : "Texto da postagem 12 atualizado!",
5   "tema" : {
6     "id" : 1
7   },
8   "usuario" : {
9     "id" : 1
10  }
11 }
```

Observe que está sendo passado dentro do JSON um **Objeto da Classe Usuario** chamado **usuario**, identificado apenas pelo **Atributo id**.

Não esqueça de inserir o token no cabeçalho da requisição.



ATENÇÃO: O Objeto Usuario deve ser persistido no Banco de dados antes de ser inserido no JSON da requisição Atualizar Postagem.