

API Testing - Métodos GET e POST

Índice

- 01 Método GET
- 02 Método POST



01

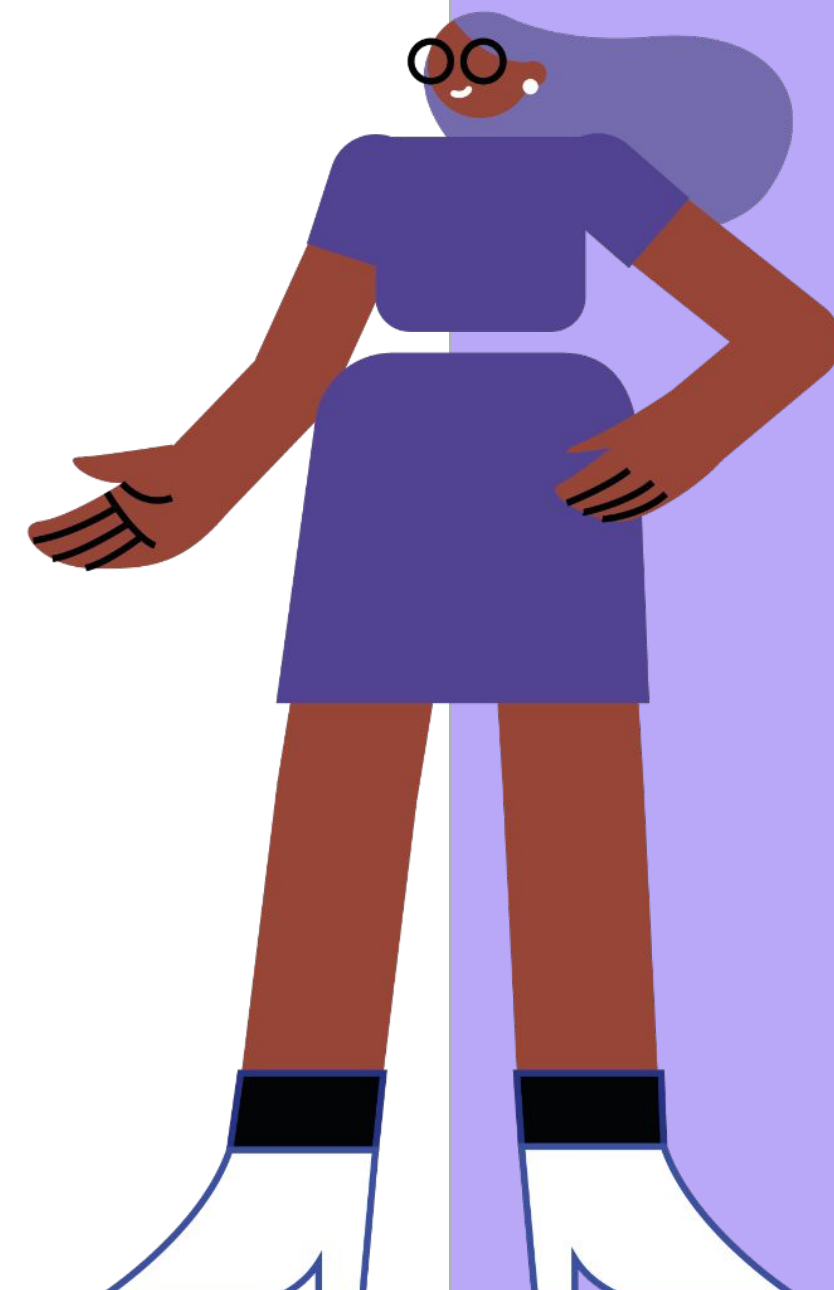
Método GET

Testing método GET

Vamos aprender como aplicar testes aos diferentes métodos HTTP.

Usaremos uma solicitação **GET** para recuperar informações de uma URL específica e analisar as informações obtidas a partir dos testes.

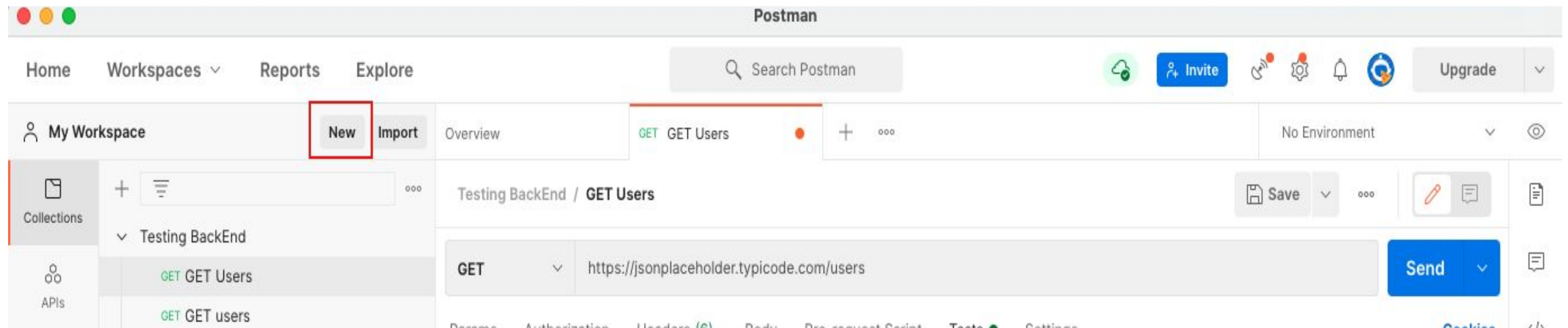
No exemplo, analisaremos a obtenção de usuários de uma API externa.



Passos

1

Primeiro, devemos criar uma nova solicitação no Postman. Para fazer isso, você deve clicar na guia “New”.



Passos

2

A próxima etapa é criar a solicitação. Como?

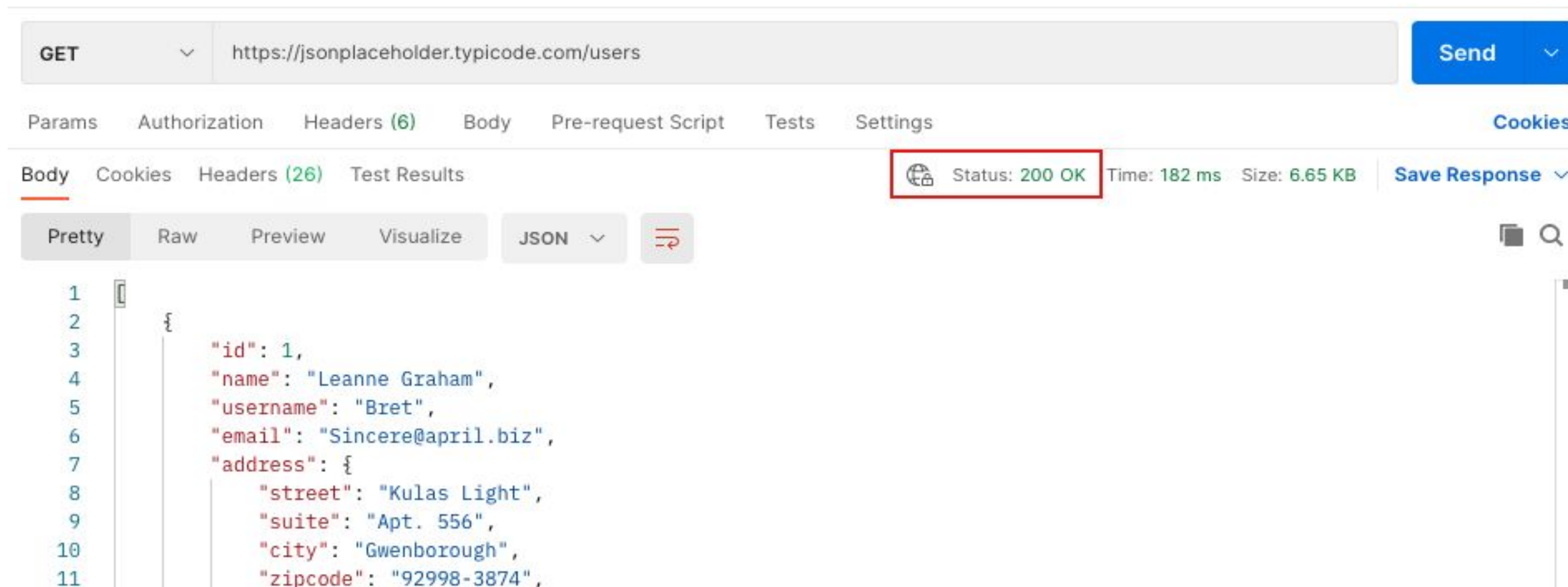
1. Configure sua solicitação HTTP em **GET**.
2. Insira o link da URL na solicitação (<https://jsonplaceholder.typicode.com/users>).
3. Clique em ENVIAR para enviar a solicitação ao servidor que hospeda a URL.



A screenshot of an HTTP client interface. It features a light gray bar with three main components: a dropdown menu on the left showing 'GET' with a red box and a red '1' next to it; a text input field in the center containing the URL 'https://jsonplaceholder.typicode.com/users' with a red box and a red '2' next to it; and a blue 'Send' button on the right with a red box and a red '3' next to it.

Passos

3 Quando a mensagem **200 OK** for exibida, significa que a solicitação foi bem-sucedida.

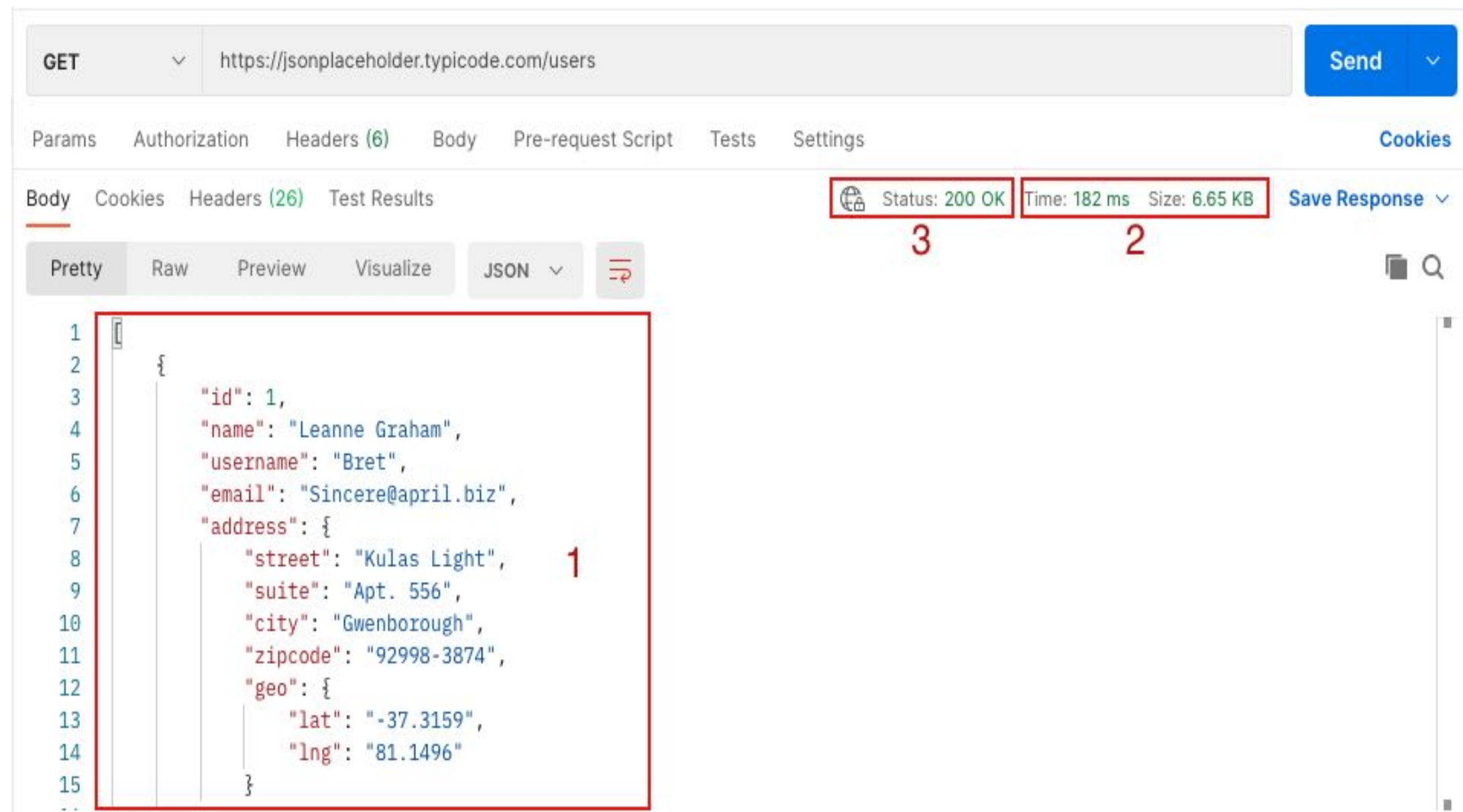


Resultado

Podemos ver vários dados relacionados à resposta do servidor:

1. *Response*: são as informações simples retornadas pelo servidor. Com isso, podemos fazer uma revisão antecipada dos dados da aplicação.

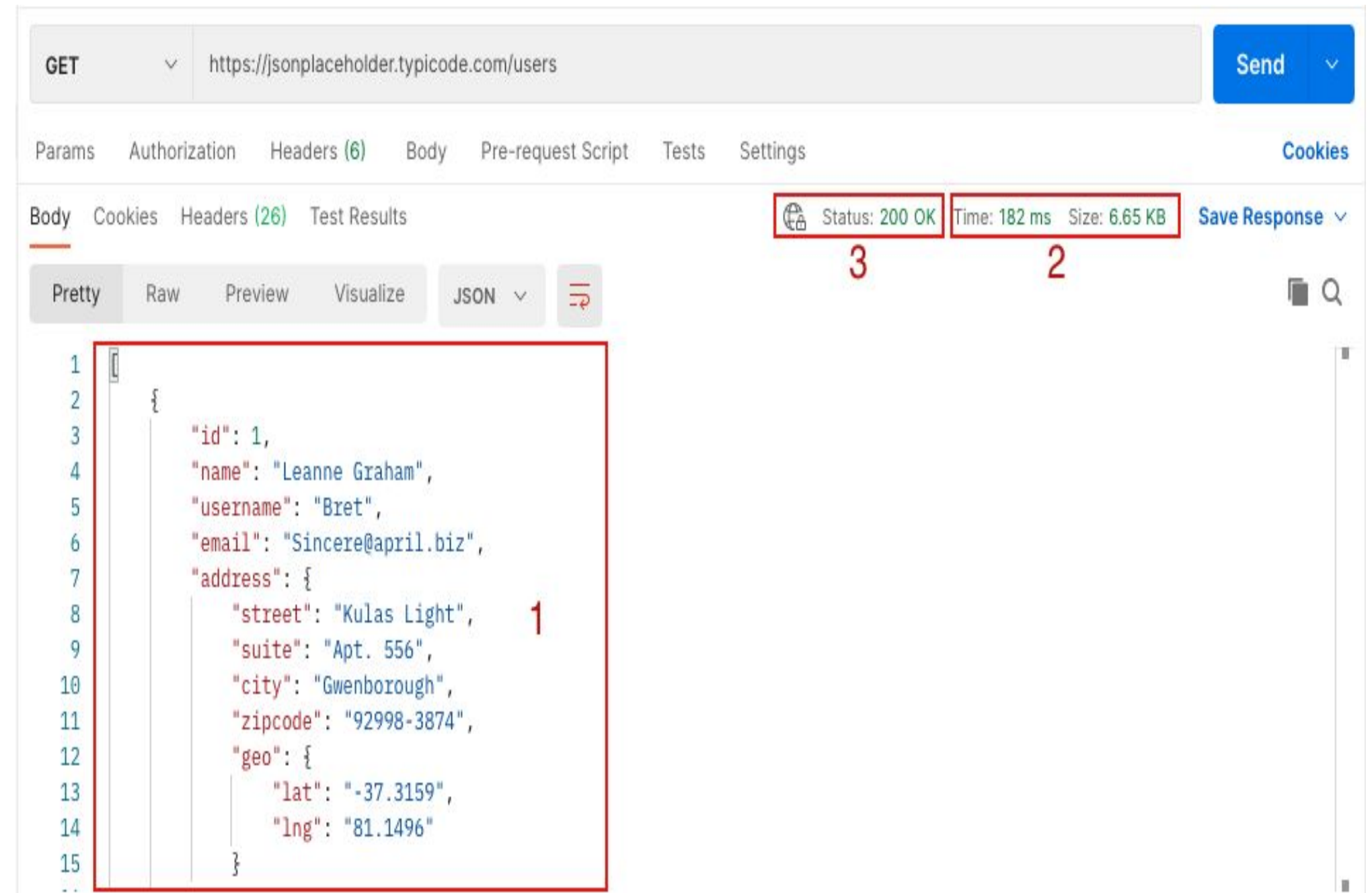
2. Tempo e tamanho de resposta: com estes dados, podemos ver se o sistema está atendendo a um dos requisitos não funcionais, como o desempenho.



Resultado

3. Código de resposta: quando você solicita informações ao servidor, ele pode responder a diferentes códigos de status que informam o que aconteceu com a solicitação.

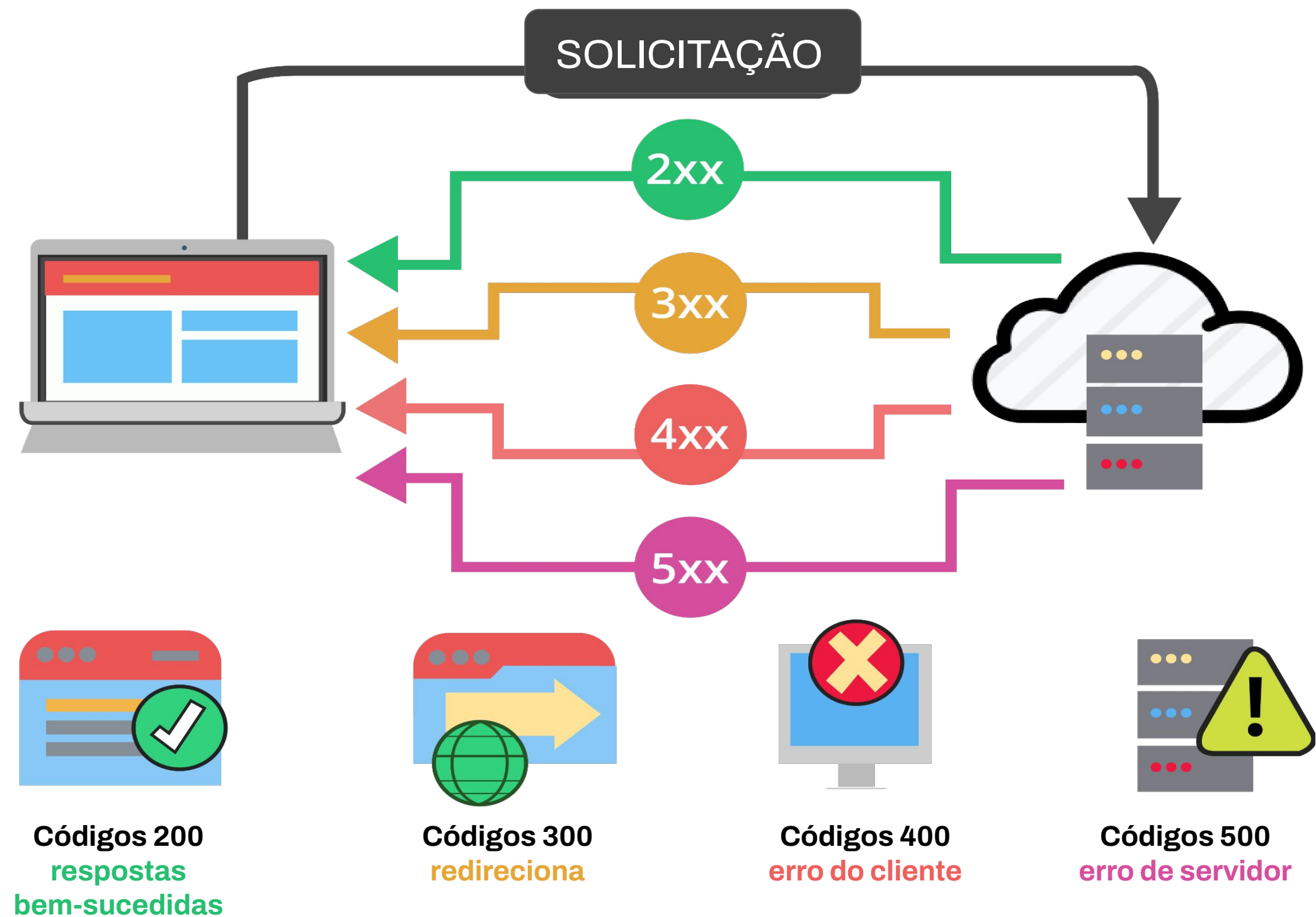
O código 200 nos indica que a solicitação foi bem-sucedida.



Resultado

Uma API retorna diferentes códigos de resposta que nos informam o que aconteceu com a solicitação. Estas respostas são agrupadas em quatro classes.

Na imagem, podemos ver as diferentes classes e seus significados.



Resultado

4. **Cookies**: nos permitem ver as informações relacionadas à sessão.

GEThttps://jsonplaceholder.typicode.com/usersSendSave

ParamsAuthorizationHeaders (9)BodyPre-request ScriptTestsSettingsCookiesCode

▼ Headers (1)

	KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets ▼
<input checked="" type="checkbox"/>	Content-Type	application/json				
	Key	Value	Description			

► Temporary Headers (8) ⓘ

BodyCookies (1)Headers (19)Test ResultsStatus: 200 OKTime: 1075msSize: 6.11 KBSave Response ▼

Name	Value	Domain	Path	Expires	HttpOnly	Secure
_cfduid	d27e2f460f649a7000038d0196eaf1e161583369076	typicode.com	/	Sat, 04 Apr 2020 00:44:36 GMT	true	false

4

Resultado

5. **Headers**: informações sobre a solicitação processada.

GEThttps://jsonplaceholder.typicode.com/users

SendSave

ParamsAuthorizationHeaders (9)BodyPre-request ScriptTestsSettingsCookiesCode

▼ Headers (1)

	KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets ▼
☰	Content-Type	application/json				X
	Key	Value	Description			

▶ Temporary Headers (8) ⓘ

BodyCookies (1)Headers (19)Test ResultsStatus: 200 OKTime: 1075msSize: 6.11 KBSave Response ▼

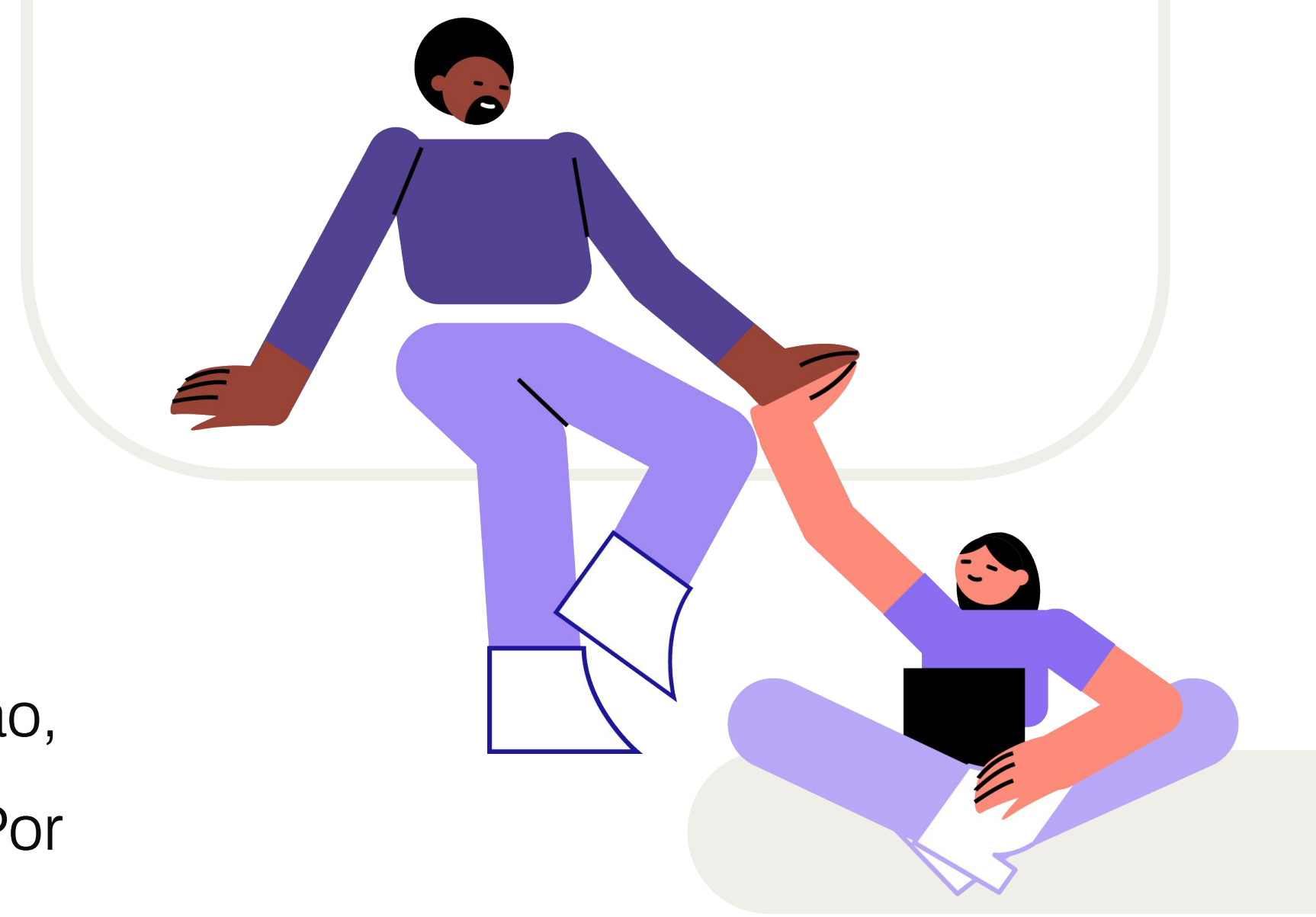
KEY	VALUE	
Date ⓘ	Thu, 05 Mar 2020 04:48:43 GMT	
Content-Type ⓘ	application/json; charset=utf-8	
Transfer-Encoding ⓘ	chunked	5
Connection ⓘ	keep-alive	
X-Powered-By ⓘ	Express	
Vary ⓘ	Origin, Accept-Encoding	

02

Método POST

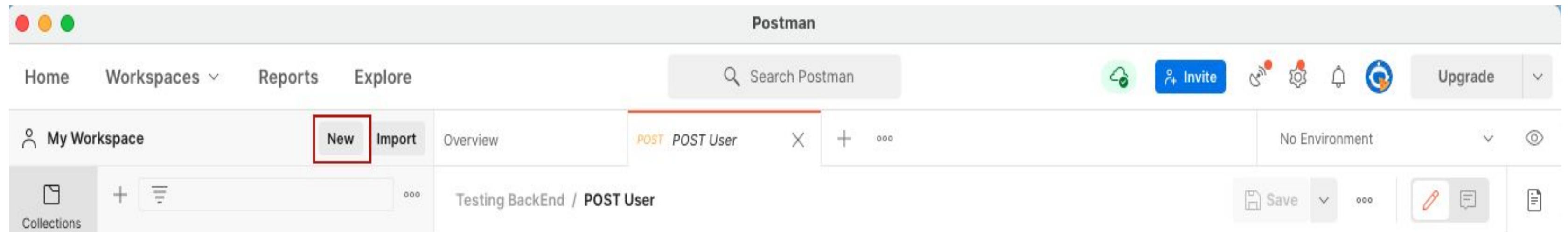
Testing método POST

Quando precisamos adicionar dados à nossa aplicação, utilizamos o método **POST** para enviar estes dados. Por meio desta solicitação, enviamos os dados e a API retorna uma resposta que valida se a criação foi bem-sucedida. No exemplo, veremos a criação de um usuário e a resposta da API.



Passos

- 1 Assim como no método GET, uma nova solicitação deve ser criada no Postman. Então, devemos clicar na guia "New".

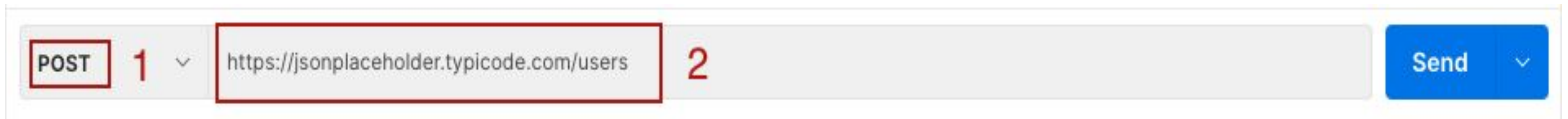


Passos

2

O próximo passo é criar a solicitação. Como?

1. Configure sua solicitação HTTP em **POST**.
2. Insira o link da **URL** na solicitação (<https://jsonplaceholder.typicode.com/users>).



A screenshot of an HTTP client interface. On the left, a dropdown menu is set to 'POST' with a red box around it and a red number '1' next to it. In the center, the URL 'https://jsonplaceholder.typicode.com/users' is entered in a text field, also with a red box around it and a red number '2' next to it. On the right, there is a blue 'Send' button with a dropdown arrow.

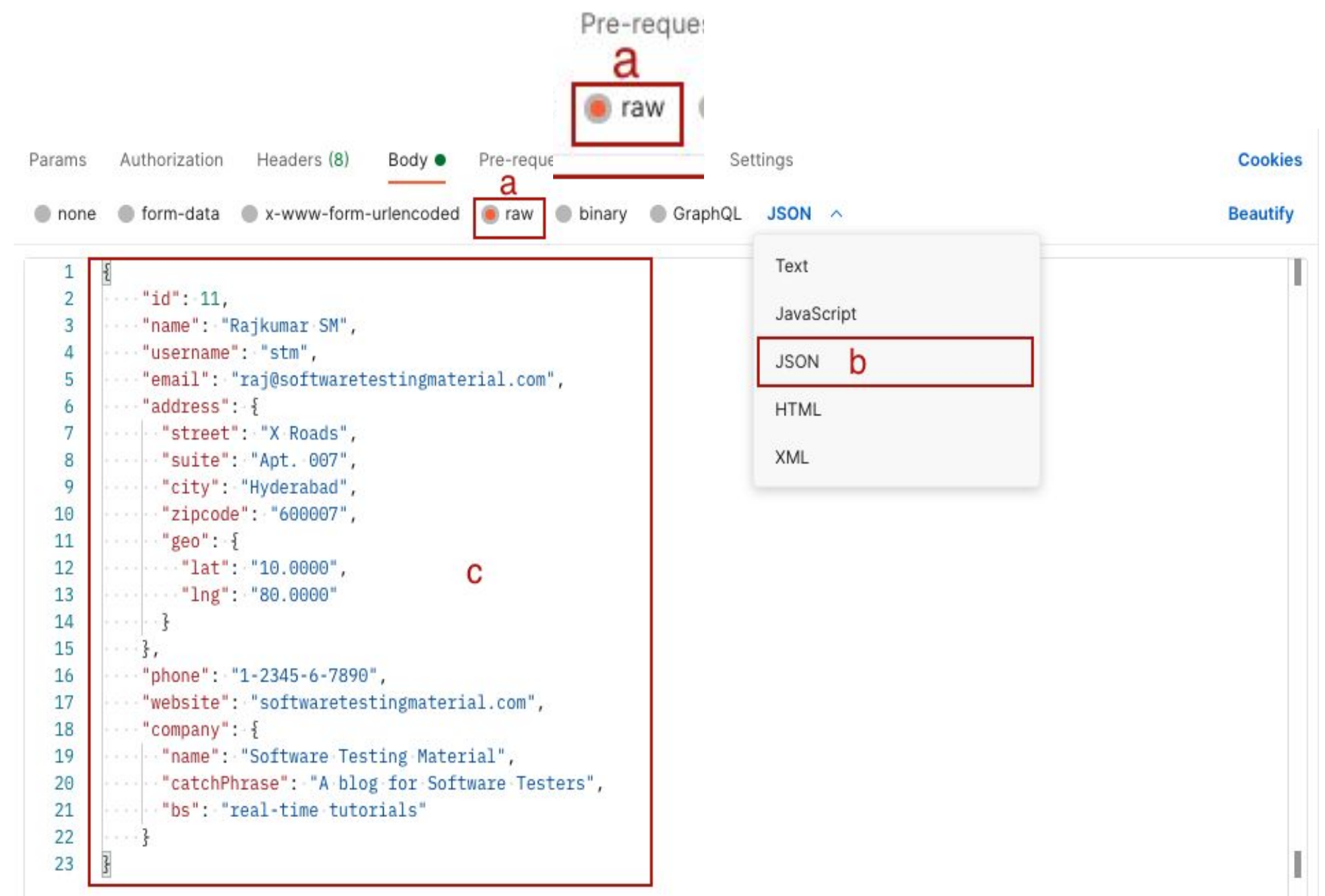
Passos

3. Os dados de uma solicitação POST não são transmitidos pela URL porque não são transmitidos com segurança, são transmitidos pelo **BODY**. Podemos enviar de diferentes formas:
 - Raw: as informações são enviadas como uma string do tipo texto, por meio de um arquivo JSON.
 - x-www-form-urlencoded: os dados são enviados como se fossem um formulário.

```
{
  "id": 11,
  "name": "Usuario SM",
  "username": "stm",
  "email": "stm@digitalhouse.com",
  "address": {
    "street": "X Roads",
    "suite": "Apt. 007",
    "city": "Hyderabad",
    "zipcode": "600007",
    "geo": {
      "lat": "10.0000",
      "lng": "80.0000"
    }
  },
  "phone": "1-2345-6-7890",
  "website": "digitalhouse.com",
  "company": {
    "name": "Testing I",
    "catchPhrase": "A blog for Software
Testers",
    "bs": "real-time tutorials"
  }
}
```

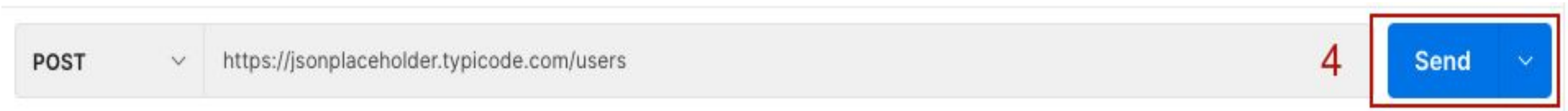
Passos

4. Neste exemplo, enviaremos os dados no formato Raw. Para isso, clique no corpo da solicitação e selecione a opção “raw” (a), depois “Json” (b). Por fim, copie e cole o exemplo fornecido no slide anterior no body (c).



Passos

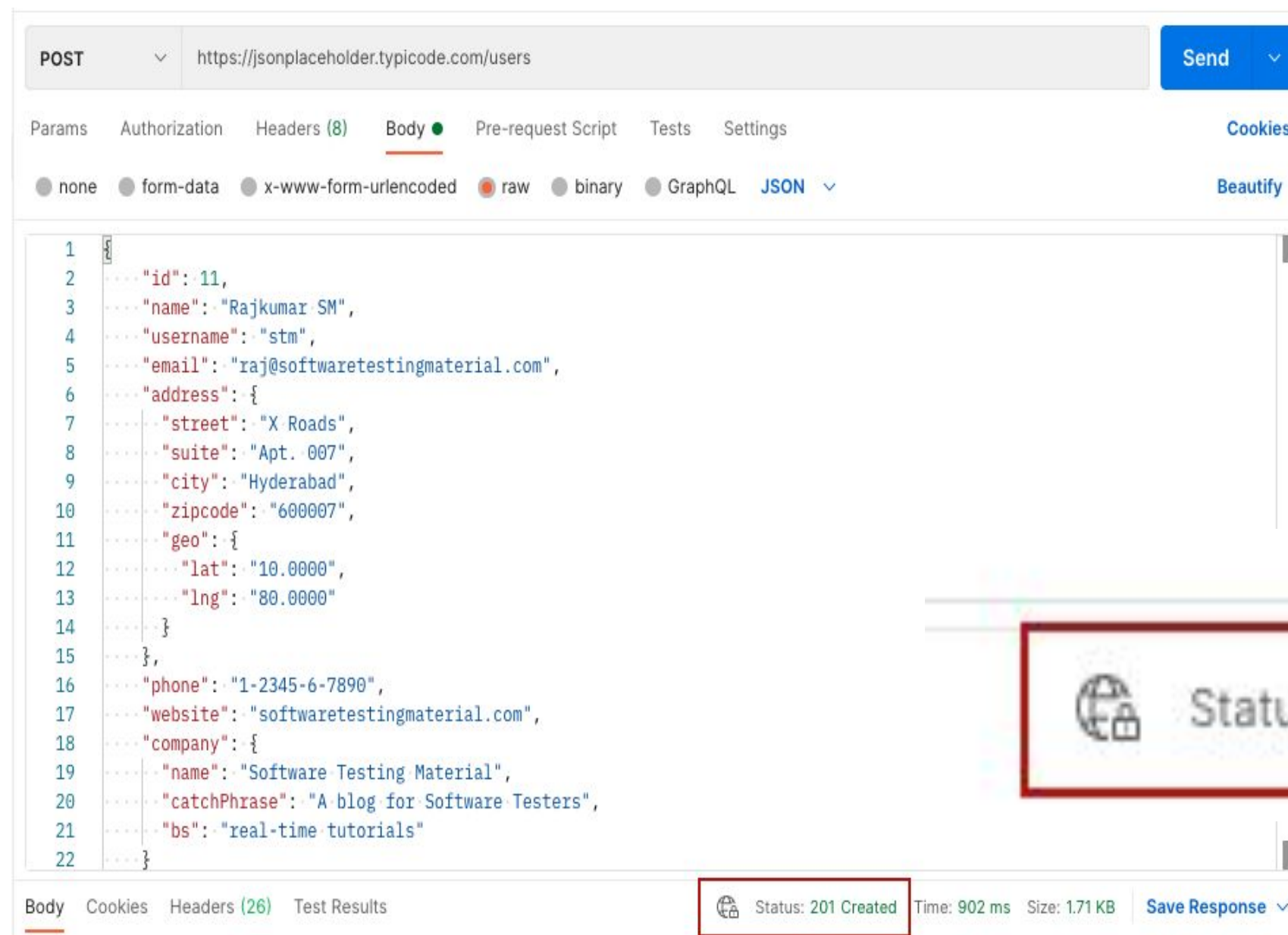
5. Clique em **ENVIAR** para enviar a solicitação ao servidor que hospeda a URL.



A screenshot of a REST client interface. The top bar shows a POST request to the URL `https://jsonplaceholder.typicode.com/users`. To the right of the URL is a red number '4'. On the far right, there is a blue button labeled 'Send' with a dropdown arrow, which is highlighted by a red rectangular box.

Passos

3 Se a mensagem **201 CREATED** for exibida, significa que a solicitação foi bem-sucedida.

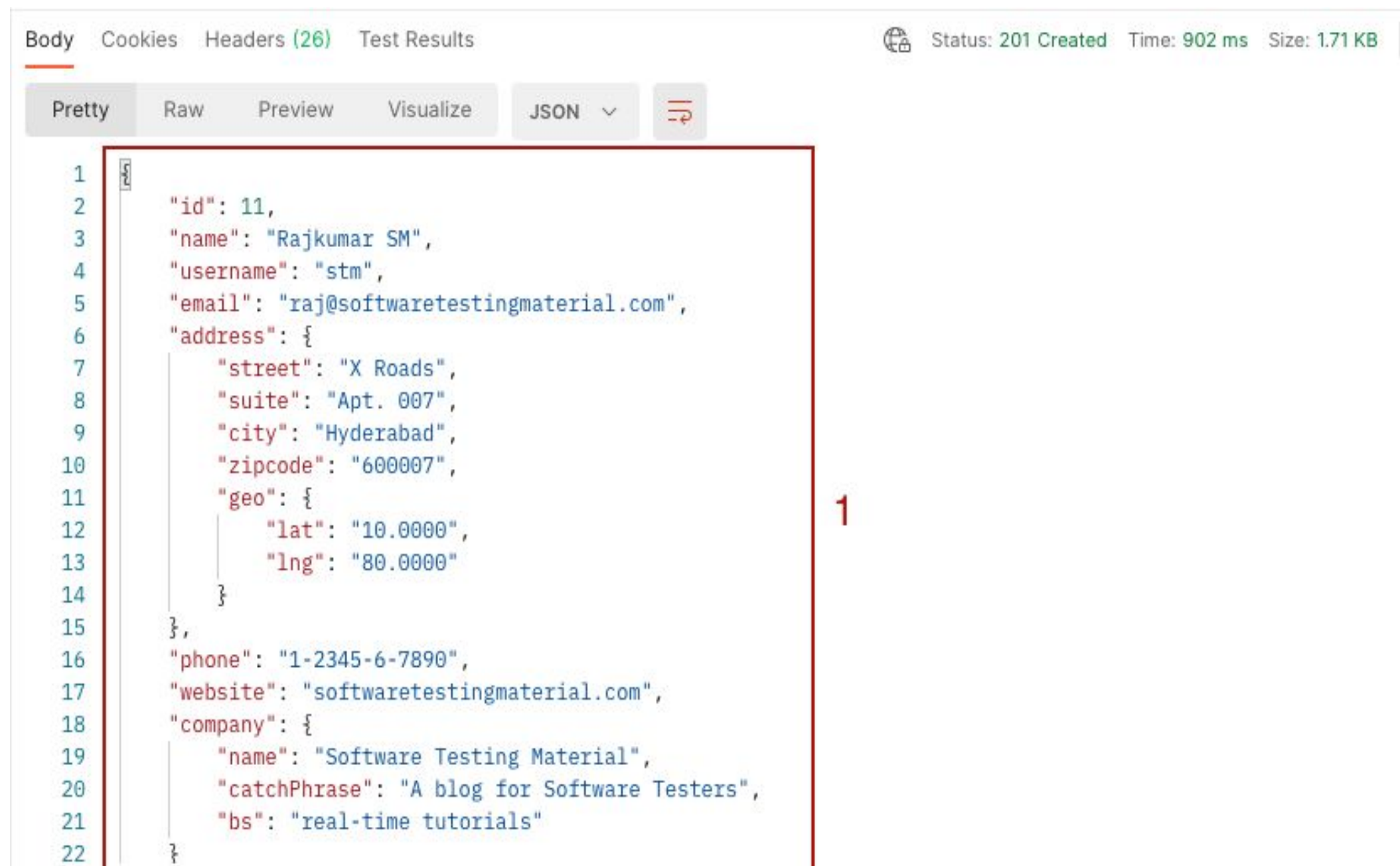


Resultado

Podemos ver vários dados relacionados à resposta do servidor:

1. **Response:** são as informações retornadas pelo servidor, que nos ajudam a validar se a criação foi bem-sucedida.

Geralmente, este método retorna os dados do usuário criado ou uma mensagem de criação bem-sucedida.



```
Body Cookies Headers (26) Test Results
Pretty Raw Preview Visualize JSON
1 {
2   "id": 11,
3   "name": "Rajkumar SM",
4   "username": "stm",
5   "email": "raj@softwaretestingmaterial.com",
6   "address": {
7     "street": "X Roads",
8     "suite": "Apt. 007",
9     "city": "Hyderabad",
10    "zipcode": "600007",
11    "geo": {
12      "lat": "10.0000",
13      "lng": "80.0000"
14    }
15  },
16  "phone": "1-2345-6-7890",
17  "website": "softwaretestingmaterial.com",
18  "company": {
19    "name": "Software Testing Material",
20    "catchPhrase": "A blog for Software Testers",
21    "bs": "real-time tutorials"
22  }
}
```


Resultado

Podemos ver vários dados relacionados à resposta do servidor:

2. **Tempo e tamanho da resposta:** com estes dados, podemos ver se o sistema está atendendo a algum dos requisitos não funcionais, como desempenho.



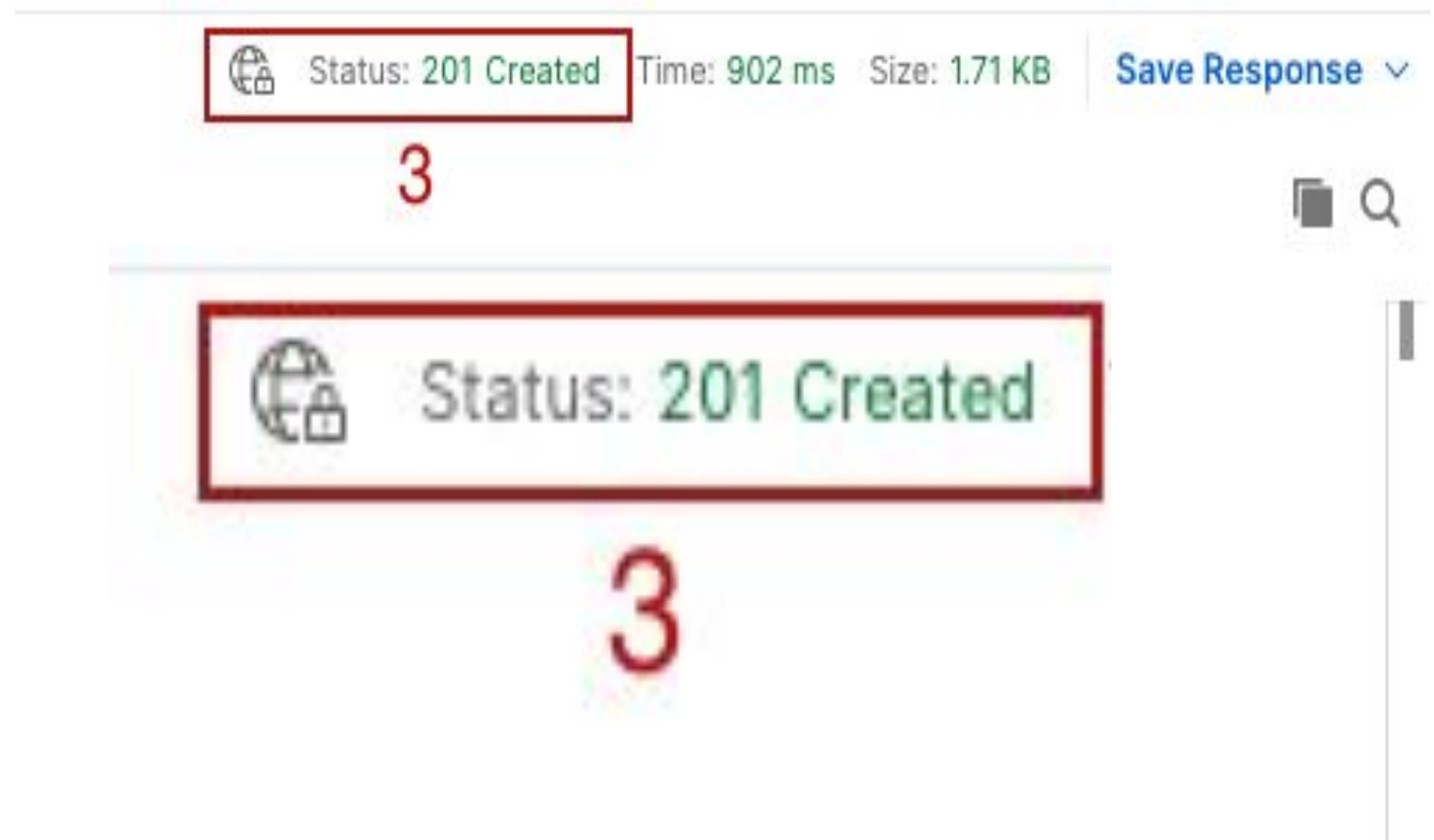
Resultado

Podemos ver vários dados relacionados à resposta do servidor:

3. Código de resposta:

quando solicita informação ao servidor, ele pode responder a diferentes códigos de status que informam o que aconteceu com a solicitação.

O código de resposta 201 nos indica que a criação foi bem-sucedida.



Muito obrigado!