

MINISTÉRIO DA SAÚDE
SECRETARIA EXECUTIVA
DEPARTAMENTO DE INFORMÁTICA DO SUS

Projeto LEITOS

Manual de Apoio para Consultar Informações
de LEITOS

Sumário

1. Introdução	3
2. Utilização da API.....	3
2.1. Campos Contidos na API	3
2.2. URL para consumo da API	4
2.3. Nomenclatura dos campos de Notificações	4
2.3.1. Dicionário dos dados de notificação.....	4
3. Configurando para Consumir Informações da API	7

1. Introdução

O objetivo deste manual é orientar as Unidades da Federação (Estados), a consumirem de forma automatizada e padronizada, os dados de notificação de COVID-19 e informações sobre o LEITOS, garantindo consistência das informações.

2. Utilização da API

API é um conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma baseado na Web. A sigla API refere-se ao termo em inglês "Application Programming Interface" que significa em tradução para o português "Interface de Programação de Aplicativos".

Através das APIs, os aplicativos podem se comunicar uns com os outros sem conhecimento ou intervenção dos usuários. Elas funcionam através da comunicação de diversos códigos, definindo comportamentos específicos de determinado objeto em uma interface.

2.1. Campos Contidos na API

Os campos abaixo, são informações disponibilizadas para LEITOS.

- "_index" : "leitos-ac",
- "_type" : "_doc",
- "_id" : "4Pm-n3lBgsCGahDPwViA",
- "_score" : 1.0,
- "_source" : {
- "estado" : "Acre",
- "estadoSigla" : "AC",
- "municipio" : "Assis Brasil",
- "cnes" : "2001020",
- "nomeCnes" : "UNIDADE MISTA DE ASSIS BRASIL",
- "dataNotificacao" : "2020-06-03T05:00:00.000Z",
- "dataAtualizacao" : "2020-06-03T20:12:40.312Z",
- "ofertaHospCli" : 12,
- "ofertaHospUti" : 0,
- "ofertaSRAGCli" : 0,
- "ofertaSRAGUti" : 0,
- "ocupHospCli" : 1,
- "ocupHospUti" : 0,
- "ocupSRAGCli" : 0,
- "ocupSRAGUti" : 0,
- "altas" : 16,
- "obitos" : 6,
- "ocupacaoInformada" : true

2.2. URL para consumo da API

Para consumir os dados disponibilizados pela API, utilizaremos a seguinte URL:

- https://elasticsearch-saps.saude.gov.br/leitos-uf/_search?pretty

2.3. Nomenclatura dos campos de Notificações

A URL retorna os campos listados abaixo no formato .JSON, o qual, poderá ser utilizado por qualquer ferramenta que consiga manipular dados em .JSON.

Exemplo: `index notificações-esusve*`

```
[
  [
    {
      "_index" : "leitos-ac",
      "_type" : "_doc",
      "_id" : "4Pm-n3IBgsCGahDPwViA",
      "_score" : 1.0,
      "_source" : {
        "estado" : "Acre",
        "estadoSigla" : "AC",
        "municipio" : "Assis Brasil",
        "cnes" : "2001020",
        "nomeCnes" : "UNIDADE MISTA DE ASSIS BRASIL",
        "dataNotificacao" : "2020-06-03T05:00:00.000Z",
        "dataAtualizacao" : "2020-06-03T20:12:40.312Z",
        "ofertaHospCli" : 12,
        "ofertaHospUti" : 0,
        "ofertaSRAGCli" : 0,
        "ofertaSRAGUti" : 0,
        "ocupHospCli" : 1,
        "ocupHospUti" : 0,
        "ocupSRAGCli" : 0,
        "ocupSRAGUti" : 0,
        "altas" : 16,
        "obitos" : 6,
        "ocupacaoInformada" : true
      }
    },
  ]
]
```

2.3.1. Dicionário dos dados de notificação

Esta seção descreve o dicionário dos dados da tela do sistema e-SUS VE relativa ao cadastramento de notificações.

Campo API	<code>_index</code>
Objetivo:	Informações do Elastic que informa qual o index está sendo consultado

Campo API	<code>_type</code>
Objetivo:	Informações do Elastic que informa qual o tipo de documento que está sendo consultado
Campo API	<code>_id</code>
Objetivo:	String que informa a chave específica do registro.
Campo API	<code>_score</code>
Objetivo:	Informação do Elastic que concede uma pontuação quando a informação é pesquisada
Campo API	<code>_source</code>
Objetivo:	Array que contem um conjunto de dados recuperados.
Campo API	<code>estado</code>
Objetivo:	String
Campo API	<code>estadoSigla</code>
Objetivo:	String
Campo API	<code>municipio</code>
Objetivo:	String
Campo API	<code>cnes</code>
Objetivo:	String
Campo API	<code>nomeCnes</code>
Valores possíveis	Datetime
Campo API	<code>dataNotificacoes</code>
Valores possíveis	Datetime
Campo API	<code>dataAtualizacao</code>
Valores possíveis	Datetime
Campo API	<code>ofertaHospCli</code>
Valores possíveis	String
Campo API	<code>ofertaHospUti</code>
Valores possíveis	String
Campo API	<code>ofertaSRAGCli</code>
Valores possíveis	String
Campo API	<code>ofertaSRAGUti</code>
Valores possíveis	String

Campo API	ocupHospCli
Valores possíveis	String

Campo API	ocupHospUti
Valores possíveis	String

Campo API	ocupSRAGCli
Valores possíveis	String

Campo API	ocupSRAGUti
Valores possíveis	String

Campo API	altas
Valores possíveis	String

Campo API	obitos
Valores possíveis	String

Campo API	ocupacaoInformada
Valores possíveis	String

3. Configurando para Consumir Informações da API

Para saber se o arquivo foi carregado com sucesso use o seguinte comando:

O documento irá auxiliar no consumo de dados da API via GET e POST utilizando o sistema do POSTMAN.

1. Ao acesso o POSTMAN, Criaremos uma aba para executar um comando em GET, esse comando ira consultar no máximo 10 mil registro.
 - a. Configurações:
 - i. **Authorization:** Basic Auth
 1. Username e Password

The screenshot shows the Postman interface for a GET request. The URL is `https://elasticsearch-saps.saude.gov.br/esusve-ce/_search?pretty`. The 'Authorization' tab is selected, and the type is set to 'Basic Auth'. A warning message states: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. [Learn more about variables](#)'. The 'Username' and 'Password' fields are visible, with a 'Show Password' checkbox checked.

b. Após o preenchimento dos dados, o sistema irá retornar até 10 mil registro.

▶ Elastic - API - 1º Passo Comments 0 | Examples 0 ▾

GET https://elasticsearch-saps.saude.gov.br/esusve-ce/_search?pretty Send ▾ | Save ▾

Params ● Authorization ● Headers (8) Body Pre-request Script Tests Settings Cookies Code

Query Params

	KEY	VALUE	DESCRIPTION	⋮	Bulk Edit
<input checked="" type="checkbox"/>	pretty				
	Key	Value	Description		

Body Cookies (1) Headers (8) Test Results Status: 200 OK Time: 62ms Size: 17.13 KB Save Response ▾

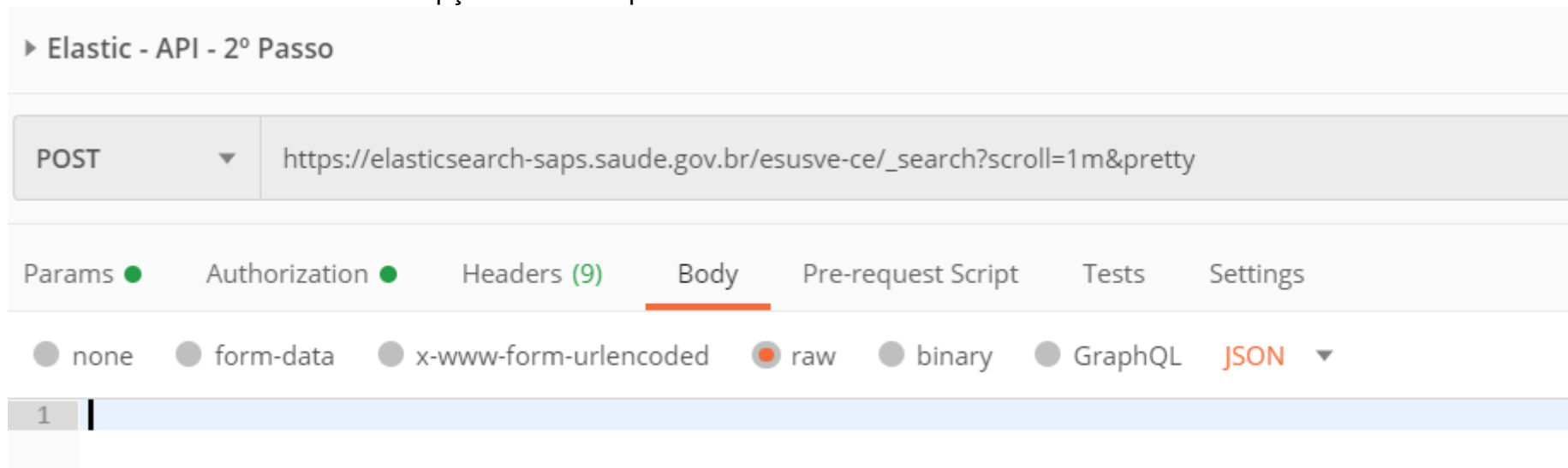
Pretty Raw Preview Visualize JSON ▾ 📄 🔍

```

1  {
2  "took": 3,
3  "timed_out": false,
4  "_shards": {
5    "total": 1,
6    "successful": 1,
7    "skipped": 0,
8    "failed": 0
9  },
10 "hits": {
11   "total": {
12     "value": 10000,
13     "relation": "gte"
14   },
15   "max_score": 1.0,
16   "hits": [
17     {
18       "_index": "esusve-ce",
19       "type": "doc".

```


2. Para consultar acima de 10 mil registro é necessário utilizar a funcionalidade do Scroll do Elastic. Essa funcionalidade irá recuperar 10 mil registro a cada requisição, logo, se tiver 70 mil registro em um índice, terá que executar uma consulta com o Scroll 7 vezes.
 - a. Utilizando o POSTMAN, criamos uma nova ABA para utilização do POST
 - b. Configurações:
 - i. **Authorization:** Basic Auth
 1. Username e Password
 - ii. Body
 1. Marcar a Opção raw
 2. Marcar a opção de escrita para JSON



c. Após esse processo temos que adicionar as seguintes variáveis no JSON

```
1 {
2   "size": 500,
3   "query": {
4     "match_all": {}
5   }
6 }
```

i. Com o código acima, poderá também realizar pesquisas para um determinado campo do índice, conforme imagem abaixo:

```
1 {
2   "size": 500,
3   "query": {
4     "match": {
5       "municipioNotificacao": "Fortaleza"
6     }
7   }
8 }
```

d. Após a inclusão do código, irá retornar os dados, conforme abaixo:

```
1 {
2   "_scroll_id": "DXF1ZXJ5QW5kRmV0Y2gBAAAAAAAAFBd8WZ1ZuMm9FRD1TdnVDN2JzUGdWRGZrZw==",
3   "took": 19,
4   "timed_out": false,
5   "_shards": {
6     "total": 1,
7     "successful": 1,
8     "skipped": 0,
9     "failed": 0
10  },
11  "hits": {
12    "total": {
13      "value": 11511,
14      "relation": "eq"
15    },
16    "max_score": 0.32738438,
17    "hits": [
18      {
19        "_index": "esusve-ce",
```

3. Após a pesquisa, caso possua mais de 10mil registros, deverá realizar a pagina do scrolling para que a API retorne os demais dados

a. Criamos uma ABA no POSTMAN e incluímos os dados:

b. Configurações:

i. **Authorization:** Basic Auth

1. Username e Password

ii. Body

1. Marcar a Opção raw

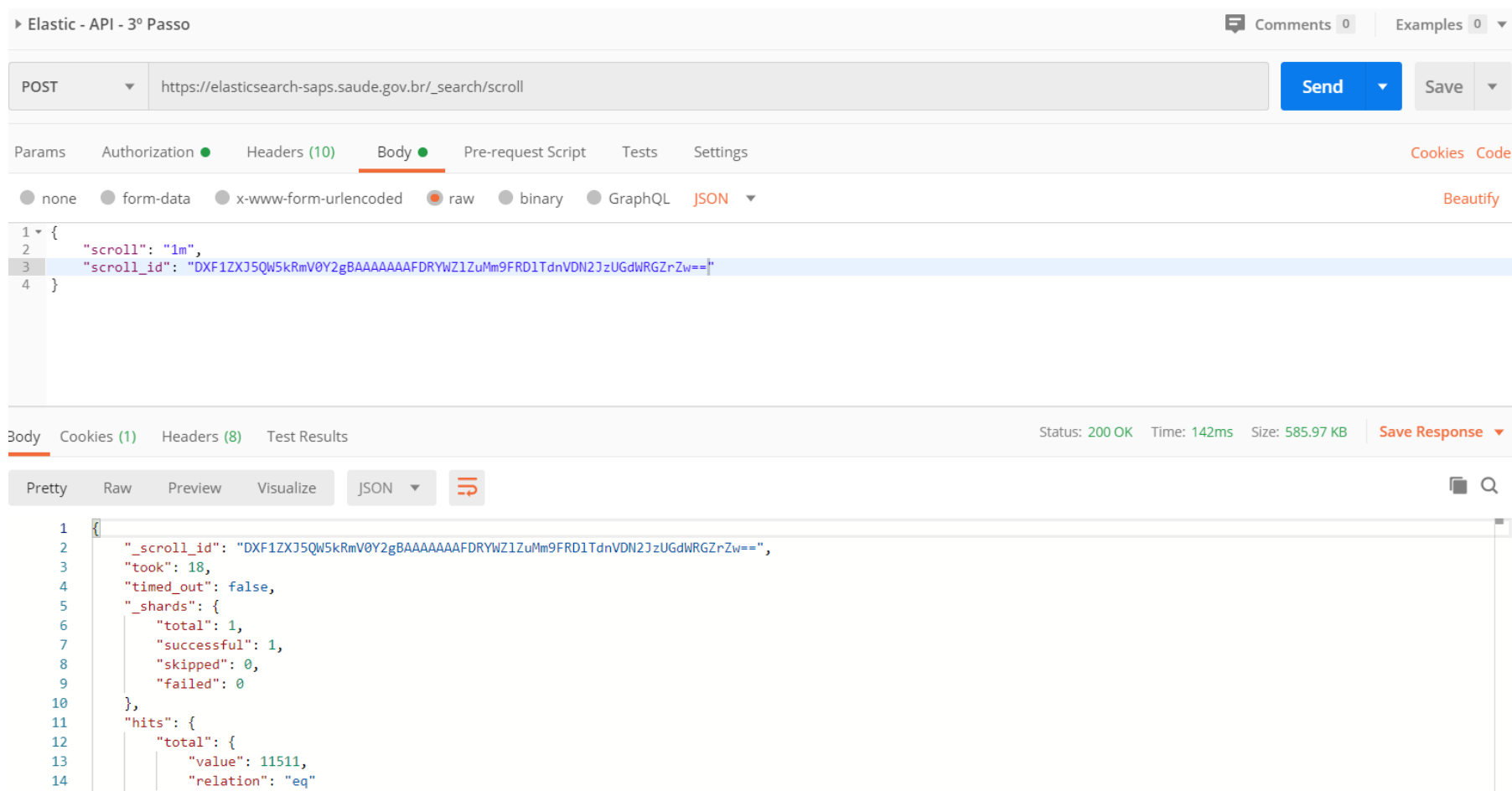
2. Marcar a opção de escrita para JSON

c. No campo de inclusão do código, colocar os seguintes dados:

```
1 {  
2   "scroll": "1m",  
3   "scroll_id": "DXF1ZXJ5QW5kRmV0Y2gBAAAAAAAAE09AWZzFXQk1SQXVRVTZmVFJvWw11anIydw=="  
4 }
```

i. A tag do SCROLL_ID ela é recuperada do PASSO 2, que retorna a tag _SCROLL_ID, essa informação que irá fazer com que o scrolling realize o retorno dos dados de 10 em 10 mil.

d. Abaixo segue a consulta e o retorno da API:



The screenshot displays a REST client interface for a POST request to the URL `https://elasticsearch-saps.saude.gov.br/_search/scroll`. The request body is a JSON object with the following structure:

```
1 {
2   "scroll": "1m",
3   "scroll_id": "DXF1ZXJ5QW5kRmV0Y2gBAAAAAAAAFDRYWZlZuMm9FRD1TdnVDN2JzUGdWRGZrZW=="
4 }
```

The response status is 200 OK, with a time of 142ms and a size of 585.97 KB. The response body is a JSON object with the following structure:

```
1 {
2   "_scroll_id": "DXF1ZXJ5QW5kRmV0Y2gBAAAAAAAAFDRYWZlZuMm9FRD1TdnVDN2JzUGdWRGZrZW==",
3   "took": 18,
4   "timed_out": false,
5   "_shards": {
6     "total": 1,
7     "successful": 1,
8     "skipped": 0,
9     "failed": 0
10  },
11  "hits": {
12    "total": {
13      "value": 11511,
14      "relation": "eq"

```

1. Nesse 3º passo a URL muda, não precisa mais passar o índice pela URL.
2. A tag SCROLL_ID ela é dinâmica, para cada consulta tem que incluir o _SCROLL_ID da anterior.