

DOCUMENTAÇÃO REST API

Este documento explica com exemplos, como utilizar os recursos disponíveis na REST API de registros. Assim como, as formas de se realizar uma requisição e suas possíveis respostas.

1. Método GET

Requisição

Requisição para exibir todos os registros da tabela no formato Json.

Method	URL
GET	/registros

Resposta

Como resposta, é obtida a lista de todos os registros da tabela.

Status	Response Body
200 Ok	<pre>[{ "id": 1, "type": "dúvida", "message": "Duis bibendum, felis sed interdum venenatis, turpis enim blandit mi, in porttitor pede justo eu massa. Donec dapibus. Duis at velit eu est congue elementum.", "is_identified": "0", "whistleblower_name": null, "whistleblower_birth": null, "created_at": "2021-06-30 18:47:23", "deleted": "1" }, { "id": 2, "type": "dúvida", "message": "Donec ut mauris eget massa tempor convallis.", "is_identified": "1", "whistleblower_name": "Carly Insko", "whistleblower_birth": "1988-03-19", "created_at": "2020-12-19 09:55:55",</pre>

	"deleted": "1" },...
--	-------------------------

Parâmetros Query (Filtros Opcionais):

- deleted (booleano): Filtra registros pelo valor do campo "deleted".

Exemplo: /registros?deleted=1

- type (string): Filtra registros pelo valor do campo "type".

Exemplo: /registros?type=denuncia

- order_by (string): Ordena a lista de registros com base no campo especificado.

Exemplo: /registros?order_by=created_at

- limit (inteiro): Limita o número de registros retornados.

Exemplo: /registros?limit=5

- offset (inteiro): Define o deslocamento para a consulta, deve ser utilizado juntamente com o filtro limit.

Exemplo: /registros?limit=10&offset=2

2. Método GET/{id}

Requisição

Requisição para exibir um registro específico da tabela.
Caso o registro não exista a API retornará um erro 404.

Method	URL
GET	/registros/{id}

Resposta

Como resposta, é obtido o registro com o id informado na requisição.

Status	Response Body
200 Ok	<pre>{ "id": 2, "type": "duvida", "message": "Donec ut mauris eget massa tempor convallis.", "is_identified": "1", "whistleblower_name": "Carly Insko", "whistleblower_birth": "1988-03-19", "created_at": "2020-12-19 09:55:55", "deleted": "1" }</pre>

3. Método POST

Requisição

Requisição para inserir um novo registro na tabela.

Method	URL
POST	/registros

Request Body
<pre>{ "type": "sugestao", "message": "Teste", "is_identified": "1", "whistleblower_name": "Felipe", "whistleblower_birth": "1999-05-04" }</pre>

Resposta

Como resposta, obtém-se o objeto criado e status code 201 Created.

Status	Response Body
201 Created	<pre>{ "deleted": 0, "type": "sugestao", "message": "Teste", "is_identified": "1", "created_at": "2023-08-27T00:49:27.712324Z", "whistleblower_name": "Felipe", "whistleblower_birth": "1999-05-04", "id": 14 }</pre>

Formato dos dados:

'type' = 'denuncia', 'sugestao' ou 'duvida',
 'message' = 'string',
 'is_identified' = 'boolean',
 'whistleblower_name' = 'string',
 'whistleblower_birth' = 'date'.

Regras de Inserção:

O campo "deleted" é definido por padrão como 0;
 O campo "created_at" é preenchido com a data de criação;
 O campo "id" é do tipo auto incremento;
 Os campos "whistleblower_name" e "whistleblower_birth" só serão preenchidos se o campo "is_identified" for true, se for falso a API retornará erro 400.

Requisição

Exemplo de requisição inválida para inserir um novo registro na tabela.

Method	URL
POST	/registros

Request Body

```
{
  "type": "sugestao",
  "message": "Teste",
  "is_identified": "1"
}
```

Resposta

Como resposta, obtém-se o erro code 400 Bad Request.

Status	Response Body
400 Bad Request	<pre>{ "message": "Erro nos dados da requisição", "errors": { "whistleblower_name": ["The whistleblower name field is required when is identified is 1."], "whistleblower_birth": ["The whistleblower birth field is required when is identified is 1."] } }</pre>

4. Método PUT

Requisição

Exemplo de Requisição para atualizar um registro: /registro/{id}

Caso o registro não exista a API retornará um erro 404.

O PUT dessa API pode alterar apenas um campo ou quantos campos forem precisos.

Method	URL
POST	/registros/{id}

Request Body

```
{
  "type": "duvida"
}
```

Resposta

Como resposta, obtém-se o objeto atualizado e status code 200 Ok.

Status	Response Body
200 OK	<pre>{ "id": 3, "type": "duvida", "message": "sadas", "is_identified": "1", "whistleblower_name": "Parsifal Dorkens", "whistleblower_birth": "1983-02-19", "created_at": "2020-09-04 12:41:53", "deleted": "0" }</pre>

Regras de Atualização:

Se o campo "is_identified" do registro for false, não será possível atualizar os campos "whistleblower_name" e "whistleblower_birth", esses campos só podem ser atualizados se o campo "is_identified" for true.

É possível atualizar um registro com campo "is_identified" de false para true, mas será preciso informar "whistleblower_name" e "whistleblower_birth".

Requisição

Exemplo de requisição inválida para atualizar um registro

Method	URL
POST	/registros/5

Request Body
<pre>{ "is_identified": 1 }</pre>

Resposta

Como resposta, obtém-se o erro code 400 Bad Request.

Status	Response Body
400 Bad Request	{ "message": "Campos whistleblower_name e whistleblower_birth são obrigatórios quando is_identified é true" }

5. Método DELETE

Requisição

Exemplo de Requisição para deletar um registro: /registro/{id}
Caso o registro não exista a API retornará um erro 404.

Method	URL
DELETE	/registros/{id}

Resposta

Como resposta, obtém-se o objeto atualizado e status code 200 Ok.

Status	Response Body
200 OK	{ "message": "Registro deletado com sucesso" }

Regras de Deletar:

O método Delete é apenas um soft delete, ele apenas altera o campo “deleted” do registro para 0.