



CRUD Local

GET TU10-admin-mostrar-detahes-avaliacoes-municipio-nao-existente

```
{{sbms-local}}/10
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Descrição: É requisitado a url 'localhost:3000/locations/1555', com o token de autenticação do usuário ADMIN, a fim de se obter os detalhes das avaliações daquele município que não existe.

Resultado: STATUS 404 NOT FOUND, é retornado "local não existente" para o id 1555.

HEADERS

Content-Type

application/json

Authorization

{{authAdminC}}

Example Request

TU10-admin-mostrar-detahes-avaliacoes-municipio-nao-existente

```
curl --location -g --request GET '{{sbms-local}}/10' \  
--header 'Content-Type: application/json' \  
--header 'Authorization: {{authAdminC}}'
```

GET TU11-admin-mostrar-detahes-avaliacoes-municipio-x

```
{{sbms-local}}/1
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Descrição: É requisitado a url 'localhost:3000/locations/1', com o token de autenticação do usuário ADMIN, a fim de se obter os detalhes das avaliações daquele município.

HEADERS

Content-Type

application/json

Authorization

{{authAdminH}}

Example Request

TU11-admin-mostrar-detalhes-avaliacoes-municipio-x

```
curl --location -g --request GET '{{sbms-local}}/1' \
--header 'Content-Type: application/json' \
--header 'Authorization: {{authAdminH}}'
```

GET TU12-admin-mostrar-locais

{{sbms-local}}

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Descrição: É requisitado a url 'localhost:3000/locations' juntamente com o JSON , com o token de autenticação do usuário ADMIN, a fim de mostrar todos os locais cadastrados do sistema.

Resultado: STATUS 200 OK, Utilizando o login do admin, é mostrado todos os locais cadastrados no sistema.

HEADERS

Content-Type

application/json

Authorization

{{authAdminH}}

Example Request

TU12-admin-mostrar-locais

```
curl --location -g --request GET '{{sbms-local}}' \
--header 'Content-Type: application/json' \
```



POST TU13-admin-criar-novo-local

```
{{sbms-local}}
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Body: São dados, o "corpo" do teste, no nosso caso o JSON referente aos dados que serão requisitados e testados.

Descrição: É requisitado a url 'localhost:3000/locations', com o token de autenticação do usuário ADMIN e o JSON com os dados do local a ser cadastrado , a fim de cadastrar um novo local.

Resultado: Status 201 created, Admin consegue criar novo local.

Relatório de Bugs:

- 16/05: Cannot POST.

HEADERS

Content-Type

application/json

Authorization

{{authAdminH}}

BODY raw

```
{
  "municipio": "Município 4",
  "escola": "Escola 4"
}
```

Example Request

TU13-admin-criar-novo-local

```
curl --location -g --request POST '{{sbms-local}}' \
--header 'Content-Type: application/json' \
--header 'Authorization: {{authAdminH}}' \
--data-raw '{
  "municipio": "Município 4",
  "escola": "Escola 4"
}'
```



```
{{sbms-local}}
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Body: São dados, o "corpo" do teste, no nosso caso o JSON referente aos dados que serão requisitados e testados.

Descrição: É requisitado a url 'localhost:3000/location', com o token de autenticação do usuário ADMIN e o JSON com os dados de um local já cadastrado .

Resultado: Status 409 Confict, Admin não consegue criar novo local com os mesmos dados já criados.

Relatorio de bugs:

- 09-05 Status 201 created, Admin consegue criar novo local com os mesmos dados já criados.(To fix)
- 18-05 Status 409 Confict, Admin não consegue criar novo local com os mesmos dados já criados.
(Fixed - Demetrius Panovitch)

HEADERS

Content-Type

application/json

Authorization

{{authAdminC}}

BODY raw

```
{
  "municipio": "Município 3",
  "escola": "Escola 3"
}
```

Example Request

TU14-admin-criar-novo-local-ja-existente

```
curl --location -g --request POST '{{sbms-local}}' \
--header 'Content-Type: application/json' \
--header 'Authorization: {{authAdminC}}' \
--data-raw '{
  "municipio": "Município 3",
  "escola": "Escola 3"
}'
```



```
{{sbms-local}}/5
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Body: São dados, o "corpo" do teste, no nosso caso o JSON referente aos dados que serão requisitados e testados.

Descrição: É requisitado a url "localhost:3000/locations/5", com o token de autenticação do usuário ADMIN, a fim de editar o local com o "id 5".

Resultado: STATUS 200 OK, Admin consegue editar os dados do local.

HEADERS

Content-Type

application/json

Authorization

{{authAdminC}}

BODY raw

```
{
  "municipio": "Campo Grande update",
  "escola": "CFCM"
}
```

Example Request

TU15-admin-editar-local

```
curl --location -g --request PUT '{{sbms-local}}/5' \
--header 'Content-Type: application/json' \
--header 'Authorization: {{authAdminC}}' \
--data-raw '{
  "municipio": "Campo Grande update",
  "escola": "CFCM"
}'
```

PUT TU16-admin-editar-local-nao-existente

```
{{sbms-local}}/9
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.



Descrição: É requisitado a url "localhost:3000/locations/9", com o token de autenticação do usuário ADMIN e o JSON com os dados a serem editados, a fim de editar o local com o "id 9" (não existente).

Resultado: STATUS 404 NOT FOUND, Admin não consegue editar local não existente.

HEADERS

Content-Type

application/json

Authorization

{{authAdminC}}

BODY raw

```
{
  "municipio": "Município 2",
  "escola": "Escola 3"
}
```

Example Request

TU16-admin-editar-local-nao-existente

```
curl --location -g --request PUT '{{sbms-local}}/9' \
--header 'Content-Type: application/json' \
--header 'Authorization: {{authAdminC}}' \
--data-raw '{
  "municipio": "Município 2",
  "escola": "Escola 3"
}'
```

GET TU17-usuario-mostrar-detahes-avaliacoes-municipio-nao-existente

{{sbms-local}}/15

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Body: São dados, o "corpo" do teste, no nosso caso o JSON referente aos dados que serão requisitados e testados.

Descrição: É requisitado a url 'localhost:3000/locations/15', com o token de autenticação do usuário (coordenador, dentista, assistente e arrolador), á fim de se obter os detalhes das avaliações daquele município que não existe.



HEADERS

Content-Type

application/json

Authorization

{{authCoordH}}

Example Request

TU17-usuario-mostrar-detallhes-avaliacoes-municipio-nao-existente

```
curl --location -g --request GET '{{sbms-local}}/15' \  
--header 'Content-Type: application/json' \  
--header 'Authorization: {{authCoordH}}'
```

GET TU18-usuario-mostrar-detallhes-avaliacoes-municipio-x

{{sbms-local}}/1

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Descrição: É requisitado a url 'localhost:3000/locations/1', com o token de autenticação do usuário (coordenador, dentista, assistente e arrolador), á fim de se obter os detalhes da avaliação daquele município de ID 1.

Resultado: STATUS 200 OK, Mostra todas as avaliações coletadas no município de ID 1.

HEADERS

Content-Type

application/json

Authorization

{{authAdminH}}

Example Request

TU18-usuario-mostrar-detallhes-avaliacoes-municipio-x

```
curl --location -g --request GET '{{sbms-local}}/1' \  
--header 'Content-Type: application/json' \  
--header 'Authorization: {{authAdminH}}'
```



GET TU19-usuario-mostrar-locais

```
{{sbms-local}}
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Descrição: É requisitado a url 'localhost:3000/locations', com o token de autenticação do usuário (coordenador, dentista, assistente e arrolador), a fim de mostrar todos os locais cadastrados do sistema.

Resultado: STATUS 200 OK, Utilizando o login do usuario, é mostrado todos os locais cadastrados no sistema.

HEADERS

Content-Type

application/json

Authorization

{{authCoordH}}

Example Request

TU19-usuario-mostrar-locais

```
curl --location -g --request GET '{{sbms-local}}' \
--header 'Content-Type: application/json' \
--header 'Authorization: {{authCoordH}}'
```

POST TU20-usuario-criar-novo-local

```
{{sbms-local}}
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Body: São dados, o "corpo" do teste, no nosso caso o JSON referente aos dados que serão requisitados e testados.

Descrição: É requisitado a url 'localhost:3000/locations', com o token de autenticação do usuário (coordenador, dentista, assistente e arrolador), a fim de cadastrar um novo local.

Resultado: Status 401 unauthorized, Usuário não consegue criar novo local.



application/json

Authorization

{{authCoordH}}

BODY raw

```
{
  "municipio": "Campo Grande",
  "escola": "CFCM"
}
```

Example Request

TU20-usuario-criar-novo-local

```
curl --location -g --request POST '{{sbms-local}}' \
--header 'Content-Type: application/json' \
--header 'Authorization: {{authCoordH}}' \
--data-raw '{
  "municipio": "Campo Grande",
  "escola": "CFCM"
}'
```

PUT TU21-usuario-editar-local

{{sbms-local}}/6

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Body: São dados, o "corpo" do teste, no nosso caso o JSON referente aos dados que serão requisitados e testados.

Descrição: É requisitado a url 'localhost:3000/locations/6', com o token de autenticação do usuário (coordenador, dentista, assistente e arrolador), a fim de editar o local com ID 6.

Resultado: Status 401 unauthorized, Usuário não consegue editar local.

HEADERS

Content-Type

application/json

Authorization

{{authCoordH}}



```
{
  "municipio": "Campo Grande",
  "escola": "CFCM"
}
```

Example Request

TU21-usuario-editar-local

```
curl --location -g --request PUT '{{sbms-local}}/6' \
--header 'Content-Type: application/json' \
--header 'Authorization: {{authCoordC}}' \
--data-raw '{
  "municipio": "Campo Grande",
  "escola": "CFCM"
}'
```

GET heroku-mostrar-locais

```
{{heroku}}locations
```

Foram executados os testes TU12 e TU19, nesse caso utilizando a URL do Heroku (<http://saude-bucal-backend.herokuapp.com/>) e são mostrados todos os locais cadastrados, com todos os usuários.

HEADERS

Authorization

Bearer

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiJEsImhhdCI6MTY1NTc0NzU2NiwiZXhwIjoxNjU1ODMzOTY2fQ.G3eYp
yRyKknkZQUhwAs-dft-7PQgIPs5H1bBkoeaa3Qo
```

Example Request

heroku-mostrar-locais

```
curl --location -g --request GET '{{heroku}}locations' \
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiJEsImhhdCI6MTY1NTc0N
```





```
{{heroku}}locations
```

Foram executados os testes TU13 e TU20, nesse caso utilizando a URL do Heroku (<http://saude-bucal-backend.herokuapp.com/>) e foi possível criar novos locais com todos os usuários (que têm permissão).

HEADERS

Authorization

Bearer

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiJEsImhhdCI6MTY1NDU0MTA4MSwiZXhwIjoxNjU0NjI3NDgxfQ.mmQBP9BdLhtKHnD2hmkEadHzHfjE-DaB4zi_4c7u6AI
```

BODY raw

```
{
  "escola": "Escola 5",
  "municipio": "Município 5"
}
```

Example Request

heroku-criar-locais

```
curl --location -g --request POST '{{heroku}}locations' \
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiJEsImhhdCI6MTY1NDU0MTA4MSwiZXhwIjoxNjU0NjI3NDgxfQ.mmQBP9BdLhtKHnD2hmkEadHzHfjE-DaB4zi_4c7u6AI' \
--data-raw '{
  "escola": "Escola 5",
  "municipio": "Município 5"
}'
```



PUT heroku-editar-locais

```
{{heroku}}locations/4
```

Foram executados os testes TU15 e TU21, nesse caso utilizando a URL do Heroku (<http://saude-bucal-backend.herokuapp.com/>) e é possível editar os locais com os usuários que têm permissão.

HEADERS

Authorization



BP9BdLhtKHnD2hmkEadHzHfjE-DaB4zi_4c7u6Al

BODY raw

```
{
  "escola": "Escola 4",
  "municipio": "Municipio 4"
}
```

Example Request

heroku-editar-locais

```
curl --location -g --request PUT '{{heroku}}locations/4' \
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiJEsImhdCI6MTY1NDU0M
--data-raw '  {
    "escola": "Escola 4",
    "municipio": "Municipio 4"
  }'
```