

Back-end para o Curso de Angular

Seviços

url: http://localhost:5000

Verbo	Url	Precisa Autenticação
POST	api/v1/authenticate	false
GET	api/v1/authenticate/user	true
PUT	api/v1/profile	true
GET	api/v1/pets/abandoned	false
RESOURCE	api/v1/pets	true
PUT	api/v1/pets/adopt/id	true
PUT	api/v1/pets/abandon/id	true

- Nas requisições POST, PUT e DEL deve ter o cabeçalho **Content-Type** e **Accept** com o valor **application/json**
- Todos serviços autenticados devem colocar no cabeçalho da requisição o token no formato **Authorization: Bearer token**
- Todas os serviços autenticados retorna o seguinte erro se o token não for informado na requisição

HTTP_CODE 400

```
{
  "error": "token_not_provided"
}
```

POST api/v1/authenticate

Request

```
{ email: "tutorial@prodeb.ba.gov.br", password: "secret" }
```

Response

200

```
{ token: "token", user: { id, name, email, created_at, updated_at, roles: [id, slug, title ] } }
```

401

```
{ error: "messages.login.invalidCredentials" }
```

422

```
{
  "email": [
    "O campo Email é obrigatório."
  ],
  "password": [
    "O campo Senha é obrigatório."
  ]
}
```

GET api/v1/authenticate/user

Request

```
{ }
```

Response

200

```
{ token: "token", user: { id, name, email, created_at, updated_at, roles: [id, slug, title ] } }
```

PUT api/v1/profile

Request

```
{ name: "String", email: "String", ?password: "String", ?password_confirm: "String" }
```

- password e password_confirm são opcionais porém se informar um tem que informar o outro

Response

200

```
{
  token: "String",
  user: {
    id: "Integer", name: "String", email: "String",
    created_at: "String Date ISO 8601", updated_at: "String Date ISO
8601",
    roles: [
      { id: "Integer", slug: "String", titleslug: "String" }
    ]
  }
}
```

GET api/v1/pet/abandoned

Request

```
{ ?page: "Integer", ?perPage: "Integer", ?name: "String", ?type: "String", ?
minAge: "Integer", ?maxAge: "Integer" }
```

Response

200

```
[ { id: "Integer", name: "String", type: "String", age: "Integer", owner: {
id: "Integer", name: "String" } } ]
```

200 (quando informado page e perPage)

```
[
  {
```

```
    items: [
      {
        id: "Integer", name: "String", type: "String",
        age: "Integer",
        owner: {
          id: "Integer",
          name: "String"
        }
      },
    ],
    total: "Integer"
  }
]
```

PUT api/v1/pet/adopt/id

Request

```
{ }
```

Response

200

```
{ id: "Integer", name: "String", type: "String", age: "Integer", owner_id:
  "Integer" }
```

PUT api/v1/pet/abandon/id

Request

```
{ }
```

Response

200

```
{ id: "Integer", name: "String", type: "String", age: "Integer", owner_id:
  "Integer" }
```

Resource: Pets

action	http verb	url	input
search	GET	api/v1/pets	{ page, per_page, name, type, minAge, maxAge }
create	POST	api/v1/pets	{ name, type, age }
update	PUT	api/v1/pets/id	{ name, type, age }
delete	DELETE	api/v1/pets/id	{ }

Response create/update

422

```
{
  "name": [
    "O campo Nome é obrigatório."
  ],
  "type": [
    "O campo type é obrigatório."
  ],
  "age": [
    "age deve ter no mínimo 1 caracteres."
  ]
}
```