TWITTER LIGHT MICROSERVICE USUÁRIOS

Documentação

API_URL**:** https://twitter-eng2-users.herokuapp.com/
Repositório: https://github.com/esthefanielanza/microservice-twitter-users

Modelo

id: Integer <primary_key>
login: String(20) <unique>
full_name: String(80)
short_bio: String(140)
followers: String()
following: String()

Obs: Os campos followers e following sofrem um parse antes de serem enviados na API. O tipo final é uma array de strings contendo o id dos usuários que seguem ou são seguidos pelo usuário corrente.

Listar Usuários

endpoint: /
método: GET

Possíveis Retornos:
Sucesso (200)

```
{
    user: {
        “followers”: [],
        “following”: [],
        “full_name”: ‘Full name’,
        “id”: 1,
        “login”: ‘Login’,
        “short_bio”: ‘Shor bio’
    }
}
```

Cadastro Usuário

endpoint: /
método: POST

body:

```
{
        login: (String, required),
        full_name: (String, required),
        short_bio: (String)
}
```

Possíveis Retornos:

Sucesso (200)

```
{
        user: {
                “followers”: [],
                “following”: [],
        ‘       “full_name”: ‘Full name’,
                “id”: 1,
                “login”: ‘Login’,
                “short_bio”: ‘Shor bio’
        }
}
```

Chave única (409)

```
{
        “type”: “UNIQUE”,
        “description”: “Field should be unique’,
        “error”: [ “username” ]
}
```

Limite Excedido (400)

```
{
        “type”: “LIMIT”,
        “description”: “Field is beyond the character limit”,
        “error”: error
}
```

Required (400)

```
{
        “type”: “REQUIRED”,
        “description”: “Please fill all the required fields”,
        “error”: {
                “login”: “Login is required”
        }
}
```

Unknown Error (500)

```
{
        “type”: “UNKNOWN”,
        “description”: “Field should be unique’,
        “error”: [ “username” ]
}
```

Recuperar Usuário por ID

endpoint: /:id
método: GET

Possíveis Retornos:

Sucesso (200)

```
{
        user: {
                “followers”: [],
                “following”: [],
        ‘       “full_name”: ‘Full name’,
                “id”: 1,
                “login”: ‘Login’,
                “short_bio”: ‘Shor bio’
        }
}
```

Usuário não existe (404)

```
{
        “type”: “NOT_EXISTS”,
        “description”: “User <1> does not exist”,
}
```

Unknown Error (500)

```
{
        “type”: “UNKNOWN”,
        “description”: “Field should be unique’,
        “error”: [ “username” ]
}
```

Deletar usuário por ID

endpoint: /:id
método: DELETE

Possíveis Retornos:

Sucesso (200)

```
{
        “description”: “User <1> was deleted”
}
```

Usuário não existe (404)

```
{
        “type”: “NOT_EXISTS”,
        “description”: “User <1> does not exist”,
}
```

Unknown Error (500)

```
{
```

```
	“type”: “UNKNOWN”,
	“description”: “Field should be unique’,
}
```

Seguir usuário

endpoint: /:id/follow
método: PUT

body:

```
{
	“follow_user”: (Integer, required)
}
```

Possíveis Retornos:
Sucesso (204)

Dados inválidos (400)

```
{
	“type”: “INVALID”,
	“description”: “User <1> already follows User <2>’
}
```

Required (400)

```
{
	“type”: “REQUIRED”,
	“description”: “Please fill all the required fields”,
	“error”: {
		“follow_user”: “Follow user is required”
	}
}
```

Usuário não existe (404)

```
{
	“type”: “NOT_EXISTS”,
	“description”: “User <1> does not exist”
}
```

Usuário que deve ser seguido não existe (404)

```
{
	“type”: “NOT_EXISTS”,
	“description”: “Unable to follow, User <1> does not exist”
}
```

Unknown Error (500)

```
{
	“type”: “UNKNOWN”,
	“description”: “error”
}
```

Deixar de seguir usuário

endpoint: /:id/unfollow
método: PUT

body:

```
{
        “unfollow_user”: (Integer, required)
}
```

Possíveis Retornos:
Sucesso (204)

Dados inválidos (400)

```
{
        “type”: “INVALID”,
        “description”: “User <1> does not follow User <2>”
}
```

Required (400)

```
{
        “type”: “REQUIRED”,
        “description”: “Please fill all the required fields”,
        “error”: {
                “unfollow_user”: “Follow user is required”
        }
}
```

Usuário não existe (404)

```
{
        “type”: “NOT_EXISTS”,
        “description”: “User <1> does not exist”
}
```

Usuário que deve ser seguido não existe (404)

```
{
        “type”: “NOT_EXISTS”
        “description”: “Unable to unfollow, User <1> does not exist”,
}
```

Unknown Error (500)

```
{
        “type”: “UNKNOWN”,
        “description”: “error”
}
```

MENSAGENS

Documentação

API_URL**:** https://messages-tpes2.herokuapp.com/
Repositório: https://github.com/ggapp1/messages-tpes2/blob/master/messages.py

Métodos

Recuperar mensagem por ID

endpoint: /:id/
método: GET

body:

```
{
        “messages”: [ ]
}
```

Possíveis Retornos:
Sucesso (204)

Mensagem não existe (404)

```
{
        “type”: “NOT_EXISTS”,
        “description”: “Message <1> does not exist”
}
```

Unknown Error (500)

```
{
        “type”: “UNKNOWN”,
        “description”: “error”
}
```

Recupera todas as mensagens

endpoint: /:
método: GET

body:

```
{
        “messages”: [ ]
}
```

Possíveis Retornos:

Sucesso (204)

Unknown Error (500)

```
{
        “type”: “UNKNOWN”,
        “description”: “error”
}
```

Cadastrar mensagem

endpoint: /
método: POST

param:

```
{
        user_id: (Integer, required),
        message: (String, required)
}
```

Possíveis Retornos:

Sucesso (204)

```
{
   "message": {
      "message": "Test",
      "message_id": 4,
      "user_id": 3
   }
}
```

Chave única (409)

```
{
        “type”: “UNIQUE”,
        “description”: “Field should be unique’,
        “error”: [ “message_id” ]
}
```

Usuário não encontrado (404)

```
{
   "description": "User not found",
   "type": "NOT FOUND"
}
```

Required (400)

```
{
        “type”: “REQUIRED”,
        “description”: “Please fill all the required fields”,
        “error”: {
                “user_id”: “user_id is required”
        }
```

}

Unknown Error (500)

{
	“type”: “UNKNOWN”,
	“description”: “An unknown error was detected’
}

TIMELINE

Documentação

API_URL**:** https://microservice-twitter-timeline.herokuapp.com/
Repositório: https://github.com/soniaAlejandra/microservice-twitter-timeline

Métodos

POSTS

endpoint: /:id/post
método: GET

body:

```
{
        “messages”: [ ]
}
```

Possíveis Retornos:
Sucesso (204)

Usuário não existe (404)

```
{
        “type”: “NOT_EXISTS”,
        “description”: “User <1> does not exist”
}
```

Unknown Error (500)

```
{
        “type”: “UNKNOWN”,
        “description”: “error”
}
```

HOME

endpoint: /:id/home
método: GET

body:

```
{
        “messages”: [ ]
}
```

Possíveis Retornos:

Sucesso (204)

Usuário não existe (404)

```
{
        “type”: “NOT_EXISTS”,
        “description”: “User <1> does not exist”
}
```

Unknown Error (500)

```
{
        “type”: “UNKNOWN”,
        “description”: “error”
}
```