Font Easy - API Documentation

Versão: 1.0.0
Base URL: `http://localhost:3000`
Última atualização: 23/12/2024

Índice

Autenticação

A API utiliza **JWT (JSON Web Token)** para autenticação. Após o login, o token deve ser enviado no header de todas as requisições protegidas.

Header de Autenticação:

```
Authorization: Bearer <token>
```

Endpoints de Usuários

1. Criar Usuário

Método	Endpoint	Autenticação
`POST`	`/users`	❌ Não

Request Body:

```
{
  "name": "string (obrigatório, min: 1 caractere)",
  "email": "string (obrigatório, email válido)",
  "password": "string (obrigatório, min: 6 caracteres)",
  "photo": "string (opcional, URL, default: 'minhafoto.com')",
  "plan_type": "string (opcional, default: 'FREE')"
}
```

Response (201):

```
{
  "message": "usuario criado com sucesso!",
  "user": { ... }
}
```

Erros:

- `400` - Dados inválidos

---

2. Login

Método	Endpoint	Autenticação
`POST`	`/login`	❌ Não

Request Body:

```
{
  "email": "string (obrigatório, email válido)",
  "password": "string (obrigatório)"
}
```

Response (200):

```
{
  "message": "Login realizado com sucesso.",
  "tokenAuth": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

Erros:

- `400` - Dados inválidos
- `401` - Credenciais inválidas

---

3. Obter Perfil

Método	Endpoint	Autenticação
`GET`	`/profile`	✅ Sim

Response (200):

```
{
  "user": {
    "id": 1,
    "name": "João Silva",
    "email": "joao@email.com",
    "photo": "https://example.com/photo.jpg",
    "plan_type": "FREE"
  }
}
```

Erros:

- `404` - Usuário não encontrado

---

4. Atualizar Usuário

Método	Endpoint	Autenticação
`PUT`	`/users`	✅ Sim

Request Body (campos opcionais):

```
{
  "name": "string",
  "email": "string",
  "password": "string",
  "photo": "string",
  "plan_type": "string"
}
```

Response (200):

```
{
  "message": "Usuário atualizado com sucesso.",
  "data": { ... }
}
```

Erros:

- `404` - Usuário não encontrado ou token inválido

---

5. Deletar Usuário

Método	Endpoint	Autenticação
`DELETE`	`/users/:id`	✅ Sim

Parâmetros:

- `id` (URL) - ID do usuário

Response (200):

```
{
  "message": "usuario deletado com sucesso!"
}
```

Erros:

- `400` - ID inválido

Endpoints de Fontes Favoritas

1. Adicionar Fonte Favorita

Método	Endpoint	Autenticação
`POST`	`/favoritefonts`	✅ Sim

Request Body:

```
{
  "font_name": "string (obrigatório, min: 1 caractere)",
  "font_variations": "number (opcional)",
  "font_type": "string (opcional)",
  "fontlinks": [
    {
      "fontLink": "string (URL válida)"
    }
  ]
}
```

Response (201):

```
{
  "message": "font adicionada sucesso!",
  "font": { ... }
}
```

Erros:

- `400` - Dados inválidos

---

2. Listar Fontes Favoritas

Método	Endpoint	Autenticação
`GET`	`/favoritefonts`	✅ Sim

Response (200):

```
[
  {
    "id_font": 1,
    "user_id": 1,
    "font_name": "Roboto",
    "font_variations": 12,
    "font_type": "sans-serif",
    "fontlinks": [
      {
        "id_link": 1,
        "font_link": "https://fonts.google.com/roboto",
        "font_id": 1
      }
    ]
```

```
  }
]
```

---

3. Deletar Fonte Favorita

Método	Endpoint	Autenticação
`DELETE`	`/favoritefonts/:id`	✅ Sim

Parâmetros:

- `id` (URL) - ID da fonte favorita

Response (200):

```
{
  "message": "font deletada com sucesso."
}
```

Erros:

- `400` - ID inválido

> ***Nota:*** *Ao deletar uma fonte favorita, todos os links associados são deletados automaticamente (cascade).*

Schemas de Validação

UserSchema (Criação)

Campo	Tipo	Obrigatório	Validação
name	string	✅	min: 1 caractere
email	string	✅	formato de email
password	string	✅	min: 6 caracteres
photo	string	❌	URL válida
plan_type	string	❌	default: "FREE"

LoginSchema

Campo	Tipo	Obrigatório	Validação
email	string	✅	formato de email
password	string	✅	-

FontSchema

Campo	Tipo	Obrigatório	Validação

font_name	string	✅	min: 1 caractere
font_variations	number	❌	-
font_type	string	❌	-
fontlinks	array	❌	array de objetos
fontlinks[].fontLink	string	✅*	URL válida

Modelos do Banco de Dados

Users

```
┌──────────────┬──────────────┬──────────────┐
│ Campo        │ Tipo         │ Constraint   │
├──────────────┼──────────────┼──────────────┤
│ id           │ Int          │ PK, Auto     │
│ name         │ String(100)  │ Nullable     │
│ email        │ String       │ Unique       │
│ password     │ String       │ Required     │
│ photo        │ String       │ Nullable     │
│ plan_type    │ String(100)  │ Nullable     │
└──────────────┴──────────────┴──────────────┘
```

FavoriteFonts

```
┌──────────────────┬──────────────┬──────────────────────┐
│ Campo            │ Tipo         │ Constraint           │
├──────────────────┼──────────────┼──────────────────────┤
│ id_font          │ Int          │ PK, Auto             │
│ user_id          │ Int          │ FK → users.id        │
│ font_name        │ String(100)  │ Required             │
│ font_variations  │ Int          │ Nullable             │
│ font_type        │ String       │ Nullable             │
└──────────────────┴──────────────┴──────────────────────┘

Relação: CASCADE DELETE com users
```

FontLinks

```
┌──────────────┬──────────┬──────────────────────────────┐
│ Campo        │ Tipo     │ Constraint                   │
├──────────────┼──────────┼──────────────────────────────┤
│ id_link      │ Int      │ PK, Auto                     │
│ font_link    │ String   │ Required                     │
│ font_id      │ Int      │ FK → favoritefonts.id_font   │
└──────────────┴──────────┴──────────────────────────────┘

Relação: CASCADE DELETE com favoritefonts
```

Códigos de Erro

Código	Descrição
200	Sucesso
201	Criado com sucesso
400	Requisição inválida (dados faltando ou inválidos)
401	Não autorizado (token inválido ou expirado)
404	Recurso não encontrado
500	Erro interno do servidor

Exemplo de Fluxo Completo

```
1. Criar usuário      →  POST /users
2. Fazer login        →  POST /login (recebe token)
3. Adicionar fonte    →  POST /favoritefonts (com token)
4. Listar fontes      →  GET /favoritefonts (com token)
5. Deletar fonte      →  DELETE /favoritefonts/:id (com token)
```

Desenvolvido por: Felipe Lino
Repositório: GitHub