

Documentação da API: Lyria

Este documento detalha todos os endpoints da API para o frontend, incluindo suas funcionalidades, parâmetros de entrada e os retornos esperados.

1. Autenticação

1.1. POST /Lyria/login

- **Descrição:** Autentica um usuário existente.

Corpo da Requisição (JSON):

```
{
  "email": "string",
  "senha_hash": "string" (opcional)
}
```

Retorno em Sucesso (200 OK):

```
{
  "sucesso": "Login realizado com sucesso",
  "usuario": "Nome do Usuário",
  "persona": "persona_escolhida"
}
```

Retorno em Erro:

- 400 Bad Request: { "erro": "Campo 'email' é obrigatório" }
- 401 Unauthorized: { "erro": "Senha incorreta" }
- 404 Not Found: { "erro": "Usuário não encontrado" }
- 500 Internal Server Error: Erro interno.

1.2. POST /Lyria/logout

- **Descrição:** Encerra a sessão do usuário.
- **Corpo da Requisição:** Vazio.

Retorno em Sucesso (200 OK):

```
{
  "sucesso": "Logout realizado com sucesso"
}
```

2. Conversa com a IA

2.1. **POST /Lyria/conversar** (Sem Login)

- **Descrição:** Permite interagir com a IA sem autenticação, especificando a persona.

Corpo da Requisição (**JSON**):

```
{
  "pergunta": "string",
  "persona": "string" (ex: "professor", "empresarial", "social")
}
```

Retorno em Sucesso (**200 OK**):

```
{
  "resposta": "string"
}
```

Retorno em Erro:

- **400 Bad Request:** { "erro": "Campo 'pergunta' ou 'persona' é obrigatório" }
- **500 Internal Server Error:** Erro interno.

2.2. **POST /Lyria/conversar-logado**

- **Descrição:** Permite que usuários autenticados conversem com a IA. As conversas são salvas no histórico.

Corpo da Requisição (**JSON**):

```
{
  "pergunta": "string"
}
```

Retorno em Sucesso (**200 OK**):

```
{
  "resposta": "string"
}
```

Retorno em Erro:

- 401 Unauthorized: { "erro": "Usuário não está logado" }
- 400 Bad Request: { "erro": "Campo 'pergunta' é obrigatório" } ou { "erro": "Usuário não tem persona definida" }
- 500 Internal Server Error: Erro interno.

3. Gerenciamento de Usuários e Personas

3.1. POST /Lyria/usuarios

- **Descrição:** Cria um novo usuário.

Corpo da Requisição (JSON):

```
{
  "nome": "string",
  "email": "string",
  "persona": "string" (ex: "professor", "empresarial", "social"),
  "senha_hash": "string" (opcional)
}
```

Retorno em Sucesso (201 Created):

```
{
  "sucesso": "Usuário criado com sucesso",
  "id": 123,
  "persona": "persona_escolhida"
}
```

Retorno em Erro:

- 400 Bad Request: { "erro": "Campos 'nome' e 'email' são obrigatórios" } ou { "erro": "Persona inválida" }
- 409 Conflict: { "erro": "Usuário já existe" }
- 500 Internal Server Error: Erro interno.

3.2. GET /Lyria/usuarios/<usuarioEmail>

- **Descrição:** Busca os dados de um usuário pelo seu e-mail.
- **Parâmetros da URL:** <usuarioEmail>

Retorno em Sucesso (200 OK):

```
{
```

```
"usuario": {
  "id": 1,
  "email": "teste@email.com",
  "nome": "Usuário Teste",
  "persona_escolhida": "professor"
}
```

Retorno em Erro:

- 404 Not Found: { "erro": "Usuário não encontrado" }
- 500 Internal Server Error: Erro interno.

3.3. GET /Lyria/PersonaEscolhida

- **Descrição:** Retorna a persona atual do usuário autenticado.

Retorno em Sucesso (200 OK):

```
{
  "persona": "professor"
}
```

Retorno em Erro:

- 401 Unauthorized: { "erro": "Usuário não está logado" }
- 404 Not Found: { "erro": "Usuário não encontrado" }
- 500 Internal Server Error: Erro interno.

3.4. POST /Lyria/PersonaEscolhida

- **Descrição:** Define a persona para o usuário autenticado.

Corpo da Requisição (JSON):

```
{
  "persona": "string" (ex: "professor", "empresarial", "social")
}
```

Retorno em Sucesso (200 OK):

```
{
  "sucesso": "Persona atualizada com sucesso"
}
```

Retorno em Erro:

- 401 Unauthorized: { "erro": "Usuário não está logado" }
- 400 Bad Request: { "erro": "Campo 'persona' é obrigatório" } ou { "erro": "Persona inválida" }

- 500 Internal Server Error: Erro interno.

3.5. GET /Lyria/personas

- **Descrição:** Retorna a lista e descrição de todas as personas disponíveis.

Retorno em Sucesso (200 OK):

```
{
  "personas": {
    "professor": "Descrição da persona professor...",
    "empresarial": "Descrição da persona empresarial...",
    "social": "Descrição da persona social..."
  }
}
```

4. Histórico de Conversas

4.1. GET /Lyria/conversas

- **Descrição:** Retorna o histórico completo das conversas do usuário autenticado.

Retorno em Sucesso (200 OK):

```
{
  "conversas": [
    { "role": "user", "content": "olá" },
    { "role": "assistant", "content": "Olá! Como posso ajudar?" },
    ...
  ]
}
```

Retorno em Erro:

- 401 Unauthorized: { "erro": "Usuário não está logado" }
- 500 Internal Server Error: Erro interno.

4.2. GET /Lyria/historico

- **Descrição:** Retorna as mensagens mais recentes do histórico do usuário autenticado.
- **Parâmetros da URL (Query String):**
 - **limite:** integer (opcional, padrão: 10, máx: 50) - Número de mensagens a retornar.
- **Exemplo de URL:** GET /Lyria/historico?limite=20

Retorno em Sucesso (200 OK):

```
{
```

```
"historico": [  
  {  
    "id": 1,  
    "pergunta": "Qual a capital da França?",  
    "resposta": "Paris.",  
    "data": "2025-10-02T15:00:00Z"  
  },  
  ...  
]  
}
```

Retorno em Erro:

- 401 Unauthorized: { "erro": "Usuário não está logado" }
- 500 Internal Server Error: Erro interno.