

API de Manutenção RecruitAI

Visão Geral

Esta API foi criada para permitir que a equipe Abacus.AI realize manutenções e correções no sistema RecruitAI após o deploy, sem precisar acessar diretamente o servidor.

Autenticação

IMPORTANTE: Todas as requisições devem incluir o header de autorização:

```
Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8fd29b98b5fc5a2697a455c1e
```

 **MANTENHA ESTE TOKEN SEGURO!** Este é o token de manutenção master que dá acesso total ao sistema.

Endpoints Disponíveis

1. Verificar Status do Sistema

GET /api/maintenance/status

Retorna informações completas sobre o status do sistema.

Exemplo de Requisição:

```
curl -X GET "https://seu-dominio.com/api/maintenance/status" \
-H "Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8f-
d29b98b5fc5a2697a455c1e"
```

Resposta:

```
{
  "success": true,
  "timestamp": "2025-11-26T02:00:00.000Z",
  "database": {
    "status": "online",
    "message": "Conexão com banco de dados OK"
  },
  "server": {
    "status": "running",
    "processes": 1,
    "uptime": 3600,
    "nodeVersion": "v22.14.0"
  },
  "memory": {
    "heapUsed": "120MB",
    "heapTotal": "180MB",
    "rss": "250MB"
  },
  "stats": {
    "users": 150,
    "companies": 45,
    "jobs": 230,
    "applications": 1500
  }
}
```

2. Executar Comandos de Manutenção

POST /api/maintenance/execute

Executa comandos específicos de manutenção no sistema.

Ações Disponíveis:

A) `restart_server` - Reiniciar o Servidor Next.js

```
curl -X POST "https://seu-dominio.com/api/maintenance/execute" \
-H "Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8f- \
d29b98b5fc5a2697a455c1e" \
-H "Content-Type: application/json" \
-d '{
  "action": "restart_server"
}'
```

Quando usar: Quando o sistema estiver travado ou não respondendo.

B) `clear_cache` - Limpar Cache do Next.js

```
curl -X POST "https://seu-dominio.com/api/maintenance/execute" \
-H "Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8f- \
d29b98b5fc5a2697a455c1e" \
-H "Content-Type: application/json" \
-d '{
    "action": "clear_cache"
}'
```

Quando usar: Após mudanças no código ou quando houver problemas de cache.

C) `check_database` - Verificar Integridade do Banco

```
curl -X POST "https://seu-dominio.com/api/maintenance/execute" \
-H "Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8f- \
d29b98b5fc5a2697a455c1e" \
-H "Content-Type: application/json" \
-d '{
    "action": "check_database"
}'
```

Quando usar: Para verificar a saúde do banco de dados.

D) `prisma_generate` - Gerar Client do Prisma

```
curl -X POST "https://seu-dominio.com/api/maintenance/execute" \
-H "Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8f- \
d29b98b5fc5a2697a455c1e" \
-H "Content-Type: application/json" \
-d '{
    "action": "prisma_generate"
}'
```

Quando usar: Após mudanças no schema do Prisma.

E) `prisma_push` - Atualizar Schema do Banco

```
curl -X POST "https://seu-dominio.com/api/maintenance/execute" \
-H "Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8f- \
d29b98b5fc5a2697a455c1e" \
-H "Content-Type: application/json" \
-d '{
    "action": "prisma_push"
}'
```

Quando usar: Para aplicar mudanças no schema do banco de dados.

⚠️ ATENÇÃO: Esta ação modifica o banco de dados. Use com cuidado!

F) `run_seed` - Executar Seed do Banco

```
curl -X POST "https://seu-dominio.com/api/maintenance/execute" \
-H "Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8f- \
d29b98b5fc5a2697a455c1e" \
-H "Content-Type: application/json" \
-d '{
    "action": "run_seed"
}'
```

Quando usar: Para popular o banco com dados iniciais.

⚠️ ATENÇÃO: Não use em produção se já houver dados reais!

G) `cleanup_orphans` - Limpar Registros Órfãos

```
curl -X POST "https://seu-dominio.com/api/maintenance/execute" \
-H "Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8f- \
d29b98b5fc5a2697a455c1e" \
-H "Content-Type: application/json" \
-d '{
    "action": "cleanup_orphans"
}'
```

Quando usar: Para remover registros órfãos do banco de dados.

H) `get_logs` - Obter Logs Recentes

```
curl -X POST "https://seu-dominio.com/api/maintenance/execute" \
-H "Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8f- \
d29b98b5fc5a2697a455c1e" \
-H "Content-Type: application/json" \
-d '{
    "action": "get_logs",
    "params": {
        "lines": 100
    }
}'
```

Quando usar: Para diagnosticar problemas verificando os logs.

3. Visualizar Histórico de Manutenções

GET `/api/maintenance/logs`

Retorna o histórico de todas as manutenções executadas.

Parâmetros de Query:

- `limit` : Número de registros (padrão: 50)

- `status` : Filtrar por status (`success` ou `error`)

Exemplo:

```
curl -X GET "https://seu-dominio.com/api/maintenance/logs?limit=20&status=success" \
-H "Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8f-
d29b98b5fc5a2697a455c1e"
```

Resposta:

```
{
  "success": true,
  "count": 20,
  "logs": [
    {
      "id": "clx...",
      "action": "restart_server",
      "params": null,
      "result": "{\"success\":true,\"message\":\"Servidor reiniciado com sucesso\"}",
      "status": "success",
      "executedAt": "2025-11-26T01:30:00.000Z"
    },
    {
      "id": "clx...",
      "action": "check_database",
      "params": null,
      "result": "{\"success\":true,\"message\":\"Banco de dados está operacional\"}",
      "status": "success",
      "executedAt": "2025-11-26T01:00:00.000Z"
    }
  ]
}
```



Cenários Comuns de Uso

Cenário 1: Sistema Parou de Responder

```
# 1. Verificar status
curl -X GET "https://seu-dominio.com/api/maintenance/status" \
-H "Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8f-
d29b98b5fc5a2697a455c1e"

# 2. Se servidor estiver parado, reiniciar
curl -X POST "https://seu-dominio.com/api/maintenance/execute" \
-H "Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8f-
d29b98b5fc5a2697a455c1e" \
-H "Content-Type: application/json" \
-d '{"action": "restart_server"}'

# 3. Verificar logs para diagnóstico
curl -X POST "https://seu-dominio.com/api/maintenance/execute" \
-H "Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8f-
d29b98b5fc5a2697a455c1e" \
-H "Content-Type: application/json" \
-d '{"action": "get_logs", "params": {"lines": 50}}'
```

Cenário 2: Erro “Prisma Client is not initialized”

```
# Gerar client do Prisma
curl -X POST "https://seu-dominio.com/api/maintenance/execute" \
-H "Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8f-\
d29b98b5fc5a2697a455c1e" \
-H "Content-Type: application/json" \
-d '{"action": "prisma_generate"}'

# Reiniciar servidor
curl -X POST "https://seu-dominio.com/api/maintenance/execute" \
-H "Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8f-\
d29b98b5fc5a2697a455c1e" \
-H "Content-Type: application/json" \
-d '{"action": "restart_server"}'
```

Cenário 3: Atualizar Schema do Banco

```
# 1. Push schema
curl -X POST "https://seu-dominio.com/api/maintenance/execute" \
-H "Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8f-\
d29b98b5fc5a2697a455c1e" \
-H "Content-Type: application/json" \
-d '{"action": "prisma_push"}'

# 2. Gerar client
curl -X POST "https://seu-dominio.com/api/maintenance/execute" \
-H "Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8f-\
d29b98b5fc5a2697a455c1e" \
-H "Content-Type: application/json" \
-d '{"action": "prisma_generate"}'

# 3. Reiniciar
curl -X POST "https://seu-dominio.com/api/maintenance/execute" \
-H "Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8f-\
d29b98b5fc5a2697a455c1e" \
-H "Content-Type: application/json" \
-d '{"action": "restart_server"}'
```



Segurança

Proteções Implementadas:

1. **Autenticação via Token:** Todas as requisições exigem o token secreto
2. **Logs de Auditoria:** Todas as ações são registradas no banco de dados
3. **Comandos Limitados:** Apenas ações pré-definidas e seguras podem ser executadas
4. **Sem Acesso SSH:** Não é necessário acesso direto ao servidor

Recomendações de Segurança:

- **NUNCA** compartilhe o token de manutenção

- **SEMPRE** use HTTPS em produção
 - **MONITORE** os logs de manutenção regularmente
 - **REVISE** o histórico de manutenções periodicamente
-



Monitoramento

Ver Últimas Manutenções

```
curl -X GET "https://seu-dominio.com/api/maintenance/logs?limit=10" \
-H "Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8f-
d29b98b5fc5a2697a455c1e"
```

Ver Apenas Erros

```
curl -X GET "https://seu-dominio.com/api/maintenance/logs?status=error" \
-H "Authorization: Bearer 3977aa7046e9bf25ce7e91d535177b4c00794ec8f-
d29b98b5fc5a2697a455c1e"
```



Variáveis de Ambiente

A API de manutenção requer a seguinte variável configurada no `.env`:

```
MAINTENANCE_SECRET=3977aa7046e9bf25ce7e91d535177b4c00794ec8fd29b98b5fc5a2697a455c1e
```

IMPORTANTE: Esta variável já foi configurada automaticamente. Ao fazer deploy no Vercel, adicione-a nas configurações de ambiente.



Checklist de Deploy

Antes de fazer o deploy, certifique-se de:

- [] Adicionar `MAINTENANCE_SECRET` nas variáveis de ambiente do Vercel
- [] Testar todos os endpoints em ambiente de staging
- [] Documentar o token em local seguro
- [] Configurar alertas para falhas de manutenção



Suporte

Em caso de problemas:

1. Verificar status: `GET /api/maintenance/status`
2. Ver logs: `POST /api/maintenance/execute com action: "get_logs"`
3. Verificar histórico: `GET /api/maintenance/logs`

4. Reiniciar servidor: `POST /api/maintenance/execute` com `action: "restart_server"`

Status Atual

-  API de manutenção criada
-  Token de autenticação configurado
-  Modelo de logs criado no banco de dados
-  Servidor Next.js reiniciado e operacional
-  Sistema pronto para deploy

Data de Criação: 26 de novembro de 2025

Versão: 1.0.0

Status: Operacional 