

# Frontend Integration Guide - AtendeAI 2.0

## Visão Geral

Este documento descreve como integrar as telas do frontend com os endpoints do backend, especificamente para a funcionalidade de Google Calendar.

## Tela de Agenda - Google Calendar

### Mapeamento de Funcionalidades

Funcionalidade	Endpoint	Método	Descrição
Autenticação OAuth	/auth/google/url	POST	Inicia fluxo OAuth2
Callback OAuth	/auth/google/callback	POST	Processa callback OAuth
Listar Integrações	google_integrations (Supabase)	SELECT	Busca integrações do usuário
Sincronizar Calendário	/google/events/sync	POST	Sincroniza eventos
Listar Eventos	/google/events	GET	Lista eventos do calendário
Criar Evento	/google/events	POST	Cria novo evento
Atualizar Evento	/google/events/{id}	PUT	Atualiza evento
Excluir Evento	/google/events/{id}	DELETE	Remove evento

### SDK Implementado

src/sdk/googleCalendarSDK.ts

```
import { googleCalendarSDK } from '@sdk/googleCalendarSDK';

// OAuth Flow
const authUrl = await googleCalendarSDK.getAuthUrl(redirectUri);
const integration = await googleCalendarSDK.handleCallback(code, state);

// Gerenciamento de Integrações
const integrations = await googleCalendarSDK.getUserIntegrations(userId);
await googleCalendarSDK.disconnect(integrationId);

// Operações de Calendário
await googleCalendarSDK.syncCalendar(integrationId);
const events = await googleCalendarSDK.getEvents(integrationId, params);
const event = await googleCalendarSDK.createEvent(integrationId, eventData);
```

src/hooks/useGoogleCalendar.ts

```
import { useGoogleCalendar } from '@hooks/useGoogleCalendar';

const {
  integration,
  events,
  loading,
  authLoading,
  syncLoading,
  calendarUrl,
  handleGoogleAuth,
  handleDisconnect,
  handleSync,
  isConnected,
  status
} = useGoogleCalendar();
```

## Fluxo de Integração

### 1. Inicialização

```
// Hook carrega automaticamente as integrações do usuário
const { integration, loading } = useGoogleCalendar();
```

### 2. Autenticação OAuth2

```
// Usuário clica em "Conectar Google Calendar"
const handleConnect = () => {
  handleGoogleAuth(); // Abre popup OAuth
};
```

### 3. Callback Processing

```
// OAuth callback é processado automaticamente
// Integração é salva no banco via Supabase
// Estado é atualizado automaticamente
```

### 4. Visualização do Calendário

```
// Iframe do Google Calendar é gerado automaticamente
{calendarUrl && status === 'active' && (
  <iframe src={calendarUrl} />
)}
```

### 5. Sincronização

```
// Sincronização manual ou automática
const handleSync = () => {
  handleSync(); // Chama endpoint de sincronização
};
```

## Configuração de Ambiente

## Variáveis Necessárias

```
# Frontend (.env.local)
VITE_API_BASE_URL=http://localhost:8000
VITE_GOOGLE_CLIENT_ID=your_client_id
VITE_GOOGLE_CLIENT_SECRET=your_client_secret

# Backend (.env)
GOOGLE_CLIENT_ID=your_client_id
GOOGLE_CLIENT_SECRET=your_client_secret
GOOGLE_REDIRECT_URI=http://localhost:3000/auth/google/callback
```

## Estrutura de Dados

### GoogleIntegration

```
interface GoogleIntegration {
  id: string;
  user_id: string;
  clinic_id: string;
  google_calendar_id: string;
  access_token: string;
  refresh_token: string;
  scope: string;
  token_expiry: string;
  calendar_name?: string;
  calendar_description?: string;
  sync_enabled: boolean;
  last_sync: string;
  status: 'active' | 'expired' | 'error';
  created_at: string;
  updated_at: string;
}
```

### GoogleEvent

```
interface GoogleEvent {
  id: string;
  summary: string;
  description?: string;
  start: {
    dateTime: string;
    timeZone: string;
  };
  end: {
    dateTime: string;
    timeZone: string;
  };
  attendees?: Array<{
    email: string;
    displayName?: string;
    responseStatus?: string;
  }>;
}
```

## Tratamento de Erros

### Tipos de Erro

- **401 Unauthorized:** Token expirado ou inválido
- **403 Forbidden:** Usuário sem permissão
- **404 Not Found:** Integração não encontrada
- **429 Rate Limited:** Muitas requisições
- **500 Internal Error:** Erro do servidor

### Implementação

```
try {
  const result = await googleCalendarSDK.getEvents(integrationId);
} catch (error) {
  if (error.status === 401) {
    // Renovar token ou redirecionar para login
    await refreshToken();
  } else if (error.status === 429) {
    // Implementar retry com backoff
    await retryWithBackoff();
  } else {
    // Mostrar erro genérico
    toast.error('Erro ao carregar eventos');
  }
}
```

## Testes de Integração

### Smoke Tests

```
// tests/integration/agenda_smoke.test.ts
describe('Agenda Integration', () => {
  test('should load integrations on mount', async () => {
    // Teste de carregamento inicial
  });

  test('should handle OAuth flow', async () => {
    // Teste do fluxo OAuth
  });

  test('should sync calendar', async () => {
    // Teste de sincronização
  });
});
```

## Monitoramento e Logs

### Métricas Importantes

- Taxa de sucesso do OAuth
- Tempo de resposta da sincronização
- Erros de autenticação
- Uso de tokens

### Logs Estruturados

```
console.log('Google Calendar Integration', {
  action: 'oauth_initiated',
  userId: user.id,
  clinicId: clinic.id,
  timestamp: new Date().toISOString()
});
```

## Segurança

### Boas Práticas

- Tokens são armazenados no backend (Supabase)
- Frontend não expõe credenciais sensíveis
- Validação de entrada em todos os endpoints
- Rate limiting implementado
- CORS configurado adequadamente

### Validações

- Verificação de permissões por clínica
- Validação de tokens antes de operações
- Sanitização de dados de entrada
- Timeout em requisições longas

## Troubleshooting

## Problemas Comuns

### 1. "Google Calendar não conectado"

- Verificar se usuário está autenticado
- Verificar se integração existe no banco
- Verificar logs do backend

### 2. "Erro ao sincronizar calendário"

- Verificar se token não expirou
- Verificar conectividade com Google API
- Verificar logs de erro

### 3. "Iframe não carrega"

- Verificar se calendarUrl está correto
- Verificar se integração está ativa
- Verificar CORS do Google Calendar

## Próximos Passos

1. **Implementar cache local** para eventos
2. **Adicionar notificações** de sincronização
3. **Implementar sincronização automática** em background
4. **Adicionar métricas** de uso
5. **Implementar testes E2E** completos

---

**Última atualização:** 27 de Janeiro de 2025

**Versão:** 1.0.0

**Status:** Implementado e Funcional