iCalendar/iCloud Calendar Integration

Overview

CitaPlanner now supports optional integration with iCloud Calendar through CalDAV protocol. This feature allows users to:

- Export appointments as .ics files
- Connect their iCloud calendar for automatic synchronization
- Bidirectional sync between CitaPlanner and iCloud
- View sync status and history

Features

1. iCalendar Export (.ics files)

Users can export their appointments as standard iCalendar (.ics) files that can be imported into any calendar application.

API Endpoint: `GET /api/calendar/export`

Query Parameters:
- `userId` (optional): User ID to export appointments for
- `startDate` (optional): Start date filter (ISO 8601)
- `endDate` (optional): End date filter (ISO 8601)

Example:

```
curl -X GET "https://your-domain.com/api/calendar/export?
userId=xxx&startDate=2025-01-01&endDate=2025-12-31"
```

2. iCloud Calendar Connection

Users can connect their iCloud calendar using an app-specific password.

Setup Instructions:

1. Go to appleid.apple.com (https://appleid.apple.com)
2. Sign in with your Apple ID
3. Navigate to “Sign-In and Security” → “App-Specific Passwords”
4. Generate a new password labeled “CitaPlanner”
5. Copy the 16-character password (format: xxxx-xxxx-xxxx-xxxx)
6. In CitaPlanner, go to Settings → Calendar Integration
7. Click “Connect iCloud Calendar”
8. Enter your Apple ID and the app-specific password
9. Select the calendar you want to sync

API Endpoint: `POST /api/calendar/icloud/connect`

Request Body:

```
{
  "appleId": "your@email.com",
  "appSpecificPassword": "xxxx-xxxx-xxxx-xxxx",
  "calendarUrl": "https://caldav.icloud.com/...",
  "calendarName": "My Calendar"
}
```

3. Manual Sync

Trigger a manual synchronization between CitaPlanner and iCloud.

API Endpoint: `POST /api/calendar/icloud/sync`

Request Body:

```
{
  "connectionId": "connection-id-here"
}
```

4. Sync Status

View the status of calendar connections and sync history.

API Endpoint: `GET /api/calendar/icloud/status`

Query Parameters:
- `connectionId` (optional): Specific connection ID to query

Response:

```
{
  "success": true,
  "connections": [
    {
      "id": "conn-123",
      "provider": "ICLOUD_CALDAV",
      "calendarName": "My Calendar",
      "syncStatus": "ACTIVE",
      "lastSyncAt": "2025-10-07T12:00:00Z",
      "syncInterval": 300,
      "bidirectionalSync": true,
      "autoExport": true
    }
  ]
}
```

Database Schema

New Tables

ExternalCalendarConnection

Stores user's calendar connections with encrypted credentials.

```
model ExternalCalendarConnection {
  id                String            @id @default(cuid())
  userId            String
  provider          CalendarProvider
  calendarUrl       String
  calendarName      String?
  encryptedUsername String            @db.Text
  encryptedPassword String            @db.Text
  syncStatus        SyncStatus        @default(ACTIVE)
  syncToken         String?           @db.Text
  ctag              String?
  lastSyncAt        DateTime?
  lastSyncError     String?           @db.Text
  syncInterval      Int               @default(300)
  bidirectionalSync Boolean           @default(true)
  autoExport        Boolean           @default(true)
  createdAt         DateTime          @default(now())
  updatedAt         DateTime          @updatedAt
}
```

CalendarSyncLog

Tracks all sync operations for debugging and audit.

```
model CalendarSyncLog {
  id                String   @id @default(cuid())
  connectionId      String
  syncType          String
  direction         String
  status            String
  eventsImported    Int      @default(0)
  eventsExported    Int      @default(0)
  eventsUpdated     Int      @default(0)
  eventsDeleted     Int      @default(0)
  conflictsResolved Int      @default(0)
  errorMessage      String?  @db.Text
  errorDetails      String?  @db.Text
  duration          Int?
  createdAt         DateTime @default(now())
}
```

Modified Tables

Appointment (Optional Fields Added)

```
model Appointment {
  // ... existing fields ...

  // iCloud integration fields (optional)
  externalConnectionId  String?
  externalEventUrl      String?
  externalEventUid      String?
  externalEtag          String?
  lastModifiedSource    String?
  icloudSyncEnabled     Boolean  @default(false)
}
```

Security

Credential Encryption

All iCloud credentials (Apple ID and app-specific passwords) are encrypted using AES-256-CBC encryption before being stored in the database.

Environment Variable Required:

```
CALENDAR_ENCRYPTION_KEY=your-32-character-encryption-key-here
```

Important: In production, use a secure secrets manager (AWS KMS, Google Cloud KMS, HashiCorp Vault) to manage the encryption key.

Authentication

All calendar API endpoints require authentication via NextAuth session. Users can only access their own calendar connections.

UI Components

ICloudConnectDialog

A dialog component for connecting iCloud calendars.

Usage:

```
import { ICloudConnectDialog } from '@/components/calendar/ICloudConnectDialog';

<ICloudConnectDialog
  open={isOpen}
  onOpenChange={setIsOpen}
  onSuccess={(connectionId) => {
    console.log('Connected:', connectionId);
  }}
/>
```

CalendarSyncStatus

A component displaying sync status and recent activity.

Usage:

```
import { CalendarSyncStatus } from '@/components/calendar/CalendarSyncStatus';

<CalendarSyncStatus userId={currentUser.id} />
```

Migration

To add the iCalendar integration to your database:

1. Copy the schema additions from `prisma/schema_additions.prisma` to your main `prisma/schema.prisma`
2. Run the migration:

```
npx prisma migrate dev --name icalendar_integration
```

Dependencies

The following packages are required:

```
{
  "ical-generator": "^7.2.0",
  "tsdav": "^2.0.5",
  "node-forge": "^1.3.1",
  "@types/node-forge": "^1.3.11"
}
```

Limitations

1. **Manual Mapping Required:** When importing events from iCloud, they need to be manually mapped to existing services and clients in CitaPlanner.
2. **Sync Interval:** Default sync interval is 5 minutes. This can be adjusted per connection.
3. **Conflict Resolution:** Currently uses “last write wins” strategy based on timestamps.
4. **No Recurrence Support:** Recurring events are not yet fully supported.

Future Enhancements

- [ ] Automatic service/client matching for imported events
- [ ] Support for recurring appointments
- [ ] Google Calendar integration
- [ ] Outlook Calendar integration
- [ ] Real-time sync via webhooks (when available)
- [ ] Advanced conflict resolution UI
- [ ] Bulk export/import operations

Troubleshooting

Connection Failed

Error: “Failed to connect to iCloud”

Solutions:
1. Verify your Apple ID is correct
2. Ensure you’re using an app-specific password, not your main Apple ID password
3. Check that Two-Factor Authentication is enabled on your Apple ID
4. Try generating a new app-specific password

Sync Errors

Error: “401 Unauthorized”

Solutions:
1. The app-specific password may have been revoked
2. Generate a new password and reconnect

No Events Imported

Possible Causes:
1. The selected calendar is empty
2. Events are outside the sync date range
3. Manual mapping is required for imported events

Support

For issues or questions about the iCalendar integration, please:

1. Check the sync logs in the database (`CalendarSyncLog` table)
2. Review the API error responses
3. Ensure all environment variables are properly set
4. Verify that the encryption key is correctly configured

License