

Class Name: OutlookCalendarIntegration	
Parent Class (if any): None Subclasses (if any): None	
Responsibilities: <ul style="list-style-type: none"> <li>• Generate Microsoft OAuth URLs for authentication</li> <li>• Test connectivity to Microsoft Graph API</li> <li>• Process calendar data to extract availability information</li> <li>• Calculate available time slots from busy periods</li> <li>• Find overlapping availability across multiple users</li> <li>• Format and display availability results</li> <li>• Ensure privacy by only accessing free/busy information</li> </ul>	Collaborators: <ul style="list-style-type: none"> <li>• TimeSlot</li> <li>• User</li> <li>• Microsoft Graph API (external)</li> </ul>

Class Name: TimeSlot	
Parent Class (if any): None Subclasses (if any): None	
Responsibilities: <ul style="list-style-type: none"> <li>• Store start and end times for a calendar slot</li> <li>• Store time zone information</li> <li>• Provide duration calculation</li> <li>• Format time slots for display</li> <li>• Support comparison operations with other time slots</li> </ul>	Collaborators: <ul style="list-style-type: none"> <li>• OutlookCalendarIntegration</li> </ul>

Class Name: User	
Parent Class (if any): None Subclasses (if any): None	
Responsibilities: <ul style="list-style-type: none"> <li>• Store user identification information</li> <li>• Track calendar connection status</li> <li>• Maintain list of busy time slots</li> <li>• Provide access to user's availability data</li> </ul>	Collaborators: <ul style="list-style-type: none"> <li>• OutlookCalendarIntegration</li> <li>• TimeSlot</li> </ul>

Class Name: OutlookConnector	
Parent Class (if any): React.Component Subclasses (if any): None	
Responsibilities: <ul style="list-style-type: none"> <li>• Provide UI for connecting Outlook calendars</li> <li>• Display connection status</li> <li>• Allow users to disconnect their calendar</li> <li>• Show available times with appropriate formatting</li> <li>• Enable date range selection for availability checks</li> <li>• Handle loading states and error messages</li> <li>• Communicate with backend API</li> </ul>	Collaborators: <ul style="list-style-type: none"> <li>• Backend API endpoints</li> <li>• React state management</li> </ul>

Class Name: CalendarAPI (Service)	
Parent Class (if any): React.Component Subclasses (if any): None	
Responsibilities: <ul style="list-style-type: none"> <li>• Make HTTP requests to backend endpoints</li> <li>• Handle authentication token management</li> <li>• Process calendar data responses</li> <li>• Provide error handling for API calls</li> </ul>	Collaborators: <ul style="list-style-type: none"> <li>• OutlookConnector</li> <li>• Backend API endpoints</li> </ul>

Class Name: TimeFormatter (Utility)	
Parent Class (if any): None Subclasses (if any): None	
Responsibilities: <ul style="list-style-type: none"> <li>• Format date and time values for display</li> <li>• Calculate and format durations</li> <li>• Handle time zone conversions</li> <li>• Provide consistent time formatting across the application</li> </ul>	Collaborators: <ul style="list-style-type: none"> <li>• OutlookConnector</li> <li>• Other UI components</li> </ul>

Class Name: <b>GoogleService</b>	
Parent Class (if any): None Subclasses (if any): None	
Responsibilities: <ul style="list-style-type: none"> <li>• Generate Google OAuth URL for authentication</li> <li>• Exchange authorization code for access and refresh tokens</li> <li>• Fetch all busy calendar events for the next month</li> <li>• Calculate and return available time slots</li> <li>• Initialize Google Calendar API client using access token</li> </ul>	Collaborators: <ul style="list-style-type: none"> <li>- Google Calendar API (external)</li> <li>- GoogleController</li> </ul>

Class Name: <b>GoogleController</b>	
Parent Class (if any): None Subclasses (if any): None	
Responsibilities: <ul style="list-style-type: none"> <li>• Redirect user to Google OAuth authentication</li> <li>• Handle OAuth callback and return access token</li> <li>• Call service method to fetch free time slots</li> <li>• Return results to frontend with appropriate error handling</li> </ul>	Collaborators: <ul style="list-style-type: none"> <li>- GoogleService</li> <li>- Frontend (React components)</li> </ul>

Class Name: <b>FreeSlots (React Component)</b>	
Parent Class (if any): None Subclasses (if any): None	
Responsibilities: <ul style="list-style-type: none"> <li>• Fetch available time slots from backend API</li> <li>• Display time slots in a user-friendly format</li> <li>• Handle loading states and display errors- Redirect to login if access token is missing or expired</li> </ul>	Collaborators: <ul style="list-style-type: none"> <li>- GoogleController (Backend API)</li> <li>- Local storage for token management</li> </ul>

Class Name: <b>App (React Component)</b>	
Parent Class (if any): None Subclasses (if any): None	
Responsibilities: <ul style="list-style-type: none"> <li>• Initiate Google Calendar connection flow</li> <li>• Exchange authorization code for access token</li> <li>• Store and manage access token in local storage</li> <li>• Redirect to free slots page after authentication</li> </ul>	Collaborators: <ul style="list-style-type: none"> <li>- GoogleController (Backend API)</li> <li>- React Router (navigate)</li> </ul>

Class Name: CalendlyController (Springboot Controller)	
Parent Class (if any): None Subclasses (if any): None	
Responsibilities: <ul style="list-style-type: none"> <li>• Handles HTTP requests from the frontend</li> <li>• Manages OAuth token exchange, user info retrieval, and availability fetching</li> <li>• Handles exceptions and errors gracefully</li> <li>• Find overlapping availability across multiple users</li> <li>• Delegates complex business logic to Calendly Service</li> </ul>	Collaborators: <ul style="list-style-type: none"> <li>• CalendlyService</li> <li>• RestTemplate</li> <li>• ExceptionHandler</li> </ul>

Class Name: CalendlyAuth (Component)	
Parent Class (if any): None Subclasses (if any): None	
Responsibilities: <ul style="list-style-type: none"> <li>• Manages OAuth authentication</li> <li>• Stores and displays users' availability data</li> <li>• Fetches Calendly schedules using axios</li> <li>• Collaborates with React hooks, axios, and local storage for state management</li> </ul>	Collaborators: <ul style="list-style-type: none"> <li>• React Hooks (useState, useEffect)</li> <li>• Axios (API calls)</li> <li>• Local Storage (token persistence)</li> </ul>



