# Documentation for the Hollies Outlook Add-in

**Table of Contents**

## Introduction

* + Overview of the Outlook Add-in
  + Purpose and Key Features

## Architecture Overview

* + High-Level Architecture Diagram
  + Key Components and Their Interactions

## Authentication and Initialization

* + MSAL Integration
  + Public Client Application (PCA) Initialization
  + Token Acquisition (Silent and Interactive)

## Event Handling and UI Initialization

* + Office.onReady Functionality
  + Event Details Extraction and Population

## Project and Task Management

* + Project Search and Selection
  + Task Search and Selection
  + Dynamic Dropdown Population

## Time Entry Creation

* + Form Submission Process
  + API Call to Dynamics 365
  + Error Handling and Notifications

## UI Components and Event Handling

* + Dropdown Lists
  + Buttons
  + Form Validation

## Error Handling and Notifications

* + Error Messages
  + User Notifications

## Detailed Process Flow

* + Initialization and Authentication
  + Event Handling
  + Project and Task Selection
  + Time Entry Creation
  + Error Handling

## Code Walkthrough

* + Key Functions and Their Roles
  + Code Snippets and Explanations

## Conclusion

* + Summary of Functionality

# Introduction

## Overview of the Outlook Add-in

The Outlook Add-in is designed to integrate seamlessly with Microsoft Outlook, allowing users to create time entries based on calendar events. It leverages Microsoft Authentication Library (MSAL) for authentication, and Dynamics 365 API for project and task management.

## Purpose and Key Features

* + **Purpose**: To streamline the process of creating time entries from calendar events.

## Key Features:

* + - Authentication using MSAL.
    - Fetching and displaying user information.
    - Dynamic project and task search and selection.
    - Creation of time entries in Dynamics 365.
    - Error handling and user notifications.

# Architecture Overview

## High-Level Architecture Diagram

**Key Components and Their Interactions**

* + **MSAL (Auth)**: Handles user authentication and token acquisition.
  + **Microsoft Graph**: Fetches user information.
  + **Dynamics 365 API**: Manages projects, tasks, and time entries.
  + **Task Pane UI**: Provides the user interface for interacting with the add-in.
  + **Error Handling**: Manages errors and notifications.

# Authentication and Initialization

## MSAL Integration

The add-in uses MSAL to authenticate users and acquire access tokens for accessing Microsoft Graph and Dynamics 365 APIs.

## Public Client Application (PCA) Initialization

The PCA is initialized with the client ID and authority URL from the authconfig file. This allows the add-in to authenticate users and acquire tokens silently or interactively if needed.

## Token Acquisition (Silent and Interactive)

* + **Silent Token Acquisition**: The add-in attempts to acquire an access token silently using the acquireTokenSilent method.
  + **Interactive Token Acquisition**: If silent token acquisition fails, the add-in prompts the user to sign in interactively using the acquireTokenPopup method.

# Event Handling and UI Initialization

## Office.onReady Functionality

The Office.onReady function initializes the add-in when it is loaded into Outlook. It sets up event handlers for new messages and appointments and initializes the UI components.

## Event Details Extraction and Population

The add-in extracts details from the current calendar event (e.g., start date, end date, duration, and subject) and populates the corresponding fields in the task pane.

# Project and Task Management

## Project Search and Selection

Users can search for projects by typing in the project search field. The add-in fetches matching projects from the Dynamics 365 API and displays them in a dropdown list.

## Task Search and Selection

Once a project is selected, the add-in fetches the associated tasks from the Dynamics 365 API and displays them in a dropdown list for the user to select.

## Dynamic Dropdown Population

The dropdown lists for projects and tasks are dynamically populated . The user must select a project to populate the options in Project Tasks.

# Time Entry Creation

## Form Submission Process

When the user clicks the "Insert Time Entry" button, the add-in collects the input values (date, project type, project, task, duration, and description) and constructs a payload.

**Note:-**

Ensure that the total duration of time entries on a specified date does not exceed 24 hours

## API Call to Dynamics 365

The add-in sends a POST request to the Dynamics 365 API to create a new time entry with the collected data.

## Error Handling and Notifications

If the API call fails, the add-in displays an error message to the user and logs the error details.

# UI Components and Event Handling

## Dropdown Lists

The add-in uses custom dropdown lists for project and task selection. These lists are dynamically populated based on user input and API responses.

## Buttons

## 1. Toggle buttons allow Users to switch between "Client" and "Internal" project types.

## A green rectangle and white rectangle AI-generated content may be incorrect.

## 2. The "Insert Time Entry" button allows users to submit data to the Time Entry Table. 3. The "Close" button closes the task pane.

## A white rectangular sign with red text AI-generated content may be incorrect.

## Form Validation

The add-in performs basic validation to ensure that required fields (project and task) are filled before submitting the form.

A screenshot of a project

AI-generated content may be incorrect.

# Error Handling and Notifications

## Error Messages

The add-in displays error messages for invalid inputs or failed API calls. These messages are displayed in the UI.

A screenshot of a phone

AI-generated content may be incorrect.

# G. Detailed Process Flow

## Initialization and Authentication

* The add-in initializes the MSAL PCA with the client ID and authority URL.
* It attempts to acquire an access token silently. If this fails, it prompts the user to sign in interactively.
* Once authenticated, the add-in fetches the user's name from Microsoft Graph using the acquired token.

## Event Handling

* When the add-in is loaded into Outlook, it checks if the current item is a calendar event.
* If it is a calendar event, the add-in extracts the event details (start date, end date, duration, and subject) and populates the corresponding fields in the task pane.

## Project and Task Selection

* The user types in the project search field, and the add-in fetches matching projects from the Dynamics 365 API.
* The projects are displayed in a dropdown list, and the user selects a project.
* Once a project is selected, the add-in fetches the associated tasks and displays them in a dropdown list for the user to select.

## Time Entry Creation

* The user fills in the required fields (date, project type, project, task, duration, and description) and clicks the "Insert Time Entry" button.
* The add-in constructs a payload with the collected data and sends a POST request to the Dynamics 365 API to create a new time entry.
* If the API call is successful, the add-in closes the task pane. If it fails, the add-in displays an error message.

## Error Handling

* The add-in handles errors during token acquisition, API calls, and form submission.
* Error messages are displayed in the UI and logged to the console for debugging.

# 10. Code Walkthrough

## Key Functions and Their Roles

* **initializePCA**: Initializes the Public Client Application for authentication.
* **fetchMatchingProjects**: Fetches projects based on user input.
* **fetchProjectTasks**: Fetches tasks for a selected project.
* **createFieldValues**: Constructs the payload and sends a POST request to create a time entry.

# 11. Conclusion

## Summary of Functionality

The Outlook Add-in provides a seamless way for users to create time entries based on calendar events. It integrates Dynamics 365 APIs to fetch projects, and tasks, and allows users to submit time entries directly from the Outlook task pane.