VIMAL KUMAR S

vimalskr97@gmail.com

Quickminutes – outlook plugin

Technical Research Document

**Table of Contents**

|  |  |
| --- | --- |
| **Contents** | **Page Number** |
| Introduction | 2 |
| Glossary of Terms | 3 |
| Technical Overview of Competitor Applications | 4 |
| Technical Overview of Microsoft Outlook key Features/ Options | 5 |
| Implementation Options | 6 |
| References | 7 |

**Introduction:**

Microsoft has developed Outlook as a personal information manager. Though it is primarily an email client, it has access to other functionalities including

* Calendaring
* Contact Managing
* Meeting Scheduling etc.

Microsoft Outlook account is available for free and with that, we can utilize most basic apps of Microsoft Office 365 suite like word, excel, and calendar. Office 365 has a market share of 38% and it is used by over 1 million companies around the world with full suite. Therefore, if QuickMinutes has an outlook plugin, it can take advantage of the Office 365 user base. The Outlook built-in distribution has been already established in enterprises and leverages its fast-growing platform and developer ecosystem.

Assuming QuickMinutes predominant user base are Microsoft users, a built-in QM plugin for outlook would be very beneficial to the existing customers as it helps them to manage their meeting agendas within their calendar

Outlook has a very extensive feature set in which QM needs to be very specific in developing the plugin and focusing on the niche market that QM holds to integrate its plugin into the Microsoft Outlook environment.

Taking the business needs and market, the plugin must be

1. **Compatible**:

As QM will be targeting the market with more than 500 employees, it must be compatible with all kinds of platforms and devices.

1. **Security:**

The outlook is a mail exchange platform where security must be the key factor. So, the QM plugin must be very secure to use.

Finally, we will discuss the technical stack that is required to build the plugin in the coming section

**Glossary of Terms:**

* API:

Application Programming Interface is a set of functions and procedures allowing the creation of applications that access the features of data of an Operating system, and other services.

* OS:

Operating Systems is a software that controls and manages the hardware and other software on the computer.

* SDK:

A Software Development Kit is an installable package that contains the compiler, debugger, and runtime environment. They are normally specific to a hardware platform and operating system combination.

* REST API:

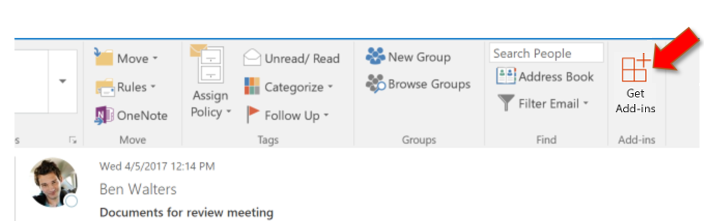
Representational State Transfer is an architectural style for distributed hypermedia systems. Web services that conform to the REST architecture style known as the RESTful Web services provide interoperability between the systems.

**Technical Overview of Competitor Applications:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Compatibility | | | | Security | | Development | |
| OS | | Devices | Integrations | Authentication | Encryption | Language | Environment |
| Desktop | Mobile |
| Docket HQ | NA | NA | Browser | [http://help.dockethq.com/en/articles/3952250-sync-your-microsoft-calendar-with-docket](#_top) | 2-factor Authentication (Must be enabled) | 256 – bit TLS and AES – 216 encryption | N/A(Not open to 3rd party development) | N/A(Not open to 3rd party development) |
| Zoom | Windows 7, Windows 8, Windows 8.1 Windows 10 | iOS 8.0, Android 5.0, and later | Phone, Desktop, Tablet, Browser | [https://support.zoom.us/hc/en-us/articles/360031592971-Getting-Started-with-the-Microsoft-Outlook-Plugin-and-Add-in](#_top) | 2-factor Authentication(Must be enabled) | 256 – bit TLS and AES – 216 encryption | Java and Swift/ Objective-C, C++ | Zoom SDK |
| GoToMeeting | Windows 7, Windows 8, Windows 8.1 Windows 10 | iOS 8.0, Android 5.0, and later | Phone, Desktop, Tablet, Browser | [https://support.goto.com/meeting](#_top) | 2-factor Authentication (Must be enabled) | End to End Encryption | .Net or Java | GoToMeeting REST API (1.0.0) |

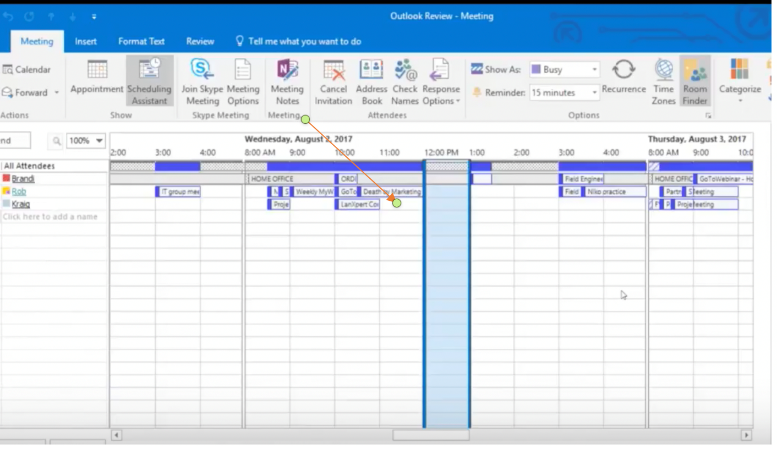
**Technical Overview of key Office Extensions Features:**

* **Creating Meetings from Outlook Mailbox:**
* A meeting in the outlook mailbox can be created by the sideloading feature. A sideloading allows a third-party application to extend its plugin functionalities in the “Get Add-ins” option. This feature differs in different versions of Outlook.
* Besides, the meetings in the mailbox can also be created from AppSource. AppSource allows one to integrate their application with the mailbox.



*Sideloading in Microsoft Outlook Mailbox*

* **Creating Meetings from Outlook Calendar:**
* Outlook calendar allows one to schedule meetings using the Scheduling Assistant. A scheduling assistant takes the input like date and time of the meeting, meeting link, invitees to schedule a meeting.
* Outlook calendars can also use the AppSource so that third-party applications can also collaborate to schedule meetings.



*Scheduling Assistant in Microsoft Calendar*

**Implementation Options:**

**QuickMinutes – Outlook Plugin:**

QuickMinutes application can create a plugin. This plugin can be accessed on both Microsoft Calendar and Microsoft Mailbox. This process involves

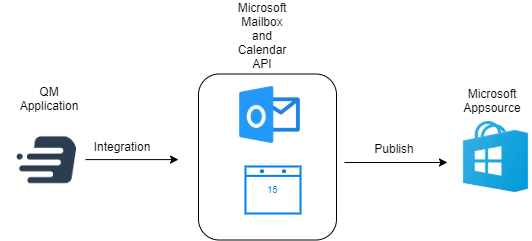
1. Update the requirements in the Manifest file of the plugin. This is the primary step of the creation of the plugin. A Manifest file holds the metadata of the plugin
2. The next step is to build the plugin frontend based on the nature of Outlook mailbox and Calendar compatibility.
3. The integration of the plugin is made using the Outlook API. This holds the functionality required for building the plugin.

* Office – Responsible and extends all the core functionalities of Microsoft Office.
* Mailbox – Allows the plugin to access Outlook Mailbox functions.
* Calendar – Create and access Microsoft Calendar events and date methods.

1. After the plugin is created, it can be hosted via platforms including,

* Sideloading – allows one to access plugin when the mail is being composed.
* AppSource – It is a storehouse or the place where all the Microsoft or the third-party applications are being hosted.

When a plugin is published in AppSource, can be accessed in both mailbox and calendar.



*Implementation of QM Plugin*

**References:**

1. [https://docs.microsoft.com/en-us/office/dev/add-ins/outlook/outlook-add-ins-overview](#_top)
2. [https://docs.microsoft.com/en-us/office/dev/add-ins/reference/requirement-sets/outlook-api-requirement-sets](#_top)
3. [https://docs.microsoft.com/en-us/office/dev/add-ins/tutorials/outlook-tutorial](#_top)
4. [https://docs.microsoft.com/en-us/office/dev/add-ins/publish/publish](#_top)
5. [https://docs.microsoft.com/en-us/office/dev/add-ins/testing/test-debug-office-add-ins](#_top)
6. [https://docs.microsoft.com/en-us/office/dev/add-ins/concepts/add-in-development-best-practices](#_top)
7. [https://docs.microsoft.com/en-us/office/dev/add-ins/outlook/apis](#_top)
8. [https://docs.microsoft.com/en-us/office/dev/add-ins/develop/understanding-the-javascript-api-for-office](#_top)