



Fachhochschule für die Wirtschaft Hannover
- FHDW -
PRAXISARBEIT

Ressourcen- und Terminplanung in Office365 mit Hilfe von Microsoft Graph API

Name: Adham Aijou
Brinker Straße 72
30851, Langenhagen

Mentor: Prof. Dr. Ing. Klinger

Studiengruppe: HFI421IN

Matrikelnummer: 600142

Ausbildungsbetrieb: DOOH media GmbH
Frankenring 18
30855 Langenhagen

Eingereicht am: xx.xx.xxxx



Inhaltsverzeichnis

1	Einleitung	2
1.1	Fragestellungen der Arbeit	2
1.2	Ziele der Arbeit	2
2	Theorie	3
2.1	Microsoft Graph API	3
3	Ist-Zustand	4
3.1	Definitionen	4
4	Soll-Zustand	5
4.1	Anforderungen	5
4.2	User Interface	5
4.3	LED Strips	5
5	Vorgehensweise	6
5.1	Prototyp	6

1 Einleitung

Die Praxisarbeit befasst sich mit der Ressourcen- und Terminplanung in Office365 mit Hilfe von Microsoft Graph API. Die Arbeit ist in drei Teile gegliedert. Im ersten Teil wird die Theorie der Ressourcen- und Terminplanung in Office365 mit Hilfe von Microsoft Graph API erläutert. Im zweiten Teil wird die praktische Umsetzung der Theorie beschrieben. Im dritten Teil wird die Arbeit abschließend bewertet.

1.1 Fragestellungen der Arbeit

Die Fragestellungen der Arbeit lauten:

- Wie funktioniert die Ressourcen- und Terminplanung in Office365 mit Hilfe von Microsoft Graph API?
- Wie kann die Ressourcen- und Terminplanung in Office365 mit Hilfe von Microsoft Graph API praktisch umgesetzt werden?
- Wie kann die Arbeit abschließend bewertet werden?

1.2 Ziele der Arbeit

Die Ziele der Arbeit sind deshalb wie folgt:

- Die Theorie der Ressourcen- und Terminplanung in Office365 mit Hilfe von Microsoft Graph API zu erläutern.
 - Die praktische Umsetzung der Theorie zu beschreiben.
 - Die Arbeit abschließend zu bewerten.
-

2 Theorie

2.1 Microsoft Graph API

Die Microsoft Graph API ist eine RESTful web API, die es einem erlaubt auf Daten von Microsoft 365 und Office 365 zuzugreifen. Mit Hilfe dieser API wurde das Projekt letztendlich umgesetzt. Weitere standen jedoch zur Verfügung:

- Microsoft Outlook API
- Microsoft Exchange API
- Microsoft SharePoint API
- Microsoft OneDrive API
- Microsoft Teams API
- Microsoft Power Automate

Einige dieser APIs sind nur für bestimmte Microsoft 365 und Office 365 Abonnements verfügbar. Die Microsoft Graph API ist jedoch für alle Abonnements verfügbar. Zudem ist die Microsoft Graph API die einzige API, die es einem erlaubt auf alle Daten von Microsoft 365 und Office 365 zuzugreifen, da sie die meisten anderen APIs integriert. Der wichtigste Faktor bei der Entscheidung war es jedoch, dass die Microsoft Graph API, mit Hilfe von Azure AD, die Authentifizierung und Autorisierung von Benutzern erlaubt. Dies ist für die Anwendung von großer Bedeutung, da es dem Benutzer ermöglicht sich mit seinem Microsoft 365 oder Office 365 Account anzumelden und somit auf seine Daten zuzugreifen.

3 Ist-Zustand

Unsere Muttergesellschaft und einige Schwesterunternehmen nutzen Microsoft 365 und Office 365. Diese Produkte sind sehr umfangreich und bieten viele Funktionen. Die heutzutage gängige und weit verbreitete Terminplanung per Outlook oder Teams ist eines dieser Funktionen. Diese Funktion ist jedoch nicht immer so praktikabel wie sie es vielleicht sein sollte, vor allem nicht, wenn mehrere Unternehmen, das gleiche Gebäude und die gleiche Organisations-E-Mail besitzen.

//UserJourney Beispiel einfügen

Falls ein User zufällig an einem Raum vorbeigeht oder vor einem Raum steht und sich fragt, ob dieser Raum belegt ist oder nicht, muss er erstmal Outlook oder Teams auf einem Gerät öffnen, zum Kalender des jeweiligen Raumes, falls er darauf überhaupt Zugriff hat und dann schauen, ob dieser Raum belegt ist oder nicht. Dies ist sehr umständlich und kann sehr viel Zeit in Anspruch nehmen.

3.1 Definitionen

- RESTful: RESTful ist ein Synonym für Representational State Transfer. RESTful ist ein Architekturstil für die Entwicklung von Webdiensten.
 - : API steht für Application Programming Interface. API ist eine Schnittstelle, die es einem erlaubt auf Daten zuzugreifen.
 - : REST steht für Representational State Transfer. REST ist ein Architekturstil für die Entwicklung von Webdiensten.
 - : HTTP steht für Hypertext Transfer Protocol. HTTP ist ein Protokoll, das es einem erlaubt auf Daten zuzugreifen.
 - : JSON steht für JavaScript Object Notation. JSON ist ein Datenformat, das es einem erlaubt Daten zu speichern und zu übertragen.
 - : OAuth ist ein Protokoll, das es einem erlaubt sich mit einem Account anzumelden und somit auf Daten zuzug
 - UserJourney: UserJourney ist ein Begriff aus der User Experience Design. UserJourney ist ein Weg, den ein User durchläuft, um ein Ziel zu erreichen.
 - UserInterface: UserInterface ist ein Begriff aus der User Experience Design. UserInterface ist die grafische Oberfläche, die ein User sieht und mit der er interagiert.
-

4 Soll-Zustand

4.1 Anforderungen

Ein User soll am Bildschirm eines Tablets, welches vor dem Raum angebracht wird, erkennen können, ob dieser Raum belegt ist oder nicht, welche Termine heute noch anstehen und spontan auch einen Termin vereinbaren können. Der User soll also nicht mehr auf Outlook oder Teams angewiesen sein, sondern kann direkt am Bildschirm des Tablets sehen, ob der Raum belegt ist oder nicht. Zudem soll der Raumstatus farbig dargestellt werden, sodass der User sofort erkennen kann, ob der Raum belegt ist oder nicht, sowohl am User Interface, als auch an den LED Strips des Tablets.

4.2 User Interface

Das User Interface soll so gestaltet sein, dass der User sofort erkennen kann, ob der Raum belegt ist oder nicht. Zudem soll das User Interface so gestaltet sein, dass der User auch Termine für den Raum vereinbaren kann. Das User Interface soll also eine Übersicht über die Termine des Raumes und einen Button zum vereinbaren eines Termins enthalten.

4.3 LED Strips

Die LED Strips sollen die Farbe des Raumes anzeigen. Wenn der Raum belegt ist, sollen die LED Strips rot leuchten. Wenn der Raum frei ist, sollen die LED Strips grün leuchten. Wenn der Raum nicht verfügbar ist, sollen die LED Strips rot leuchten. Falls jedoch der Raum in den nächsten 15 Minuten belegt sein wird, sollen die LED Strips gelb leuchten.

5 Vorgehensweise

5.1 Prototyp

Erst wurde ein Prototyp mithilfe von vuejs entwickelt, basierend auf folgendem Mockup:

