

Features

Johann Haag, Tarik Hosic und Simon Blaha

20.12.2018, Leonding

Project Name	Smart Organizer
Project Leader	Simon Blaha
Version	1.0
Document state	In process

Contents

1	Feature List	3
1.1	Rest-Server	3
2	Feature Descriptions	3
2.1	Rest-Server	3
2.1.1	Save appointment on Rest-Server	3
2.1.2	Edit appointment on Rest-Server	3
2.1.3	Delete appointment on Rest-Server	4
2.1.4	Create group on Rest-Server	4
2.1.5	Delete group on Rest-Server	4
2.1.6	Get groups of user	4
2.1.7	Get all appointments	5
2.1.8	Login user	5
2.1.9	Logout user	5
2.1.10	Check user	5

1 Feature List

1.1 Rest-Server

- Save appointment on Rest-Server
- Edit appointment on Rest-Server
- Delete appointment on Rest-Server
- Create group on Rest-Server
- Delete group on Rest-Server
- Get group from User
- Get all appointments
- Login user
- Logout user
- Check user

2 Feature Descriptions

2.1 Rest-Server

2.1.1 Save appointment on Rest-Server

Per POST-Request kann man einen Termin an den Server schicken. Der Termin muss als JSON übermittelt werden. Des Weiteren ist ein gültiger Benutzername sowie Token zu übergeben. Wenn diese stimmen, so wird der Termin für den entsprechenden User auf dem Server gespeichert.

2.1.2 Edit appointment on Rest-Server

Es kann per POST-Request ein existierender Termin bearbeitet werden. Hierfür muss der Termin als JSON an den Server übermittelt werden. Des Weiteren sind noch Benutzername sowie Token zu übergeben. Wenn der übergebene Termin existiert und der Username sowie Token gültig ist, wird die Änderung des Termins

am Server durchgeführt. Bei dem übergebenen Termin muss es sich nicht um einen vollständiges Terminobjekt handelt, es müssen nur die Felder ausgefüllt werden, die geändert werden sollen. Hierbei bildet nur das ID-Feld des Termins eine Ausnahme, weil dieses zum Suchen des zu ändernden Termins verwendet wird. Falls ein Fehler auftreten sollte, so wird dieser als JSON an den POST-Request-Sender zurückgegeben.

2.1.3 Delete appointment on Rest-Server

Zur Verwendug des Features muss hierbei ein Username, Token sowie die ID des zu löschenden Termins übergeben werden. Auf dem Server wird überprüft, ob es einen Termin für den übergebenen User existiert und ob der übergebene Token gültig ist. Ist dies der Fall, so wird der Termin gelöscht. Falls ein Fehler auftreten sollte, so wird dieser als JSON an den POST-Request-Sender zurückgegeben.

2.1.4 Create group on Rest-Server

Auf dem Server kann per POST-Request eine Gruppe erstellt werden. Hierbei muss die zu erstellende Gruppe als JSON übertragen werden. Des Weiteren muss Username sowie Token übertragen werden. Mittels dieser Informationen wird überprüft, ob der User existiert und die Identität per Token festgestellt. Es wird anschließend, falls die Überprüfungen erfolgreich waren, eine Gruppe für den User erstellt. Der Besitzer der Gruppe ist der Ersteller. Bei Fehlerfall bzw. Erfolg wird die entsprechende Rückmeldung an den POST-Request-Sender als JSON zurückgeschickt.

2.1.5 Delete group on Rest-Server

Um dieses Feature zu nutzen muss per POST-Request die Gruppen-ID sowie Username und Token übergeben werden. Beim übergebenen User muss es sich um den Ersteller der Gruppe handeln. Falls diese Bedingungen erfüllt werden, wird die Gruppe gelöscht und per JSON der Erfolg zurückgegeben. Wenn das Löschen fehlschlägt, so wird eine Fehlermeldung als JSON zurückgegeben.

2.1.6 Get groups of user

Der Server antwortet auf einen POST-REQUEST. Bei dem Request muss per JSON ein Username übergeben werden. Falls dieser User existiert, werden alle

Gruppen zurückgegeben, zu denen der User gehört.

2.1.7 Get all appointments

Um diese Funktionalität zu verwenden, muss an den Rest-Server ein POST-Request mit Username und Token geschickt werden. Falls, der Username sowie Token gültig ist, so werden alle Termine als JSON an den Sender zurückgeschickt. Bei einem Fehlerfall wird als JSON der Fehler an den Sender zurückgegeben.

2.1.8 Login user

Um diese Funktionalität des Rest-Servers zu verwenden, muss mittels POST-Request eine JSON-Datei mit gültigen Username und Passwort geschickt werden. Falls das Passwort als auch Username gültig waren, so wird ein Token zurückgeschickt. Bei Misserfolg wird eine Fehlermeldung als JSON an den Sender zurückgeschickt.

2.1.9 Logout user

Um diese Funktionalität des Rest-Servers in Anspruch zu nehmen, muss mittels POST-Request ein JSON mit Username sowie Token geschickt werden. Wenn sich Token sowie Username als gültig herausstellen, wird der Erfolg als JSON an den Sender zurückgeschickt. Bei Misserfolg wird eine Fehlermeldung an den Sender geschickt.

2.1.10 Check user

Mit dieser Funktionalität des Servers kann festgestellt werden, ob ein Username sich bereits in Verwendung befindet. Um die Funktionalität des Servers zu verwenden muss ein POST-Request mit dem zu überprüfenden Usernamen übermittelt werden. Der Server gibt anschließend zurück, ob dieser Username frei ist oder sich bereits in Verwendung befindet.