# MyNote API-Dokumentation

## Vorwort

Jede Funktion, auch solche, bei denen als Rückgabe nur void stehen, geben dem User ein Resultat zurück, welches detailliert, ob die Funktion geklappt hat, oder wegen einer Exception (habe ich noch nicht geschrieben lol) fehlgeschlagen ist.

Format ist JSON, alles wird über POST-Requests gemacht.

## getAuthToken

### Parameter:

string username

string password

### Rückgabe:

string token

### Definition:

Gibt für einen Benutzernamen und ein Password ein Authentifizierungstoken zurück, welches nach Generierung für 24h oder bis ein neues generiert wurde gültig ist. Dieses Token ist die Basis für alle Aktionen und wird in der User-Tabelle aufgeführt.

## uploadSemester

### Parameter:

string auth – Authentifizierungstoken

string semester – Das als JSON serialisierte Semester

### Rückgabe:

int semesterID – ID des eventuell neu erstellten Semesters

### Definition:

Lädt ein Semester hoch.

## uploadCourse

### Parameter:

string auth – Authentifizierungstoken

string course – Der als JSON serialisierte Kurs

### Rückgabe:

int courseID – ID des eventuell neu erstellten Kurses

### Definition:

Lädt einen Kurs hoch. Falls die Online-ID des Kurses bereits existiert, und der User Zugriffsrecht auf diesen Kurs besitzt, wird er lediglich aktualisiert.

## uploadCanvas

### Parameter:

string auth – Authentifizierungstoken

int courseID – ID des Kurses

string canvas – als JSON serialisiertes Canvas

### Rückgabe:

int courseID

### Definition:

Lädt ein Canvas in einen Kurs hoch und erhält die neue ID des Canvasses zurück.

## deleteCanvas

### Parameter:

string auth – Authentifizierungstoken

int canvasID

### Rückgabe:

void

### Definition:

Löscht ein Canvas. Diese Funktion kann nur vom Ersteller des Canvasses ausgeführt werden. Falls ein Canvas von einem Admin aus einem Kurs entfernt werden soll, muss es privat gestellt werden.

## setCoursePassword

### Parameter:

string auth – Authentifizierungstoken

int courseID – die ID des Kurses

string password – neues Passwort des Kurses

### Rückgabe:

void

### Definition:

Setzt ein Passwort für einen Kurs. Der Benutzer muss Admin dieses Kurses sein, um diese Aktion durchführen zu können. Bereits bestehende Benutzer werden nicht entfernt.

## accessCourse

### Parameter:

string auth – Authentifizierungstoken

int courseID – die ID des Kurses

string password – bestehendes Kurspasswort

### Rückgabe:

void

### Definition:

Gibt einem Benutzer Zugriff auf einen Kurs.

## getUser

### Parameter:

string auth – Authentifizierungstoken

int userID – ID des Benutzers

### Rückgabe:

string username – Name des Benutzers

### Definition:

Gibt für eine UserID den zugehörigen Benutzernamen zurück.

## userIsAdmin

### Parameter:

string auth – Authentifizierungstoken

int courseID – ID des Kurses

int userID – ID des Benutzers

### Rückgabe:

bool isAdmin – Wahrheitswert, ob der User Admin ist oder nicht

### Definition:

Bringt in Erfahrung, ob ein Benutzer in einem bestimmten Kurs Admin ist oder nicht.

## setUserAdmin

### Parameter:

string auth – Authentifizierungstoken

int courseID – ID des Kurses

int userID – ID des Benutzers

bool admin – Wahrheitswert, ob der User Admin sein soll oder nicht

### Rückgabe:

void

### Definition:

Stellt ein, ob ein Benutzer Admin eines Kurses ist, oder nicht. Der Ersteller eines Kurses ist und bleibt immer Admin.

## removeUserFromCourse

### Parameter:

string auth – Authentifizierungstoken

int courseID – ID des Kurses

string username – Name des zu entfernenden Users

### Rückgabe:

void

### Definition:

Entfernt einen Benutzer aus einem Kurs. Dieser Befehl kann nur von Admins verwendet werden.

## moveCourseToSemester

### Parameter:

string auth – Authentifizierungstoken

int courseID – ID des Kurses

int semesterID – ID des neuen Semesters

### Rückgabe:

void

### Definition:

Setzt einen Kurs in ein bestimmtes Semester.

## setCanvasPublicOrPrivate

### Parameter:

string auth – Authentifizierungstoken

int canvasID – ID des Canvas

bool public – ob das Canvas Public oder Private sein soll

### Rückgabe:

void

### Definition:

Markiert ein Canvas als öffentlich oder privat. Öffentliche Canvasses können von allen Nutzern in einem Kurs gesehen werden, private nur vom Ersteller.

## getCourseCanvasIDs

### Parameter:

string auth – Authentifizierungstoken

int courseID – ID des Kurses

### Rückgabe:

int[] IDs

### Definition:

Gibt dem Benutzer die IDs von allen für ihn sichtbaren Canvasses eines Kurses zurück.

## getOwnCanvasIds

### Parameter:

string auth – Authentifizierungstoken

### Rückgabe:

int[] IDs – Array von CanvasIDs

### Definition:

Gibt alle IDs von den Canvassen des Benutzers zurück.

## getCanvas

### Parameter:

string auth – Authentifizierungstoken

int canvasID – ID des Canvas

### Rückgabe:

string serializedCanvas – Als Binary serialisiertes Canvas

### Definition:

Gibt dem Benutzer, soweit das Canvas für ihn sichtbar ist, ein serialisiertes Canvas zurück.

## interpretCourse

### Parameter:

string auth – Authentifizierungstoken

int courseID – ID des Kurses

string serializedCourse – der interpretierte Kurs

### Rückgabe:

void

### Definition:

Wenn ein User einen Kurs, in dem er ist, verändert (Namen, Farbe, Icon etc), interpretiert er ihn und kann dies mit diesem Befehl hochladen. Diese Interpretation ist nur für ihn sichtbar und er muss kein Admin sein.

## uploadTest

### Parameter:

string auth – Authentifizierungstoken

int courseID – ID des Kurses

string serializedTest – als JSON serialisiertes Test-Objekt

### Rückgabe:

void

### Definition:

Lädt einen Test in einen Kurs hoch. Falls die Online-ID des Tests bereits existiert, wird er nur verändert. Dieser Befehl kann nur von Admins ausgeführt werden.

## deleteTest

### Parameter:

string auth – Authentifizierungstoken

int testID – ID des Tests

### Rückgabe:

void

### Definition:

Löscht einen Test aus einem Kurs. Dieser Befehl kann nur von Admins ausgeführt werden.