

BACKEND FÜR APPLIKATIONEN REALISIEREN

Name: Mouayad Alnhlawe

Datum: 28. Oktober 2022

Version: 1.0

Inhaltsverzeichnis

1	Ausgangslage	2
2	Analyse	2
3	Namensschema	2
4	Endpoints	4
5	Authentifizierung.....	4
6	Arbeitsjournal	5
6.1	Arbeitsjournal 24.10.2022.....	5
6.2	Arbeitsjournal 25.10.2022.....	6
6.3	Arbeitsjournal 26.10.2022.....	7
6.4	Arbeitsjournal 27.10.2022.....	8
6.5	Arbeitsjournal 28.10.2022.....	9

Tabellenverzeichnis

Tabelle 1	category_function.....	2
Tabelle 2	product_function	3
Tabelle 3	Dateiennamen	3
Tabelle 4	Endpoints.....	4
Tabelle 5	Username & Password	4
Tabelle 6	Arbeitsjournal 24.10.2022	5
Tabelle 7	Arbeitsjournal 25.10.2022	6
Tabelle 8	Arbeitsjournal 26.10.2022	7
Tabelle 9	Arbeitsjournal 27.10.2022	8

Quellenangabe

Ich verwendete den Prinzip, den ich in der Schule von Herr Sollberger und die Hausaufgaben gelernt habe.

1 Ausgangslage

Ich muss eine API zur Verwaltung von Produkten eines Online-Shops erstellen. Alle Endpoints (erstellen, lesen, ändern und löschen) müssen dem REST-Standard entsprechen. Ich muss die Datenbank mit den nötigen Tabellen und Eigenschaften einrichten. Die Anfrage und Antwort müssen in JSON strukturiert werden. Ich muss JWT benutzen, damit ich alle Endpoints schützen kann.

2 Analyse

Die erste Schritt ist, ein Code zu schreiben, um die Zugriff auf die Datenbank zu erhalten, damit ich neue Daten erstellen, lesen, ändern und löschen kann.

Danach muss ich ein anderen Datei machen und definieren, was die Endpoints machen sollen.

Einen dritten Datei brauche ich für die REST-API. Dort muss ich die Request und Response Funktionen definieren.

Ich fange an, alle API-Code vorzubereiten. Ich werde mit Authenticate anfangen, weil es sehr einfach ist. Ich muss ein neuen Datei (config) erstellen, um die Username und Password, ohne dass jemanden sie sehen zu kann, speichern. Ein Dritten Datei (require_authentication) muss ich noch machen, damit ich nicht den gleichen Code bei jeden Endpoints zu schreiben. So muss ich nur beim Anfang der Endpoints require zu den dritten Datei zu machen.

Für die Error-Meldungen werde ich auch ein Datei dafür machen, damit ich nicht immer den gleichen Code kopieren.

Alles muss ich noch einmal machen, aber für die Product.

3 Namensschema

Man muss immer snake_case benutzen, wenn man Webseite programmiert, deswegen werde ich alle Namens als snake_case machen. Für die Functions werde ich zwei Dateien machen.

category_function	Beschreibung
get_category	Die ausgewählte Daten aus Category auflisten.
get_all_categorys	Listet alle Daten aus DB Category
create_new_category	Wird die eingegebene Daten an MySQL geschickt, um neue Daten in Category zu erstellen
update_category	Aktualisiert alle eingegeben Daten in Category
delete_category	Es löscht die eingegebene Daten in Category

Tabelle 1 category_function

product_function	Beschreibung
get_product	Die ausgewählte Daten aus die product auflisten.
get_all_products	Listet alle Daten aus DB Product
create_new_product	Wird die eingegebene Daten an MySQL geschickt, um neue Daten in Product zu erstellen
update_product	Aktualisiert alle eingegeben Daten in Product
delete_product	Es löscht die eingegebene Daten

Tabelle 2 product_function

Die Dateinamen muss man auch als snake_case schreiben.

Dateinamen	Beschreibung
index.php	Hauptdatei
swagger.php	Ist für die Kommentare definiert
database.php	Anfrage für die Zugriff auf die DB
function_category	Alle Functions des Category werden definiert
function_product.php	Alle Functions des Products werden definiert
endpoints_category.php	Alle API und Kommentare werden hier schreiben
endpoints_product.php	Alle API und Kommentare werden hier schreiben
error_category	Errordatei, damit ich alle Error für die Category machen kann
error_product	Errordatei, damit ich alle Error für die Product machen kann
require_authentication.php	Wenn nicht authentisiert ist, frage diese Datei nach Authentication
config.php	Alle Usernames und Passwörter werden hier gespeichert, damit niemand sie sehen kann.

Tabelle 3 Dateinamen

Für die Variablen muss man auch snace_case machen, deswegen alle meine Variablen, die ich erstelle, habe, mit snace_case gemacht.

4 Endpoints

Damit man Daten auflisten, braucht man immer die Endpoint GET. Nachdem man die GET-Endpoint geschrieben hat, muss man definieren, welchen Daten aufgelistet werden soll. Man braucht also den KEY und VALUE bzw. Parameter. Aber mit REST_API muss man die ID in den http schreiben.

Und damit man neue Daten in die Datenbank erstellen kann, muss man POST benutzen. Man muss kein ID schreiben, weil man ein neuen ID erstellen will. Wenn man ein ID in http schreibt, dann ist es nicht mehr Erstellung von Daten, sondern Aktualisierung der Daten.

PUT braucht man, wenn man die Daten aktualisieren will. Man muss ein ID in den http schreiben, sonst werden alle Daten aktualisiert.

DELETE ist für die Löschen von Daten. Man muss auch den ID eingeben, damit man nicht alle Daten der Datenbank löscht.

Endpoints	HTTP-Anfragemethode
POST	/Authenticate
POST	/Product
POST	/Category
GET	/Product/{product_id}
GET	/Products
GET	/Category /{category_id}
GET	/Categorys
PUT	/Product/{product_id}
PUT	/Category /{category_id}
DELETE	/Product/{product_id}
DELETE	/Category /{category_id}

Tabelle 4 Endpoints

5 Authentifizierung

Für die Authentifizierung benutze ich JWT (JSON Web Token). So kann man die Endpoints schützen, bis man sich mit den Username und Password anmeldet.

Username	Password
admin	sec!ReT423*&

Tabelle 5 Username & Password

6 Arbeitsjournal

6.1 Arbeitsjournal 24.10.2022

Tätigkeiten Von Mouayad Alnhlawe	Personen	Zeitaufwand	
		SOLL	IST
Während der Unterrichtszeit geleistet			
Projekt lesen	1	20	20
Ausserhalb der Unterrichtszeit geleistete			
Ausgangslage dokumentiert	1	20	20
Anfangen zu planen	1	30	30
Gesamte Tagesleistung	1	70	70
Tagesverlauf			
Wie können ab Heute mit den Projekt anzufangen.			
Probleme			
Keine Probleme gehabt			
Hilfestellung			
Word und LBV			
Reflexion			
Ich habe ein Plan in meinen Kopf und glaube, es wird funktionieren.			
Nächste Schritte			
Analyse			

Tabelle 6 Arbeitsjournal 24.10.2022

6.2 Arbeitsjournal 25.10.2022

Tätigkeiten Von Mouayad Alnhlawe	Personen	Zeitaufwand	
		SOLL	IST
Während der Unterrichtszeit geleistet			
Analyse dokumentieren	1	45	50
Endpoints dokumentieren	1	30	30
Authentifizierung dokumentieren	1	20	20
Datenbank in phpmyadmin erstellt und Beziehung gemacht			
Ausserhalb der Unterrichtszeit geleistete			
Namensschema dokumentieren	1	20	20
Gesamte Tagesleistung	1	115	120
Tagesverlauf			
Heute habe ich mit den Dokumentation angefangen. Ich habe die Analyse, Endpoints, Authentifikation und Namensschema fertig dokumentiert. Ich habe die Datenbank in phpmyadmin erstellt und die Beziehung gemacht.			
Probleme			
Keine Probleme gehabt			
Hilfestellung			
Word und phpMyAdmin			
Reflexion			
Ich habe heute gelernt, wie man die Beziehung in phpMyAdmin machen kann.			
Nächste Schritte			
Programmieren			

Tabelle 7 Arbeitsjournal 25.10.2022

6.3 Arbeitsjournal 26.10.2022

Tätigkeiten Von Mouayad Alnhlawe	Personen	Zeitaufwand	
		SOLL	IST
Während der Unterrichtszeit geleistet			
Authenticate programmiert	1	200	200
GET programmiert	1	20	20
Ausserhalb der Unterrichtszeit geleistete			
GET fertig programmiert	1	20	20
Gesamte Tagesleistung	1	240	240
Tagesverlauf			
Heute habe ich angefangen zu programmieren. Ich habe heute Probleme mit die Authenticate gehabt, deswegen habe ich viel Zeit benutzt, als was ich geplant habe.			
Probleme			
Authenticate nicht funktioniert			
Hilfestellung			
VS Code, phpMyAdmin			
Reflexion			
Ich habe heute GET fertig gemacht und kann sagen, dass es einfach war, da wir es gut mit Herr Sollberger gelernt haben			
Nächste Schritte			
POST und PUT			

Tabelle 8 Arbeitsjournal 26.10.2022

6.4 Arbeitsjournal 27.10.2022

Tätigkeiten Von Mouayad Alnhlawe	Personen	Zeitaufwand	
		SOLL	IST
Während der Unterrichtszeit geleistet			
POST programmiert	1	100	100
PUT programmiert	1	60	60
Swagger Kommentare gemacht		40	40
Ausserhalb der Unterrichtszeit geleistete			
POST fertig programmiert	1	40	40
PUT fertig programmiert	1	40	40
Authenticate fertig programmiert	1	40	40
Swagger-Kommentare fertig gemacht	1	90	90
Gesamte Tagesleistung	1	410	410
Tagesverlauf			
Heute habe ich viel Probleme gehabt, weil ich viele Sachen gleichzeitig gemacht, aber habe direkt Hilfe von Herr Sollberger bekommen.			
Probleme			
Swagger, Authenticate			
Hilfestellung			
VS Code, Word, Herr Sollberger, Swagger-ui			
Reflexion			
Ich habe heute viel gelernt. Am Anfang habe ich gelernt, wie man Swagger-Errors korregieren kann. Danach POST und PUT haben viel Zeit gebraucht, aber sind am Ende gegangen. Authenticate ist nicht an mein Prinzip gegangen, aber als ich mit Herr Sollberger gefragt habe, hat er mir gesagt, wie ich ein Prinzip in Authenticate machen kann			
Nächste Schritte			
Product und die Kommentare fertig machen			

Tabelle 9 Arbeitsjournal 27.10.2022

6.5 Arbeitsjournal 28.10.2022

Tätigkeiten Von Mouayad Alnhlawe	Personen	Zeitaufwand	
		SOLL	IST
Während der Unterrichtszeit geleistet			
Product fertig programmiert	1	150	150
Swagger fertig programmiert	1	50	50
Getestet, ob alles funktioniert	1	20	20
Ausserhalb der Unterrichtszeit geleistete			
Gesamte Tagesleistung	1	220	220
Tagesverlauf			
Heute habe ich alles kopiert und für die Product gemacht. Swagger habe ich heute auch fertig gemacht. Am Ende habe ich alles getestet und alles funktioniert bei mir			
Probleme			
POST geht nicht, weil category existiert nicht			
Hilfestellung			
VS Code, phpMyAdmin, Swagger, Herr Sollbergrer			
Reflexion			
Heute habe ich gelernt, dass man den Foreign Key nicht erstellen kann, Herr Sollberger hat mir das gesagt und habe direkt mein Code geändert und hat am Ende funktioniert.			
Nächste Schritte			
Abgeben			