

ÜK 295 BACKEND FÜR APPLIKATIONEN REALISIEREN

Morhaf Alnhlawe
Inf21
CsBe (Computerschule Bern)
Version: 1.5

Inhaltsverzeichnis

Planung	2
Zeitplan	2
Ausgangslage	2
Analyse	2
Endpoints	3
Namensschema	3
Arbeitsjournal	4

Tabellenverzeichnis

Tabelle 1 Zeitplan	2
Tabelle 2 Endpoints	3
Tabelle 3 JWT	3
Tabelle 4 Namensschema_function_category	3
Tabelle 5 Namensschema_function_product	4
Tabelle 6 Namensschema_Dateien	4
Tabelle 7 Arbeitsjournal_Montag	4
Tabelle 8 Arbeitsjournal_Dienstag	5
Tabelle 9 Arbeitsjournal_Mittwoch	5
Tabelle 10 Arbeitsjournal_Donnerstag	6
Tabelle 11 Arbeitsjournal_Freitag	6

Quellenverzeichnis

Ich benötigte das Code_Prinzip, das wir in der Schule von Herr Manuel Sollberger gelernt haben.

Ich habe die Hilfe von unseren Hausaufgaben Dokumente benötigt.

[zircote/swagger-php - Packagist](#) – Vom diese Website habe ich Swagger-php mit Hilfe von Herr Manuel Sollberger installiert.

Planung

Zeitplan

Tabelle 1 Zeitplan

Arbeitspaket	Zeit	Dauer	
	in Stunden	Start	Ende
		Datum	Datum
Auftrag lesen und verstehen	0.5	24.10.2022	24.10.2022
Mit Dokumentation anfangen	1.5	24.10.2022	24.10.2022
Zeitplan erstellen	0.5	24.10.2022	24.10.2022
Ausgangslage	1	24.10.2022	24.10.2022
Analyse	2	25.10.2022	25.10.2022
Endpoints	2	25.10.2022	25.10.2022
Namenschema	0.5	28.10.2022	28.10.2022
Code	20	26.10.2022	28.10.2022
Arbeitsjournal	0.75	24.10.2022	28.10.2022

Ausgangslage

Ich muss eine API erstellen, um Online-Shop-Produkte zu verwalten. Die Endpoints (Erstellen, Lesen, Ändern, Löschen) müssen dem REST-Standard entsprechen. Ich muss die Datenbank mit den erforderlichen Tabellen und Eigenschaften einrichten. Die Anfrage und Antwort müssen in JSON organisiert sein. Ich muss JWT verwenden, damit ich alle Endpunkte schützen kann.

Analyse

Am Anfang muss man durch einen Code den Zugriff auf die Datenbank haben, damit man die Daten erstellen, lesen, ändern und löschen kann.

Der zweite Schritt ist, dass man eine Datei erstellen, indem sind die Endpoints definiert.

Für die REST-API muss man in einer dritten Datei die Request und Response Funktionen definieren.

Im Code beginne ich mit Authentifizierung, es ist einfach, deswegen kann ich es schnell machen. Daneben muss ich eine neue Datei (Konfiguration) erstellen, um meinen Benutzernamen und meine Kennwörter zu speichern. Eine dritte Datei muss ich noch erstellen (require_authentication), damit ich den Code nicht immer wiederholen muss.

Eine mehrere Dateien muss ich für die Error-Meldungen machen, damit ich den Code nicht immer wiederholen muss.

Endpoints

Zum Einfügen von Daten benötigt man immer einen GET-Endpoint. Nachdem man den GET-Endpoint eingegeben hat, sollte man angeben, welche Daten eingefügt werden sollen. man benötigt also den Schlüssel und den Wert oder die Parameter. Aber mit REST_API muss man die ID in http eingeben.

Um neue Daten in der Datenbank zu einfügen, müssen Sie POST verwenden. In http muss man kein ID eingeben, da wir die Daten einfügen müssen und nicht aktualisieren.

PUT benötigt man, um Daten zu aktualisieren, in den http muss man das ID eingeben, danach werden die Daten aktualisiert.

DELETE benötigt man, um Daten zu löschen. Man gibt das ID in den http, um die ausgewählten Daten aus der Datenbank zu löschen.

Tabelle 2 Endpoints

Endpoints	HTTP-Anfragemethode
POST	/Authenticate
POST	/Product
POST	/Product_category
GET	/Product/{product_id}
GET	/Products
GET	/Product_category/{product_id}
GET	/Products_category
PUT	/Product/{product_id}
PUT	/Product_category/{product_id}
DELETE	/Product/{product_id}
DELETE	/Product_category/{product_id}

Authentifizierung

JWT (JSON Web Token) werde ich benötigen, so werden die Endpoints durch Benutzername und Kennwort geschützt.

Tabelle 3 JWT

Benutzername	Kennwort
Admin	Uek-295@M

Namensschema

Tabelle 4 Namensschema_function_category

Category_function	Was es macht:
get_category	Einzelne Daten aus Datenbank auflisten.
get_all_category	Alle Daten aus Datenbank auflisten.
create_new_category	Erstellt Daten in der Datenbank.
update_category	Daten aus der Datenbank können bearbeitet werden.
Delete_category	Löscht Daten aus der Datenbank.

Tabelle 5 Namensschema_function_product

Product_function	Was es macht:
get_prudect	Einzelne Daten aus Datenbank auflisten.
Get_all_prudect	Alle Daten aus Datenbank auflisten.
create_new_prudect	Erstellt Daten in der Datenbank.
update_prudect	Daten aus der Datenbank können bearbeitet werden.
Delete_prudect	Löscht Daten aus der Datenbank.

Tabelle 6 Namensschema_Dateien

Dateiennamen	Was das Datei macht:
Index.php	Hauptdatei
swagger.php	Ist für Kommentare zuständig.
config.php	Alle privaten Daten sind im Config geschrieben.
error_category	Ist für alle Fehler zuständig.
require_authentication.php	Ist für Authentication zuständig.
routes_Category.php	Für API Category zuständig.
Routes_product.php	Für API Product zuständig.
Database.php	Verbinden Datenbank mit dem Code
registration_category	Definiert die Funktionen des API_category
registration_product	Definiert die Funktionen des API_product

Arbeitsjournal

Tabelle 7 Arbeitsjournal_Montag

Tätigkeiten Von Montag, 24.10.2022	Personen	Zeitaufwand	
		SOLL	IST
Während der Unterrichtszeit geleistet			
Auftrag durchlesen	1	20	20
Ausserhalb der Unterrichtszeit geleistete			
Mit Dokumentation angefangen	1	60	45
Zeit geplant	1	30	25
Ausgangslage klar geschrieben	1	60	45
Gesamte Tagesleistung	1	170	135
Tagesverlauf			
Auftrag gelesen, verstanden und mit Dokumentation angefangen			
Probleme			
Keine			
Hilfestellung			
Word			
Reflexion			
Zeit gut geplant			
Nächste Schritte			
Mit Analyse und Endpoints anfangen			

Tabelle 8 Arbeitsjournal_Dienstag

Tätigkeiten Von Dienstag, 25.10.2022	Personen	Zeitaufwand	
		SOLL	IST
Während der Unterrichtszeit geleistet			
Analyse und Endpoints in der Dokumentation geschrieben	1	210	210
Ausserhalb der Unterrichtszeit geleistete			
Gesamte Tagesleistung	1	210	210
Tagesverlauf			
Analyse und Endpoints in der Dokumentation geschrieben			
Probleme			
Keine			
Hilfestellung			
Word			
Reflexion			
Endpoints definiert			
Nächste Schritte			
Mit Code anfangen			

Tabelle 9 Arbeitsjournal_Mittwoch

Tätigkeiten Von Mittwoch, 26.10.2022	Personen	Zeitaufwand	
		SOLL	IST
Während der Unterrichtszeit geleistet			
Mit Code angefangen	1	210	210
Ausserhalb der Unterrichtszeit geleistete			
Code weiterschreiben	1	180	200
Gesamte Tagesleistung	1	390	410
Tagesverlauf			
Ich habe mit Code angefangen			
Probleme			
errors			
Hilfestellung			
Word, Visual Studio Code			
Reflexion			
PHP besser kennengelernt			
Nächste Schritte			
Code weiterschreiben			

Tabelle 10 Arbeitsjournal_Donnerstag

Tätigkeiten Von Donnerstag, 27.10.2022	Personen	Zeitaufwand	
		SOLL	IST
Während der Unterrichtszeit geleistet			
Code weiterschreiben	1	210	210
Ausserhalb der Unterrichtszeit geleistete			
Code weiterschreiben	1	200	400
Gesamte Tagesleistung	1	410	610
Tagesverlauf			
Ich habe mit Code weitergearbeitet			
Probleme			
errors			
Hilfestellung			
Word, Visual Studio Code			
Reflexion			
Swagger kennengelernt			
Nächste Schritte			
Code fertigschreiben			

Tabelle 11 Arbeitsjournal_Freitag

Tätigkeiten Von Freitag, 28.10.2022	Personen	Zeitaufwand	
		SOLL	IST
Während der Unterrichtszeit geleistet			
Dokumentation fertig schreiben	1	150	180
Gesamte Tagesleistung	1	150	180
Tagesverlauf			
Ich habe Dokumentation fertiggeschrieben			
Probleme			
Keine			
Hilfestellung			
Word, Visual Studio Code			
Reflexion			
Ich habe zu ersten mal Namensschema gamacht.			
Nächste Schritte			
Code fertig schreiben			