



# BACKEND FÜR APPLIKATIONEN REALISIEREN

ÜK Modul 295

Nutharshan Mohanabalan  
nutharshan.mohanabalan@ict.csbe.ch

## Inhaltsverzeichnis

1. Informieren.....	2
2. Ziele und Aufgaben .....	2
3. Projekt Datenmodell Definition .....	2
4. Kontrollieren .....	6
5. API Testing Mit Thunder Client .....	7
6. Arbeitsjournale .....	8
7. Quellenverzeichnis .....	11
8. Tabellenverzeichnis .....	12
9. Abbildungsverzeichnis.....	12

## 1. Informieren

### Einführung

Das Hauptziel des Projekts besteht darin, eine robuste API für das effiziente Management von Produkten in einem Online-Shop zu entwickeln. Um dies zu erreichen, ist es unerlässlich, die erforderlichen Endpunkte gemäß dem REST (Representational State Transfer) Standard zu implementieren. Dies erfordert die Erstellung einer Datenbank, die sich nahtlos in die API integrieren lässt und eine reibungslose und genaue Datenverwaltung ermöglicht. Alle eingehenden Anfragen und darauffolgenden Antworten müssen der korrekten JSON (JavaScript Object Notation) Struktur entsprechen.

Sicherheit ist in diesem Zusammenhang von größter Bedeutung. Daher werden strenge Maßnahmen umgesetzt, um die Integrität und Vertraulichkeit des Systems zu schützen. Insbesondere sind alle API-Endpunkte, mit Ausnahme des Authentifizierungsendpunkts, durch JSON Web Tokens (JWT) geschützt. Dies stellt sicher, dass nur autorisierte Benutzer mit gültigen Tokens auf die API zugreifen und mit ihr interagieren können, was zu einer sicheren und vertrauenswürdigen Online-Shopping-Umgebung beiträgt.

## 2. Ziele und Aufgaben

- Eine RESTful API für das Online-Shop-Management entwickeln.
- Benutzerauthentifizierung mit JWT umsetzen.
- Klare und umfassende Dokumentation für das Projekt erstellen.

## 3. Projekt Datenmodell Definition

Folgende Tabellen werden in der Datenbank "product\_manager" erstellt.

**Tabelle : person**

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	
first_name	varchar(255)	YES		NULL	
last_name	varchar(255)	YES		NULL	
password	varchar(255)	YES		NULL	
user_name	varchar(255)	YES	UNI	NULL	

*Tabelle 1 Person*

**Tabelle : product**

Field	Type	Null	Key	Default	Extra
active	tinyint(4)	NO		NULL	
price	float	NO		NULL	
stock	int(11)	NO		NULL	
id	bigint(20)	NO	PRI	NULL	auto_increment
description	mediumtext	NO		NULL	
image	varchar(1000)	NO		NULL	
name	varchar(500)	NO		NULL	
sku	varchar(100)	NO		NULL	

*Tabelle 2 product***Tabelle: product\_category**

Field	Type	Null	Key	Default	Extra
active	bit(1)	NO		NULL	
id	bigint(20)	NO	PRI	NULL	auto_increment
name	varchar(255)	NO	UNI	NULL	

*Tabelle 3 product\_category*

**Funktionen und deren Definitionen:****Tabelle 1: Funktionen des ProductCategoryController**

Function Name	Description
getAllCategories()	Get all product categories.
getCategoryById()	Get product category by ID.
createCategory()	Create a new product category.
updateCategory()	Update an existing product category by ID.
deleteCategory()	Delete a product category by ID.

*Tabelle 4 ProductCategoryController***Tabelle 2: Funktionen des PersonKontroller.**

Function Name	Description
getAllPeople()	Get all people.
getPersonById(id)	Get person by ID.
createPerson(...)	Create a new person.
updatePerson(id, ...)	Update an existing person by ID.
deletePerson(id)	Delete a person by ID.

*Tabelle 5 PersonKontroller***Tabelle 3: Funktionen des ProductKontroller.**

Function Name	Description
getAllProducts()	Get all products.
getProductById(id)	Get product by ID.
createProduct(...)	Create a new product.
updateProduct(id, ...)	Update an existing product by ID.
deleteProduct(id)	Delete a product by ID.

Tabelle 6 ProductController.

## Endpoints

Endpoint	HTTP Method	Description
/api/categories	GET	Get all product categories.
/api/categories/{id}	GET	Get product category by ID.
/api/categories	POST	Create a new product category.
/api/categories/{id}	PUT	Update an existing product category by ID.
/api/categories/{id}	DELETE	Delete a product category by ID.
/api/people	GET	Get all people.
/api/people/{id}	GET	Get person by ID.
/api/people	POST	Create a new person.
/api/people/{id}	PUT	Update an existing person by ID.
/api/people/{id}	DELETE	Delete a person by ID.

Endpoint	HTTP Method	Description
/api/products	GET	Get all products.
/api/products/{id}	GET	Get product by ID.
/api/products	POST	Create a new product.
/api/products/{id}	PUT	Update an existing product by ID.
/api/products/{id}	DELETE	Delete a product by ID.
/auth/login	POST	Authenticate and get an access token (include token in header for authentication).

Tabelle 7 EndPoints

## 4. Kontrollieren

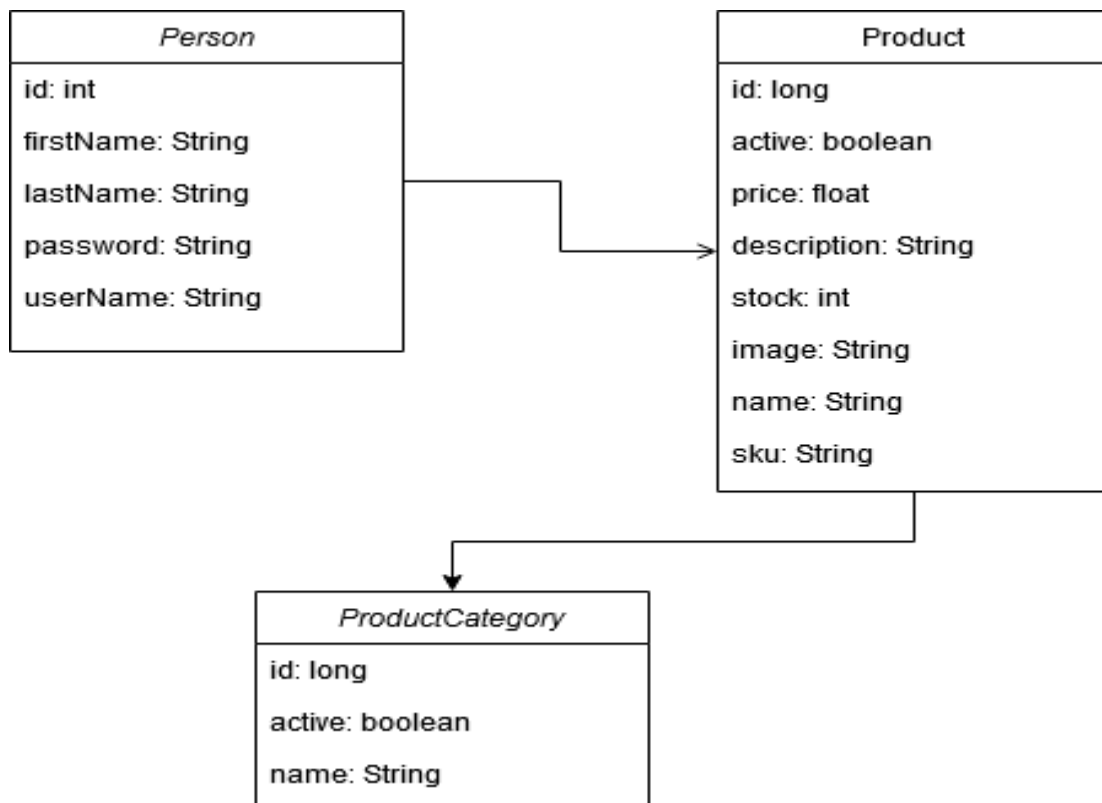


Abbildung 1 Kontrollieren

## 5. API Testing Mit Thunder Client

### a. GET-Anfrage (Produkte oder Kategorien lesen):

- Verwenden Thunder Client, um GET-Anfragen an Endpunkte zu senden, die Produkte oder Kategorien abrufen.
- Überprüfe, ob die Antworten die erwarteten Daten und Statuscodes enthalten.

### b. POST-Anfrage (Produkte oder Kategorien erstellen):

- Sende POST-Anfragen an Endpunkte, um neue Produkte oder Kategorien zu erstellen.
- Füge die erforderlichen Parameter im Anfragekörper hinzu.
- Überprüfe, ob die neuen Daten erfolgreich in der Datenbank erstellt wurden und du die entsprechende Antwort erhältst.

### c. DELETE-Anfrage (Produkte oder Kategorien löschen):

- Verwenden mit Thunder Client, um DELETE-Anfragen an Endpunkte zu senden, die Produkte oder Kategorien löschen.
- Füge die notwendigen Kennungen oder Parameter hinzu, um den zu löschenden Artikel zu spezifizieren.
- Bestätige, dass der Artikel aus der Datenbank entfernt wurde und die Antwort eine erfolgreiche Löschung anzeigt.

### Antworten überprüfen

Untersuche die Antworten, die im Thunder Client erhalten. Stelle sicher, dass die API wie erwartet funktioniert und die entsprechenden Antworten für jeden Anfragetyp (GET, POST, DELETE) zurückgibt.

### Mit Authentifizierung arbeiten:

- Überprüfen, ob Authentifizierungsprozess korrekt funktioniert. Fordere ein JWT-Token an und verwende es in den Headern deiner GET, POST und DELETE Anfragen zu authentifizierten Endpunkten.



## 6. Arbeitsjournale

Arbeitsjournal - Arbeitstag 1	Person	Aufwand geplant (Std)	Aufwand effektiv (Std)
Auftrag lesen	NM	0.5	0.5
Gedanken darüber machen	NM	1.0	1.0
Arbeitsjournale Schreiben	NM	0.5	0.5
Dokumentation bearbeiten	NM	1.0	1.0
<b>Geplante Tagesreserve</b>		0	0
<b>Total:</b>		<b>4.0</b>	<b>4.0</b>
<b>Abweichung SOLL / IST:</b>		<b>0</b>	
<b>Probleme</b>			
keine			
<b>Hilfestellungen</b>			
Keine			
<b>Reflexion</b>			
Aufgrund der Aufgaben, die wir durchgeführt haben, kamen mir sofort Ideen, die ich in der Planung verwenden werde.			
<b>Nächste Schritte</b>			
Projektdokumentation vorbereiten; Planung			

Tabelle 8 Arbeitsjournal - Arbeitstag 1

Arbeitsjournal - Arbeitstag 2	Person	Aufwand geplant (Std)	Aufwand effektiv (Std)
Auftrag lesen	NM	0.5	0.5
Dokumentation vorbereitet	NM	1.0	1.0
Arbeitsjournale Schreiben	NM	0.5	0.5

Alle notwendigen Softwares auf eigenen PC installiert und GitRepository erstellt	NM	2.0	2.0
<b>Geplante Tagesreserve</b>		0	0
<b>Total:</b>		<b>5.0</b>	<b>5.0</b>
<b>Abweichung SOLL / IST:</b>		<b>0</b>	
<b>Probleme</b>			
keine			
<b>Hilfestellungen</b>			
Ich hatte die falsche Version von IntelliJ. Danach habe ich Herrn Techo um Hilfe gebeten, und er hat mir geholfen, die richtige Version von IntelliJ zu installieren.			
<b>Reflexion</b>			
----			
<b>Nächste Schritte</b>			
Weiter Projekt arbeiten und Projektdokumentation vorbereiten.			

Tabelle 9 Arbeitsjournal - Arbeitstag 2

Arbeitsjournal - Arbeitstag 3	Person	Aufwand geplant (Std)	Aufwand effektiv (Std)
Auftrag lesen	NM	0.5	0.5
Dokumentation vorbereitet	NM	1.0	1.0
Arbeitsjournale Schreiben	NM	0.5	0.5
Gemäß dem Datenmodell habe ich die Datenbank und die zugehörigen Tabellen erstellt und miteinander verbunden.	NM	2.0	2.0
<b>Geplante Tagesreserve</b>		0	0
<b>Total:</b>		<b>5.0</b>	<b>5.0</b>
<b>Abweichung SOLL / IST:</b>		<b>0</b>	
<b>Probleme</b>			
keine			
<b>Hilfestellungen</b>			
Ich hatte Probleme beim Verbinden des Primary Key und Foreign Key. Jedoch gelang es mir mithilfe eines YouTube-Videos, diese korrekt zu verknüpfen..			

<b>Reflexion</b>
Beim Erstellen der Datenbank und der dazugehörigen Tabellen habe ich darüber nachgedacht, wie ich die Funktionen gestalten möchte. Dabei habe ich bemerkt, dass ich eine leichte Unsicherheit oder Angst verspürte.
<b>Nächste Schritte</b>
Weiter Projekt arbeiten und Projektdokumentation vorbereiten.

Tabelle 10 Arbeitsjournal - Arbeitstag 3

Arbeitsjournal - Arbeitstag 4	Person	Aufwand geplant (Std)	Aufwand effektiv (Std)
Auftrag lesen	NM	0.5	0.5
Endpoints beschrieben	NM	1.0	1.0
Arbeitsjournale Schreiben	NM	0.5	0.5
Realisierung: Script	NM	16	16
<b>Geplante Tagesreserve</b>		0	0
<b>Total:</b>		<b>19</b>	<b>19</b>
<b>Abweichung SOLL / IST:</b>		<b>0</b>	
<b>Probleme</b>			
<p>Ich hatte ReallySimpleJWT\Token; nicht installiert, da ich dies übersehen hatte. Deswegen habe ich viel Zeit verloren, weil ich die Installation selbst durchführen wollte.</p> <p>Bei jedem meiner Probleme hat mir ChatGPT geholfen. Ich war überrascht, wie gut es mit ChatGPT funktioniert hat.</p>			
<b>Reflexion</b>			
---			
<b>Nächste Schritte</b>			
Weiter Projekt arbeiten und Projektdokumentation vorbereiten.			

Tabelle 11 Arbeitsjournal - Arbeitstag 4

Arbeitsjournal - Arbeitstag 5	Person	Aufwand geplant (Std)	Aufwand effektiv (Std)
Auftrag lesen	NM	0.5	0.5
Code beschriften	NM	1.0	1.0
Arbeitsjournale Schreiben	NM	0.5	0.5
Projektdokumentation fertigstellen	NM	1.0	1.0
Code beschriften	NM	2	2
Abgabebereit machen			
<b>Geplante Tagesreserve</b>		0	0
<b>Total:</b>		<b>6</b>	<b>6</b>
<b>Abweichung SOLL / IST:</b>		<b>0</b>	
<b>Probleme</b>			
Bei jedem Problem habe ich YouTube und ChatGPT verwendet und habe immer eine Lösung gefunden.			
<b>Reflexion</b>			
In diesem Projekt geriet ich unter Stress, da ich zu spät begonnen habe und nicht frühzeitig mit vollem Einsatz daran gearbeitet habe. Das muss ich definitiv ändern. Zudem habe ich festgestellt, dass die Suche nach Fehlern in der Informatik sehr mühsam und entmutigend sein kann, besonders wenn ein kleiner Tippfehler, wie ein fehlendes Komma, alles zum Einsturz bringt.			
<b>Nächste Schritte</b>			
Projekt Abgabe.			

Tabelle 12 Arbeitsjournal - Arbeitstag 5

## 7. Quellenverzeichnis

<https://www.youtube.com/watch?v=WXsD0ZgxjRw>

<https://www.chatgpt.com>

[https://www.youtube.com/watch?v=O\\_XL9oQ1\\_To](https://www.youtube.com/watch?v=O_XL9oQ1_To)

<https://www.youtube.com/watch?v=mPPhcU7oWDU>

## 8. Tabellenverzeichnis

Tabelle 1 Person.....	2
Tabelle 2 product.....	3
Tabelle 3 product_category.....	3
Tabelle 4 ProductCategoryController.....	4
Tabelle 5 PersonKontroller .....	4
Tabelle 6 ProductController.....	5
Tabelle 7 EndPoints.....	6
Tabelle 8 Arbeitsjournal - Arbeitstag 1.....	8
Tabelle 9 Arbeitsjournal - Arbeitstag 2.....	9
Tabelle 10 Arbeitsjournal - Arbeitstag 3.....	10
Tabelle 11 Arbeitsjournal - Arbeitstag 4.....	10
Tabelle 12 Arbeitsjournal - Arbeitstag 5.....	11

## 9. Abbildungsverzeichnis

Abbildung 1 Kontrollieren .....	6
---------------------------------	---