



# HSB

Hochschule Bremen  
City University of Applied Sciences

Dokumentation

---

MICROSERVICE WARENWIRTSCHAFT

---

Hochschule Bremen – Fakultät für Elektrotechnik und Informatik  
Studiengang Komplexe Softwaresysteme (M.Sc.)

Eingereicht von  
MERLE LABUSCH und MARTIN MÜLLER

Dozenten  
MATTHIAS STOCK, RENZO KOTTMANN

Bremen, 30.06.2017

# 1 Microservice-Steckbrief

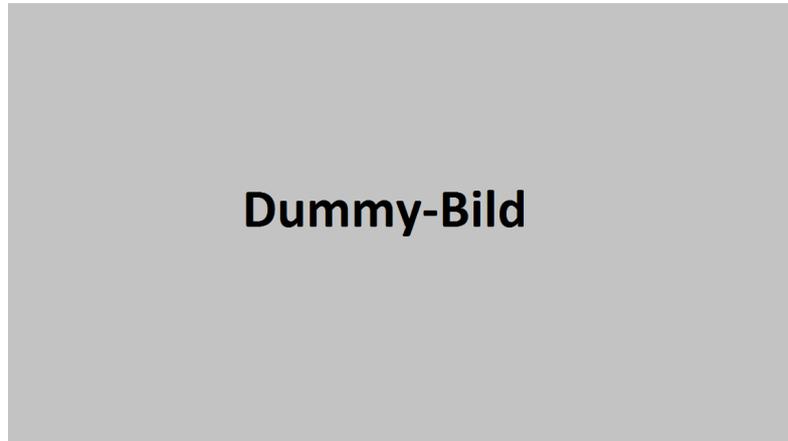


Abbildung 1: Microservice Warenwirtschaft

- **ToDo: kurze Beschreibung**

# Inhaltsverzeichnis

<b>1</b>	<b>Microservice-Steckbrief</b>	<b>I</b>
<b>2</b>	<b>Dokumentationsstruktur</b>	<b>1</b>
<b>3</b>	<b>Definition der Anforderungen</b>	<b>2</b>
<b>4</b>	<b>Struktur des Microservice</b>	<b>3</b>
4.1	Presentation Layer . . . . .	3
4.2	Application Layer . . . . .	3
4.3	Persistent Layer . . . . .	3
4.4	Integrierte Tests . . . . .	3
4.5	Admin-Frontend . . . . .	3
4.6	Anpassung des Monolithen . . . . .	3
<b>5</b>	<b>Backbox-Testfälle</b>	<b>4</b>
	<b>Anhang</b>	<b>5</b>
<b>A</b>	<b>Handout zum Admin-Frontend</b>	<b>5</b>
<b>B</b>	<b>Testprotokoll vom XX.06.2017</b>	<b>6</b>

## 2 Dokumentationsstruktur

Für die Dokumentation des Microservice Warenwirtschaft wurden eine Kombination aus zwei Dokumenten gewählt. Zum einen beschreibt ein Handout auf einer Seite die Funktionen des Admin-Frontends für den Benutzer. Diese sehr kurze Dokumentenform wurde gewählt, da Benutzer häufig nicht gewillt sind, umfangreiche Anleitungen zu lesen um eine Anwendung nutzen zu können. Vielmehr wollen sie schnell einen Überblick der Kernfunktionalitäten erhalten. Zu diesem Zweck arbeitet das Handout mit einem aufbereiteten Screenshot des Admin-Frontends und einer Beschreibung der Funktionen in Stichpunkten.

Auf der anderen Seite muss der Microservice auch für Entwickler dokumentiert sein, hierfür wurde diese Dokumentation angelegt. Sie beginnt anstelle eines Abstract mit einem Steckbrief des Microservice, der dessen grundlegende Struktur und Funktionalität kurz beschreibt. In dem eigentlichen Dokument werden dann zunächst die Anforderungen an den Microservice beschrieben, da ihre Umsetzung das primäre Ziel der Entwicklung ist. Weiter werden der Microservice und seine Code-Struktur detaillierter beschrieben sowie die gewählten Blackbox-Testfälle dokumentiert. Der Anhang dieser Dokumentation umfasst das Handout des Admin-Frontends sowie ein beispielhaftes Testprotokoll. Auf Details wie ein Abkürzungs- oder Literaturverzeichnis wurde in dieser Dokumentation bewusst verzichtet, um sie kurz zu halten. Zitate und Verweise werden hier in Form von Fußnoten integriert.

### 3 Definition der Anforderungen

Der Microservice Warenwirtschaft dient der Verwaltung der einzelnen Waren pro Produkt für den Webshop Mosh, sodass zum Beispiel neue Waren erfasst und keine Waren verkauft werden können, die sich nicht mehr im Lagerbestand befinden. Dieser Microservice basiert auf den nachfolgend aufgeführten Anforderungen. Mit dem Begriff Produkt werden im Folgenden die über Mosh angebotenen Fruchtarten, zum Beispiel Kiwis bezeichnet, während mit dem Begriff Ware die einzelnen Positionen pro Fruchtart im Lagerbestand bezeichnet werden.

- Speicherung der Waren pro Produkt mit einem Zeitstempel, wann sie erfasst wurden sowie ihrem Lagerort
- Admin-Frontend
  - Hinzufügen neuer Waren zum Bestand
  - Manuelles Entfernen von Waren aus dem Bestand, zum Beispiel wenn der Bestand verdorben ist
  - Entfernen von Waren aus dem Bestand, wenn diese an einen Kunden versendet werden
  - Blockieren von Waren in dem Bestand, wenn ein Kunde sie in seinen Warenkorb gelegt hat
  - Automatische Freigaben von blockierten Waren, wenn diese nach 30 Minuten nicht an den Versand überstellt wurden
- Kunden-Frontend
  - Anzeige der Produktverfügbarkeit über ein Ampelsystem, mehr als zehn Waren entsprechen grün, fünf bis zehn Waren entsprechen gelb und weniger als fünf Waren entsprechen rot
- Optionale Zusatz-Funktionen
  - Ausgabe einer Statistik, wie viele Waren im letzten Monat aus dem Bestand versandt und wie viele manuell entfernt wurden im Admin-Fontend
  - Ampeldarstellung pro Waren, die Anzeigt ab die Ware bereits überaltert ist, im Admin-Frontend

## 4 Struktur des Microservice

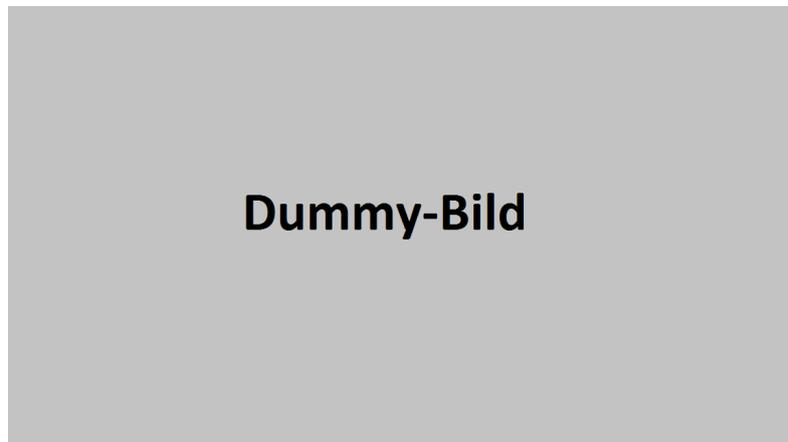


Abbildung 2: Struktur des Microservice

- Der Microservice wurde in der Programmiersprache Go<sup>1</sup> entwickelt.
- **ToDo: kurze Beschreibung der Programmstruktur und genauere Beschreibung der Einzelteile in Unterkapiteln**

### 4.1 Presentation Layer

### 4.2 Application Layer

### 4.3 Persistant Layer

### 4.4 Integrierte Tests

### 4.5 Admin-Frontend

### 4.6 Anpassung des Monolithen

---

<sup>1</sup><https://golang.org/doc>

## 5 Backbox-Testfälle

Die in den Microservice integrierten Tests, prüfen ob jedes Stück Code auch wirklich ausgeführt wird (Code Coverage) und die Anforderungen funktionell erfüllt werden. Da diese Tests mit der Kenntnis des Codes beschrieben wurden, nennt man sie auch Whitebox-Tests. Neben der Sicht des Entwicklers sollte beim Test aber auch die Sicht des Benutzers nicht außer Acht gelassen und ein sogenannter Blackbox-Test, ohne Kenntnis des Codes durchgeführt werden. Dies ermöglicht die Prüfung, ob die Anforderungen auch aus der Sicht des Benutzers, der weder den Code noch die internen Abläufe innerhalb des Microservice kennt, erfüllt werden. Zu diesem Zweck wurden die folgenden Testfälle anhand der Anforderungen erstellt. Sie definieren zunächst das Vorgehen bei der Testdurchführung und anschließend den erwarteten Soll- sowie den eingetretenen Ist-Zustand. Das Protokoll eines durchgeführten Blackbox-Tests findet sich unter Anhang B.

Tabelle 1: Blackbox-Testfälle

Vorgehen	Soll	Ist
Aufruf der URL xxx	Anzeige des Login-Feldes für das Admin-Frontend	
Anmeldung mit dem Benutzernamen X und dem Passwort Y	Erfolgreicher Login, Anzeige des Warenwirtschaft-Frontend	
-	Das Frontend zeigt eine Liste der Waren pro Produkt mit ihrem Zeitstempel	
-	Es wird angezeigt, dass keine Kirschen mehr vorhanden sind	
Hinzufügen von 5 Kiwis über den Button +	Es werden 5 Kiwis mit der aktuellen Zeit zu dem Bestand hinzugefügt	
Entfernen von 2 Tomaten über den Button -	Es werden die angegebenen 2 Tomaten aus dem Bestand entfernt	
Aufruf der URL xxx	Anzeige des Kunden-Frontend für Kirschen, Anzeige der Verfügbarkeit mit einer roten Ampel	
Aufruf der URL xxx	Anzeige des Kunden-Frontend für Tomaten, Anzeige der Verfügbarkeit mit einer gelben Ampel	
Aufruf der URL xxx	Anzeige des Kunden-Frontend für Kiwis, Anzeige der Verfügbarkeit mit einer grünen Ampel	
Bestellen von 2 Kiwis	Im Admin-Frontend werden 2 Kiwis als blockiert angezeigt	
Transportbestätigung für die 2 Kiwis	Im Admin-Frontend werden die 2 Kiwis automatisch aus dem Bestand gelöscht	

## A Handout zum Admin-Frontend

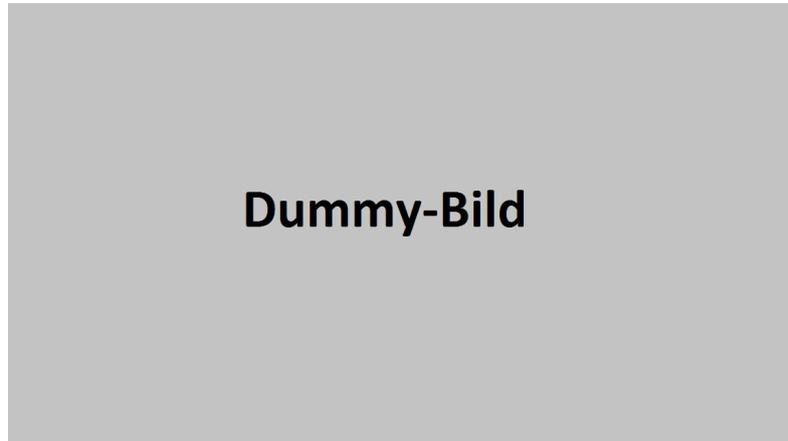


Abbildung 3: Handout zum Admin-Frontend

## B Testprotokoll vom XX.06.2017

Tabelle 2: Testprotokoll vom XX.06.2017

Vorgehen	Soll	Ist
Aufruf der URL xxx	Anzeige des Login-Feldes für das Admin-Frontend	
Anmeldung mit dem Benutzernamen X und dem Passwort Y	Erfolgreicher Login, Anzeige des Warenwirtschaft-Frontend	
-	Das Frontend zeigt eine Liste der Waren pro Produkt mit ihrem Zeitstempel	
-	Es wird angezeigt, dass keine Kirschen mehr vorhanden sind	
Hinzufügen von 5 Kiwis über den Button +	Es werden 5 Kiwis mit der aktuellen Zeit zu dem Bestand hinzugefügt	
Entfernen von 2 Tomaten über den Button -	Es werden die angegebenen 2 Tomaten aus dem Bestand entfernt	
Aufruf der URL xxx	Anzeige des Kunden-Frontend für Kirschen, Anzeige der Verfügbarkeit mit einer roten Ampel	
Aufruf der URL xxx	Anzeige des Kunden-Frontend für Tomaten, Anzeige der Verfügbarkeit mit einer gelben Ampel	
Aufruf der URL xxx	Anzeige des Kunden-Frontend für Kiwis, Anzeige der Verfügbarkeit mit einer grünen Ampel	
Bestellen von 2 Kiwis	Im Admin-Frontend werden 2 Kiwis als blockiert angezeigt	
Transportbestätigung für die 2 Kiwis	Im Admin-Frontend werden die 2 Kiwis automatisch aus dem Bestand gelöscht	