

Titel: Protokoll GK711  
Autor: Sebastian Pollak  
Datum: 2023-09-19

# Protokoll GK771

---

Es wurde in der WarehouseData Klasse die einige Eigenschaften für das Warehouse Hinzugefügt. Ebenso wurde ein Attribut productData welches ein Objekt Array ist. Das Objekt für diesen Array enthält die Produktdaten.

## WarehouseData Klasse:

```
private String warehouseID;  
private String warehouseName;  
private String warehouseAddress;  
private String warehousePostalCode;  
private String warehouseCity;  
private String warehouseCountry;  
private String timestamp;  
private ProductData[] productData;
```

## ProductData Klasse:

```
private String productID;  
private String productName;  
private String productCategory;  
private String productQuantity;  
private String productUnit;
```

Als nächstes wurde im WarehouseSimulator einen Multidimensionalen Array von ChatGPT generieren lassen. In diesem sind 15 Zufällige Produkte mit Einheit und Kategorie.

```
String[][] produkte = {  
    {"Apfel", "Obst", "Stück"},  
    {"Banane", "Obst", "Stück"},  
    {"Karotte", "Gemüse", "Stück"},  
    {"Tomate", "Gemüse", "Stück"},  
    {"Huhn", "Fleisch", "Gramm"},  
    {"Lachs", "Fisch", "Gramm"},  
    {"Milch", "Milchprodukte", "Liter"},  
    {"Käse", "Milchprodukte", "Gramm"},  
    {"Brot", "Backwaren", "Stück"},  
    {"Reis", "Getreideprodukte", "Gramm"},  
    {"Ei", "Eier", "Stück"},  
    {"Joghurt", "Milchprodukte", "Gramm"},  
}
```

```

        {"Zucker", "Backwaren", "Gramm"},
        {"Kartoffel", "Gemüse", "Stück"},
        {"Erdbeere", "Obst", "Stück"}
    };

```

Folgend habe ich für einen zuvor erstellten Produkt Array eine For-Schleife geschrieben. In dieser Vorschleife habe ich eine zufällige Zahl generieren lassen welche ein Zufälliges Produkt aus dem Produkte Array auswählt. Jetzt wird der Name, die Kategorie und die Einheit zum Produkt hinzugefügt. Als Nächstes hab ich eine Zufällige Menge für das Produkt generieren lassen. Zuletzt wird der Array zu den Produkten hinzugefügt.

```

ProductData[] products = new ProductData[4];

for(int i = 0; i < products.length; i++){
    ProductData product = new ProductData();
    product.setProductID(String.valueOf(i));

    Random rand = new Random();
    int index = rand.nextInt(produkte.length);

    String[] randomProduct = produkte[index];

    product.setProductName(randomProduct[0]);
    product.setProductCategory(randomProduct[1]);
    product.setProductUnit(randomProduct[2]);

    int randomQuantity = rand.nextInt(501);
    product.setProductQuantity(String.valueOf(randomQuantity));

    products[i] = product;
}

```

## Request Mapping

Um die Objekte als XML und JSON auf den Spring Server zu bekommen muss man folgenden Code verwenden:

```

@RequestMapping(value="/warehouse/{inID}/data", produces =
MediaType.APPLICATION_JSON_VALUE)
public WarehouseData warehouseData( @PathVariable String inID ) {
    return service.getWarehouseData( inID );
}

@RequestMapping(value="/warehouse/{inID}/xml", produces =
MediaType.APPLICATION_XML_VALUE)
public WarehouseData warehouseDataXML( @PathVariable String inID ) {

```

```
        return service.getWarehouseData( inID );  
    }
```

Als erstes wird mit @RequestMapping gesagt unter welchem Pfad die WarehouseData wie angezeigt werden soll. Hier ist auch ein {inID} im Pfad welches ein Platzhalter für die ID ist.