

## WAREHOUSE-REST

David Socaciu 4DHIT 16.09.2025

---

### Grundlagen

Bevor ich angefangen habe, habe ich zuerst das Projekt auf Github geforked und anschließend in IntelliJ geklont. Anschließend habe ich das build.gradle ausgeführt und angefangen die Schritte zu erledigen.

### Implementierung eines Generator zur Erzeugung der Standort- und Warendaten

Als erstes habe ich die Klasse ProductData erzeugt die Attribute, Konstruktor sowie die Setter- und Gettermethoden implementiert, folgende Attribute habe ich verwendet:

```
private String productID;  
private String productName;  
private String productCategory;  
private int productQuantity;  
private String productUnity;
```

Anschließend habe ich die Methode getData aus der Klasse WarehouseSimulation bearbeitet um eine Generierung der Produkte zu ermöglichen.

Ich habe in einer Arrayliste 6 Produkte hinzugefügt und anschließend habe ich mithilfe der nextBoolean() Methode folgendes implementiert:

```
while (actualProducts.size() < 2) {  
    for(ProductData productData : productDataList) {  
        if (random.nextBoolean()) {  
            actualProducts.add(productData);  
        }  
    }  
}
```

Ich habe eine zweite Arrayliste erstellt, in welche mindestens 2 der 6 Produkte hinzugefügt werden mithilfe der while Bedingung. Die nextBoolean() Methode hat mir geholfen um mir zufällig die Anzahl der Produkte auszuwählen.

### Daten im JSON und XML Format zur Verfügung stellen

Die Daten waren im JSON bereits abrufbar unter: <http://localhost:8080/warehouse/001/json>

Um die Daten im XML-Format anzuzeigen zu lassen, fehlt mir bislang eine Dependency die ich in die build.gradle einfügen musste

```
implementation 'com.fasterxml.jackson.dataformat:jackson-dataformat-xml'
```

und anschließend die build.gradle neu ausgeführt, sowie eine neue Methode im Controller die fast identisch mit der Methode für die JSON ist.

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

Der Unterschied war das man beim Value /xml hinschreibt und bei MediaType XML statt JSON.

### Implementierung der REST Schnittstelle

Nun kann man unter localhost:8080 folgendes sehen:

---

**This is the warehouse application! (DEZSYS\_WAREHOUSE\_REST)**

[Link to JSON warehouse/001/json](#)

[Link to XML warehouse/001/xml](#)

Hier sind 2 Beispiele, eines vom XML und eines vom JSON Dateiformat:

```
{
  "warehouseID": "001",
  "warehouseName": "Wien Bahnhof",
  "warehouseAddress": "Am Hauptbahnhof 1",
  "warehousePostalCode": 1100,
  "warehouseCity": "Wien",
  "warehouseCountry": "Austria",
  "timestamp": "2025-09-16 16:33:45.364",
  "products": [
    {
      "productID": "00-871895",
      "productName": "Bio Apfelsaft Gold",
      "productCategory": "Getraenk",
      "productQuantity": 293,
      "productUnity": "Packung 1L"
    },
    {
      "productID": "00-316253",
      "productName": "Persil Discs Color",
      "productCategory": "Waschmittel",
      "productQuantity": 383,
      "productUnity": "Packung 3KG"
    },
    {
      "productID": "00-315252",
      "productName": "Milka Alpenmilch Schokolade",
      "productCategory": "Süßwaren",
      "productQuantity": 941,
      "productUnity": "Tafel 100g"
    }
  ]
}
```

```
▼ <WarehouseData>
  <warehouseID>001</warehouseID>
  <warehouseName>Wien Bahnhof</warehouseName>
  <warehouseAddress>Am Hauptbahnhof 1</warehouseAddress>
  <warehousePostalCode>1100</warehousePostalCode>
  <warehouseCity>Wien</warehouseCity>
  <warehouseCountry>Austria</warehouseCountry>
  <timestamp>2025-09-16 16:33:48.434</timestamp>
  ▼ <products>
    ▼ <products>
      <productID>00-315252</productID>
      <productName>Milka Alpenmilch Schokolade</productName>
      <productCategory>Süßwaren</productCategory>
      <productQuantity>62</productQuantity>
      <productUnity>Tafel 100g</productUnity>
    </products>
    ▼ <products>
      <productID>00-316253</productID>
      <productName>Persil Discs Color</productName>
      <productCategory>Waschmittel</productCategory>
      <productQuantity>157</productQuantity>
      <productUnity>Packung 3KG</productUnity>
    </products>
    ▼ <products>
      <productID>01-526733</productID>
      <productName>Milka Nussini Riegel</productName>
      <productCategory>Süßwaren</productCategory>
      <productQuantity>960</productQuantity>
      <productUnity>Tafel 100g</productUnity>
    </products>
  </products>
</WarehouseData>
```

---

## Quellen

[XML Serialization and Deserialization with Jackson | Baeldung](#)