

MidEng 7.1 Warehouse Rest & Dataformats

Yildirim Aran, 4DHIT

WarehouseData

```
3 usages
private String warehouseID;
2 usages
private String warehouseName;
4 usages
private String timestamp;
2 usages
private String wareHouseCity;
2 usages
private String address;
2 usages
private String country;
2 usages
private ProductData[] products;
```

Ich habe damit begonnen die nötigen Attribute hinzuzufügen, darunter auch ein Objekt-Array der Klasse ProductData. Die Attribute in dieser Klasse habe ich davor schon hinzugefügt, welche Id, Name, Kategorie und Anzahl sind. Daraufhin habe ich für alle Attribute Getter- und Setter-Methoden erstellt.

ProductData

```

10     1 usage
    public String getProductName() {
11         return productName;
12     }
    1 usage
13     public void setProductName(String productName) {
14         this.productName = productName;
15     }
    no usages
16     public String getProductId() {
17         return this.productId;
18     }
    1 usage
19     public void setProductId(String productId) {
20         this.productId = productId;
21     }
    no usages
22     public String getProductAmount() {
23         return this.productAmount;
24     }
    1 usage
25     public void setProductAmount(String amount) {
26         this.productAmount = amount;
27     }
    no usages
28     public String getProductCategory() {
29         return this.productCategory;
30     }

```

Hier habe ich die vorhin genannten Attribute erstellt und für diese Getter- und Setter-Methoden erstellt.

WarehouseSimulation

```

warehouseData.setWarehouseName("Bipa");
warehouseData.setCountry("Austria");
warehouseData.setWareHouseCity("Wien");
warehouseData.setAddress("Rennbahnweg 50");
warehouseData.setCountry("Österreich");

```

Zuerst habe ich damit begonnen die neu erstellten Attribute zu setzen über die Setter-Methoden der Klasse WarehouseData. Dadurch wurden jetzt Straße, Stadt und Land mit den von mir hard gecodeten Werten zu der XML- und JSON-Ansicht hinzugefügt.

```

ProductData[] productData = new ProductData[4];
productData[0] = generateRandomProductData();
productData[1] = generateRandomProductData();
productData[2] = generateRandomProductData();
productData[3] = generateRandomProductData();

```

Da die Anforderung mindestens 4 “products” waren, habe ich 4 Product-Objekte erstellt.

```

usage
public static String generateRandomProductName() {
    // You can replace this with a list of actual product names
    String[] productNames = {"Orange", "Zitrone", "Mandarine", "Wassermelone", "Zuckermelone", "Honigmelone"};
    Random random = new Random();
    return productNames[random.nextInt(productNames.length)];
}

usage
public static String assignProductCategory(String productName) {
    // Assign the category based on the product name
    if (productName.equals("Orange") || productName.equals("Zitrone") || productName.equals("Mandarine")) {
        return "Zitrusfrucht";
    } else if (productName.equals("Wassermelone") || productName.equals("Zuckermelone") || productName.equals("Honigmelone")) {
        return "Melone";
    } else {
        return "Unknown";
    }
}

```

Mit diesen zwei Methoden gebe ich den Produkten einen Namen, und basierend auf diesem Namen eine Kategorie. Wenn der Name des Produktes Orange, Zitrone oder Mandarine ist, dann kriegt es als Kategorie “Zitrusfrucht”.

4 usages

```
public static ProductData generateRandomProductData() {  
    ProductData productData = new ProductData();  
    Random random = new Random();  
  
    productData.setProductId(String.format("IN%03d", random.nextInt( bound: 100000));  
    productData.setProductName(generateRandomProductName());  
    productData.setProductCategory(assignProductCategory(productData.getProductName()));  
    productData.setProductAmount(Integer.toString(random.nextInt( bound: 100)));  
  
    return productData;  
}
```

Hier gebe ich dem Produkt noch seine ID, Namen, Kategorie und seine Anzahl.

```

▼<WarehouseData>
  <warehouseID>593</warehouseID>
  <warehouseName>Bipa</warehouseName>
  <timestamp>2023-09-20 13:46:16.068</timestamp>
  <wareHouseCity>wareHouseCity</wareHouseCity>
  <address>Rennbahnweg 50</address>
  <country>Österreich</country>
  ▼<products>
    ▼<products>
      <productName>Honigmelone</productName>
      <productId>IN197026</productId>
      <productAmount>87</productAmount>
      <productCategory>Melone</productCategory>
    </products>
    ▼<products>
      <productName>Orange</productName>
      <productId>IN260551</productId>
      <productAmount>59</productAmount>
      <productCategory>Zitrusfrucht</productCategory>
    </products>
    ▼<products>
      <productName>Mandarine</productName>
      <productId>IN364146</productId>
      <productAmount>50</productAmount>
      <productCategory>Zitrusfrucht</productCategory>
    </products>
    ▼<products>
      <productName>Orange</productName>
      <productId>IN510981</productId>
      <productAmount>49</productAmount>
      <productCategory>Zitrusfrucht</productCategory>
    </products>
  </products>
</WarehouseData>

```

Das ganze sieht dann so fertig aus in der XML.