

Warehouse Doku

Code

Wenn die Website aufgerufen wird kommt man auf die Main page wo man zu den Verschiedenen Subpages weitergeleitet werden kann. Jenachdem ob die data oder xml Seite aufgerufen wird, wird der Output als JSON oder XML angezeigt.

```
1 @RequestMapping("/")
2 public String warehouseMain() {
3     String mainPage = "This is the warehouse application! (DEZSYS_WAREHOUSE_REST) <br/><br/>" +
4         "<a href='http://localhost:8080/warehouse/001/data'>Link to warehouse/001/data</a><br/>" +
5         "<a href='http://localhost:8080/warehouse/001/xml'>Link to warehouse/001/xml</a><br/>" +
6         "<a href='http://localhost:8080/warehouse/001/transfer'>Link to warehouse/001/transfer</a><br/>";
7     return mainPage;
8 }
9
10 @RequestMapping(value="/warehouse/{inID}/data", produces = MediaType.APPLICATION_JSON_VALUE)
11 public WarehouseData warehouseData( @PathVariable String inID ) {
12     return service.getWarehouseData( inID );
13 }
14
15 @RequestMapping(value="/warehouse/{inID}/xml", produces = MediaType.APPLICATION_XML_VALUE)
16 public WarehouseData warehouseDataXML( @PathVariable String inID ) {
17     return service.getWarehouseData( inID );
18 }
19
20 @RequestMapping("/warehouse/{inID}/transfer")
21 public String warehouseTransfer( @PathVariable String inID ) {
22     return service.getGreetings("Warehouse.Transfer!");
23 }
```

Es gibt eine Liste mit Standorten und es wird ein zufälliger Standort ausgewählt. Dann werden von der WarehouseDATA die verschiedenen Attribute gesetzt.

```

1  public WarehouseData getData( String inID ) {
2
3      String[][] cities = {
4          {"Wien", "1010", "Jägerstraße 1", "Wien Hauptlager", "Österreich"},
5          {"Salzburg", "5020", "Hauptstraße 5", "Salzbug Lager", "Österreich"},
6          {"Graz", "8010", "Weinbergstraße", "Graz Lager", "Österreich"}
7      };
8
9      int r = new Random().nextInt(2);
10
11      WarehouseData data = new WarehouseData();
12      data.setWarehouseID(inID);
13      data.setWarehouseName(cities[r][3]);
14      data.setStreet(cities[r][2]);
15      data.setPlz(cities[r][1]);
16      data.setCity(cities[r][0]);
17      data.setCountry(cities[r][4]);
18
19      return data;
20
21  }

```

Die WarehouseData hat eine zufällige Anzahl zwischen 4 und 10 Produkten, welche in einer ArrayList gespeichert werden. Es wird auch der Aktuelle Timestamp hinzugefügt.

```

1  private ArrayList<Product> productData;
2
3      /**
4       * Constructor
5       */
6      public WarehouseData() {
7
8          int r = new Random().nextInt(6) + 4;
9
10         productData = new ArrayList<Product>(r);
11
12         for(int i = 0; i < r; i++) {
13             productData.add(new Product());
14         }
15
16         this.timestamp = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss.SSS").format(new Date());
17
18     }

```

Ein Produkt wird erstellt indem eine Zufällige ID erstellt wird und ein Zufälliger Array aus dem Array der möglichen Produkt Daten ausgewählt wird.

```
1  private String id;
2  private String name;
3  private String category;
4  private String amount;
5  private String unit;
6
7  public Product() {
8      String[][] products = {
9          {"Bio Orangensaft", "Getränk", "1L Pakung"},
10         {"Bio Apfelsaft", "Getränk", "1L Pakung"},
11         {"Ariel Waschmittel Color", "Waschmittel", "Packung 3Kg"},
12         {"Persil Discs Color", "Waschmittel", "Packung 700g"}
13     };
14
15     String r = String.valueOf(new Random().nextInt(1000000, 9999999));
16     this.id = r.substring(0, 2) + "-" + r.substring(2, r.length() - 1);
17
18     int rInt = new Random().nextInt(3);
19     this.name = products[rInt][0];
20     this.category = products[rInt][1];
21     this.amount = String.valueOf(new Random().nextInt(5000));
22     this.unit = products[rInt][2];
23 }
```

Probleme

- Anzeige bei XML war:

```
...
<productData>
  <product>
    <product>
      <id>..
```

```
</product>  
</productData>
```

- Lösung:
 - Keine ProductData Klasse verwenden sondern einen Array an Products in der WarehouseData Klasse verwenden