

# MySQLReceptKomponentDAO.java

```

1 package webapplication.datalayer;
2
3 import java.sql.ResultSet;
13
14 /**
15  * @author Theis
16  *
17  */
18 public class MySQLReceptKomponentDAO implements ReceptKompDAO {
19     SQLMapper map = new SQLMapper();
20
21     public MySQLReceptKomponentDAO(){
22         try { new Connector(); }
23         catch (InstantiationException e) { e.printStackTrace(); }
24         catch (IllegalAccessException e) { e.printStackTrace(); }
25         catch (ClassNotFoundException e) { e.printStackTrace(); }
26         catch (SQLException e) { e.printStackTrace(); }
27     }
28
29     @Override
30     public ReceptKompDTO getReceptKomp(int receptId, int raavareId) throws DAException {
31         String statement = map.getStatement("rec_komponent_SELECT");
32         String[] values = new String[]{Integer.toString(receptId),
Integer.toString(raavareId)};
33         statement = map.insertValuesIntoString(statement, values);
34         System.out.println("Query: "+statement);
35         ResultSet rs = Connector.doQuery(statement);
36
37         try {
38             if (!rs.first()) throw new DAException("Receptkomponenten for recept " +
receptId + " og raavare " + raavareId + " findes ikke");
39             return new ReceptKompDTO (rs.getInt("recept_id"), rs.getInt("raavare_id"),
rs.getDouble("nom_netto"), rs.getDouble("tolerance"));
40         }
41         catch (SQLException e) {throw new DAException(e); }
42     }
43
44     @Override
45     public List<ReceptKompDTO> getReceptKompList(int receptId) throws DAException {
46         List<ReceptKompDTO> list = new ArrayList<ReceptKompDTO>();
47         String statement = map.getStatement("rec_komponent_SELECT_ALL_rec_id");
48         statement = map.insertValuesIntoString(statement, new
String[]{Integer.toString(receptId)});
49         ResultSet rs = Connector.doQuery(statement);
50         try
51         {
52             while (rs.next())
53             {
54                 list.add(new ReceptKompDTO(rs.getInt("recept_id"), rs.getInt("raavare_id"),
rs.getDouble("nom_netto"), rs.getDouble("tolerance")));
55             }
56         }
57         catch (SQLException e) { throw new DAException(e); }
58         return list;
59     }
60
61     @Override
62     public List<ReceptKompDTO> getReceptKompList() throws DAException {
63         List<ReceptKompDTO> list = new ArrayList<ReceptKompDTO>();
64         ResultSet rs = Connector.doQuery(map.getStatement("rec_komponent_SELECT_ALL"));
65         try
66         {

```

```

67         while (rs.next())
68         {
69             list.add(new ReceptKompDTO(rs.getInt("recept_id"), rs.getInt("raavare_id"),
rs.getDouble("nom_netto"), rs.getDouble("tolerance")));
70         }
71     }
72     catch (SQLException e) { throw new DAException(e); }
73     return list;
74 }
75
76 @Override
77 public void createReceptKomp(ReceptKompDTO receptkomponent) throws DAException {
78     String statement = map.getStatement("rec_komponent_INSERT");
79     String[] values = new String[]{
80         Integer.toString(receptkomponent.getReceptId()),
81         Integer.toString(receptkomponent.getRaavareId()),
82         Double.toString(receptkomponent.getNomNetto()),
83         Double.toString(receptkomponent.getTolerance())
84     };
85     statement = map.insertValuesIntoString(statement, values);
86     System.out.println(statement);
87     Connector.doUpdate(statement);
88
89 }
90
91 @Override
92 public void updateReceptKomp(ReceptKompDTO receptkomponent) throws DAException {
93     String statement = map.getStatement("rec_komponent_UPDATE");
94     String[] values = new String[]{
95         Double.toString(receptkomponent.getNomNetto()),
96         Double.toString(receptkomponent.getTolerance()),
97         Integer.toString(receptkomponent.getReceptId()),
98         Integer.toString(receptkomponent.getRaavareId())
99     };
100     statement = map.insertValuesIntoString(statement, values);
101     System.out.println(statement);
102     Connector.doUpdate(statement);
103
104 }
105
106 }
107

```