

SimpleHibernateTableModel

NOTE: This is outdated information that applies only to Tapestry 4.

This is a simple implementation of a [BasicTableModel](#) for the contrib:Table to list Hibernate entities.

```
package de.hsofttec.core4.component.table;

import java.util.Iterator;
import java.util.regex.Pattern;

import org.apache.commons.lang.StringUtils;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.apache.hivemind.ApplicationRuntimeException;
import org.apache.tapestry.contrib.table.model.IBasicTableModel;
import org.apache.tapestry.contrib.table.model.ITableColumn;

import org.hibernate.Query;
import org.hibernate.Session;

/**
 * a simple tablemodel with hibernate integration
 */
public class Core4TableModel implements IBasicTableModel
{
    private static Log _logger = LogFactory.getLog(Core4TableModel.class);
    private Session _hbmSession;
    private String _sqlString;
    private int _rowCount = -1;

    /**
     * Constructor.
     */
    public Core4TableModel(Session hbmSession, String sqlString)
    {
        _hbmSession = hbmSession;
        _sqlString = sqlString;
    }

    /**
     * how many table rows should be display.
     */
    public int getRowCount()
    {
        if (_rowCount < 0)
            _rowCount = getMaximumResultObjects();

        return _rowCount;
    }

    /**
     * get the table rows that should display.
     */
    public Iterator getCurrentPageRows(int nFirst, int nPageSize, ITableColumn iTableColumn, boolean sortable)
    {
        try
        {
            Query query = _hbmSession.createQuery(_sqlString);
            query.setMaxResults(nPageSize);
            query.setFirstResult(nFirst);

            if (_logger.isDebugEnabled())
                _logger.debug(query.getQueryString());

            return query.list().iterator();
        }
        catch (Exception e)
        {
        }
    }
}
```

```

        {
            throw new ApplicationRuntimeException(e);
        }
    }

/**
 * get the maximum count of rows to display.
 */
private int getMaximumResultObjects()
{
    try
    {
        String tempQuery1 = convertQueryString(_sqlString);
        Query query = _hbmSession.createQuery(tempQuery1);
        return (Integer) query.uniqueResult();
    }
    catch (Exception e)
    {
        throw new ApplicationRuntimeException(e);
    }
}

/**
 * den SQL-Query so Aufbauen, das wir einen Count auf die Tabelle absetzen koennen.
 */
private String convertQueryString(String originalQueryString)
{
    String tempQueryString = StringUtils.substring(originalQueryString, StringUtils.indexOf(StringUtils.
upperCase(originalQueryString), "FROM"));

    String convertedQueryString = "SELECT COUNT(*) ";

    // Split input with the pattern
    Pattern p = Pattern.compile("[\\s]+");
    String[] result = p.split(tempQueryString);

    for (String queryWord : result)
    {
        //
        // ist queryWord ein Query-Fragment, was in einem Count-Query nicht auftauchen darf ?
        //
        // if (queryWord.equalsIgnoreCase("LEFT") ||
        // queryWord.equalsIgnoreCase("JOIN") ||
        // queryWord.equalsIgnoreCase("FETCH"))

        if (queryWord.equalsIgnoreCase("FETCH"))
            continue;

        if (queryWord.equalsIgnoreCase("ORDER"))
            break;

        convertedQueryString += queryWord + " ";
    }

    if (_logger.isInfoEnabled())
        _logger.info("source: " + originalQueryString + System.getProperty("line.separator") +
            "dest: " + convertedQueryString);

    return convertedQueryString;
}
}

```