

```
package fi.paatti.paattidatabaseutils.dbsevice.querydelegates;

import com.vaadin.data.util.sqlcontainer.RowItem;
import com.vaadin.data.util.sqlcontainer.query.generator.StatementHelper;
import fi.paatti.paattidatabaseutils.names.PaattiColumnNames;
import fi.paatti.paattidatabaseutils.names.PaattiTableNames;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

/**
 * Get all the events (with event times that) are available to the user.
 * Schedule and usergroup are also joined to the list.
 * <p/>
 * Available events are those that have no link to task data. Read only access
 * to data.
 * <p/>
 * @author Lauri Satokangas, lauri.n.satokangas@student.jyu.fi
 * @since 0.0.4
 */
public class UserUnfinishedEventsDelegate extends PaattiQueryDelegate {

    private static final long serialVersionUID = 1L;
    private final Object userID;

    /**
     * Constructor.
     *
     * @param userID The ID of the user.
     */
    public UserUnfinishedEventsDelegate(Object userID) {
        super(PaattiTableNames.EVENT, PaattiColumnNames.EVENT_eventID,
            PaattiColumnNames.getColumnNamesInArray(PaattiTableNames.EVENT));
        this.userID = userID;
    }

    /**
     * {@inheritDoc }
     * <p/>
     * Not in use.
     */
    @Override
    public int storeRow(Connection conn, RowItem row) throws SQLException {
        throw new UnsupportedOperationException("Not supported yet.");
    }
}
```

```

}

/**
 * {@inheritDoc }
 * <p/>
 * Not in use.
 */
@Override
protected void setRowValues(PreparedStatement statement, RowItem row) throws SQLException {
    throw new UnsupportedOperationException("Not supported yet.");
}

/**
 * Returns the SELECT-query. This can be found from the project --> Other Sources sqlqueries-package.
 * <p/>
 * @param offset not in use
 * @param limit not in use
 * @return The StatementHelper with the select-query
 */
public StatementHelper getQueryStatement(int offset, int limit) {
    StatementHelper queryHelper = new StatementHelper();
    StringBuilder querySb = new StringBuilder();

    querySb.append("SELECT * FROM " + PaattiTableNames.EVENT);
    querySb.append(
        " LEFT OUTER JOIN (SELECT SCHEDULE_scheduleID, SCHEDULE_description, USERGROUP_usergroupId, USERGROUP_description");
    querySb.append(" FROM ").append(PaattiTableNames.SCHEDULE).append(" LEFT OUTER JOIN (SELECT ").append(
        PaattiColumnNames.USERGROUP_usergroupId + ", " + PaattiColumnNames.USERGROUP_description);
    querySb.append(" FROM ").append(PaattiTableNames.BELONGS).append(" LEFT OUTER JOIN ");
    querySb.append(
        PaattiTableNames.USERGROUP + " ON " + PaattiColumnNames.BELONGS_USERGROUP_usergroupId + " = "
        + PaattiColumnNames.USERGROUP_usergroupId);
    querySb.append(" WHERE " + PaattiColumnNames.BELONGS_USER_userID + " = ").append(userID).append(" ) as Usergroup");
    querySb.append(
        " ON " + PaattiColumnNames.SCHEDULE_USERGROUP_usergroupId + " = " + PaattiColumnNames.USERGROUP_usergroupId
        + " )");
    querySb.append(
        " as eventschedule ON " + PaattiColumnNames.SCHEDULE_scheduleID + " = "
        + PaattiColumnNames.EVENT_SCHEDULE_scheduleID);
    querySb.append(" LEFT OUTER JOIN ").append(PaattiTableNames.EVENTTIME).append(" ON ").append(
        PaattiColumnNames.EVENTTIME_eventTimeID + " = " + PaattiColumnNames.EVENT_EVENTTIME_eventTimeID);
    querySb.append(" WHERE ").append(PaattiColumnNames.EVENTTIME_eventTimeID).append(" > -1 AND " /
    ).append(PaattiColumnNames.EVENT_eventID).append(
        " NOT IN ( " );
    querySb.append(" SELECT DISTINCT " + PaattiColumnNames.EVENT_eventID + " FROM " + PaattiTableNames.EVENT);

```

```
querySb.append(
    " LEFT OUTER JOIN " + PaattiTableNames.TASK + " ON " + PaattiColumnNames.TASK_EVENT_eventID + " = "
    + PaattiColumnNames.EVENT_eventID);
querySb.append(
    " LEFT OUTER JOIN " + PaattiTableNames.CHOICE + " ON " + PaattiColumnNames.CHOICE_TASK_contains_taskID + " = "
    + PaattiColumnNames.TASK_taskID);
querySb.append(
    " LEFT OUTER JOIN " + PaattiTableNames.TASKDATA + " ON " + PaattiColumnNames.TASKDATA_CHOICE_choiceID + " = "
    + PaattiColumnNames.CHOICE_choiceID);
querySb.append(" WHERE " + PaattiColumnNames.TASKDATA_USER_userID + " = ").append(userID).append(
    " AND " + PaattiColumnNames.EVENT_rowStatus + " = ' " + ROWSTATUS_ACTIVE + "' ");
querySb.append(" AND ").append(PaattiColumnNames.EVENT_SCHEDULE_scheduleID).append(" IN ");
querySb.append(" (SELECT ").append(PaattiColumnNames.SCHEDULE_scheduleID).append(" FROM ").append(
    PaattiTableNames.SCHEDULE);
querySb.append(" LEFT OUTER JOIN (SELECT ").append(PaattiColumnNames.BELONGS_USERGROUP_usergroupID).append(" AS 'userGroupID'
FROM ").append(
    PaattiTableNames.BELONGS);
querySb.append(" WHERE ").append(PaattiColumnNames.BELONGS_USER_userID).append(" = ").append(userID).append(
    ") BELONGS_UG");
).append(PaattiColumnNames.SCHEDULE_USERGROUP_usergroupID).append(" = userGroupID WHERE "
" = userGroupID ");
querySb.append(" ORDER BY ").append(PaattiColumnNames.EVENTTIME_year).append(", "
).append(PaattiColumnNames.EVENTTIME_month).append(", ").append(
    PaattiColumnNames.EVENTTIME_day);
querySb.append(", ").append(PaattiColumnNames.EVENTTIME_fromHour).append(", "
).append(PaattiColumnNames.EVENTTIME_minute).append(
    " ASC");
return queryHelper.setQueryString(querySb.toString());
return queryHelper;
}
```