Barista Security Object

Provides an object with the ability to retrieve the current security settings as set in Barista Security Administration.

use ::sys/prog/bao_security.bbj::Security
declare Security Security!
security! = new Security("<table_id>")
Creates a new security object for specified table

security!.setSecurityID("<menu_option_id>")
Sets the menu security id if different from the alias

security!.getTableColumns()
Retrieves all columns for the specified form by accessing the Barista database tables.

security!.setTableColumns("column_1:c(16),column_2:c(30*),column_3:c(20*)")*
Specifies selected columns using a string template

cols!=BBjAPI().makeVector()
cols!.addItem("ADM_USER.NAME")
cols!.addItem("ADM_USER.DEPARTMENT")
cols!.addItem("ADM_USER.EMAIL_ADDRESS")
security!.setTableColumns(cols!)
Specifies selected columns using a vector.

security!.getSettings()
Retrieves security and audit settings for the table and all specified columns.

Table Level Security Methods

security!.getTaskAllowView()
Specifies if the user has rights to view the form.

security!.getTaskAllowAdd()
Specifies if the user has rights to add new records.

security!.getTaskAllowModify()
Specifies if the user has rights to modify existing records.

security!.getTaskAllowDelete()
Specifies if the user has rights to delete existing records.

security!.getTaskAllowPrint()
Specifies if the user has rights to print the table records.

Column Level Security Methods

security!.getColumnAllowView("<table_id>.<column_id>")
Specifies if the user has rights to view the specified column.

security!.getColumnAllowModify("<table_id>.<column_id>")
Specifies if the user has rights to modify the specified column.

security!.getColumnAuditAdd("<table_id>.<column_id>")
Specifies if the data for the specified column is auditied when a new record is entered.

security!.getColumnAuditModify("<table_id>.<column_id>")
Specifies if the data for the specified column is audited during if modified on an existing record.

security!.getColumnAuditDelete("<table_id>.<column_id>")
Specifies if the data for the specified column is audited when an existing record is deleted.

Creating Audit Records

The Barista Security Object provides a way to create records in the Barista Audit System.

security!.setAuditRecordKey("<record_key>")
Specifies the audit record primary key

security!.setAuditIOAction("<action>")
Specifies the audit IO action being performed
        W = Writing a new or modified record
        D = Deleting an existing record

security!.setAuditRecordStatus("<status>")
Specifies the current status of the record
        A = New record being added
        C = Existing record with modifications

security!.addAuditCompare("<table>.<column>", "<orig_val>", "<new_val>")
Creates compare records for specified columns. This method may be repeated any number of times for multiple columns. Example:
        security!.addAuditCompare("ADM_USER.NAME", "", "Test Name Change")
        security!.addAuditCompare("ADM_USER.DEPARTMENT", "", "DeptB")
        security!.addAuditCompare("ADM_USER.EMAIL_ADDRESS", "", "svulopas@basis.com")

security!.processAudit()
Processes the compare entries for all columns and creates the appropriate records

Example - Writing a modified record

```
security!.setAuditRecordKey("ADMIN_BAR")
security!.setAuditIOAction("W")
security!.setAuditRecordStatus("C")
security!.addAuditCompare("ADM_USER.NAME", "Test Name", "Test Name Change")
security!.addAuditCompare("ADM_USER.DEPARTMENT", "DeptA", "DeptB")
security!.addAuditCompare("ADM_USER.EMAIL_ADDRESS", "sgv@basis.com", "svulopas@basis.com")
security!.processAudit()
```

Example - Deleting an existing record

```
security!.setAuditRecordKey("ADMIN_DEV")
security!.setAuditIOAction("D")
security!.setAuditRecordStatus("C")
security!.addAuditCompare("ADM_USER.NAME", "Test Name", "")
security!.addAuditCompare("ADM_USER.DEPARTMENT", "DeptA", "")
security!.addAuditCompare("ADM_USER.EMAIL_ADDRESS", "sgv@basis.com", "")
```

```
security!.processAudit()
```