Text

Description automatically generated

November. 2022

PRAKTISI MENGAJAR

POLITEKNIK NEGERI BALI

Materi :   
Dimata QDEP Layer Controller

Applying Java : Model-View-Controller Software Architecture

Implementation

of

Object Oriented Programming

as   
Quick Development Pattern

in Software Industry

Dipl.-Ing. I Ketut Kartika Tanjana

PT Dimata Sora Jayate

www.dimata.com

# Controller :

package com.dimata.myqdep.form;

/\*\*

\*

\* @author iketutkartikatanjana

\*/

/\* java package \*/

import java.util.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

/\* dimata package \*/

import com.dimata.util.\*;

import com.dimata.util.lang.\*;

/\* qdep package \*/

import com.dimata.qdep.system.\*;

import com.dimata.qdep.form.\*;

import com.dimata.qdep.db.\*;

/\* project package \*/

//import com.dimata.qdep.db.\*;

import com.dimata.myqdep.entity.BillMain;

import com.dimata.myqdep.entity.PstBillMain;

public class CtrlBillMain extends Control implements I\_Language {

private int start;

private String msgString;

private BillMain billMain;

private PstBillMain pstBillMain;

private FrmBillMain frmBillMain;

int language = LANGUAGE\_DEFAULT;

public CtrlBillMain(HttpServletRequest request) {

msgString = "";

billMain = new BillMain();

try {

pstBillMain = new PstBillMain(0);

} catch (Exception e) {;

}

frmBillMain = new FrmBillMain(request, billMain);

resultText[LANGUAGE\_DEFAULT][RSLT\_DATA\_EXIST] = "Nota sudah ada";

resultText[LANGUAGE\_FOREIGN][RSLT\_DATA\_EXIST] = "Bill already exists";

}

private String getSystemMessage(int msgCode) {

switch (msgCode) {

case I\_DBExceptionInfo.MULTIPLE\_ID:

this.frmBillMain.addError(frmBillMain.FRM\_BILL\_NUMBER, resultText[language][RSLT\_DATA\_EXIST]);

return resultText[language][RSLT\_DATA\_EXIST];

default:

return resultText[language][RSLT\_UNKNOWN\_ERROR];

}

}

private int getControlMsgId(int msgCode) {

switch (msgCode) {

case I\_DBExceptionInfo.MULTIPLE\_ID:

return RSLT\_DATA\_EXIST;

default:

return RSLT\_UNKNOWN\_ERROR;

}

}

public int getLanguage() {

return language;

}

public void setLanguage(int language) {

this.language = language;

}

public BillMain getBillMain() {

return billMain;

}

public FrmBillMain getForm() {

return frmBillMain;

}

public String getMessage() {

return msgString;

}

public int getStart() {

return start;

}

public int action(int cmd, long oidBillMain) {

msgString = "";

int excCode = I\_DBExceptionInfo.NO\_EXCEPTION;

int rsCode = RSLT\_OK;

switch (cmd) {

case Command.ADD:

msgString = FRMMessage.getMsg(FRMMessage.MSG\_INPUT\_DATA);

break;

case Command.SAVE:

if (oidBillMain != 0) {

try {

billMain = PstBillMain.fetchExc(oidBillMain);

} catch (Exception exc) {

}

}

frmBillMain.requestEntityObject(billMain);

if (frmBillMain.errorSize() > 0) {

msgString = FRMMessage.getMsg(FRMMessage.MSG\_INCOMPLATE);

return RSLT\_FORM\_INCOMPLETE;

}

if (billMain.getOID() == 0) {

try {

long oid = pstBillMain.insertExc(this.billMain);

msgString = FRMMessage.getMsg(FRMMessage.MSG\_SAVED);

return RSLT\_OK;

} catch (DBException dbexc) {

excCode = dbexc.getErrorCode();

msgString = getSystemMessage(excCode);

return getControlMsgId(excCode);

} catch (Exception exc) {

msgString = getSystemMessage(I\_DBExceptionInfo.UNKNOWN);

return getControlMsgId(I\_DBExceptionInfo.UNKNOWN);

}

} else {

try {

long oid = pstBillMain.updateExc(this.billMain);

msgString = FRMMessage.getMsg(FRMMessage.MSG\_SAVED);

return RSLT\_OK;

} catch (DBException dbexc) {

excCode = dbexc.getErrorCode();

msgString = getSystemMessage(excCode);

} catch (Exception exc) {

msgString = getSystemMessage(I\_DBExceptionInfo.UNKNOWN);

}

}

break;

case Command.ASK:

if (oidBillMain != 0) {

try {

billMain = PstBillMain.fetchExc(oidBillMain);

msgString = FRMMessage.getMsg(FRMMessage.MSG\_ASKDEL);

return RSLT\_OK;

} catch (DBException dbexc) {

excCode = dbexc.getErrorCode();

msgString = getSystemMessage(excCode);

} catch (Exception exc) {

msgString = getSystemMessage(I\_DBExceptionInfo.UNKNOWN);

}

}

break;

case Command.DELETE:

if (oidBillMain != 0) {

try {

long oid = PstBillMain.deleteExc(oidBillMain);

if (oid != 0) {

msgString = FRMMessage.getMessage(FRMMessage.MSG\_DELETED);

excCode = RSLT\_OK;

} else {

msgString = FRMMessage.getMessage(FRMMessage.ERR\_DELETED);

excCode = RSLT\_FORM\_INCOMPLETE;

}

} catch (DBException dbexc) {

excCode = dbexc.getErrorCode();

msgString = getSystemMessage(excCode);

} catch (Exception exc) {

msgString = getSystemMessage(I\_DBExceptionInfo.UNKNOWN);

}

}

break;

case Command.EDIT:

default:

if (oidBillMain != 0) {

try {

billMain = PstBillMain.fetchExc(oidBillMain);

} catch (DBException dbexc) {

excCode = dbexc.getErrorCode();

msgString = getSystemMessage(excCode);

} catch (Exception exc) {

msgString = getSystemMessage(I\_DBExceptionInfo.UNKNOWN);

}

}

break;

}

return rsCode;

}

}