Create update dialog

April 13, 2023

11:39 AM

1. Create a new window in the form of a dialog (the above is to create a new page)

In this case, you can choose to use JDialog , because it is more suitable as a modal dialog box, usually used to pop up a secondary window in the main window to perform some actions. JFrame is typically used to create an independent main window, while JInternalFrame is typically used to create movable, resizable subwindows within a desktop window.

In order to create a new JDialog , follow these steps:

* 1. In Eclipse 's WindowBuilder , create a new Swing Designer class (e.g. right click on the project -> New -> Other... -> WindowBuilder -> Swing Designer -> Application Window ).
  2. Change the parent of the new class from JFrame to JDialog .
  3. Add the components and layout you need to JDialog , and write logic to update the information in the table.

Now, when you want to open a new window, you can instantiate and show that JDialog from the update button's actionPerformed method . For example:

JButton btnUpdate = new JButton("Update");

btnUpdate. addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// Get the information in the form, if necessary

//...

// Create and display a new JDialog

MyUpdateDialog dialog = new MyUpdateDialog(); // replace with the name of the JDialog class you created

dialog.setModal(true); // Sets the dialog as modal, which prevents the user from interacting with the parent window while the dialog is open

dialog.setVisible(true);

// Process the results returned by the dialog box here, such as updating the information in the form

//...

}

});

Second, retrieve the data that needs to be updated on the booksupdate page.

JButton btnNewButton\_1\_2 = new JButton ( "Update" );

btnNewButton\_1\_2. addActionListener ( new ActionListener () {

public void actionPerformed( ActionEvent e) {

//create new dialog for update

int selectedRow = table.getSelectedRow (); // Find the target object that needs to be updated in the table

if (selectedRow !=- 1 ) {

int ISN =( int )table.getValueAt (selectedRow, 0 ); // Read the first column of the target row, and the table.getValueAt method finds the target object. Here isn is the first column, so if you find this column, you will find the isn of the target , because isn is a unique code, so use this as the identity of the book

// Here, use the read isn information as an actual parameter, import the dialog page file, and create a dialog object to obtain the isn

BookInformation dialog = new BookInformation (ISN);

// dialog.setModal(true); Set the dialog as a modal dialog, which means that when the dialog is opened, it will block all parent windows and all other open modal dialogs until the dialog is opened until closed.

dialog.setModal ( true ) ;

// Let the dialog page be displayed

dialog.setVisible ( true ) ;

} else {

JOptionPane . *showMessageDialog* ( null , "Please select a row in the table!" );

}

}

});

3. Point to the data that needs to be updated from the table of booksupdate , click update , and import the original data into the newly created blank dialog box

Machine generated alternative text:
publicBooklnformation（intISN){
,//invokfromuserServicemethodtogetdatafromdatabase
ne》了US汇S汇V主C（）；
US巳汇S巳汇V工C巳
B00k5book=us汇S汇V主c．gtBookIn《0m彐t主onsByIS(ISN)；

Create a constructor with a parameter of ISN

Now create a method in the userservice page

getBookInformationsByISN (ISN)

Machine generated alternative text:
publicB00k5getBookInformationsByISN(intISN){
String5q1
"SELECT*FROMb00k5WHEREISN
returnbooksDAO.querySing1e（5q1，B00k5．cl彐55，
ISN)；

Use the reference in basicDao to realize the method of calling sql in a single line

Machine generated alternative text:
//selectsingle汇ow
publicTquerySing1e(String5q1，Class<T>clazz,Object.
null;
ConnectIOnconnectIOn
try{
connection=JDBCUtiIsByDruid．0；
p彐汇彐m巳t巳汇5）
return(connection,5q1，newBeanHandIer<T>(clazz)，
p彐汇彐m巳t巳汇5）；
catch(Exception）{
th0树n巳树RuntimeException（）；
《主n彐11y{
JDBCUti1sByDruid．close(null,null，
connection)

Return a single row of data of a generic class, pass in a sql statement, a class file, and parameters

Machine generated alternative text:
岛UpdateInformation
ISN
Title
Author
price
Publisher
BookType
Available

Machine generated alternative text:
textFieId=newJTextFieId()；
//getisnfromupdatepagei5n．
textFie1d.setText(String。valueOf(book.getIsn0））；
textFieId.addActionListener(newct主onL主St巳n巳汇0{

set the data obtained in the database into the blank column

The other blank columns have the same logic and are filled with data

When executing SQL query statement, QueryRunner will use ResultSetHandler to process the result. ResultSetHandler is an interface, which defines the method of processing the result set, such as encapsulating each row in the result set into an object, and encapsulating these objects into a List to return. In this example, the return value type of the booksDAO.querySingle method is Books.class, which means that the QueryRunner will encapsulate the data in the result set into a Books object and return the object.

Specifically, QueryRunner will execute the SQL query statement and obtain the result set, then encapsulate the data in each row of the result set into the Books object, and finally return the object. This process is accomplished through the BeanHandler class that implements the ResultSetHandler interface. The BeanHandler class encapsulates the first row of data in the result set into a Java object of the specified type and returns the object. In this example, Books.class specifies the type of Java object that the BeanHandler should encapsulate the result set, the Books type.

Machine generated alternative text:
publiccl彐55B彐5主CD0<T>{
p汇上V彐t巳QueryRunnerqr
newQueryRunner0；

Fourth, change the content according to the content of the blank box, and save

first create a

The updateBookInformation method is used to judge whether there is a new row change by whether there is an increase in the row.

Pay special attention to the sql statement here. It took a lot of time to compare the sql statement here. The placeholder symbol must match the updatedbook of the subsequent Book object .

Machine generated alternative text:
public boolean updateBookInformation (Books updatedBook) { 
String sq] 
- "UPDATE books SET title 
? , author 
prxce 
+ "publisher 
? , bookTypeId =?, available 
? WHERE ISN 
int rowsAffected = booksDAO.update (sq], updatedBook.getTit1e () , 
updatedBook.getAuthor (), updatedBook.getPrice (), updatedBook.getPub1isher () , 
updatedBook. getBookTypeId () , updatedBook. getAvai1ab1e () , updatedBook. getlsn () ) ; 
System. out. println ("Rows affected: 
' + rowsAffected); // Print the number of rows affected 
rowsAffected 
return 

Screen clipping taken: 2023-04-13 9:41 PM

When judging whether there is a changed row, the content of each value must be obtained. At this time, save the content under the button.

Machine generated alternative text:
JButton okButton 
new JButton ("Save") ; 
okButton. addActionListener (new ActionListener () { 
public void actionPerformed (ActionEvent e) { 
int isn = Integer . parseInt(textFie1d.getText ()) ; 
String title 
textFie1d 1 .getText () ; 
String author 
textFie1d 2 .getText () ; 
double price = Double . parseDoub1e (textFie1d 3.getText ) ; 
String publisher 
textFie1d 4 .getText () ; 
int bookTypeId 
int available 
Integer . parseInt(textFie1d 5 .getText ) ; 
Integer . parseInt(textFie1d 6.getText ) ; 
// Create a new Books object with the updated information 
Books updatedBook = new Books (isn, title, author, price, publisher, 
// Call the UserService to update the book information 
boolean isUpdated = userService . updateBookInformation (updatedBook) ; 
bookTypeId, 
available) ; 
if (isUpdated) { 
JOptionPane . showMessageDia10g(nu11, 
) else { 
JOptionPane . showMessageDia10g(nu11, 
"Book information updated successfully! " 
"Fai1ed to update book information! ") ; 

Create a constructor suitable for this book , and pay attention to the corresponding number and order of formal parameters and actual parameters.