sign up log in tour help

Saving an image in MySQL from Java

I am trying to save images in MySQL database from a Java swing application. I am using JFileChsoser to get the path of the image. Then after that converting the file so that it can be saved in the MySQL column which is of BLOB type. But every image I try to save does not save properly or get converted properly. Could someone tell me what I'm doing wrong over here?

```
private void btn_choosepicActionPerformed(java.awt.event.ActionEvent evt) {
    JFileChooser picchooser = new JFileChooser();
    picchooser.setDialogTitle("Select Image");
    picchooser.showOpenDialog(null);
    File pic=picchooser.getSelectedFile();
    path= pic.getAbsolutePath();
    txt_path.setText(path.replace('\\','/'));
    try{
        File image = new File(path);
        FileInputStream fis = new FileInputStream(image);
        ByteArrayOutputStream baos= new ByteArrayOutputStream();
        byte[] buff = new byte[1024];
        for(int readNum; (readNum=fis.read(buff)) !=-1 ; ){
            baos.write(buff,0,readNum);
        }
        userimage=baos.toByteArray();
    }
    catch(Exception e){
        JOptionPane.showMessageDialog(null, e);
    }
}
```

And then after this Im saving it to the database like so.

```
private void btn_saveActionPerformed(java.awt.event.ActionEvent evt) {
    String user= txt_username.getText();
    try{
        String sql="insert into imgtst (username,image) values
('"+user+"','"+userimage+"')";
        pst=con.prepareStatement(sql);
        pst.executeUpdate();
        JOptionPane.showMessageDialog(null, "Saved");
    }
    catch(Exception e){
        JOptionPane.showMessageDialog(null, e);
    }
}
```

and I have declared the variable userimage and path as a global variables

```
String path=null;
byte[] userimage=null;
```

java mysql image swing blob

asked Feb 23 '13 at 1:15

Raed Shahid
85 2 6 12

1 It is not a good practice to store image as a BLOB in MySQL database. You should first copy the file, in this case image, to your application folder and then just store the path to that image in database instead of image as BLOB. – dzeno Feb 23 '13 at 1:39

Do not store images in DB. Instead place them in a folder and store the path in DB. – Che Feb 23 '13 at 1:45

@dzeno I am doing a project for university. And I did the exact way how you just told. But a friend of mind is adamant on storing the images in MySQL database. So was just helping her out. – Raed Shahid Feb 23 '13 at 2:00

@Che I did not store the path of images in the database. Actually I am using the items in a JComboBox which are of the same name of the images in the folder to call the images in the application. I thought this way would better for my project. And while saving the images with the help of JFileChsoser to the application folder . I rename the image files with the same name of the item which it will be connected too. And those items in the JComboBox are populated from the MySQL database. – Raed Shahid Feb 23 '13 at 2:06

1 Answer

You are converting the byte[] to a String in your sql statement, and you will end up with incorrect data.

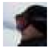
The right way to use a BLOB would be to pass the `InputStream` itself. You can use the `FileInputStream` you are using to read the file.

```
File image = new File(path);
FileInputStream fis = new FileInputStream ( image );

String sql="insert into imgtst (username,image) values (?, ?)";
pst=con.prepareStatement(sql);

pst.setString(1, user);
pst.setBinaryStream (2, fis, (int) file.length() );
```

When you retrieve it back you can similarly get an `InputStream` from the `ResultSet` :

```
InputStream imgStream = resultSet.getBinaryStream(2);
```

answered Feb 23 '13 at 1:39



jbx
6,175 3 34 70

Ah thanks alot for the hint. Its working now. I did this. `String sql="insert into imgtst (username,image) values (?,?)"; pst=con.prepareStatement(sql); String user= txt_username.getText(); pst.setString(1, user); pst.setBytes(2, userimage); pst.executeUpdate();` – Raed Shahid Feb 23 '13 at 1:56

Yes that works too. Its just a bit slower since you are first reading the file into the byte array, and then reading the byte array again to pass it to the database, and you consume more memory if the file is large, since you need to load it all first. While with my example its 'piping' through the data from the file to the DB. You could just keep the `File image` as a member variable set through `btn_choosepicActionPerformed()` and then in `btn_saveActionPerformed()` you open the stream and send it to the database. – jbx Feb 23 '13 at 13:47