

Play Swing and JDBC with **Rubeus**

Takeshi AKIMA @akm2000
Ruby Business Commons

Who am I?

I'm one of the authors, "Ruby Tettei Nyumon
(Ruby徹底入門)"



<http://www.amazon.co.jp/JRuby-徹底入門-株式会社イーシー・ワン/dp/4881666452>

I wrote all of the code in the last chapter.

Very famous book in the world

Look at **this!**

Video Message From JRuby guys 20100727



一番面白いのは私も関わった
Andoroidへの取り組みです

YouTube

0:43 / 4:07

<http://rubybizcommons.jp/ja/articles/2010/07/28/2010727-rubydays-2010/>

GJ, Tom!

Ruby Business Commons

<http://rubybizcommons.jp>

I'm a member of the staff.

In Fukuoka since 2007.

We hold hands-on workshops
about once per a month recently.

Rubeus

<http://code.google.com/p/rubeus/>

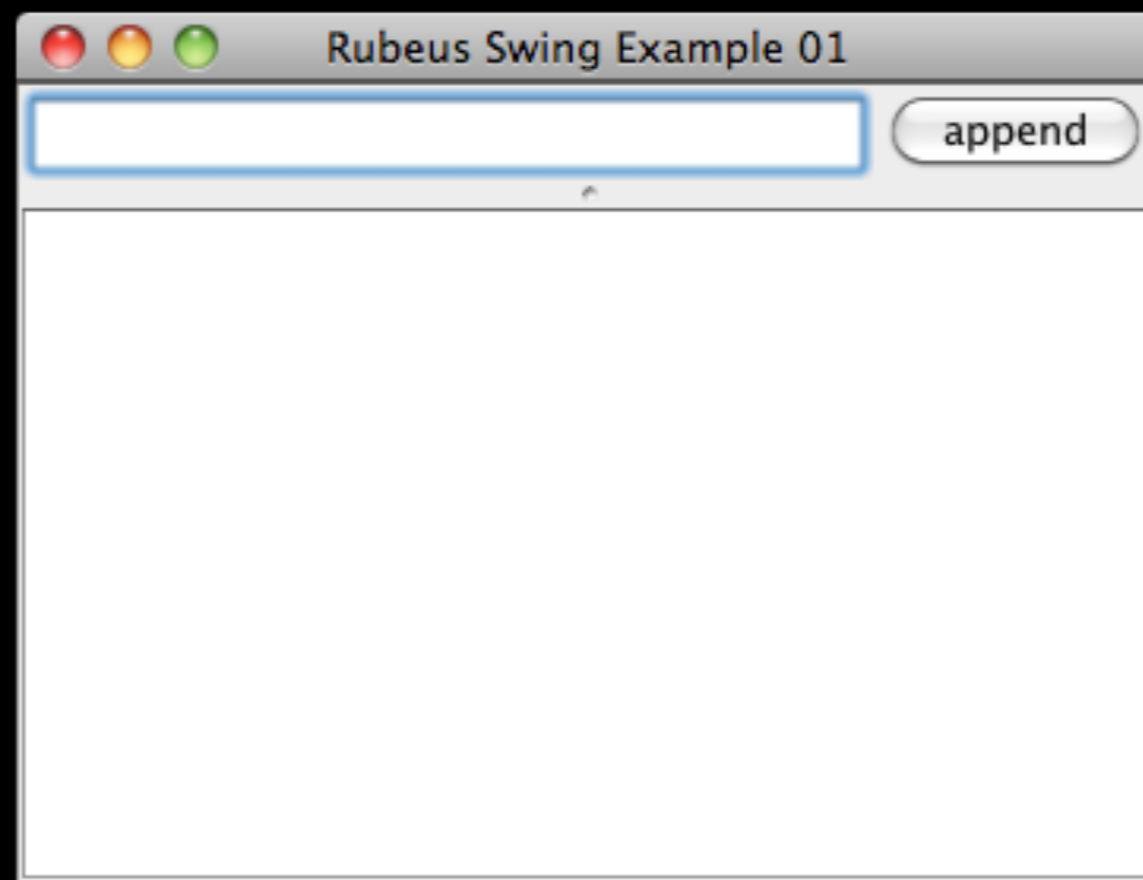
"Rubeus provides you an easy access to Java objects from Ruby scripts on JRuby."

We made a gem "Rubeus" in 2008.

It provides a DSL for Java **Swing** and **JDBC**

Swing with Rubeus

If you want to write a code of window like this ...



```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class JavaSwingExample01 {

    public static void main(String[] args) {
        JFrame frame = new JFrame("Rubeus Swing Example 01");
        frame.setLayout(new BorderLayout(frame.getContentPane(), BorderLayout.Y_AXIS));
        JSplitPane splitPane = new JSplitPane(JSplitPane.VERTICAL_SPLIT);
        frame.add(splitPane);
        JPanel panel = new JPanel();
        splitPane.setTopComponent(panel);
        panel.setLayout(new BorderLayout(panel, BorderLayout.X_AXIS));
        final JTextField textField = new JTextField();
        panel.add(textField);
        final JTextPane textPane = new JTextPane();
        textField.addKeyListener(new KeyAdapter() {
            public void keyPressed(KeyEvent event) {
                if (event.getKeyCode() == 10) {
                    textPane.setText(textPane.getText() + textField.getText() + "\n");
                    textField.setText("");
                }
            }
        });
        JButton button = new JButton("append");
        panel.add(button);
        button.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent event) {
                textPane.setText(textPane.getText() + textField.getText() + "\n");
                textField.setText("");
            }
        });
        JScrollPane scrollPane = new JScrollPane(textPane);
        splitPane.setBottomComponent(scrollPane);
        scrollPane.setPreferredSize(new Dimension(400, 250));
        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```



```
require 'rubygems'  
require "rubeus"
```

in JRuby with Rubeus

```
Rubeus::Swing.irb
```

```
JFrame.new("Rubeus Swing Example 01") do |frame|  
  frame.layout = BorderLayout.new(:Y_AXIS)  
  JSplitPane.new(JSplitPane::VERTICAL_SPLIT) do  
    JPanel.new do |panel|  
      panel.layout = BorderLayout.new(:X_AXIS)  
      @text_field = JTextField.new do |event|  
        if event.key_code == 10 # RETURN  
          @textpane.text += @text_field.text + "\n"  
          @text_field.text = ''  
        end  
      end  
      JButton.new("append") do  
        @textpane.text += @text_field.text  
        @text_field.text = ''  
      end  
    end  
  end  
  JScrollPane.new(:preferred_size => [400, 250]) do |pane|  
    @textpane = JTextPane.new  
  end  
end  
frame.visible = true  
end
```

JDBC with Rubeus

```
require 'rubygems'  
require 'rubeus'  
Rubeus::Jdbc.irb  
DriverManager.connect("jdbc:derby:test_db") do |conn|  
  conn.query("SELECT * FROM TEST ORDER BY ID") do |rs|  
    rs.each do |row|  
      puts row.to_a.join(",")  
    end  
  end  
end  
end
```

JDBC Metadata

```
conn = Rubeus::Jdbc::DriverManager.connect("jdbc:derby:test_db")  
#=> #<Java::OrgApacheDerbyImplJdbc::EmbedConnection40:0x478dabf1 >
```

```
table = conn.meta_data.tables[:FLIGHTS]  
#=> #<Rubeus::Jdbc::Table FLIGHTS  
      (FLIGHT_ID,SEGMENT_NUMBER,ORIG_AIRPORT,DEPART_TIME,  
      DEST_AIRPORT,ARRIVE_TIME,MEAL)>
```

```
table.columns  
#=> [#<Rubeus::Jdbc::Column FLIGHT_ID CHAR(6) NOT NULL>,  
      #<Rubeus::Jdbc::Column SEGMENT_NUMBER INTEGER(10) NOT  
      NULL>,#<Rubeus::Jdbc::Column ORIG_AIRPORT CHAR(3) NULL>,  
      #<Rubeus::Jdbc::Column DEPART_TIME TIME(8) NULL>...]
```

Easy usage Swing and JDBC

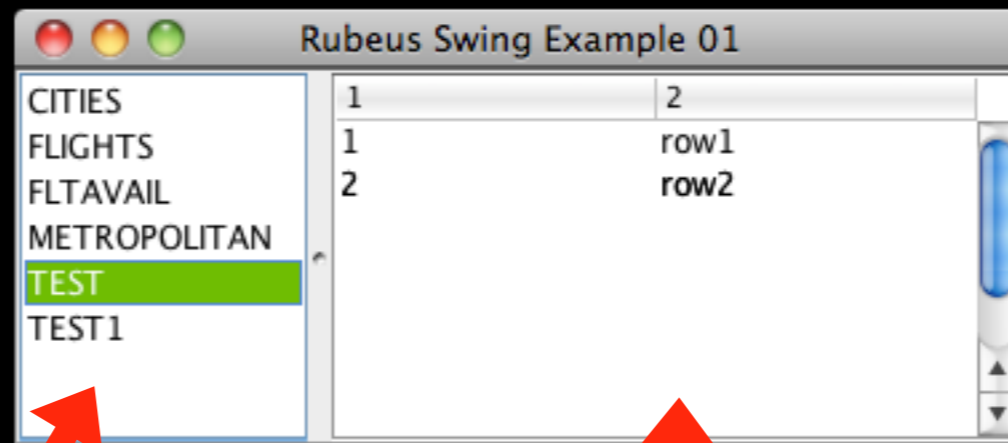


table names

result set

```
require 'rubygems'  
require "rubeus"  
Rubeus::Swing.irb
```

```
@conn = Rubeus::Jdbc::DriverManager.  
  connect("jdbc:derby:test_db;create = true")  
tables = @conn.meta_data.tables(:schema => "APP")
```

```
JFrame.new("Rubeus Swing Example 01") do |frame|  
  frame.layout = BorderLayout.new(:X_AXIS)  
  JSplitPane.new(JSplitPane::HORIZONTAL_SPLIT) do  
    JScrollPane.new(:preferred_size => [150, 250]) do |pane|  
      @list = JList.new(tables.map(&:name).to_java) do |event|  
        table_name = @list.selected_value.inspect  
        @conn.query("select * from #{table_name}") do |rs|  
          @table.model = rs.map{|row| row.to_a}  
        end  
      end  
    end  
  end  
  JScrollPane.new{|pane| @table = JTable.new}  
end  
frame.visible = true  
end
```

JRuby + irb

It's a good Java Object Inspector.

Now I always use irb to inspect Java classes before using it.

I can't remember a lot of Java method names and their arguments.

listing String methods

```
> java.lang.String.java_class.declared_class_methods
```

```
#=> [#<Java::JavaMethod/indexOf([C,int,int,[C,int,int,int])>,
#<Java::JavaMethod/checkBounds([B,int,int])>,
#<Java::JavaMethod/lastIndexOf([C,int,int,[C,int,int,int])>,
#<Java::JavaMethod/format(java.util.Locale,java.lang.String,
[Ljava.lang.Object;)>, #<Java::JavaMethod/format
(java.lang.String,[Ljava.lang.Object;)>, #<Java::JavaMethod/
valueOf([C)>, #<Java::JavaMethod/valueOf(double)>,
#<Java::JavaMethod/valueOf(java.lang.Object)>,
#<Java::JavaMethod/valueOf([C,int,int])>, ...
```

It's difficult to read :-)

When I want public static methods at first

```
require 'rubeus'
```

```
> Rubeus::Reflection.irb
```

```
#=> Object
```

```
> java.lang.String.java_public_class_methods
```

```
#=> ["format", "valueOf", "copyValueOf"]
```


And I'll check the method arguments with Rubeus

```
> java.lang.String.java_class.
```

```
?> java_public_class_methods.
```

```
?> select{|m| m.name == "copyValueOf"}
```

```
#=> [#<Java::JavaMethod/copyValueOf([C,int,int)>,  
#<Java::JavaMethod/copyValueOf([C)>]
```

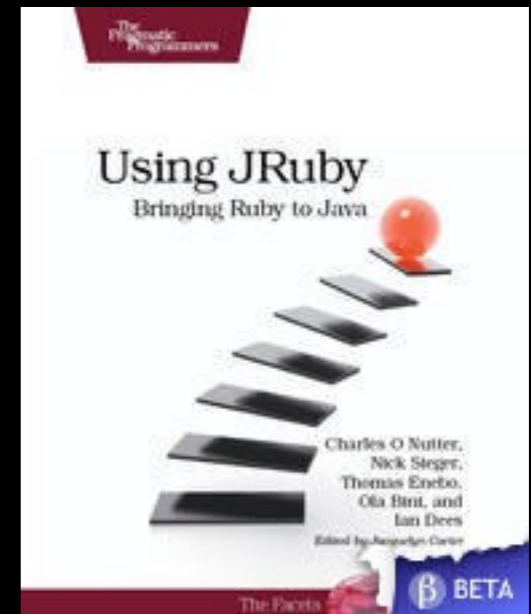
Rubeus in JRuby book!?

JRuby guys are writing a book.

Rubeus and other libraries are introduced in the GUI part.

GET IT!

<http://pragprog.com/titles/jruby/using-jruby>



Thanks



Tom gave us some patches.

We appreciate your kindness.