

소개글

자바의 socket + io + Robo + swing 이용해 작업한 GUI 프로그램입니다.

TeamView나 원격제어 프로그램들을 보면서 생각해보고 만들어본

동영상 스트리밍 구현해본 샘플 소스입니다.

Robo class의 **createScreenCapture()** 화면 캡처해 실시간으로 서버에 전송하는 예제 소스입니다.

java.socket + java.io + java.awt.Robo + javax.swing 을 이용한 샘플입니다.

테스트 환경은 사설아이피(**localhost**)에서 실행되었습니다.

아이피 변경은 해당 아이피 컴퓨터에서 서버클래스를 실행하시고

클라이언트 컴퓨터에서 **TeamViewClient.java** 안의 소스중 **new Socket("localhost", 9000);** 소스 부분에서 **localhost => '테스트할 아이피'** 변경하셔야 됩니다.

데이터베이스나 기타 환경 설정을 할 필요 없이 바로 실행 가능합니다.

이클립스,넷빈즈에서 자바 GUI를 공부하실 때 사용하실 수 있을 겁니다.

컴파일 실행환경 및 도구들 정보

운영체제 - Window7

자바SDK - java version "1.8.0_102"

Java(TM) SE Runtime Environment (build 1.8.0_102-b14)

Java HotSpot(TM) 64-Bit Server VM (build 25.102-b14, mixed mode)

자바 Tool

1) Eclipse Java EE IDE for Web Developers.

ersion: Neon Release (4.6.0)

uild id: 20160613-1800

2) NetBeans IDE 8.2

Java API 사용목록

Server 사용 API

```
package TeamViewer2;

import java.awt.Image;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.net.ServerSocket;
import java.net.Socket;
import java.net.SocketTimeoutException;
import java.util.Random;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JScrollPane;
import javax.swing.JFrame;
```

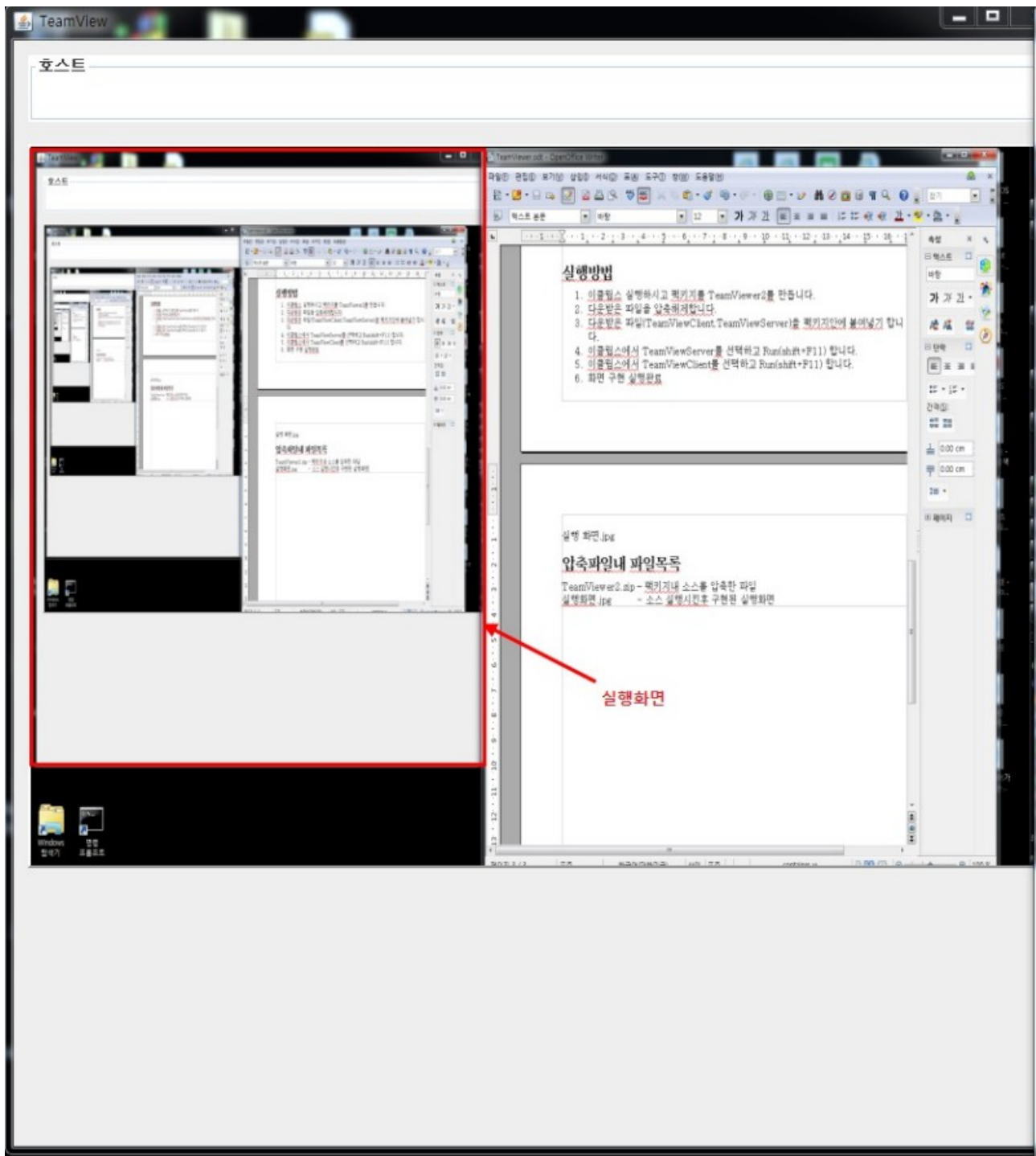
Client 사용 API

```
import java.awt.AWTException;
import java.awt.Dimension;
import java.awt.Rectangle;
import java.awt.Robot;
import java.awt.Toolkit;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import java.io.ObjectOutputStream;
import java.net.Socket;
import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
```

실행방법

1. 이클립스 실행하시고 패키지를 TeamViewer2를 만듭니다.
2. 아래하단에 TeamViewClient.java, TeamViewServer.java 를 TeamView2 패키지에 신규생성합니다.
3. 이클립스에서 TeamViewServer를 선택하고 Run(shift+ F11) 합니다.
4. 이클립스에서 TeamViewClient를 선택하고 Run(shift+ F11) 합니다.
5. 화면 구현 실행완료

실행화면



실행소스1

TeamViewClient.java

```
package TeamViewer2;

import java.awt.AWTException;
import java.awt.Dimension;
import java.awt.Rectangle;
import java.awt.Robot;
import java.awt.Toolkit;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import java.io.ObjectOutputStream;
import java.net.Socket;

import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JLabel;

public class TeamViewClient {
    JLabel label; // 화상출력
    boolean running = false; // 쓰레드 확인
    Socket socket; // 소켓

    public TeamViewClient(){}

    public static void main(String[] args){
        try {
            TeamViewClient tClient = new TeamViewClient();
            Robot r = new Robot(); // AWTException
            for(int i=0; i < 50; i++){

                // 빈파일객체 만들기
                File file = new File( "view.jpg" );
                file.createNewFile();//IOException

                // 스크린샷 정리
                Dimension scrSize = Toolkit.getDefaultToolkit().getScreenSize();
                // 사용자 스크린 사이즈 알아서 저장하기
                BufferedImage snapshot = r.createScreenCapture(new
Rectangle(scrSize)); // 화면사이즈대로 캡처해서 이미지 전용 버퍼 저장
                //
                https://docs.oracle.com/javase/7/docs/api/javax/imageio/ImageIO.html
                ImageIO.write( snapshot, "jpg", file);
                // snapshot 버퍼를 file에 적고 헤더를 jpg
                로 설정

                ImageIcon imgIcon = new ImageIcon( snapshot );
                // 스윙 레이블안에 삽입할 이미지아이콘객체 포장(랩기능)

                // 소켓정리
                Socket socket = new Socket("localhost", 9000);
                // 소켓연결

                ObjectOutputStream oos = new
ObjectOutputStream(socket.getOutputStream()); // 객체보내는 스트림사용
```

```

        oos.writeObject( imgIcon );
// imgIcon

객체 스트림에 보내기
        oos.flush();

        // 잔류 내용 보내기
        oos.close();

        Thread.sleep(300); // 0.3초 채우기
    }
} catch (AWTException | IOException | InterruptedException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}
}

```

실행소스2

TeamViewServer.java

```

package TeamViewer2;

import java.awt.Image;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.net.ServerSocket;
import java.net.Socket;
import java.net.SocketTimeoutException;
import java.util.Random;

import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JScrollPane;
import javax.swing.JFrame;
//import javax.swing.*;

// 쓰레드를 구현하고 JFrame을 상속한 TeamViewServer 클래스
public class TeamViewServer extends JFrame implements Runnable {
    JLabel label; // 화상출력
    boolean running = false; // 쓰레드 확인
    ServerSocket socket; // 서버소켓
    private int index = 0; // 동기화변수// 동기화*****

    public TeamViewServer(){

        initComponents();// 디자인모드
        setVisible(true);// 디자인모드

        try {
            socket = new ServerSocket(9000);// 서버 9000포트로 연결 객체 생성
            start();// 쓰레드 시작
            System.out.println("TeamView Thread 생성자----- index = " + index);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

// 디자인 initComponents();
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    vScreen = new javax.swing.JLabel();
    vHost = new javax.swing.JTextField();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("TeamView");

    vScreen.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));

```

```

vHost.setBorder(javax.swing.BorderFactory.createTitledBorder("호스트"));

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(vHost)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(18, 18, 18)
                    .addComponent(vScreen, javax.swing.GroupLayout.PREFERRED_SIZE, 836,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(0, 0, Short.MAX_VALUE))
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(0, 0, Short.MAX_VALUE))
            )
        )
);
jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(vHost, javax.swing.GroupLayout.PREFERRED_SIZE, 54,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(vScreen, javax.swing.GroupLayout.DEFAULT_SIZE, 581,
Short.MAX_VALUE)
            .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE))
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE))
        )
);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE))
            .addGroup(layout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE))
        )
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE))
            .addGroup(layout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE))
        )
);

pack();
}

```

// 스레드 시작(종료방지) , 채팅과 같이 구현

```
private void start() {  
    Thread team = new Thread(this);  
    running = true;  
    System.out.println("TeamView Thread start()----- index = " + index);  
    team.start();  
} //start -----
```

@Override

```
public void run() {  
    while( running ) {  
        try {
```

```
            System.out.println("TeamView Thread run() - running----- index = "  
+ index);
```

```
            Random r = new Random(System.currentTimeMillis());  
            long s = r.nextInt(300);  
            Thread.sleep(s); // 네트워크 딜레이
```

```
            ObjectInputStream ois = new  
ObjectInputStream(socket.accept().getInputStream()); // 소켓 인풋스트림기반으로 객체스트림사용
```

```
            Object obj = ois.readObject(); // 객  
체별로 읽어낸다.
```

```
            if( obj instanceof ImageIcon ) { //  
ImageIcon객체만 따로 떼낸다.
```

```
                ImageIcon snapshot = new ImageIcon( ((ImageIcon)  
obj).getImage().getScaledInstance(836, 581, Image.SCALE_SMOOTH) ); //  
ClassNotFoundException
```

```
                //ImageIcon snapshot = new ImageIcon( ((ImageIcon)  
obj).getImage().getScaledInstance((int)vScreen.getWidth()*1, (int)vScreen.getHeight()*1,  
Image.SCALE_SMOOTH) ); // 동적사이즈 조정 Height bug
```

```
                vScreen.setIcon(snapshot); //vScreen  
revalidate(); // 화면검증  
repaint(); // 화면다시그리기
```

```
            } else { // else if ( obj instanceof Object ) //
```

```
            }
```

```
        } catch ( SocketTimeoutException e) { // 소켓타임아웃
```

```
            System.out.println("SocketTimeout  
SocketTimeoutException-----index = " + index--);  
            e.printStackTrace();
```

```
        } catch (InterruptedException e) {
```

```
            index--; // 동기화*****  
            e.printStackTrace();
```

```
        } catch (IOException | ClassNotFoundException e) {
```

```
            System.out.println("SocketTimeout IOException |  
ClassNotFoundException-----index = " + index--);
```



```

        e.printStackTrace();

    } catch ( Exception e) {
        System.out.println("SocketTimeout Exception-----index = " +
index--);
        e.printStackTrace();

        }// try -----
    }// while ----- Thread Check
    addIndex();// 동기화*****
}// run -----

public void stop() {
    System.out.println("TeamView Thread Stop()-----index = " + index);
    running = false;
} // stop -----

// 동기화구문// 동기화*****
synchronized void addIndex(){
    index++;
    System.out.println("current index value: -----index = " + index);
}

public static void main( String[] arg ) {
    System.out.println("TeamView main() -----");
    new TeamViewServer();
}// main -----

// Variables declaration - do not modify
private javax.swing.JPanel jPanel1;
private javax.swing.JTextField vHost;
private javax.swing.JLabel vScreen;
// End of variables declaration

}// TeamView class-----

/*
    TeamView main() -----
    TeamView Thread start()----- index = 0
    TeamView Thread 생성자----- index = 0
    TeamView Thread run() - running----- index = 0
*/

```

