

네트워킹





○ 네트워킹

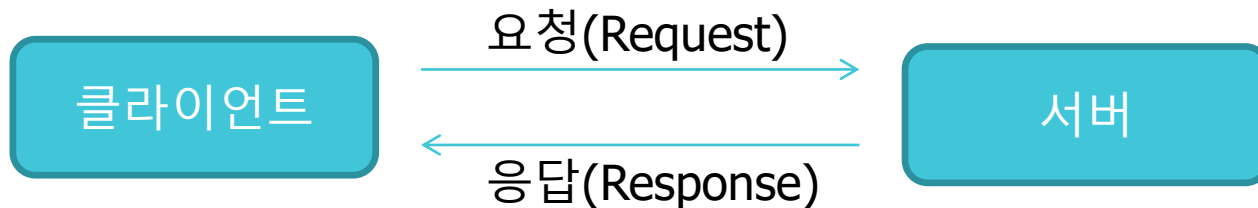
- ① 인터넷 망에 연결되어 있는 원격지의 서버 또는 원격지의 단말과 통신을 통해 데이터를 주고 받는 일반적인 일
- ② 최근 정보를 주고 받기 위해 많이 사용하는 소셜 네트워크서비스(SNS)가 네트워킹을 활용하는 대표적인 예



○ 원격지의 서버를 연결하는 방식

◎ 2-tier C/S 모델

- 클라이언트와 서버가 일대일로 연결하는 방식



◎ 3-tier 모델

- 서버를 좀 더 유연하게 구성
- 응용 서버와 데이터 서버로 구성하는 경우, 데이터베이스를 분리시킴





○ 네트워크에 대한 이해

- TCP/IP 수준의 통신 방식을 제공하는 소켓을 이용해 서버에 연결해 보면 이해하기 쉬움
- 일반적인 프로그래밍에서는 대부분 TCP 연결 사용
- 비연결성 특성으로 인해 실시간으로 데이터를 처리하는 애플리케이션의 경우, 응답 속도를 높이기 위해 HTTP 보다 소켓 연결 선호

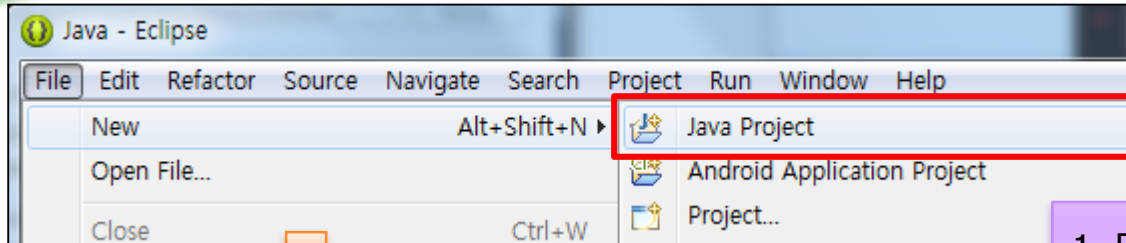
○ 소켓 연결 방식

- 안드로이드에서는 표준 자바의 소켓을 그대로 사용할 수 있음
- 서버 : 서버 소켓을 만들어 실행함(포트 지정)
- 클라이언트: 소켓을 만들어 서버 소켓으로 연결함(IP와 포트 지정)
- Stream 객체를 이용해 데이터를 보내거나 받을 수 있음

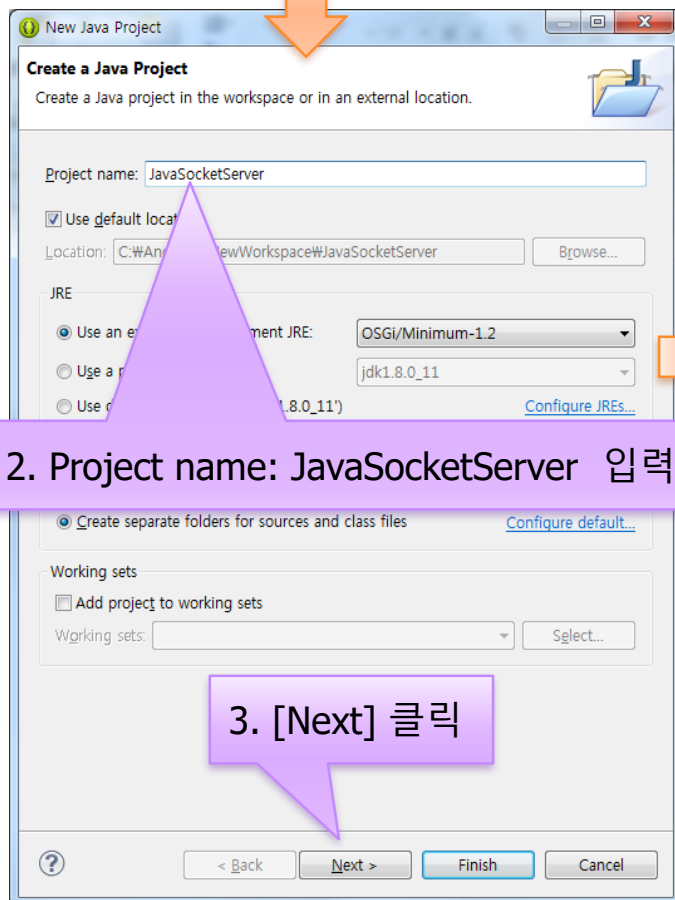


자바 소켓 프로그램

5

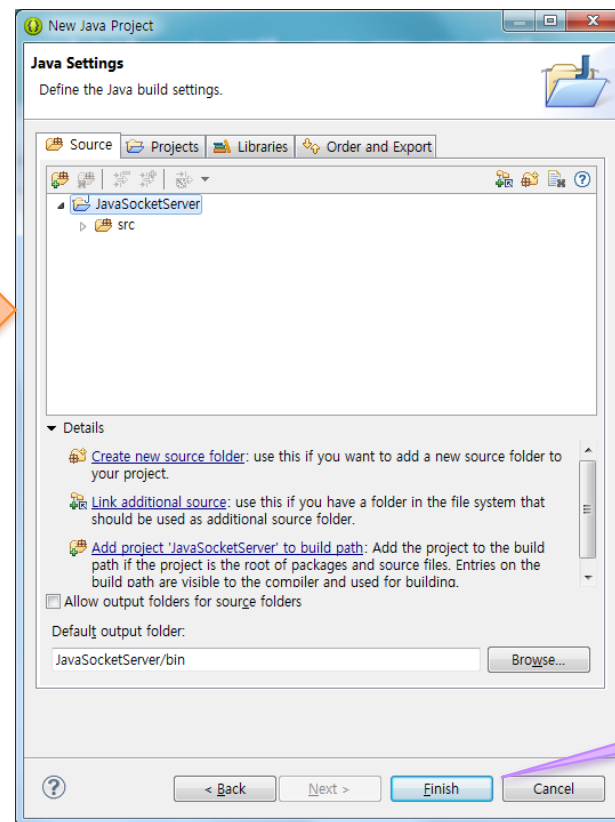


1. [File]→[New]→[Java Project] 클릭

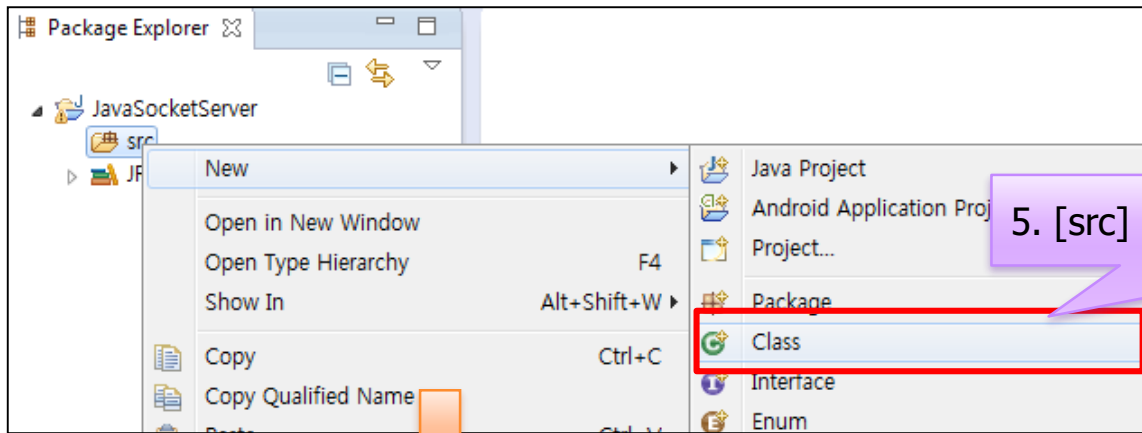


2. Project name: JavaSocketServer 입력

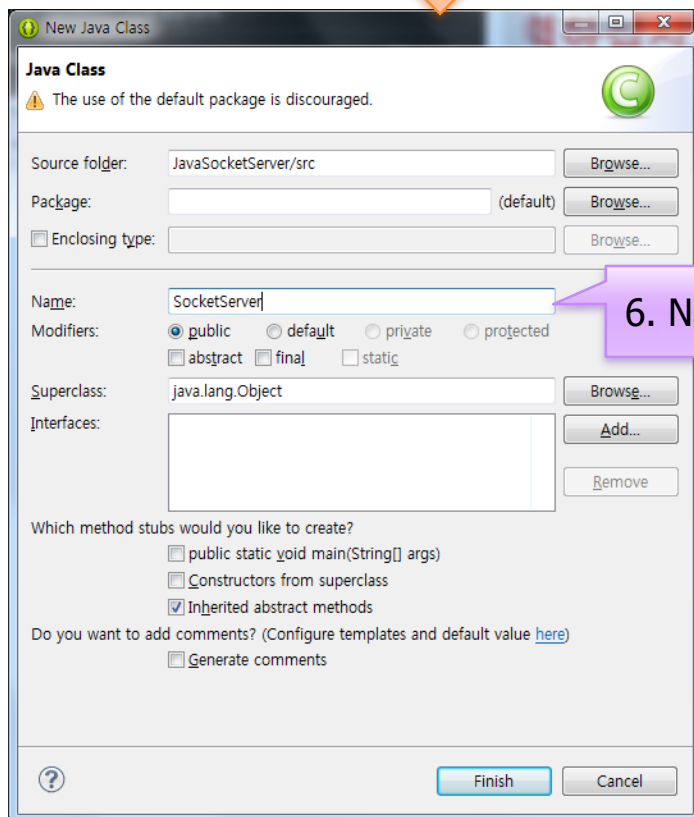
3. [Next] 클릭



4. [Finish] 클릭



5. [src] 오른쪽 마우스 클릭 후 [class] 클릭



6. Name: SocketServer 입력

7. 코딩

```
1 import java.io.ObjectInputStream;
2 import java.io.ObjectOutputStream;
3 import java.net.InetAddress;
4 import java.net.ServerSocket;
5 import java.net.Socket;
6
7
8 public class SocketServer {
9     public static void main(String[] args){
10         try{
11             int portNumber = 5001;
12
13             System.out.println("Starting Java Socket Server ...");
14             ServerSocket aServerSocket = new ServerSocket(portNumber);
15             System.out.println("Listening at port " + portNumber + "...");
16             while(true){
17                 Socket sock = aServerSocket.accept();
18                 InetAddress ClientHost = sock.getLocalAddress();
19                 int clientPort = sock.getPort();
20                 System.out.println("A client connected. host : " +
21                                     ClientHost + ", port : " + clientPort);
22                 ObjectInputStream instream = new ObjectInputStream(sock.getInputStream());
23                 Object obj = instream.readObject();
24                 System.out.println("Input : " + obj);
25
26                 ObjectOutputStream outstream = new ObjectOutputStream(sock.getOutputStream());
27                 outstream.writeObject(obj + " form Server.");
28                 outstream.flush();
29                 sock.close();
30             }
31         }catch(Exception ex){
32             ex.printStackTrace();
33         }
34     }
35 }
36
37 }
```

```
8 public class SocketServer {
9     public static void main(String[] args){
10         try{
11             int portNumber = 5001;
12
13             System.out.println("Starting Java Socket Server ...");
14             ServerSocket aServerSocket = new ServerSocket(portNumber);
15             System.out.println("Listening at port " + portNumber + "...");
16             while(true){
17                 Socket sock = aServerSocket.accept();
18                 InetAddress ClientHost = sock.getLocalAddress();
19                 int clientPort = sock.getPort();
20                 System.out.println("A client connected. host : " +
21                                     ClientHost + ", port : " + clientPort);
22                 ObjectInputStream instream = new ObjectInputStream(sock.getInputStream());
23                 Object obj = instream.readObject();
24                 System.out.println("Input : " + obj);
25
26                 ObjectOutputStream outstream = new ObjectOutputStream(sock.getOutputStream());
27                 outstream.writeObject(obj + " form Server.");
28                 outstream.flush();
29                 sock.close();
30             }
31         }catch(Exception ex){
32             ex.printStackTrace();
33         }
34     }
35 }
36
37 }
```

서버 소켓 생성

클라이언트가 접속할 때 소켓 개체 생성

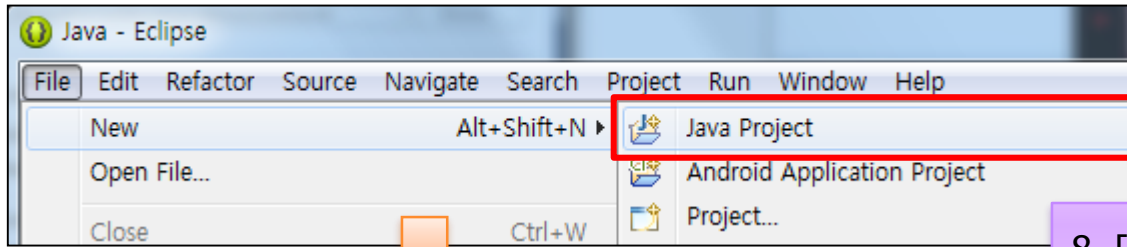
반복문으로 클라이언트 연결 대기

클라이언트가 보낸 오브젝트를 받는다.

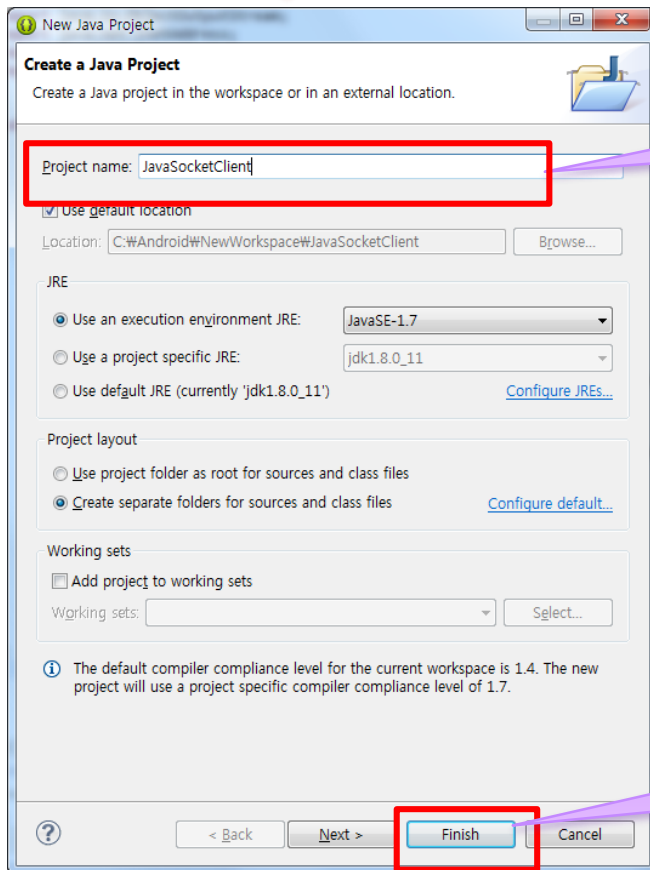
클라이언트에 오브젝트를 보낸다.

클라이언트에 오브젝트를 보낸다.

클라이언트 만들기

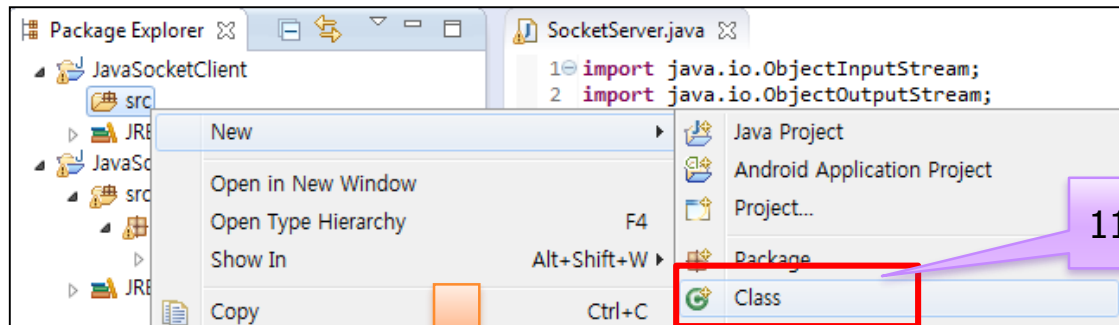


8. [File]→[New]→[Java Project] 클릭

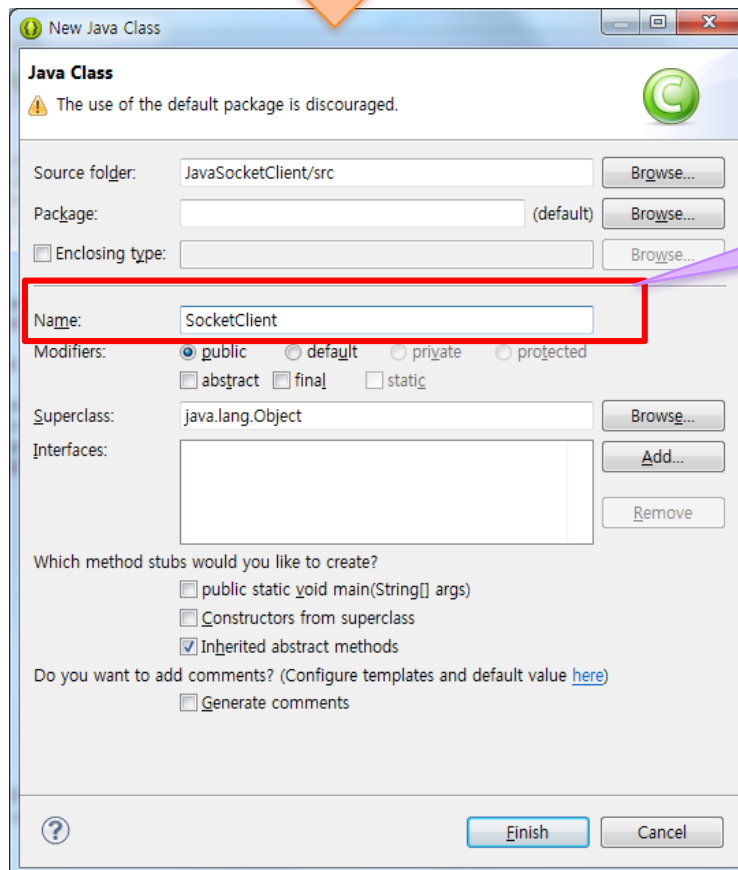


9. "Project name: JavaSocketClient" 입력

10. 클릭



11. [src] 오른쪽 마우스 클릭 후 [Class] 클릭



12. Name : SocketClient 입력

```
SocketClient.java ✕  
1 import java.net.Socket;  
2  
3  
4 public class SocketClient {  
5     public static void main(String[] args){  
6         int portNumber = 5001;  
7         Socket sock = new Socket("localhost", portNumber);  
8  
9  
10    }  
11 }
```

13. 클릭

↓

```
SocketClient.java ✕  
1 import java.net.Socket;  
2  
3  
4 public class SocketClient {  
5     public static void main(String[] args){  
6         int portNumber = 5001;  
7         Socket sock = new Socket("localhost", portNumber);  
8  
9  
10    }  
11 }  
12
```

Unhandled exception type
3 quick fixes available:
• Add throws declaration
• Surround with try/multi catch
• Surround with try/catch

14. 클릭

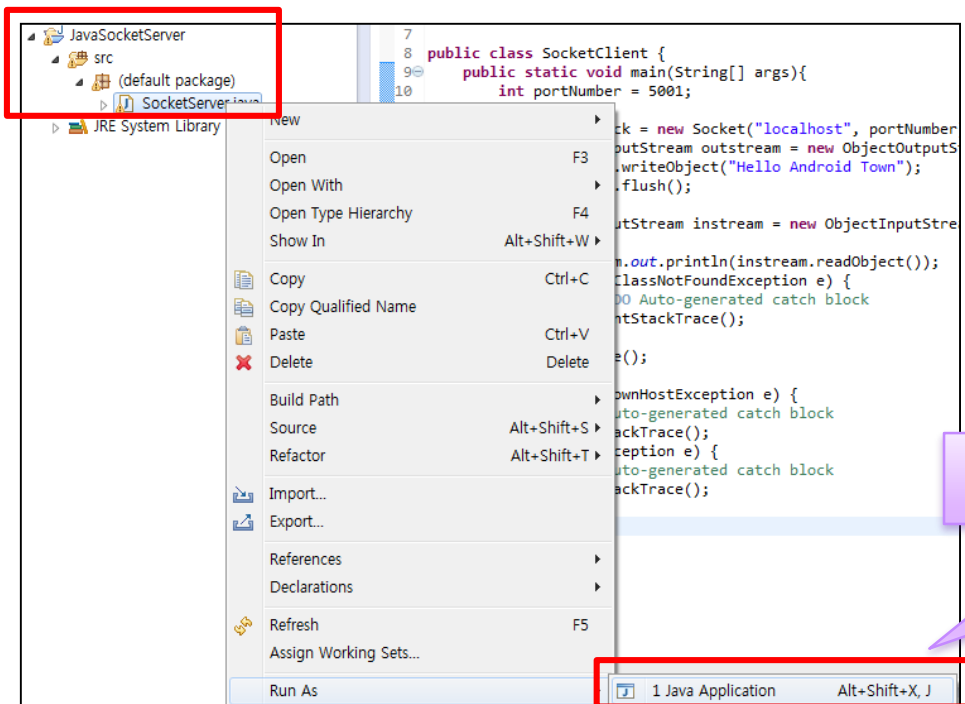
15. 코딩

```
1 import java.io.IOException;
2 import java.io.ObjectInputStream;
3 import java.io.ObjectOutputStream;
4 import java.net.Socket;
5 import java.net.UnknownHostException;
6
7
8 public class SocketClient {
9     public static void main(String[] args){
10         int portNumber = 5001;
11         try {
12             Socket sock = new Socket("localhost", portNumber);
13             ObjectOutputStream outstream = new ObjectOutputStream(sock.getOutputStream());
14             outstream.writeObject("Hello Android Town");
15             outstream.flush();
16
17             ObjectInputStream instream = new ObjectInputStream(sock.getInputStream());
18             try {
19                 System.out.println(instream.readObject());
20             } catch (ClassNotFoundException e) {
21                 // TODO Auto-generated catch block
22                 e.printStackTrace();
23             }
24             sock.close();
25
26         } catch (UnknownHostException e) {
27             // TODO Auto-generated catch block
28             e.printStackTrace();
29         } catch (IOException e) {
30             // TODO Auto-generated catch block
31             e.printStackTrace();
32         }
33     }
34 }
35 }
```

```
8 public class SocketClient {
9     public static void main(String[] args){
10         int portNumber = 5001;
11         try {
12             Socket sock = new Socket("localhost", portNumber);
13             ObjectOutputStream outstream = new ObjectOutputStream(sock.getOutputStream());
14             outstream.writeObject("Hello Android Town");
15             outstream.flush();
16
17             ObjectInputStream instream = new ObjectInputStream(sock.getInputStream());
18             try {
19                 System.out.println(instream.readObject());
20             } catch (ClassNotFoundException e) {
21                 // TODO Auto-generated catch block
22                 e.printStackTrace();
23             }
24             sock.close();
25
26         } catch (UnknownHostException e) {
27             // TODO Auto-generated catch block
28             e.printStackTrace();
29         } catch (IOException e) {
30             // TODO Auto-generated catch block
31             e.printStackTrace();
32         }
33     }
34 }
35 }
```

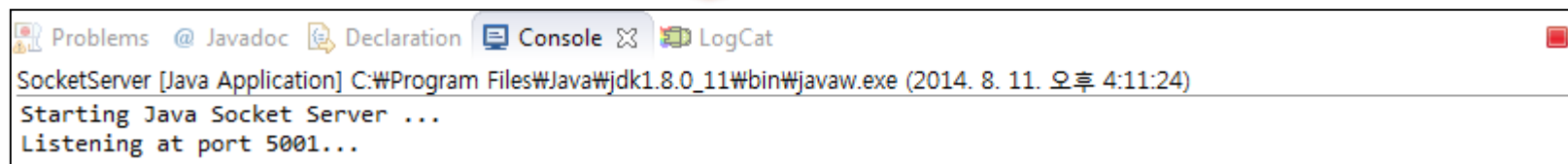
서버에 오브젝트를 보낸다.

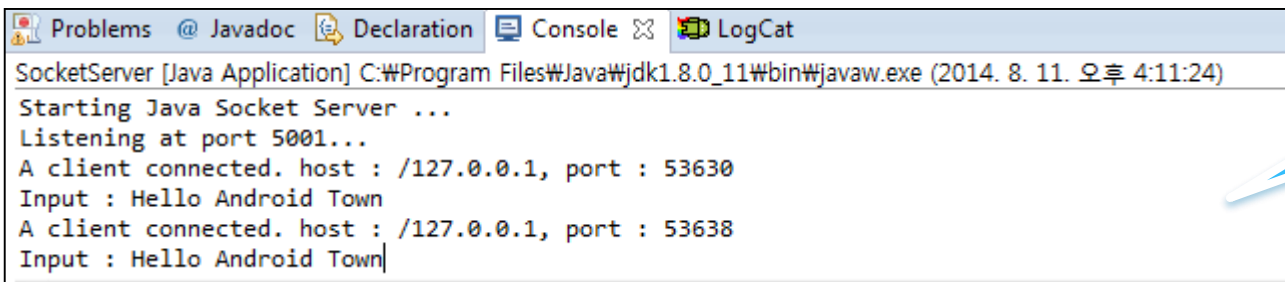
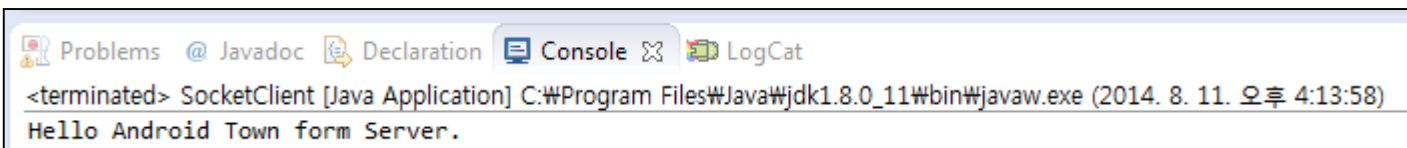
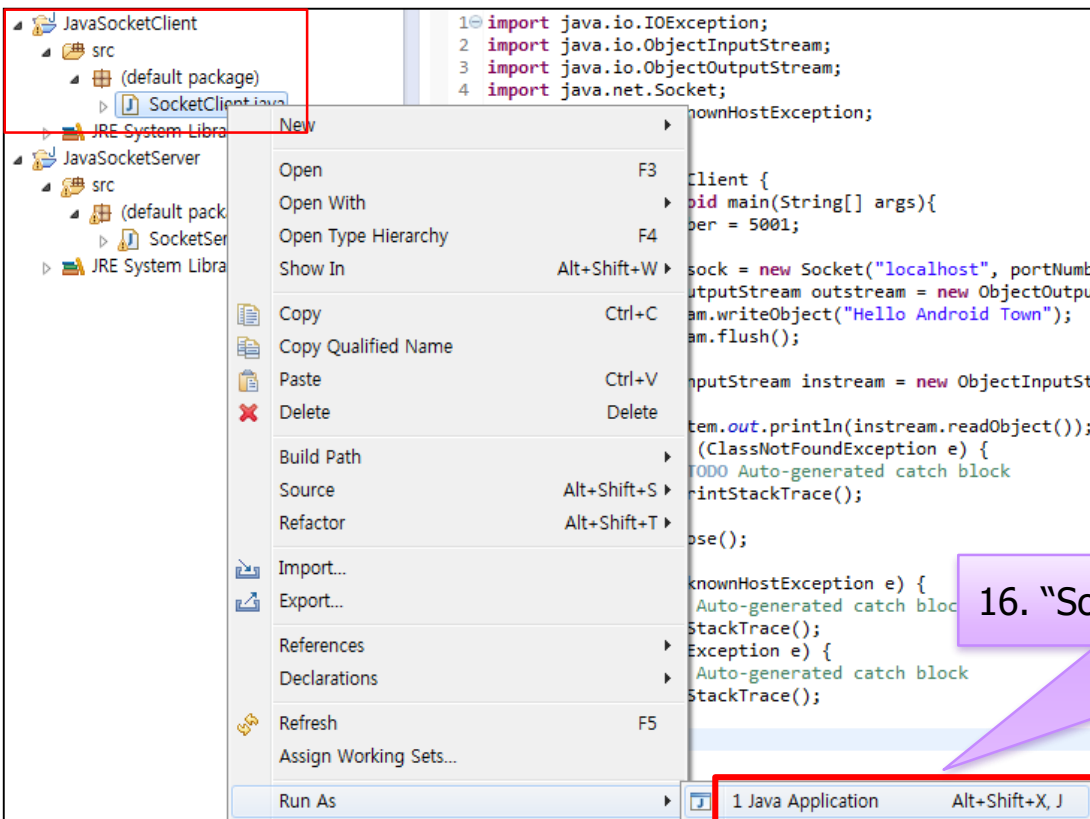
서버가 보낸 오브젝트를 받는다.



15. "SocketServer" 오른쪽 마우스 클릭 후 실행

실행





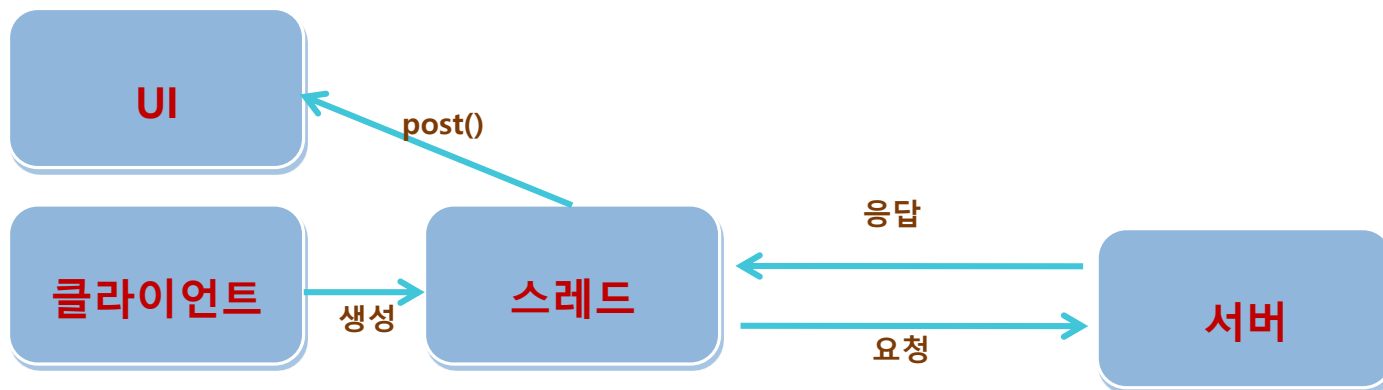
안드로이드 소켓 프로그래밍

■ 네트워킹을 사용할 때는 반드시 스레드 사용

- 최신 버전의 안드로이드에서는 네트워킹을 사용할 때는 반드시 스레드를 사용하도록 변경되었음 (이전에는 스레드 없이도 가능했음)

■ 스레드를 사용하므로 UI 업데이트를 위해서는 반드시 핸들러 사용

- 네트워킹을 위해 새로 만든 스레드 안에서 그 결과를 보여주기 위해 UI 업데이트를 하는 경우 스레드 부분에서 공부한 바와 같이 핸들러를 사용해야 함
- 가장 간단한 방법으로 post() 메소드 사용 권장





Application name: **AndroidSocketClient**

Company Domain: **admid.example.com**

Project location: **C:\Android\StdWork\AndroidSocketClient**

Target Android Devices

- **Phone and Tabet**
- **Minimun SDK API 15 : Android 4.0.3**

Add an activity to Mobile : **Empty Activity**

Customize the Activity

Activity Name: **MainActivity**

Layout Name: **activity_main**



```
activity_main.xml ×
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="com.example.administrator.androidsocketclient.MainActivity">

    <TextView
        android:id="@+id/text01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:text="버튼을 누르면 소켓이 연결됩니다.
        메시지는 로그를 확인하세요." />

    <Button
```

1. 코딩

activity_main.xml x

메시지는 로그를 확인하세요." />

2. 코딩

<Button

```
    android:id="@+id/button01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/text01"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"
    android:text="소켓 연결하기"
    android:textSize="20dp"
    android:textStyle="bold" />
```

<EditText

```
    android:id="@+id/input01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/button01"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"
    android:hint="연결할 소켓 서버 IP"
    android:text="192.168.1.100"
    android:textSize="16dp"
    android:textStyle="bold" />
```

</RelativeLayout>



MainActivity.java ×

```
1 package com.example.administrator.androidsocketclient;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
5 import android.util.Log;
6 import android.view.View;
7 import android.widget.Button;
8 import android.widget.EditText;
9
10 import java.io.ObjectInputStream;
11 import java.io.ObjectOutputStream;
12 import java.net.Socket;
13
14 public class MainActivity extends AppCompatActivity {
15
16     EditText input01;
17     Button button01;
18
19     @Override
```

3. 추후 import 확인

4. 코딩

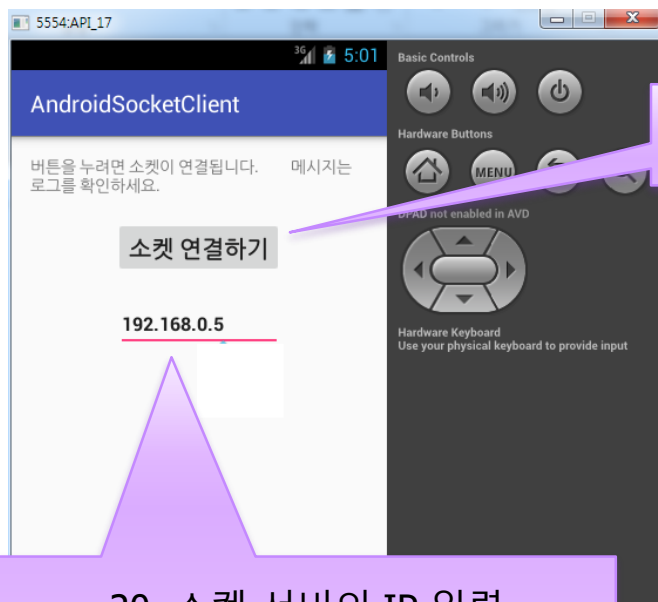
MainActivity.java x

```
19      @Override
20      protected void onCreate(Bundle savedInstanceState) {
21          super.onCreate(savedInstanceState);
22          setContentView(R.layout.activity_main);
23
24          input01 = (EditText)findViewById(R.id.input01);
25          button01 = (Button)findViewById(R.id.button01);
26
27          button01.setOnClickListener(new View.OnClickListener() {
28              @Override
29              public void onClick(View v) {
30                  String addr = input01.getText().toString().trim();
31                  ConnectThread thread= new ConnectThread(addr);
32                  thread.start();
33              }
34          });
35      }
36      class ConnectThread extends Thread{
37          String hostname;
```

5. 코딩

6. 코딩

```
34     });
35 }
36 class ConnectThread extends Thread{
37     String hostname;
38     public ConnectThread(String addr){
39         hostname = addr;
40     }
41     public void run(){
42         try{
43             int port = 5001;
44             Socket sock = new Socket(hostname, port);
45             ObjectOutputStream outputStream = new ObjectOutputStream(sock.getOutputStream());
46             outputStream.writeObject("Hello AndroidTown on Android");
47             outputStream.flush();
48
49             ObjectInputStream instream = new ObjectInputStream(sock.getInputStream());
50             String obj = (String)instream.readObject();
51
52             Log.d("MainActivity", "서버에서 받은 메시지: " +obj);
53             sock.close();
54         } catch (Exception ex){
55             ex.printStackTrace();
56         }
57     }
58 }
59 }
```



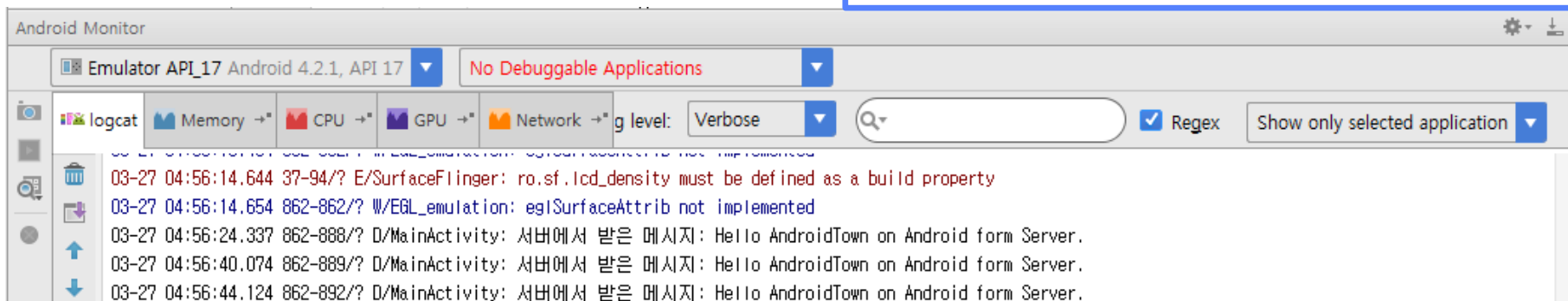
21. 클릭

(이클립스) 서버 로그

```
Problems @ Javadoc Declaration Console LogCat
SocketServer [Java Application] C:\Program Files\Java\jdk1.8.0_11\bin\javaw.exe (2016. 3. 27. 오후 1:51:37)
A client connected. host : /192.168.0.5, port : 49892
Input : Hello AndroidTown on Android
A client connected. host : /192.168.0.5, port : 49897
Input : Hello AndroidTown on Android
```

20. 소켓 서버의 IP 입력

(안드로이드 스튜디오) 클라이언트 로그





■ HTTP 연결 방식

- 예전 휴대 단말은 데이터 통신의 송수신 속도가 느려서 소켓으로 연결하거나 웹 페이지를 보기 위해서는 많이 기다려야 함
- 비연결성(stateless)인 HTTP 프로토콜은 새로 연결을 만드는 데 따른 지연 시간이 길게 발생
- 최근 스마트폰 및 무선 네트워크 환경이 좋아져서 HTTP 프로토콜을 이용한 웹의 사용이 자연스러울 뿐만 아니라 일반 웹 사이트를 보는 풀 브라우징(full browsing)도 가능함
- 자바에서 사용하던 HTTP 관련 클래스를 그대로 사용할 수 있음





Application name: **HTTP01**

Company Domain: **admid.example.com**

Project location: **C:\Android\StdWork\HTTP01**

Target Android Devices

- **Phone and Tabet**
- **Minimun SDK API 15 : Android 4.0.3**

Add an activity to Mobile : **Empty Activity**

Customize the Activity

Activity Name: **MainActivity**

Layout Name: **activity_main**



1. 코딩

```
activity_main.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      android:orientation="vertical">
7
8      <TextView
9          android:id="@+id/text01"
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:text="버튼을 누르면 HTTP 연결됩니다. "
13         android:textSize="16dp" />
14
15     <LinearLayout
16         android:layout_width="wrap_content"
```



```
activity_main.xml x
13 android:textSize="16dp" />
14
15 <LinearLayout
16     android:layout_width="wrap_content"
17     android:layout_height="wrap_content"
18     android:layout_marginTop="20dp"
19     android:orientation="horizontal">
20
21     <EditText
22         android:id="@+id/inout01"
23         android:layout_width="wrap_content"
24         android:layout_height="wrap_content"
25         android:hint="사이트 주소 입력 ..."
26         android:textSize="18dp" />
27
28     <Button
29         android:id="@+id/requestBtn"
30         android:layout_width="wrap_content"
31         android:layout_height="wrap_content"
32         android:text="HTTP 요청"
33         android:textSize="20dp"
34         android:textStyle="bold" />
35 </LinearLayout>
36
```

2. 코딩



```
activity_main.xml x
31 android:layout_height="wrap_content"
32 android:text="HTTP 요청"
33 android:textSize="20dp"
34 android:textStyle="bold" />
35 </LinearLayout>
36
37 <ScrollView
38     android:layout_width="wrap_content"
39     android:layout_height="wrap_content"
40     android:layout_marginTop="20dp">
41
42     <TextView
43         android:id="@+id/txtMsg"
44         android:layout_width="match_parent"
45         android:layout_height="match_parent"
46         android:background="#ff99ccee"
47         android:textColor="#ff0000ff"
48         android:textSize="12dp" />
49 </ScrollView>
50
51 </LinearLayout>
```

3. 코딩



C MainActivity.java x

```
1 package com.example.administrator.http01;
2
3 import android.os.Bundle;
4 import android.os.Handler;
5 import android.support.v7.app.AppCompatActivity;
6 import android.util.Log;
7 import android.view.View;
8 import android.widget.Button;
9 import android.widget.EditText;
10 import android.widget.TextView;
11
12 import java.io.BufferedReader;
13 import java.io.InputStreamReader;
14 import java.net.HttpURLConnection;
15 import java.net.URL;
16
17  public class MainActivity extends AppCompatActivity {
```

4. 추후 확인

5. 코딩

```
MainActivity.java x
16
17 public class MainActivity extends AppCompatActivity {
18
19     EditText input01;
20     TextView txtMsg;
21     Button requestBtn;
22
23     public static String defaultUrl = "http://m.naver.com";
24
25     Handler handler = new Handler();
26
27     @Override
28     protected void onCreate(Bundle savedInstanceState) {
29         super.onCreate(savedInstanceState);
30         setContentView(R.layout.activity_main);
31
32         input01 = (EditText)findViewById(R.id.inout01);
33         input01.setText(defaultUrl);
34
35         txtMsg = (TextView)findViewById(R.id.txtMsg);
36         requestBtn = (Button)findViewById(R.id.requestBtn);
37
38         requestBtn.setOnClickListener(new View.OnClickListener() {
```

6. 코딩

```
MainActivity.java x
36 requestBtn = (Button)findViewById(R.id.requestBtn);
37
38 requestBtn.setOnClickListener(new View.OnClickListener() {
39     @Override
40     public void onClick(View v) {
41         String urlStr = input01.getText().toString();
42
43         ConnectThread thread = new ConnectThread(urlStr);
44         thread.start();
45     }
46 });
47
48
49 class ConnectThread extends Thread{
```

MainActivity.java x

```
47     }
48
49     class ConnectThread extends Thread{
50         String urlStr;
51
52         public ConnectThread(String inStr){
53             urlStr = inStr;
54         }
55         public void run(){
56             try{
57                 final String output = request(urlStr);
58                 handler.post(new Runnable() {
59                     @Override
60                     public void run() {
61                         txtMsg.setText(output);
62                     }
63                 });
64             } catch (Exception ex){
65                 ex.printStackTrace();
66             }
67         }
68         private String request(String urlStr){
```

7. 코딩

MainActivity.java x

```
66  
67  
68 @  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
  
private String request(String urlStr){  
    StringBuilder output = new StringBuilder();  
    try{  
        URL url = new URL(urlStr);  
        HttpURLConnection conn = (HttpURLConnection)url.openConnection();  
        if(conn != null){  
            conn.setConnectTimeout(10000);  
            conn.setRequestMethod("GET");  
            conn.setDoInput(true);  
            conn.setDoOutput(true);  
            int resCode = conn.getResponseCode();  
            if(resCode == HttpURLConnection.HTTP_OK){  
                BufferedReader reader  
                    = new BufferedReader(new InputStreamReader(conn.getInputStream()))  
                String line = null;  
                while (true){  
                    line = reader.readLine();  
                    if(line == null){  
                        break;  
                    }  
                    output.append(line + "\n");  
                }  
            }  
        }  
    }  
}
```

8. 코딩

```
MainActivity.java x
82 String line = null;
83 while (true){
84     line = reader.readLine();
85     if(line == null){
86         break;
87     }
88     output.append(line + "\n");
89 }
90 reader.close();
91 conn.disconnect();
92 }
93 }
94 }catch (Exception ex){
95     Log.e("SampleHTTP", "Exception in processing response.", ex);
96 }
97 return output.toString();
98 }
99 }
100 }
```

9. 코딩

AndroidManifest.xml

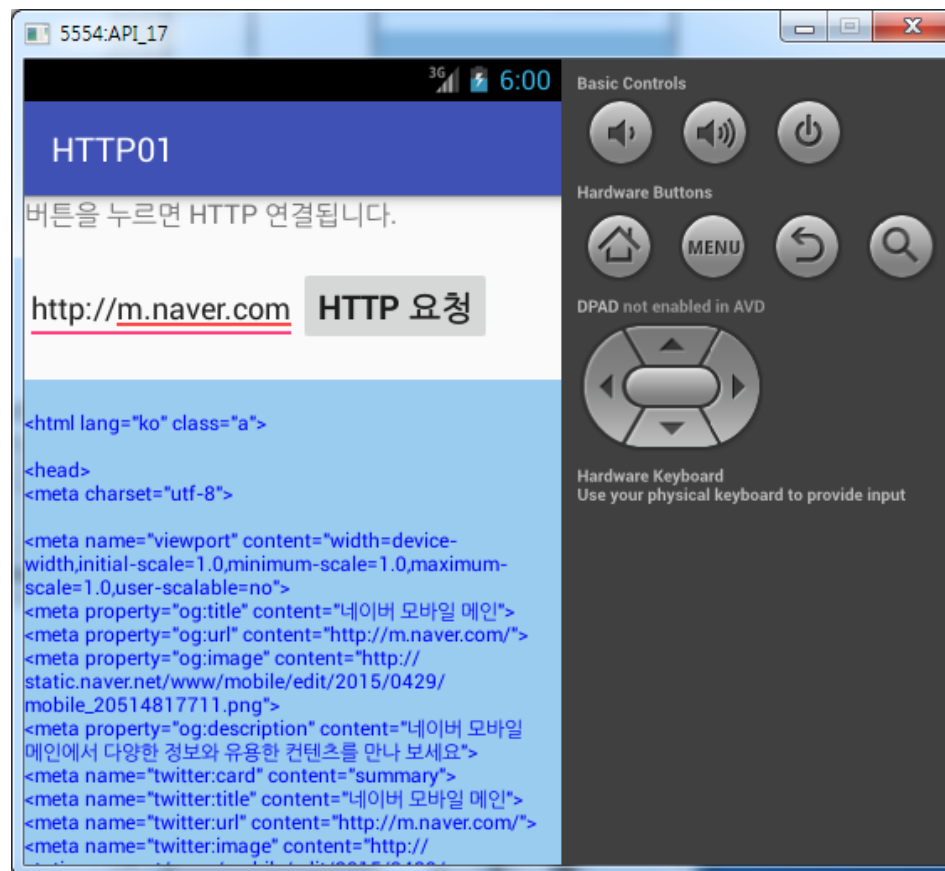
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.administrator.http01">

    <uses-permission android:name="android.permission.INTERNET" />

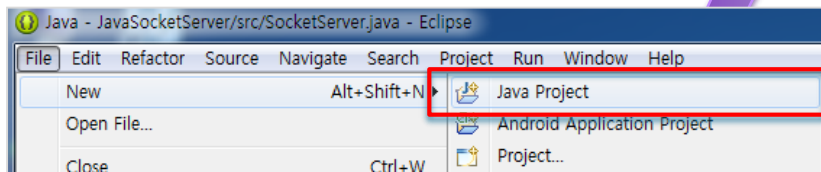
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="HTTP01"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

10. 코딩

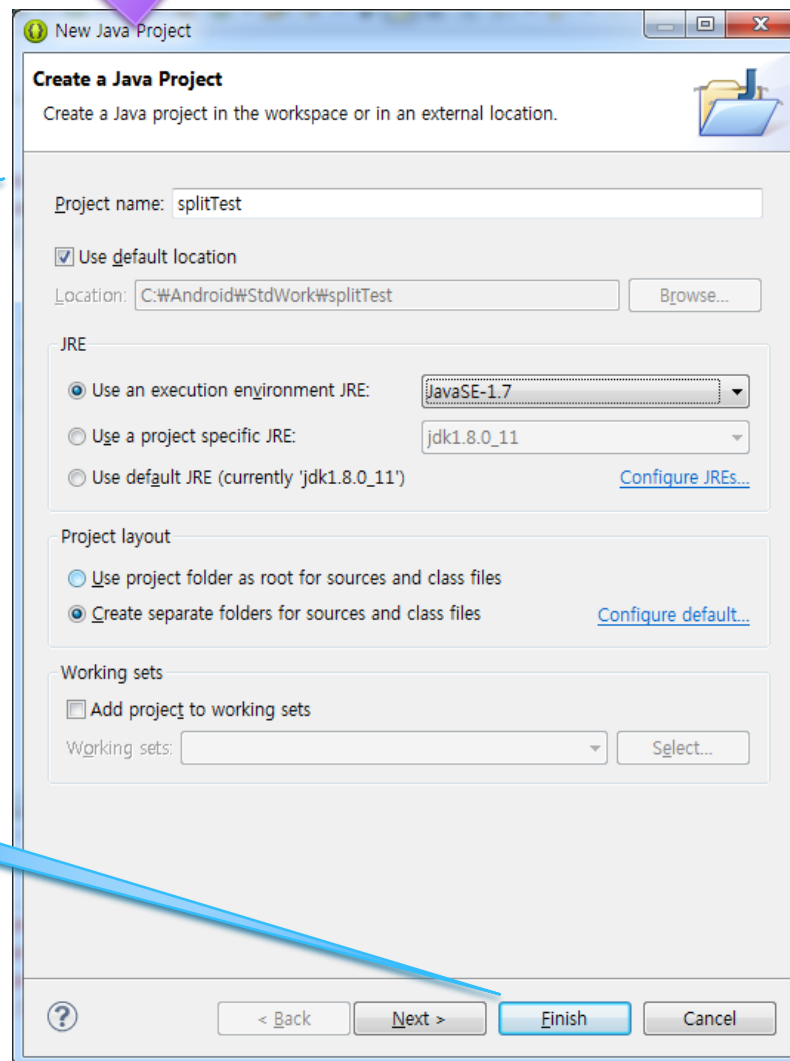


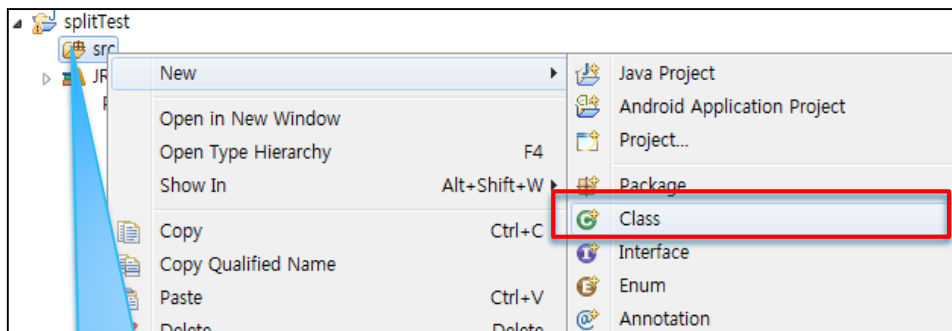
이클립스 자바 프로젝트 생성



자바 프로젝트 생성
"splitTest"

클릭

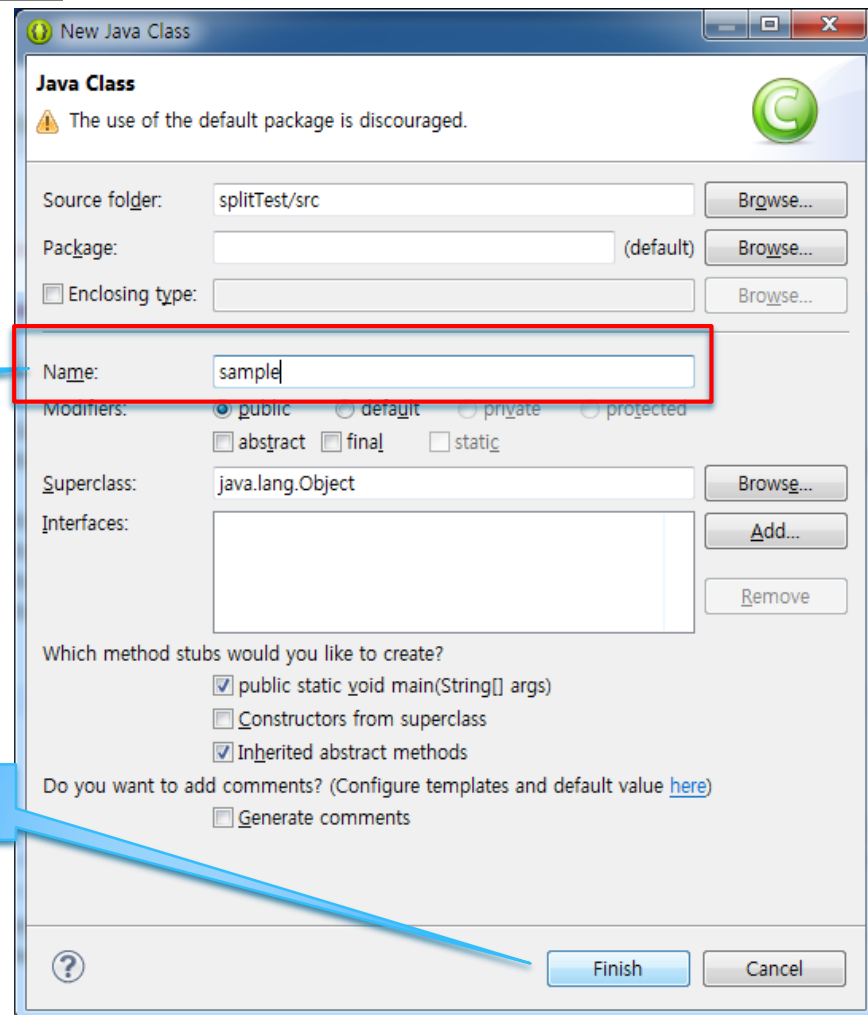




'src' 클릭 후 오른쪽
마우스 클릭

파일명 입력
"sample"

클릭



```
sample.java ✕  
  
public class sample {  
    public static void main(String[] args) {  
        String test = "One and Two and Three and Four";  
  
        String[] array = test.split("and");  
        System.out.println(" =>" + array[0] + array[1]+ array[2]+array[3]);  
  
        System.out.println(array[0].length());  
        System.out.println(array[1].length());  
        System.out.println(array[2].length());  
        System.out.println(array[3].length());  
    }  
}
```

코딩

```
=>One  Two  Three  Four  
4  
5  
7  
5
```

```
3  
3  
5  
4
```

글자만 추출하기 위해서는
어떻게 할 것인가?

```
sample.java ✖  
  
public class sample {  
    public static void main(String[] args) {  
        String test = "One and Two and Three and Four";  
  
        String[] array = test.split(" and ");  
        System.out.println(" =>" + array[0] + array[1] + array[2] + array[3]);  
  
        System.out.println(array[0].length());  
        System.out.println(array[1].length());  
        System.out.println(array[2].length());  
        System.out.println(array[3].length());  
    }  
}
```

공백 " "까지 포함하여 추출

=>OneTwoThreeFour

3

3

5

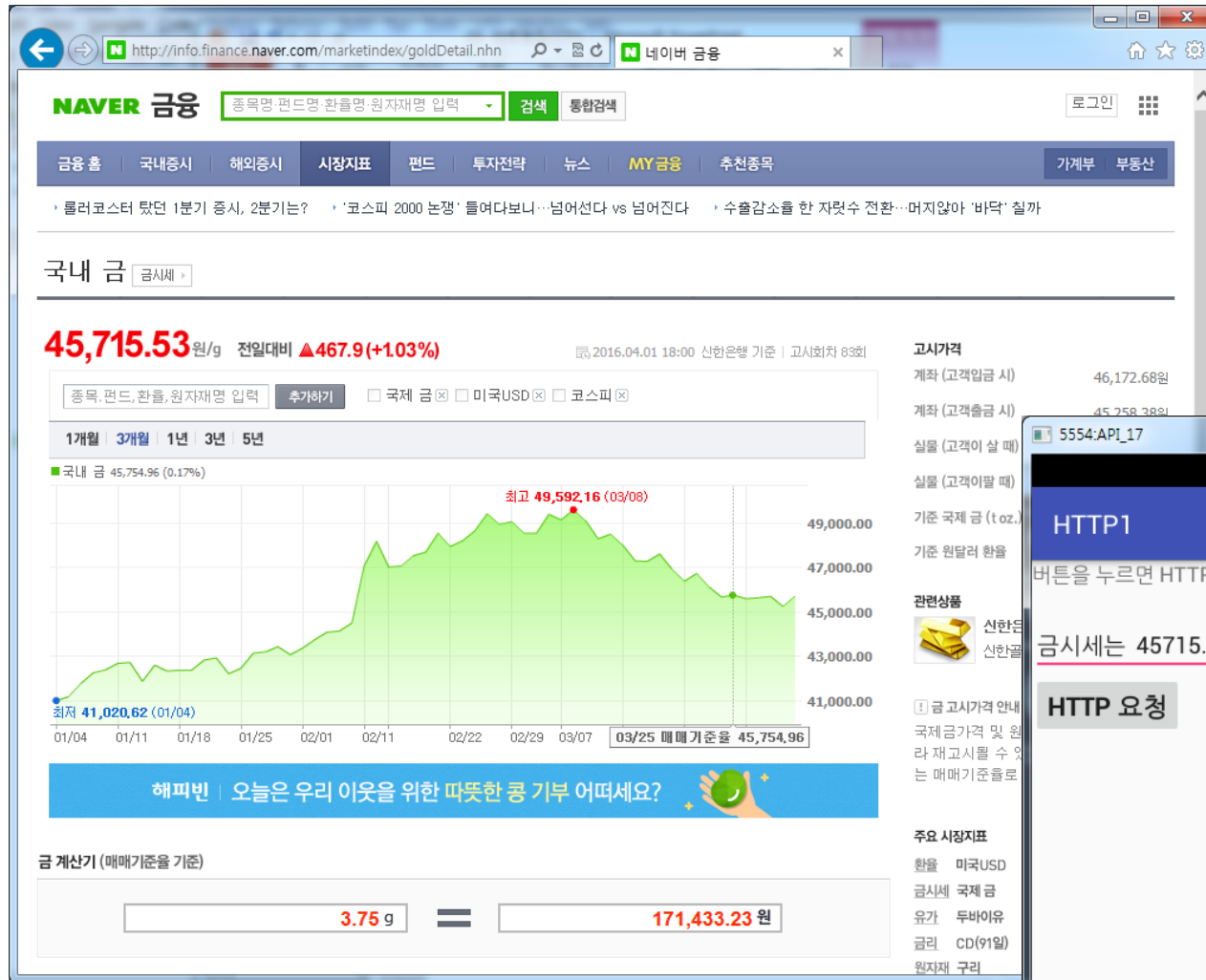
4

sample.java

```
public class sample {  
  
    public static void main(String[] args) {  
        String test = "문자열 검색을 위한 예제 입니다.";  
  
        if(test.contains("예제")){  
            System.out.println("문자 있어요.");  
        }else{  
            System.out.println("문자 없어요.");  
        }  
  
        if(test.indexOf("예제")>-1){  
            System.out.println("문자 있어요.");  
        }else{  
            System.out.println("문자 없어요.");  
        }  
  
        if(test.matches(".*예제.*")){  
            System.out.println("문자 있어요.");  
        }else{  
            System.out.println("문자 없어요.");  
        }  
    }  
}
```

문자 있어요.
문자 있어요.
문자 있어요.

http://info.finance.naver.com/marketindex/goldDetail.nhn

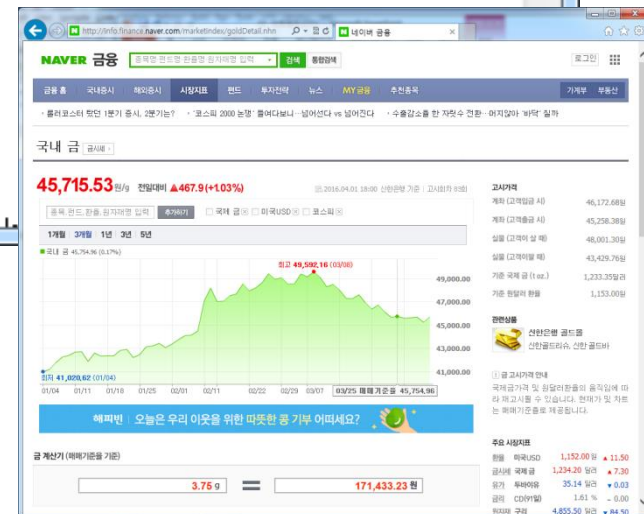


```

http://info.finance.naver.com/marketindex/goldDetail.nhn - 원본

파일(F) 편집(E) 형식(O)
191 <script type="text/javascript" language="javascript" charset="euckr"
src="/js/jindoComponent/jindo.WatchInput.1.0.1.js?20160218161951"></script>
192 <script type="text/javascript" language="javascript" charset="euckr"
src="/js/jindoComponent/jindo.Timer.1.0.1.js?20160218161951"></script>
193 <script type="text/javascript" language="javascript" charset="euckr"
src="/js/jindoComponent/jindo.Formatter.1.0.1.js?20160218161951"></script>
194 <script type="text/javascript" language="javascript" charset="euckr"
src="/js/jindoComponent/jindo.NumberFormatter.1.0.1.js?20160218161951"></script>
195 <script type="text/javascript" language="javascript" src="/js/ichart.js?20160218161951"></script>
196 <script type="text/javascript" language="javascript">
197
198 var calcInput, calcOutput;
199 var DEAL_VAL = 45715.53;
200
201 function calcGoldSise(event) {
202     if (event.keyCode != 13 && event.keyCode != 27 && event.keyCode != 37 && event.keyCode != 38 &&
event.keyCode != 110 && event.keyCode != 229) {
203         var regExp = new RegExp("[^0-9#.#,]");
204         var sInputValue = jindo.$("calcInput").value;
205
206         if(regExp.test(sInputValue)){
207             alert("숫자, 소수점(.)만 입력하실 수 있습니다.");
208             hide("#calcOutput");

```



```
17 public class MainActivity extends AppCompatActivity {
```

```
18
```

```
19     EditText input01;
```

```
20     TextView txtMsg;
```

```
21     Button requestBtn;
```

```
22
```

```
23     // public static String defaultUrl = "http://m.naver.com";
```

```
24     public static String defaultUrl = "http://info.finance.naver.com/marketindex/goldDetail.nhn"
```

```
25
```

```
26     Handler handler = new Handler();
```

수정

```
67 @
68 private String request(String urlStr){
69     StringBuilder output = new StringBuilder();
70     try{
71         URL url = new URL(urlStr);
72         HttpURLConnection conn = (HttpURLConnection)url.openConnection();
73         if(conn != null){
74             conn.setConnectTimeout(10000);
75             conn.setRequestMethod("GET");
76             conn.setDoInput(true);
77             conn.setDoOutput(true);
78             int resCode = conn.getResponseCode();
79             if(resCode == HttpURLConnection.HTTP_OK){
80                 BufferedReader reader = new BufferedReader
81                     (new InputStreamReader(conn.getInputStream()));
82                 String line = null;
83                 while (true){
84                     line = reader.readLine();
85                     if(line == null){
86                         break;
87                     }
88                     if(line.contains("DEAL_VAL =")){
89                         String[] outData = line.split("DEAL_VAL =");
90                         String[] outWon = outData[1].split(";");
91                         output.append("금시세는 " + outWon[0] + "원");
92                         break;
93                     }
94                 }
95             }
96             reader.close();
97         }
98     }
99 }
```

수정 및 추가

