

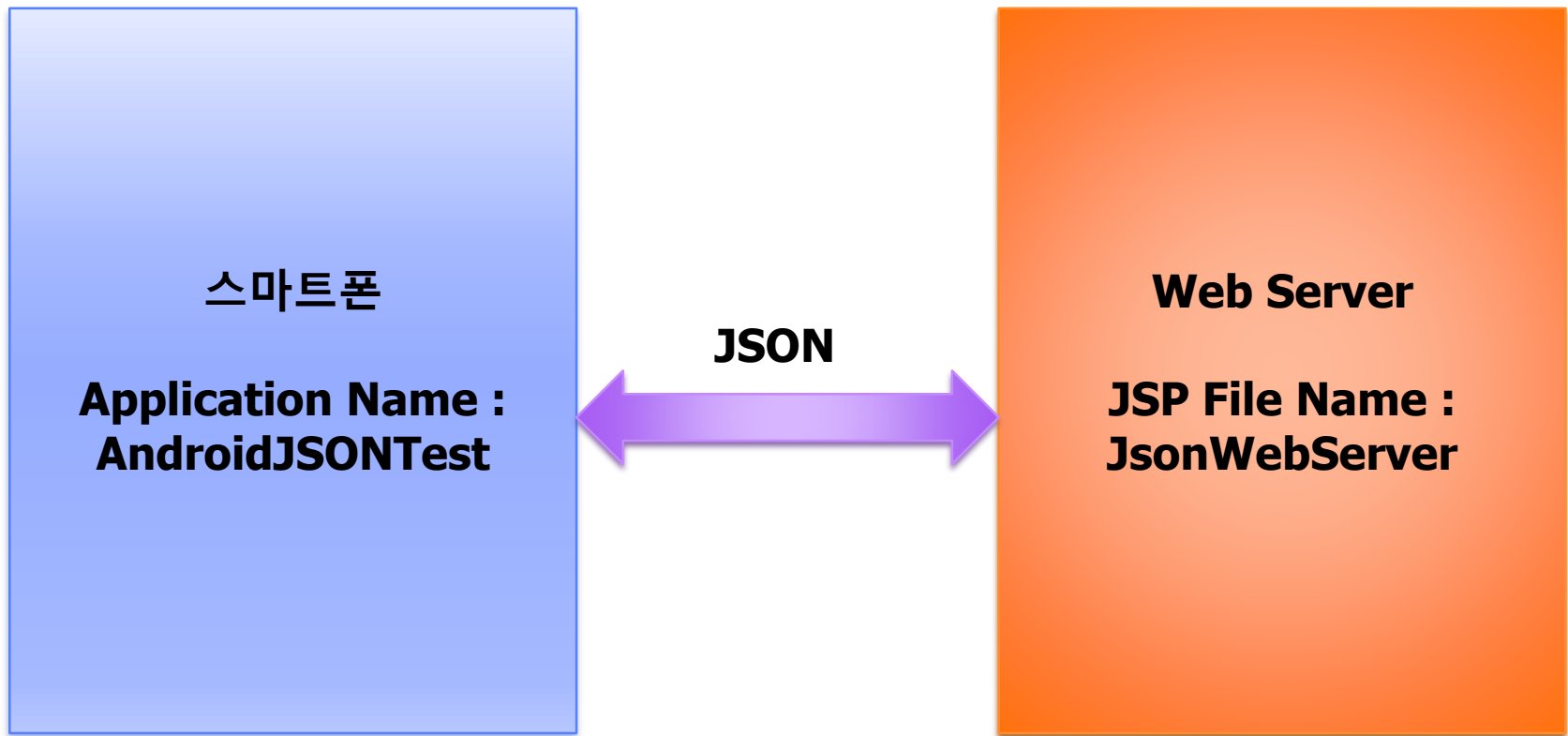


JSON을 활용한 ANDROID와 WEB SERVER

안드로이드의 JSON을 이용하여 웹 서버에 데이터를
주고 받기



JSON: Smart Phone과 JSP



프로젝트 만들기



Application name: **AndroidJSONTest**

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

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

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**

○ 준비사항:

- **"json_simple-1.1.jar"** 파일 다운로드



```
activity_main.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:orientation="vertical">
6
7      <LinearLayout
8          android:layout_width="wrap_content"
9          android:layout_height="wrap_content"
10         android:orientation="horizontal">
11
12         <TextView
13             android:layout_width="wrap_content"
14             android:layout_height="wrap_content"
15             android:text="보내는 메시지 : " />
16
17         <EditText
18             android:id="@+id/et_message"
19             android:layout_width="wrap_content"
20             android:layout_height="wrap_content"
21             android:hint="보내는 메시지 입력" />
22     </LinearLayout>
23
```



```
activity_main.xml x
24 <Button
25     android:id="@+id/btn_sendData"
26     android:layout_width="wrap_content"
27     android:layout_height="wrap_content"
28     android:layout_margin="20dp"
29     android:text="Send" />
30
31 <LinearLayout
32     android:layout_width="match_parent"
33     android:layout_height="wrap_content"
34     android:orientation="horizontal">
35
36     <TextView
37         android:layout_width="wrap_content"
38         android:layout_height="wrap_content"
39         android:text="받은 메시지 : " />
40
41     <TextView
42         android:id="@+id/tv_recvData"
43         android:layout_width="wrap_content"
44         android:layout_height="wrap_content"
45         android:text="받은 메시지가 없습니다." />
46 </LinearLayout>
47 </LinearLayout>
```



```
MainActivity.java x
1  package com.example.admin.androidjsontest;
2
3  import android.os.Build;
4  import android.os.Bundle;
5  import android.os.StrictMode;
6  import android.support.v7.app.AppCompatActivity;
7  import android.util.Log;
8  import android.view.View;
9  import android.view.View.OnClickListener;
10 import android.widget.Button;
11 import android.widget.EditText;
12 import android.widget.TextView;
13 import android.widget.Toast;
14
15 import org.apache.http.HttpResponse;
16 import org.apache.http.client.methods.HttpPost;
17 import org.apache.http.impl.client.DefaultHttpClient;
18 import org.apache.http.params.HttpConnectionParams;
19 import org.apache.http.params.HttpParams;
20 import org.json.JSONArray;
21 import org.json.JSONException;
22 import org.json.JSONObject;
23
24 import java.io.BufferedReader;
25 import java.io.InputStreamReader;
26
```

Import 확인

C MainActivity.java x

코딩

```
26
27 public class MainActivity extends AppCompatActivity {
28
29     private EditText etMessage;
30     private Button btnSend;
31     private TextView tvRecvData;
32
33     @Override
34     protected void onCreate(Bundle savedInstanceState) {
35         super.onCreate(savedInstanceState);
36         setContentView(R.layout.activity_main);
37
38         StrictMode.setThreadPolicy(new StrictMode.ThreadPolicy.Builder()
39             .detectDiskReads().detectDiskWrites().detectNetwork()
40             .penaltyLog().build());
41
42         if(Build.VERSION.SDK_INT > 9 ){
43             StrictMode.ThreadPolicy policy =
44                 new StrictMode.ThreadPolicy.Builder().permitAll().build();
45             StrictMode.setThreadPolicy(policy);
46         }
47
48         etMessage = (EditText) findViewById(R.id.et_message);
```

네트워크 스레드 관리 메소드

MainActivity.java ×

코딩

```
47
48     etMessage = (EditText) findViewById(R.id.et_message);
49     btnSend = (Button) findViewById(R.id.btn_sendData);
50     tvRecvData = (TextView) findViewById(R.id.tv_recvData);
51
52     /* Send 버튼을 눌렀을 때 서버에 데이터를 보내고 받는다 */
53     btnSend.setOnClickListener(new OnClickListener() {
54         public void onClick(View v) {
55             // 보내는 메시지를 받아옴
56             String sMessage = etMessage.getText().toString();
57             // 메시지를 서버에 보냄
58             String result = SendByHttp(sMessage);
59             // JSON 데이터 파싱
60             String[][] parsedData = jsonParserList(result);
61             Toast.makeText(getApplicationContext(),
62                 result, Toast.LENGTH_SHORT).show();
63
64             tvRecvData.setText(result); // 받은 메시지를 화면에 보여주기
65         }
66     });
67 }
68
```


MainActivity.java x

```
68
69
70 /**
71  * 서버에 데이터를 보내는 메소드
72  * @param msg
73  * @return
74  */
75 private String SendByHttp(String msg) {
76     if(msg == null)
77         msg = "";
78
79     String URL = "http://192.168.0.8:8090/jsonWebServer/webjson.jsp";
80
81     DefaultHttpClient client = new DefaultHttpClient();
82
83     try {
84         /* 체크할 id와 pwd값 서버로 전송 */
85         HttpPost post = new HttpPost(URL+"?msg="+msg);
86
87         /* 지연시간 최대 5초 */
88         HttpParams params = client.getParams();
89         HttpConnectionParams.setConnectionTimeout(params, 3000);
90         HttpConnectionParams.setSoTimeout(params, 3000);
91
92         /* 데이터 보낸 뒤 서버에서 데이터를 받아오는 과정 */
```

웹 서버 파일 경로

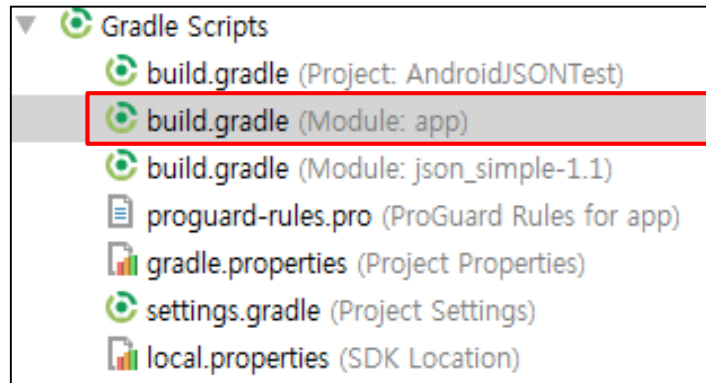
MainActivity.java ×

```
90
91      /* 데이터 보낸 뒤 서버에서 데이터를 받아오는 과정 */
92      HttpResponse response = client.execute(post);
93      BufferedReader bufrader = new BufferedReader(
94          new InputStreamReader(response.getEntity().getContent(),
95              "utf-8"));
96
97      String line = null;
98      String result = "";
99
100     while ((line = bufrader.readLine()) != null) {
101         result += line;
102     }
103
104     return result;
105 } catch (Exception e) {
106     e.printStackTrace();
107     client.getConnectionManager().shutdown();    // 연결 지연 종료
108     return "";
109 }
110 }
111
```

MainActivity.java x

```
111
112
113  /*
114   * 받은 JSON 객체를 파싱하는 메소드
115   * @param page
116   * @return
117   */
117  @
118  private String[][] jsonParserList(String pRecvServerPage) {
119
120      Log.i("서버에서 받은 전체 내용 : ", pRecvServerPage);
121
122      try {
123          JSONObject json = new JSONObject(pRecvServerPage);
124          JSONArray jArr = json.getJSONArray("List");
125
126          // 받아온 pRecvServerPage를 분석하는 부분
127          String[] jsonName = {"msg1", "msg2", "msg3"};
128          String[][] parseredData = new String[jArr.length()][jsonName.length];
129
130          for (int i = 0; i < jArr.length(); i++) {
131              json = jArr.getJSONObject(i);
132
133              for(int j = 0; j < jsonName.length; j++) {
134                  parseredData[i][j] = json.getString(jsonName[j]);
135              }
136          }
137      }
```

```
MainActivity.java x
135     }
136
137     // 분해 된 데이터를 확인하기 위한 부분
138     for(int i=0; i<parsedData.length; i++){
139         Log.i("JSON을 분석한 데이터 "+i+" : ", parsedData[i][0]);
140         Log.i("JSON을 분석한 데이터 "+i+" : ", parsedData[i][1]);
141         Log.i("JSON을 분석한 데이터 "+i+" : ", parsedData[i][2]);
142     }
143
144     return parsedData;
145 } catch (JSONException e) {
146     e.printStackTrace();
147     return null;
148 }
149 }
150 }
151 }
```



```
dependencies {  
    compile fileTree(include: ['*.jar'], dir: 'libs')  
    testCompile 'junit:junit:4.12'  
    compile 'com.android.support:appcompat-v7:23.2.0'  
    compile project(':json_simple-1.1')  
    provided 'org.jbundle.util.osgi.wrapped:org.jbundle.util.osgi.wrapped.org.apache.http.client:4.1.2'  
}
```

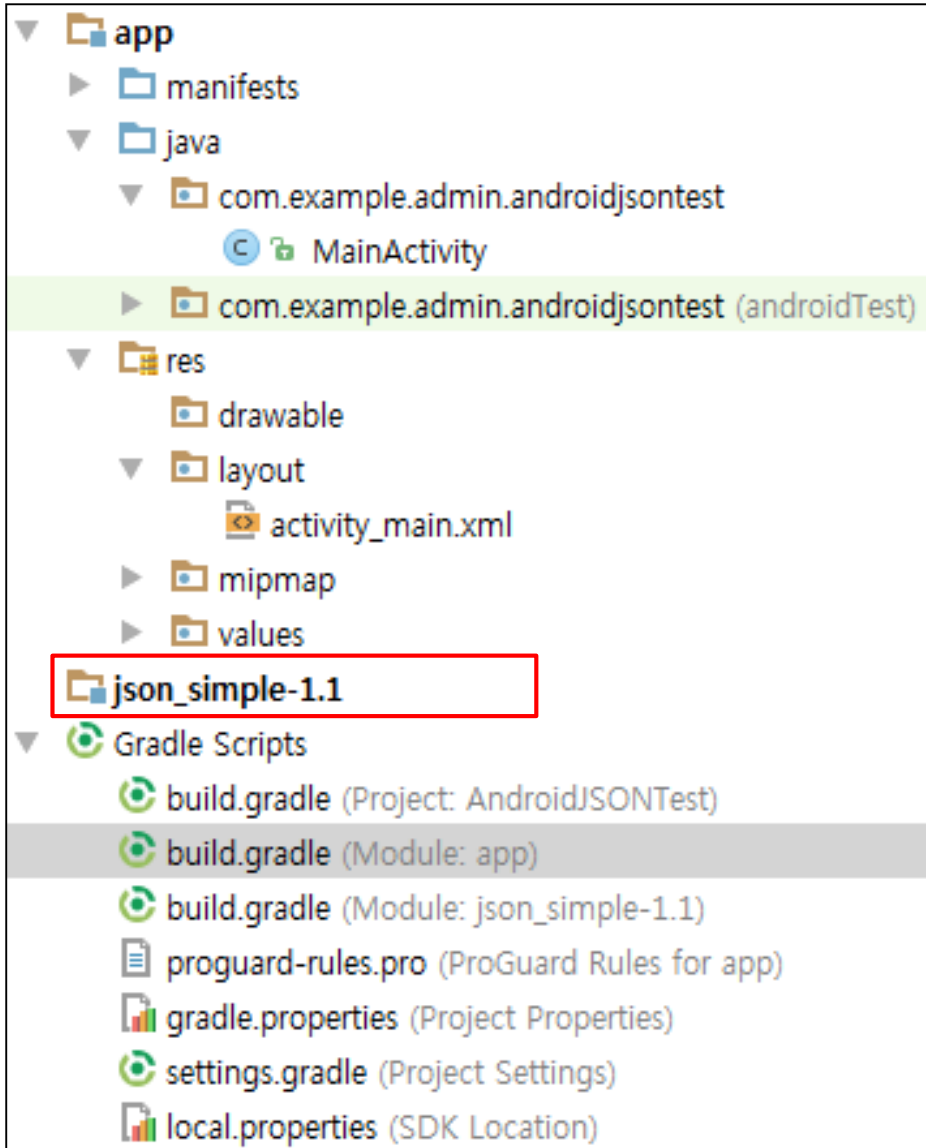
provided

'org.jbundle.util.osgi.wrapped:org.jbundle.util.osgi.wrapped.org.apache.http.client:4.1.2'

```

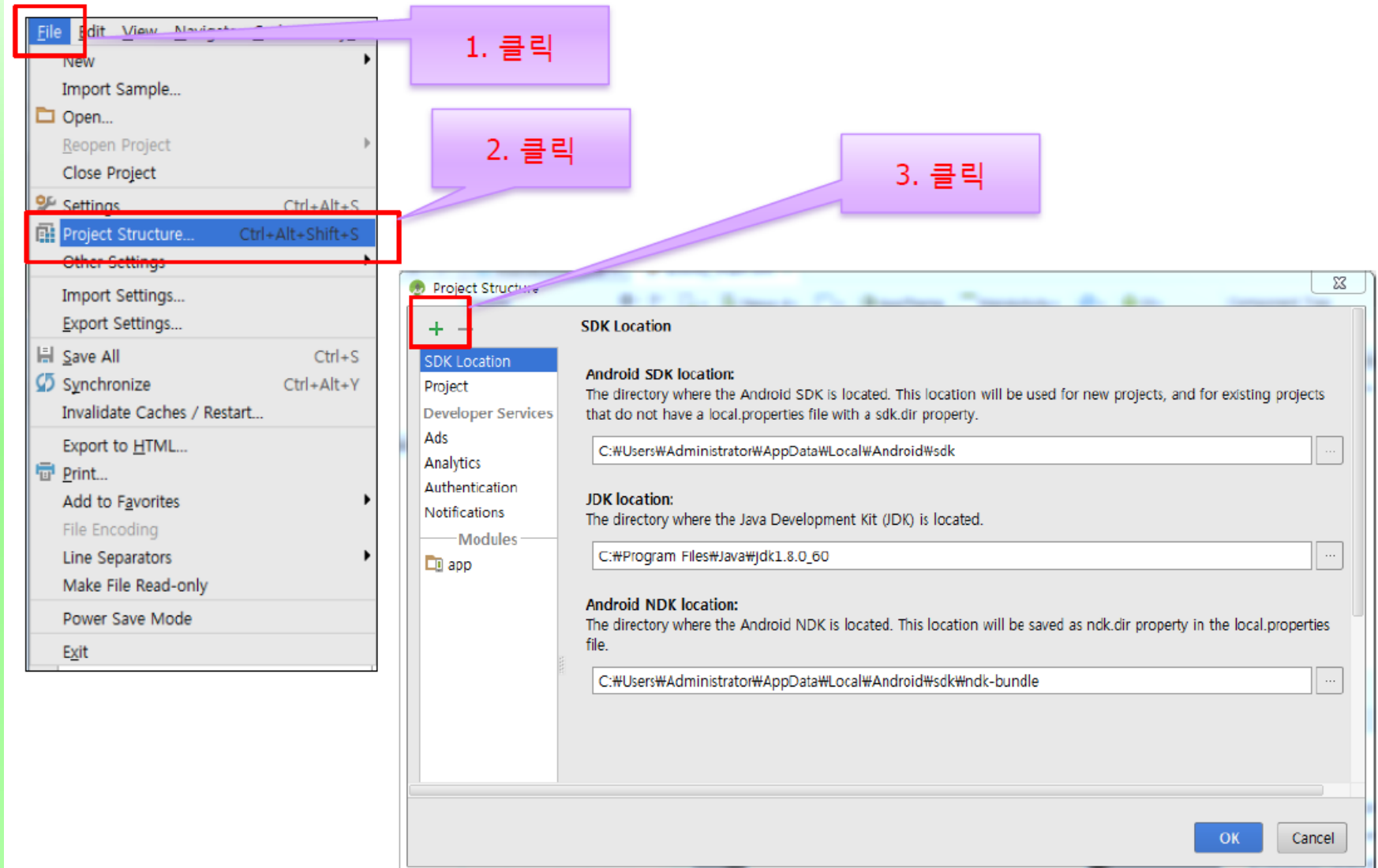
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.example.admin.androidjsontest">
4
5      <uses-permission android:name="android.permission.INTERNET" />
6
7      <application
8          android:allowBackup="true"
9          android:icon="@mipmap/ic_launcher"
10         android:label="AndroidJSONTest"
11         android:supportsRtl="true"
12         android:theme="@style/AppTheme">
13         <activity android:name=".MainActivity">
14             <intent-filter>
15                 <action android:name="android.intent.action.MAIN" />
16
17                 <category android:name="android.intent.category.LAUNCHER" />
18             </intent-filter>
19         </activity>
20     </application>
21
22 </manifest>

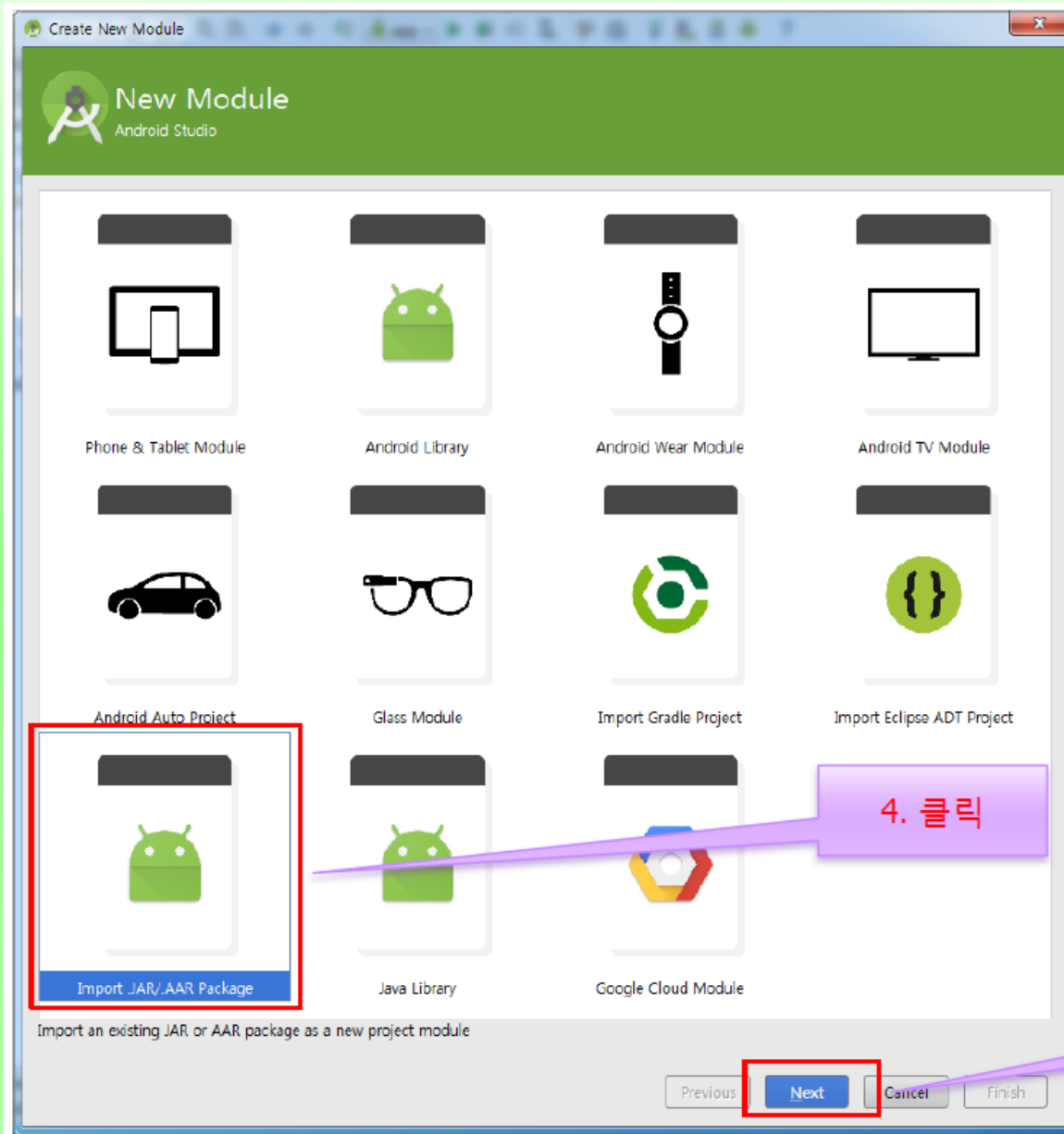
```

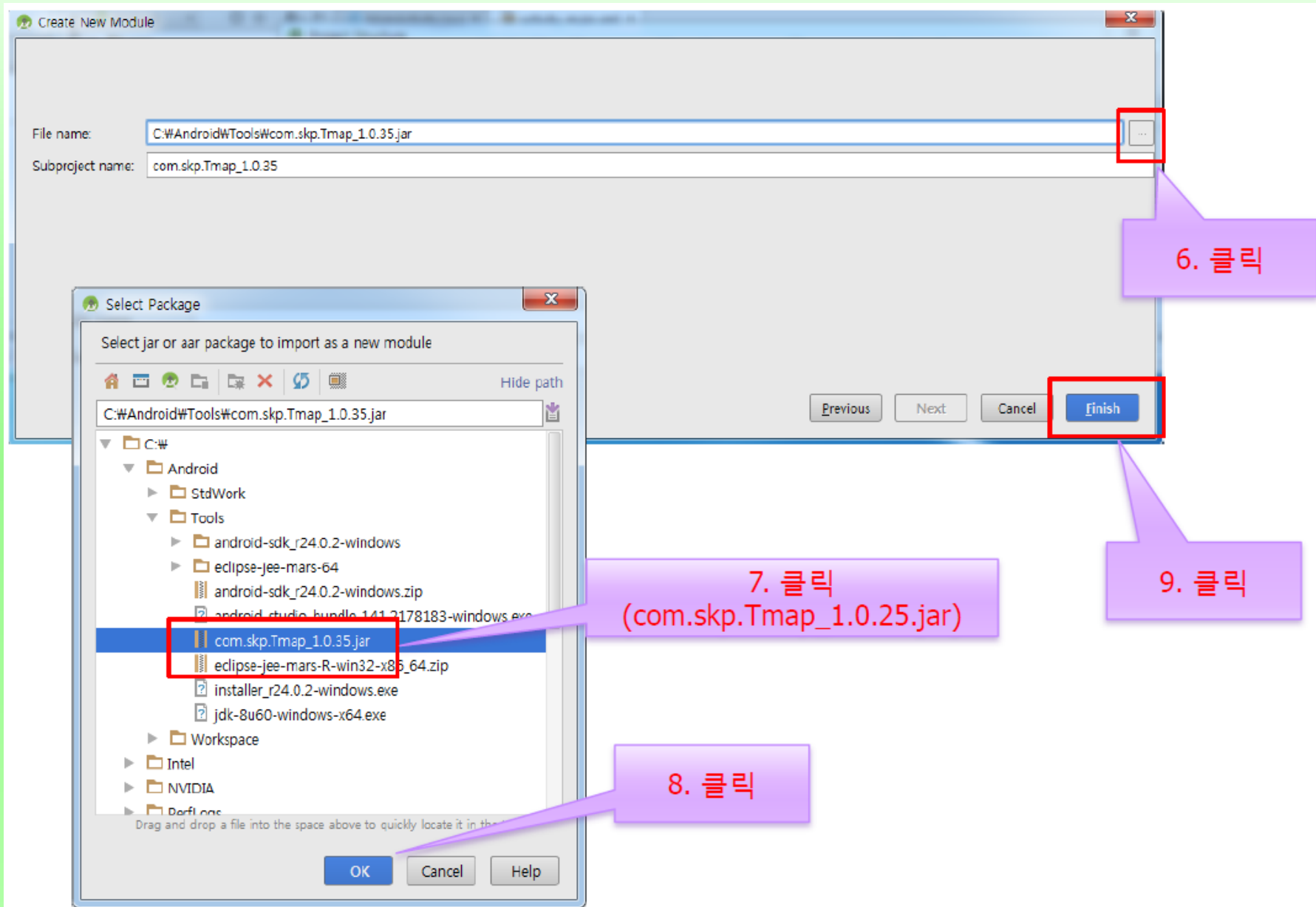


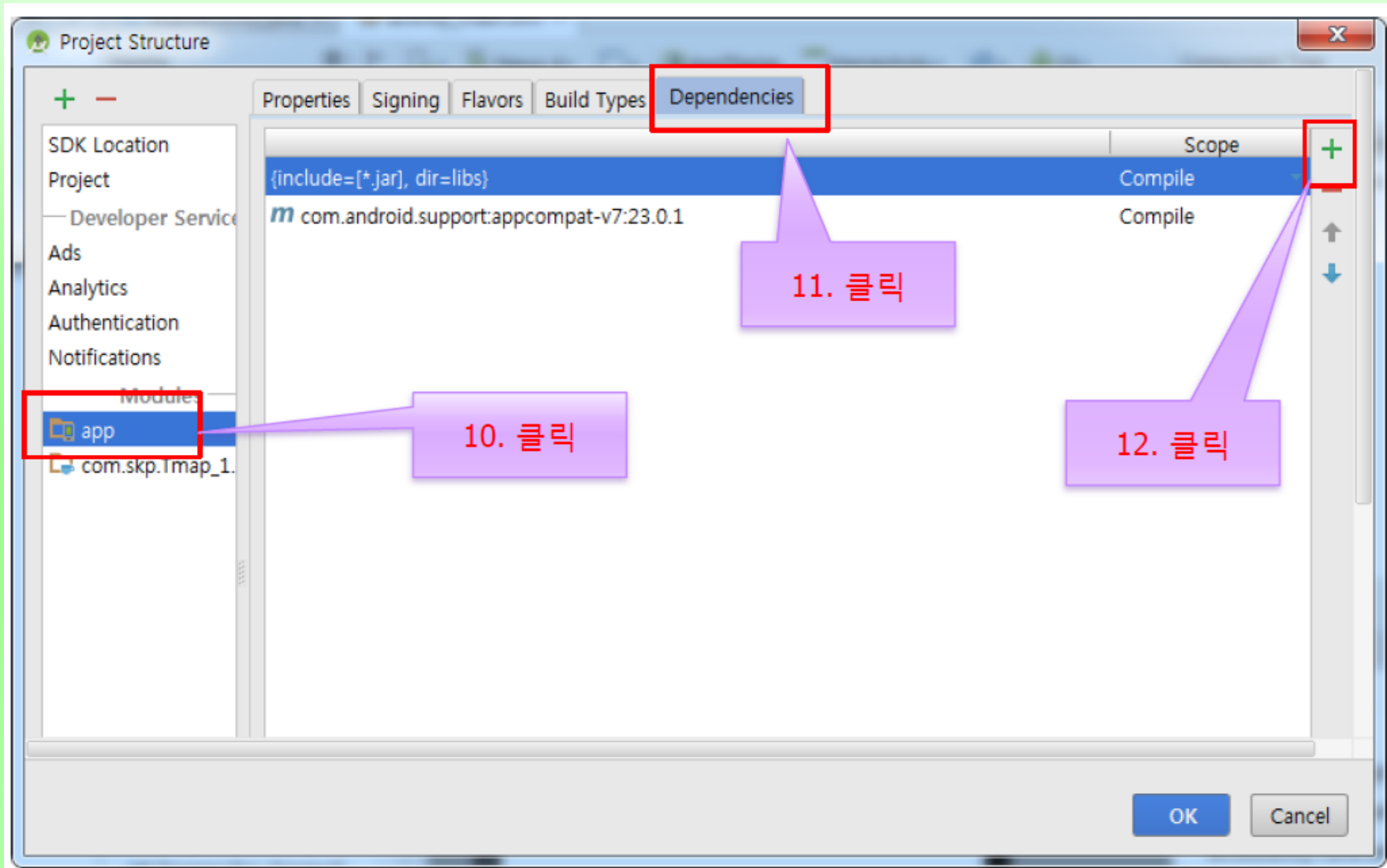
[참고] Android Studio에 **JAR** 파일 추가

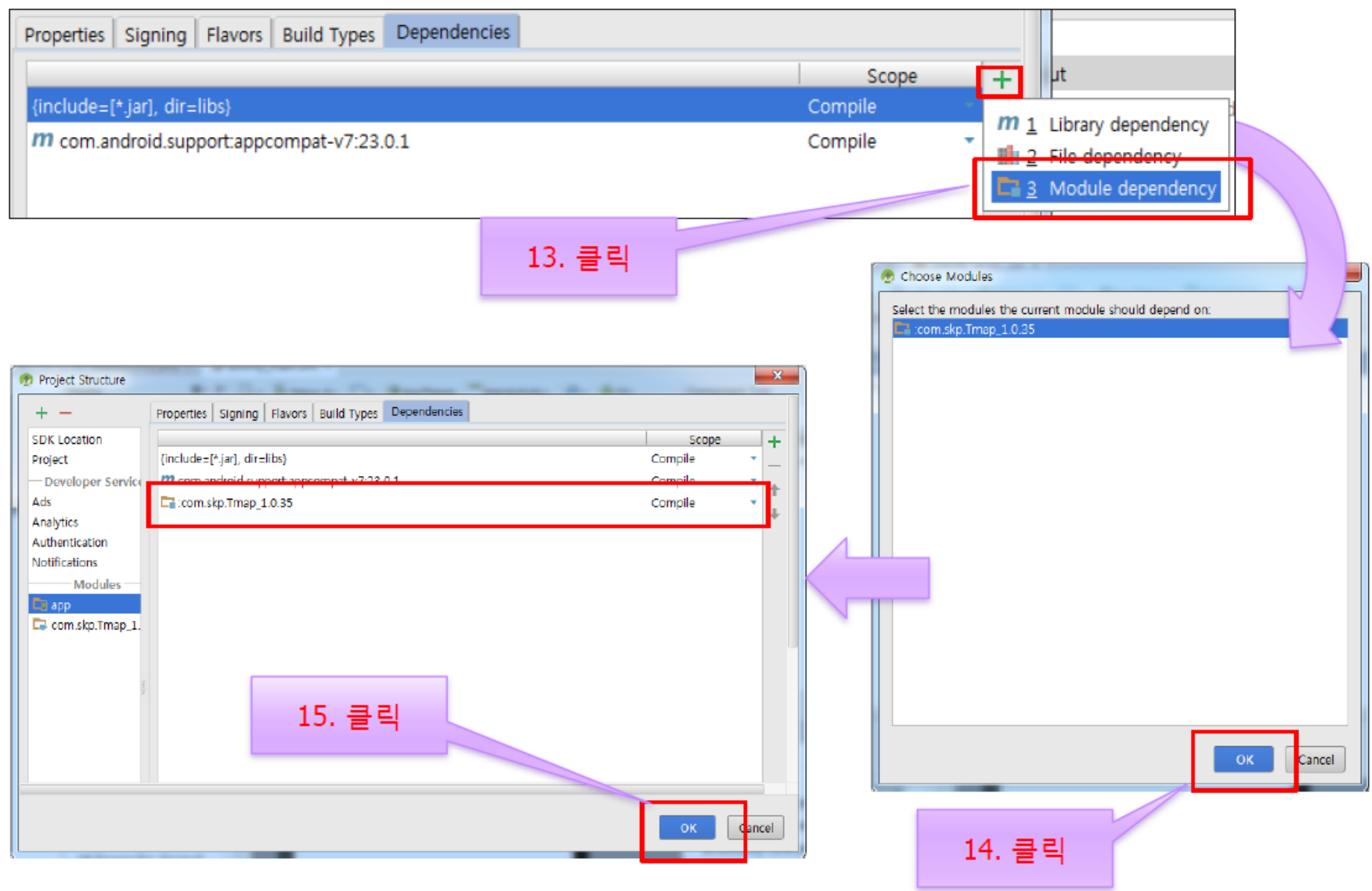
16



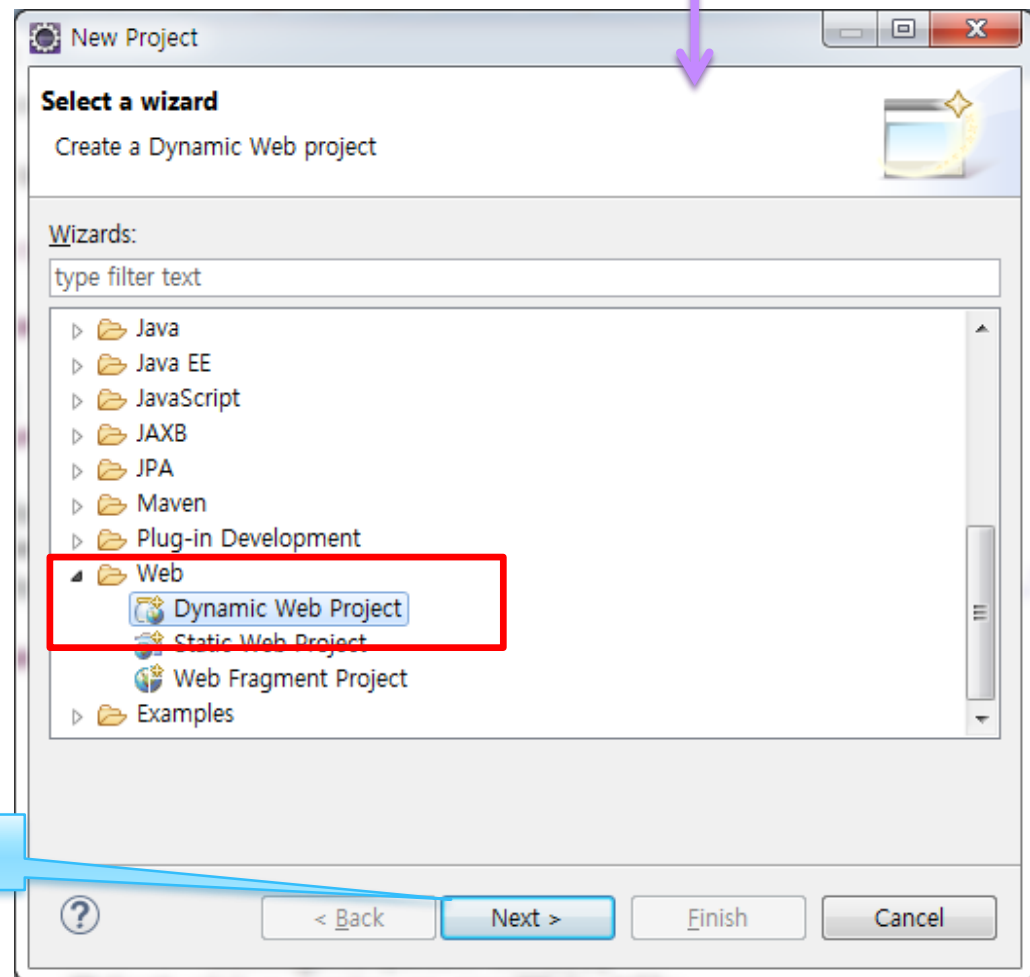
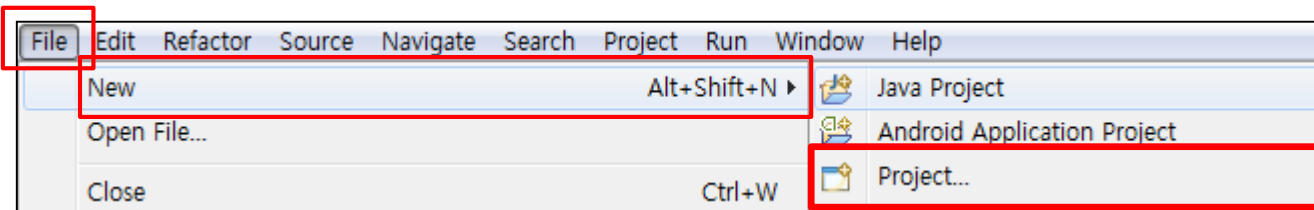


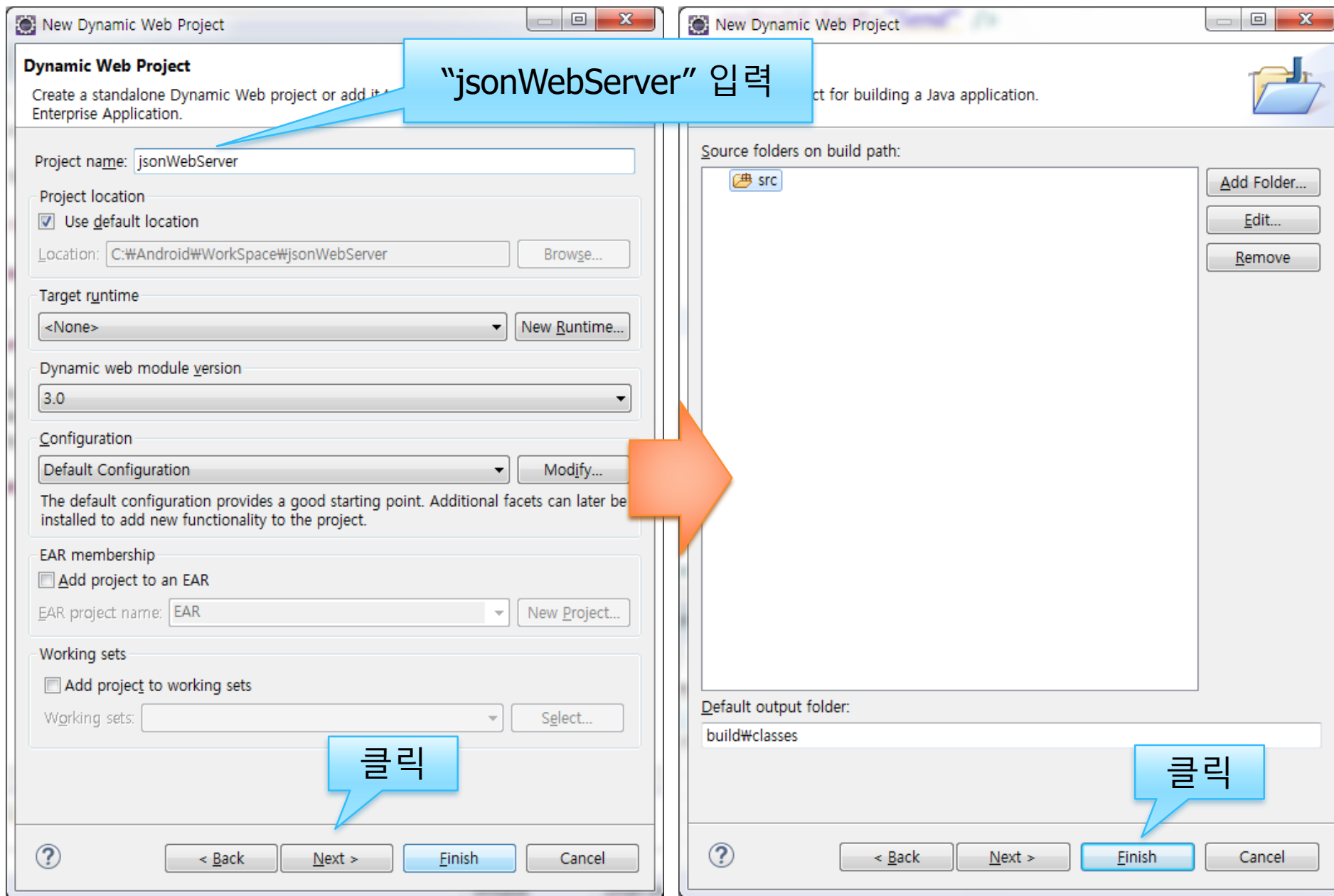


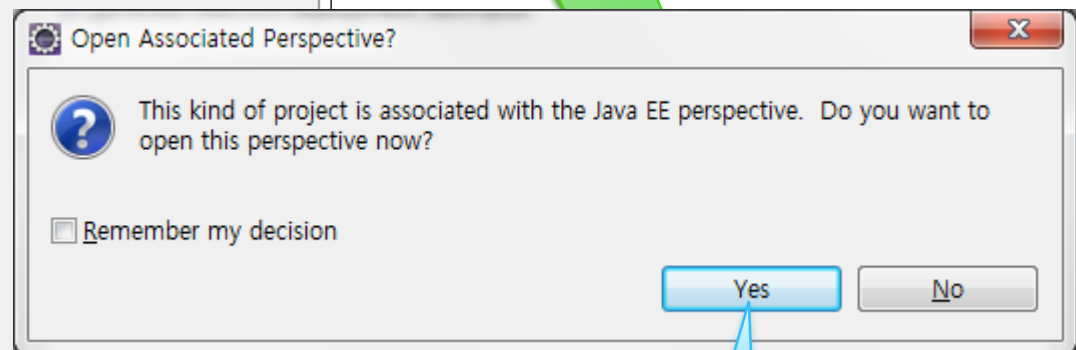
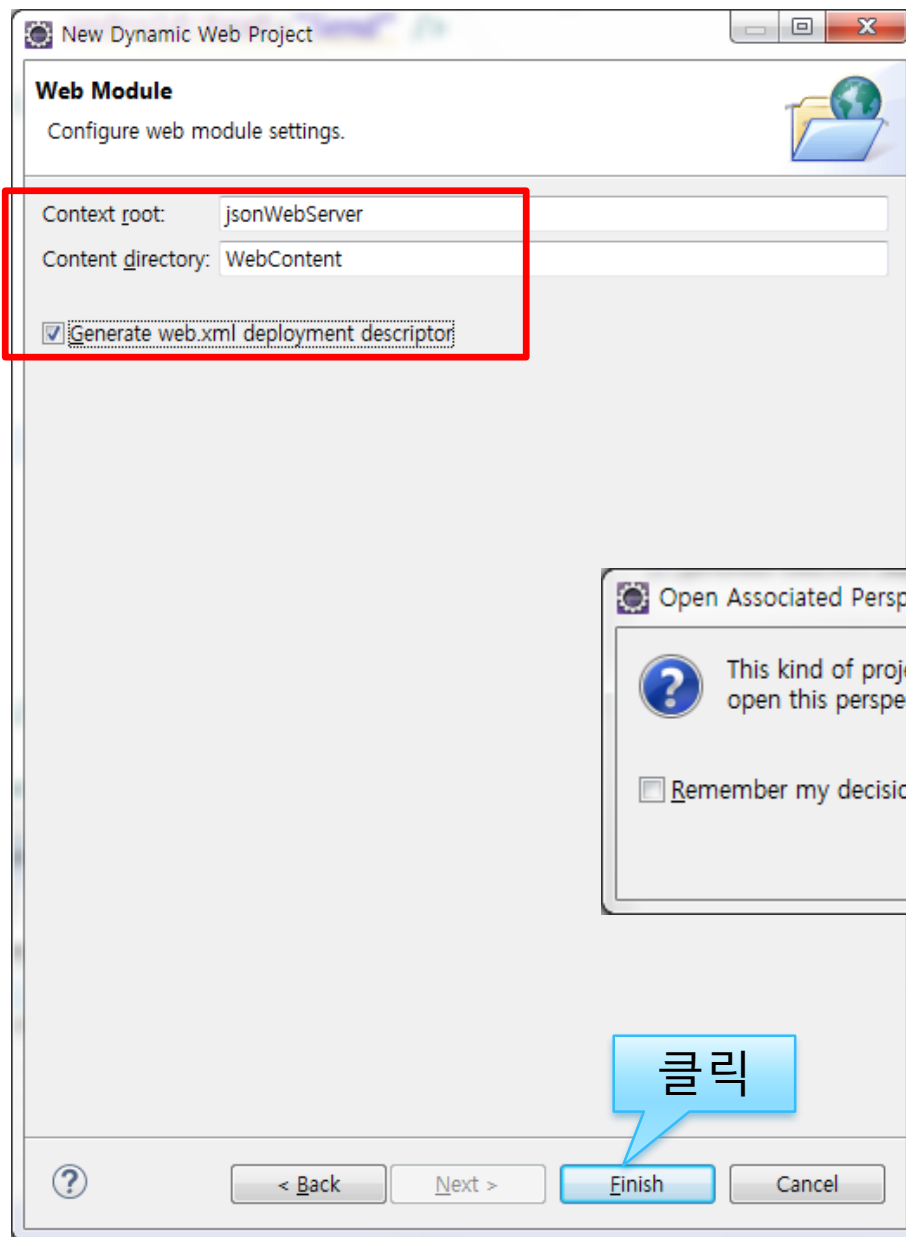


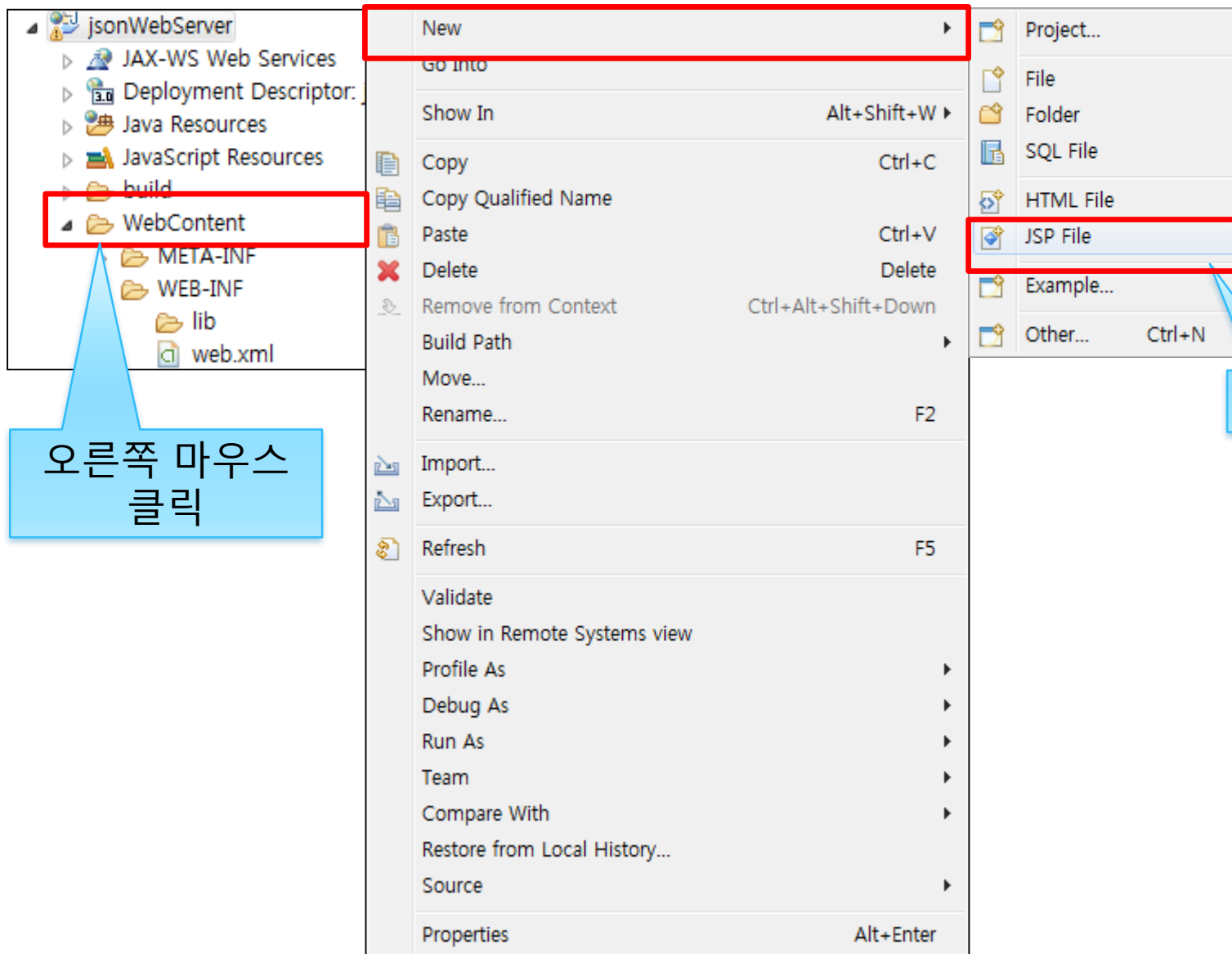


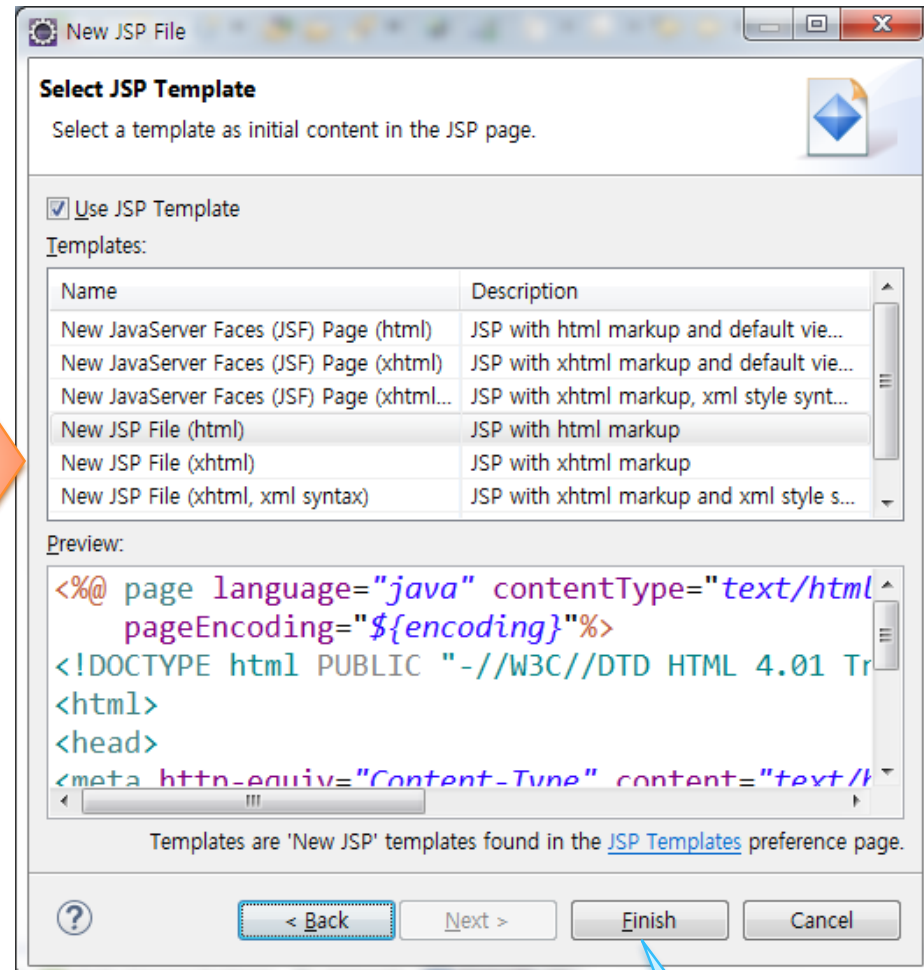
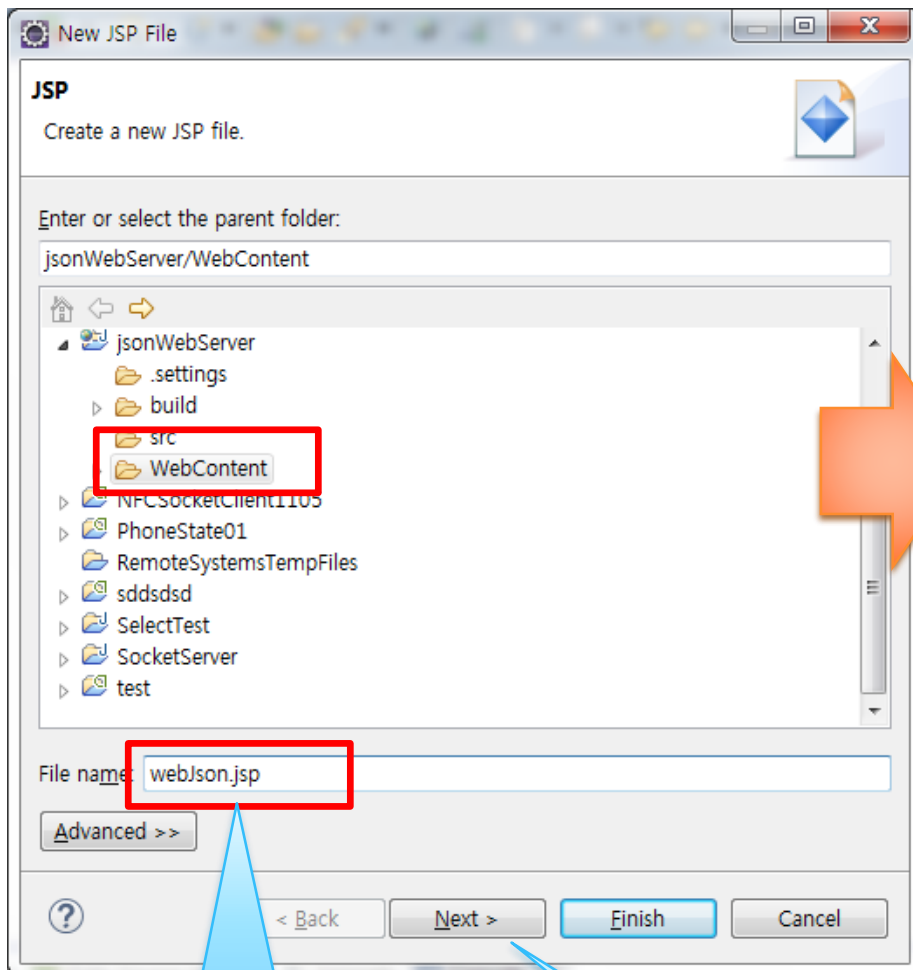
서버 프로그래밍(JSP 프로그래밍)





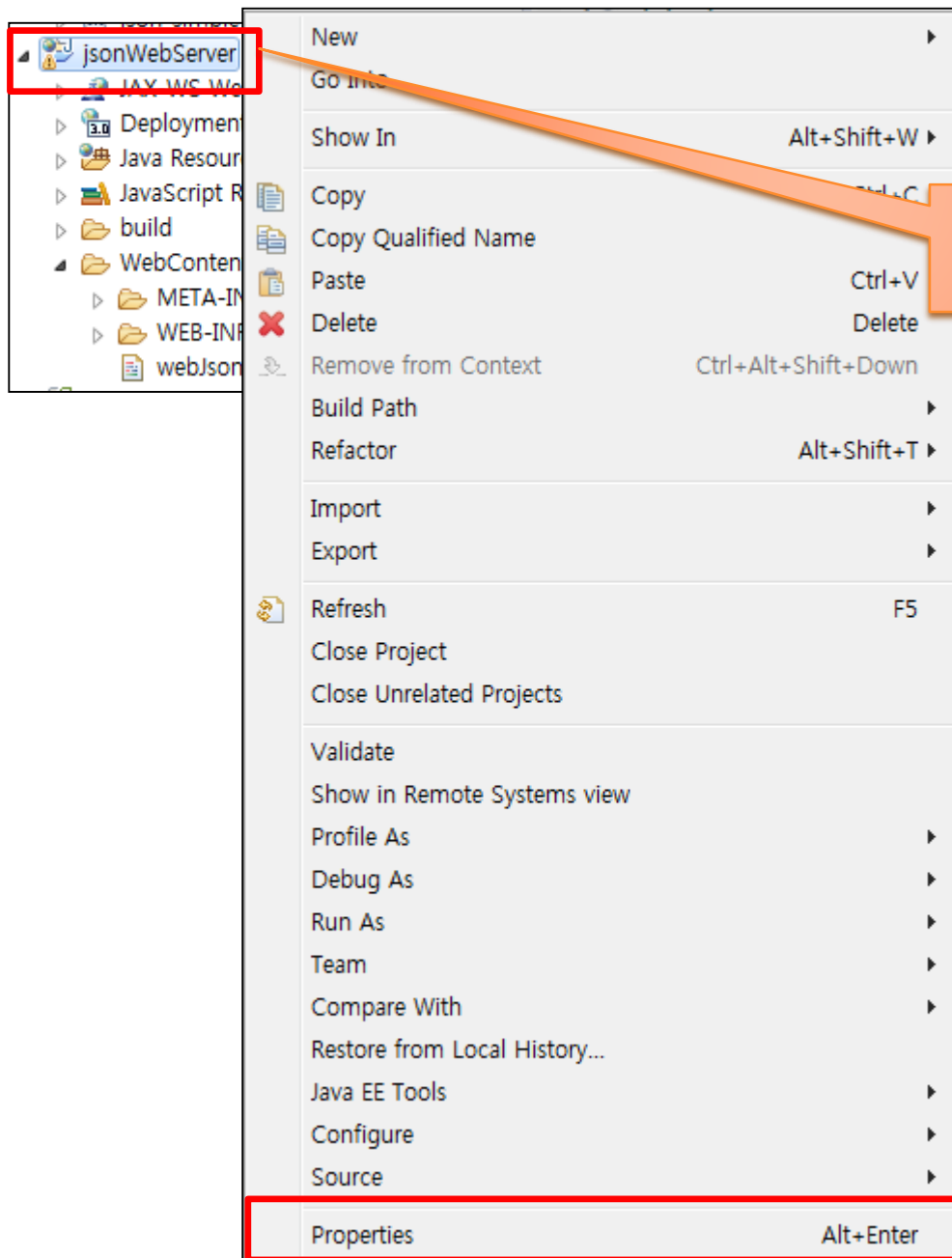






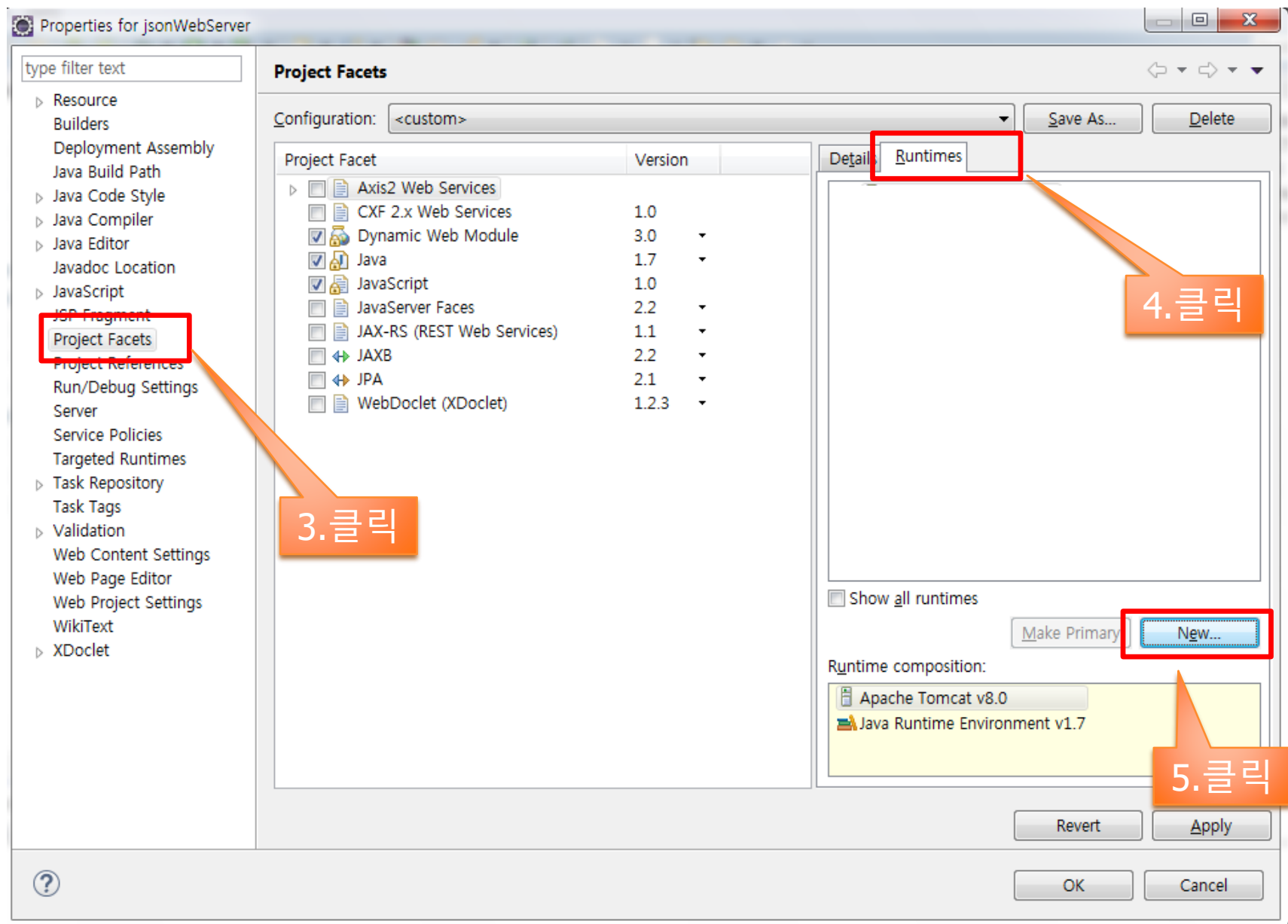
```
webJson.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8" import="org.json.simple.*" import="java.sql.*"
3   import="java.util.*"%>
4
5 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR
6 <html>
7 <head>
8 <meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
9 <title>Insert title here</title>
10 </head>
11 <body>
12
13 </body>
14 </html>
```

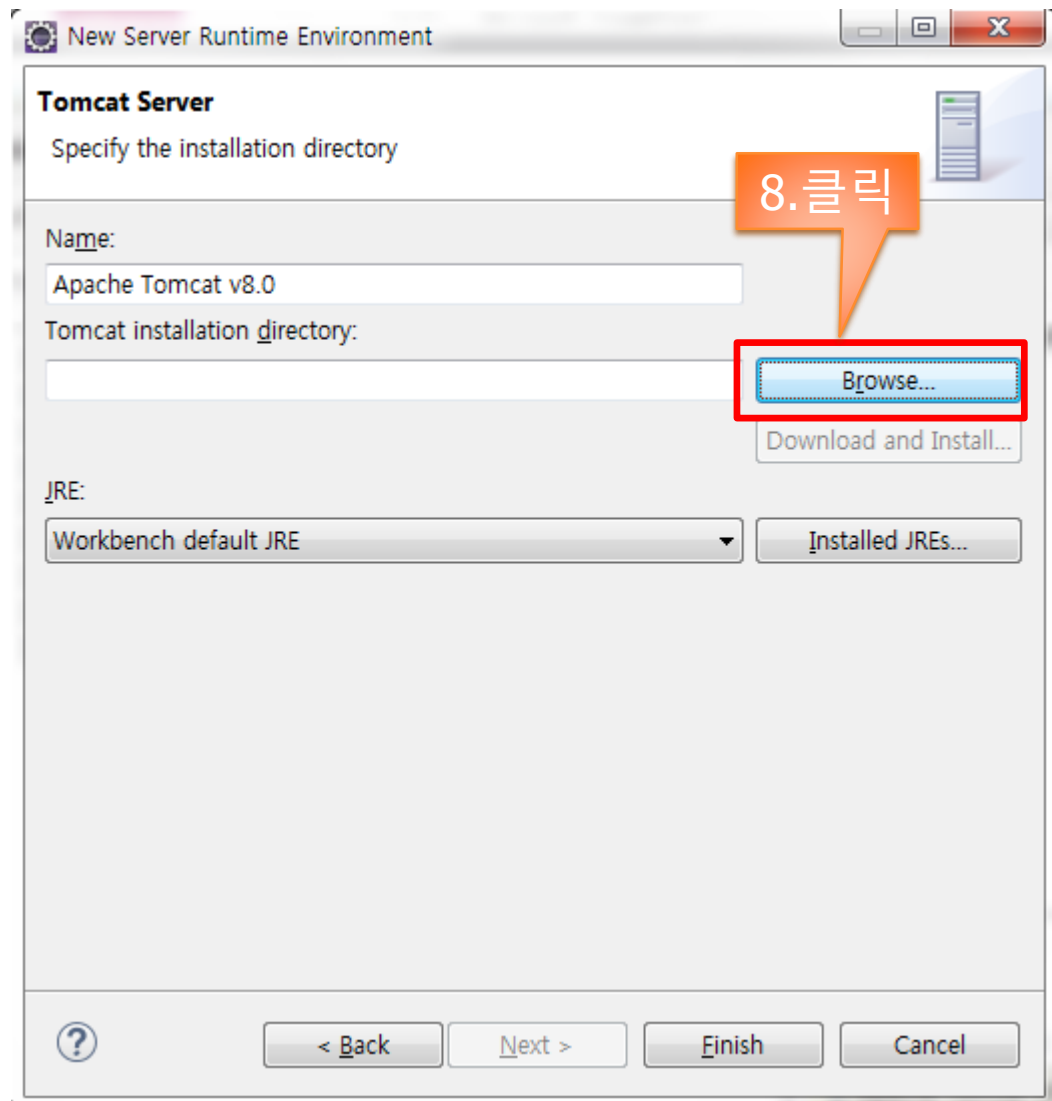
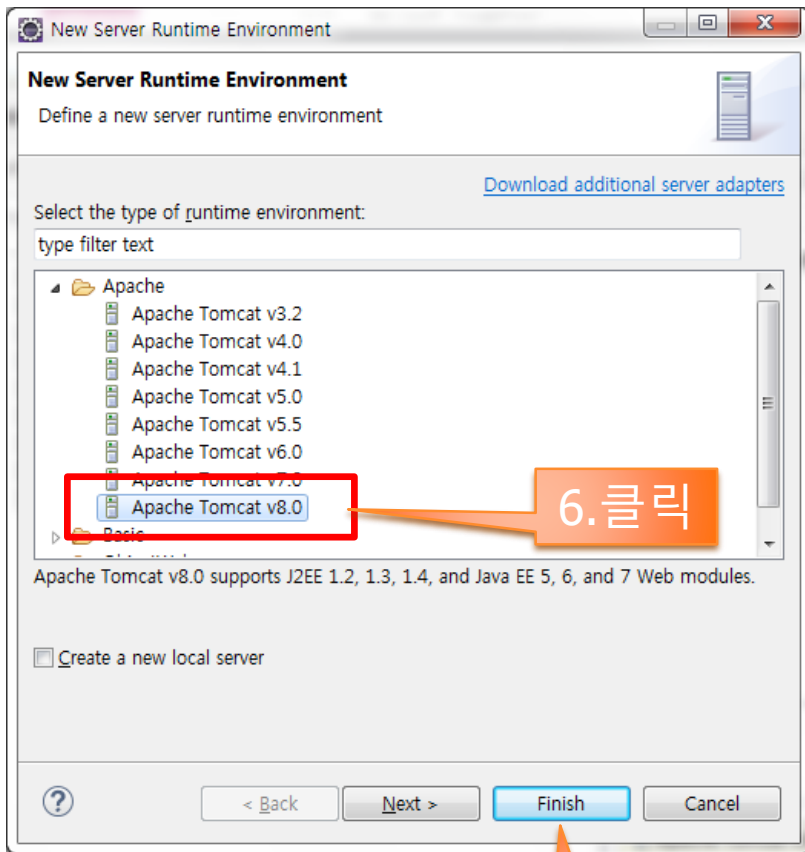
혹? JSP 파일 생성시 "The superclass
"java.servlet.http.HttpServlet" was not found on the Java
Build Path " 에러 발생시 톰캣 연결을 해줘야함.
다음 **1~12**까지 따라서 세팅

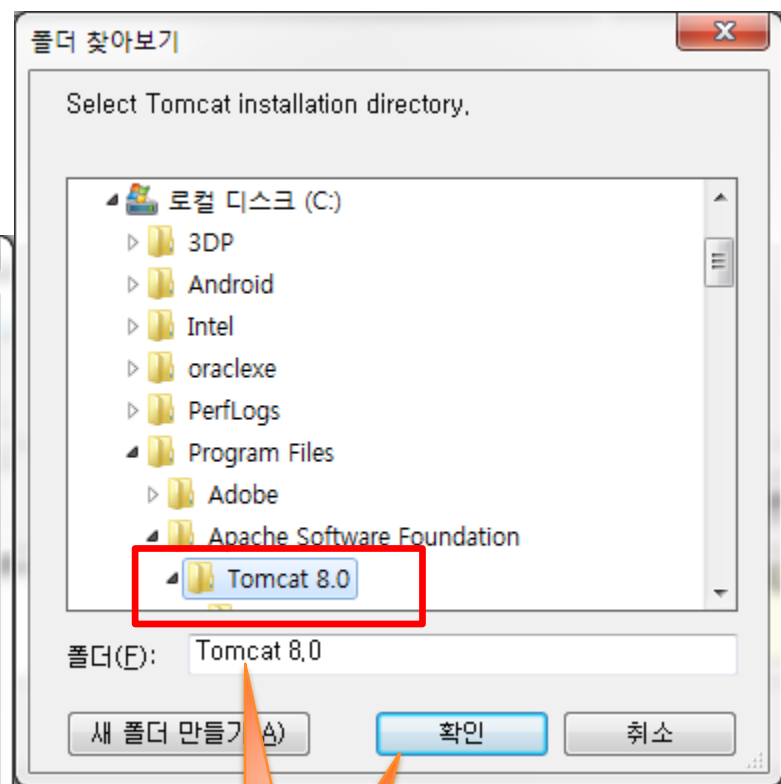
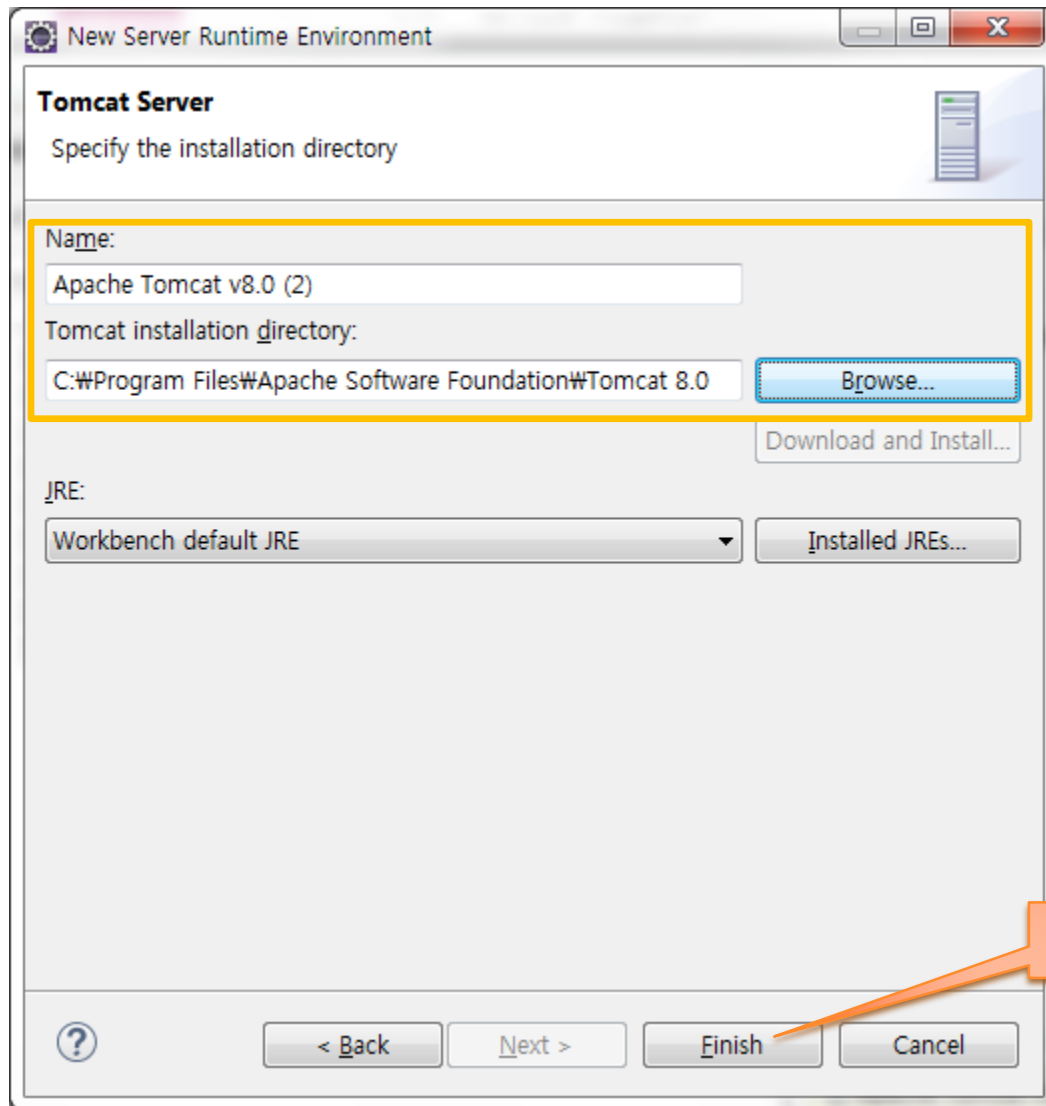


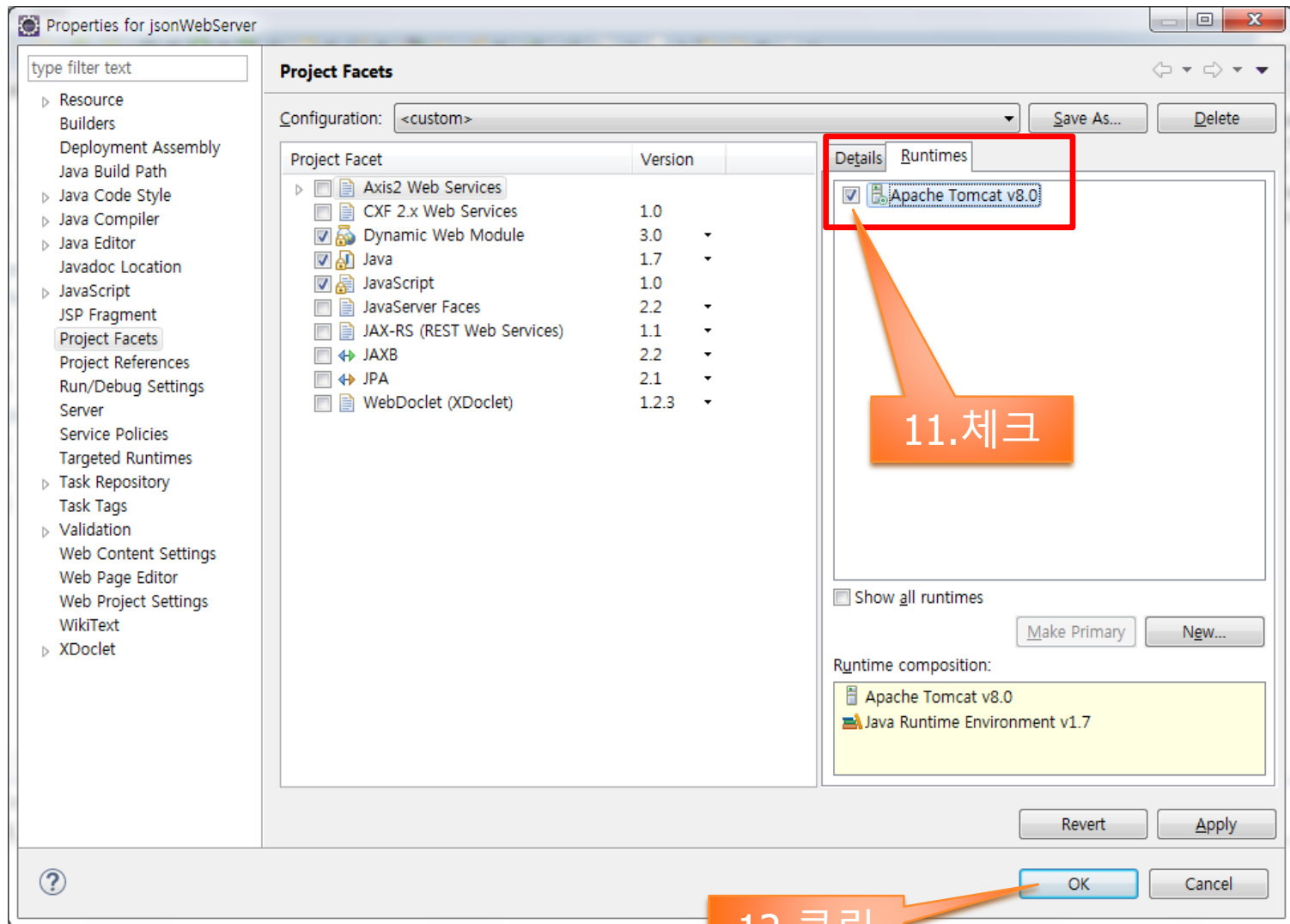
1. "jsonWebServer"
오른쪽 마우스 클릭

2. 클릭







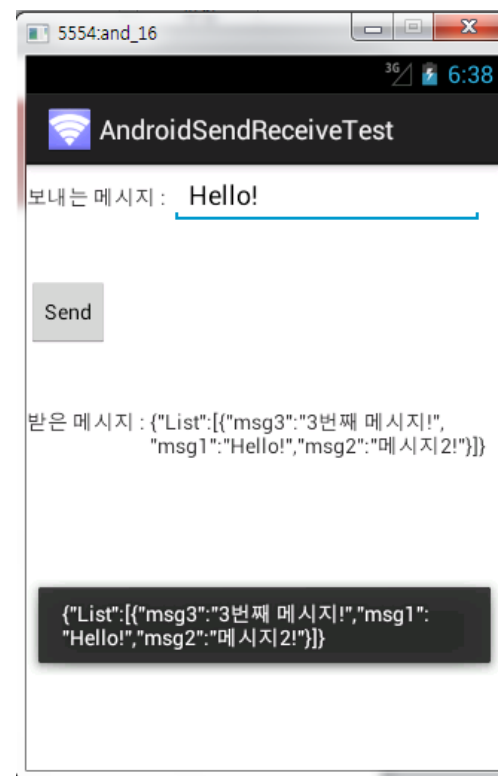
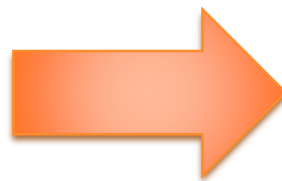
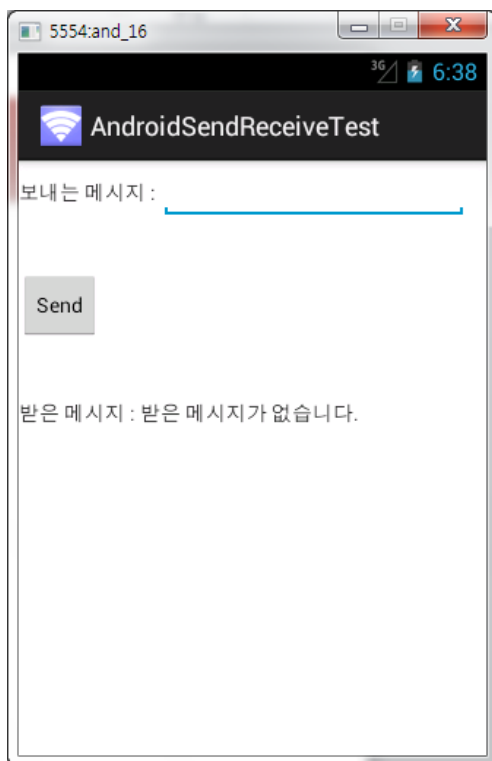


```
webJson.jsp http://localhost:8090/jsonWebServer/webJson.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8" import="org.json.simple.*"%>
3
4 <%
5   String recvMessage = request.getParameter("msg");
6
7   JSONObject jsonMain = new JSONObject();
8   JSONArray jArray = new JSONArray();
9
10  JSONObject jobject = new JSONObject();
11
12  jobject.put("msg1", recvMessage);
13  jobject.put("msg2", "메시지 2!");
14  jobject.put("msg3", "3번째 메시지!");
15
16  jArray.add(0, jobject);
17
18  jsonMain.put("List", jArray);
19
20  out.println(jsonMain.toJSONString());
21 %>
```

받는 부분

보내는 부분

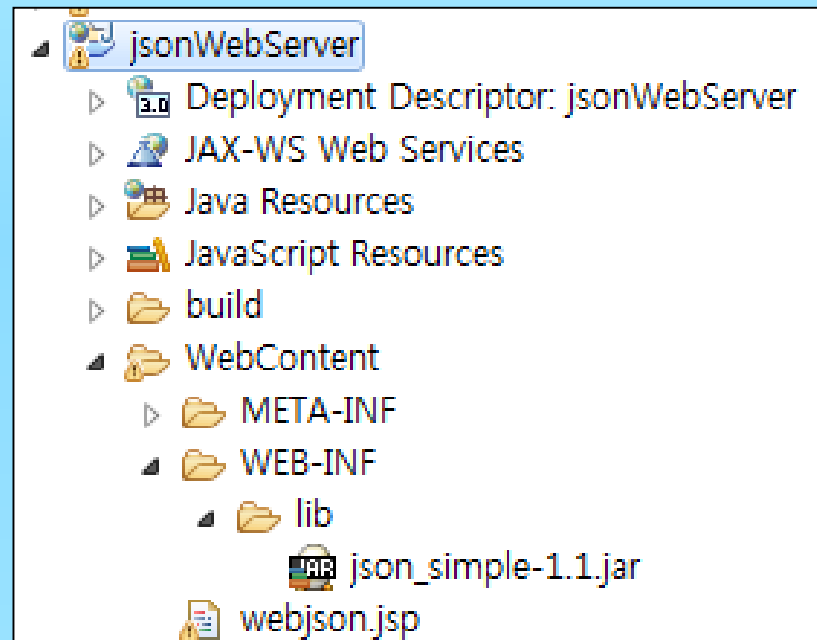
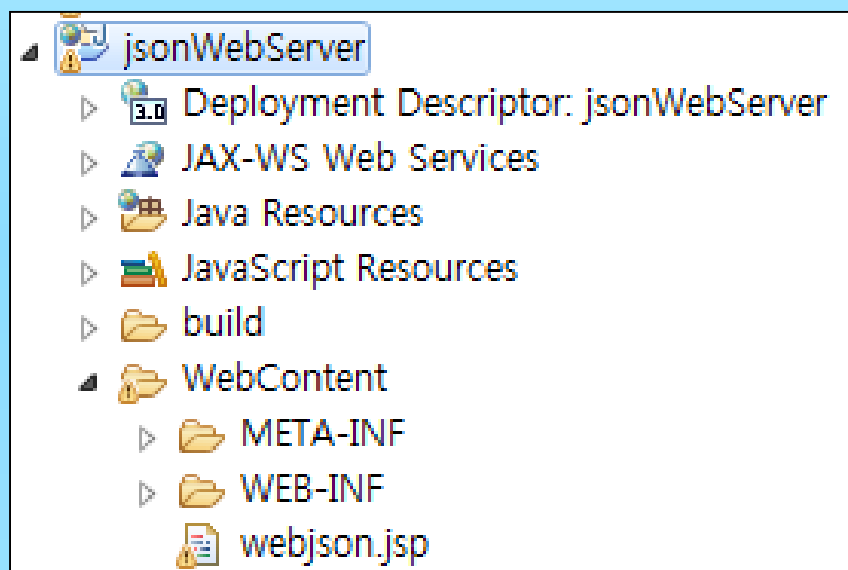
실행 결과



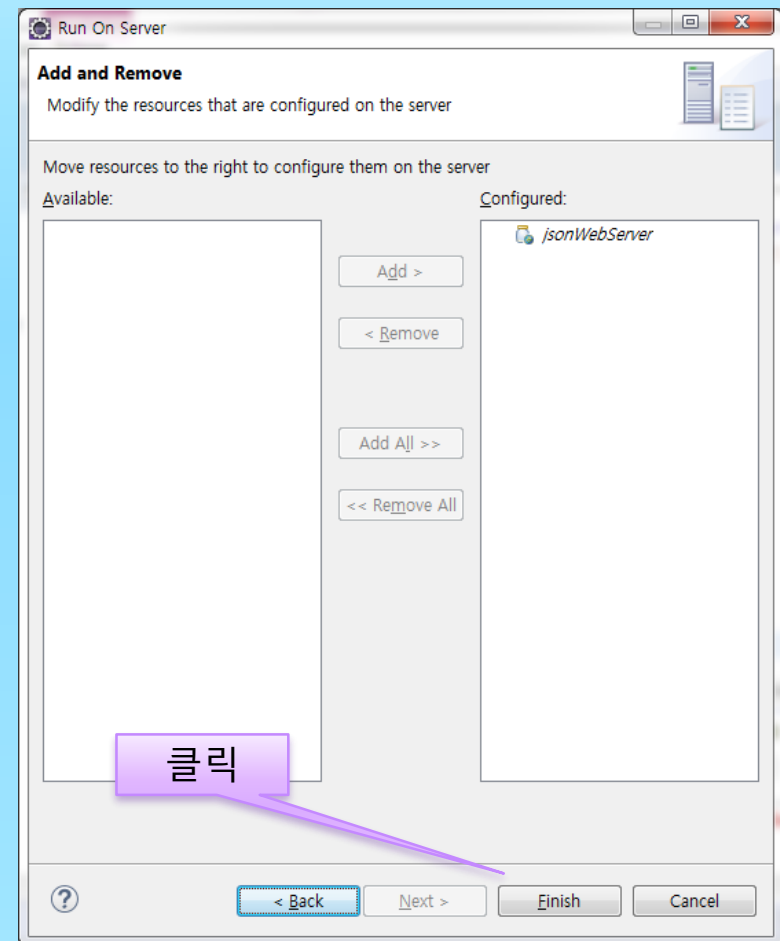
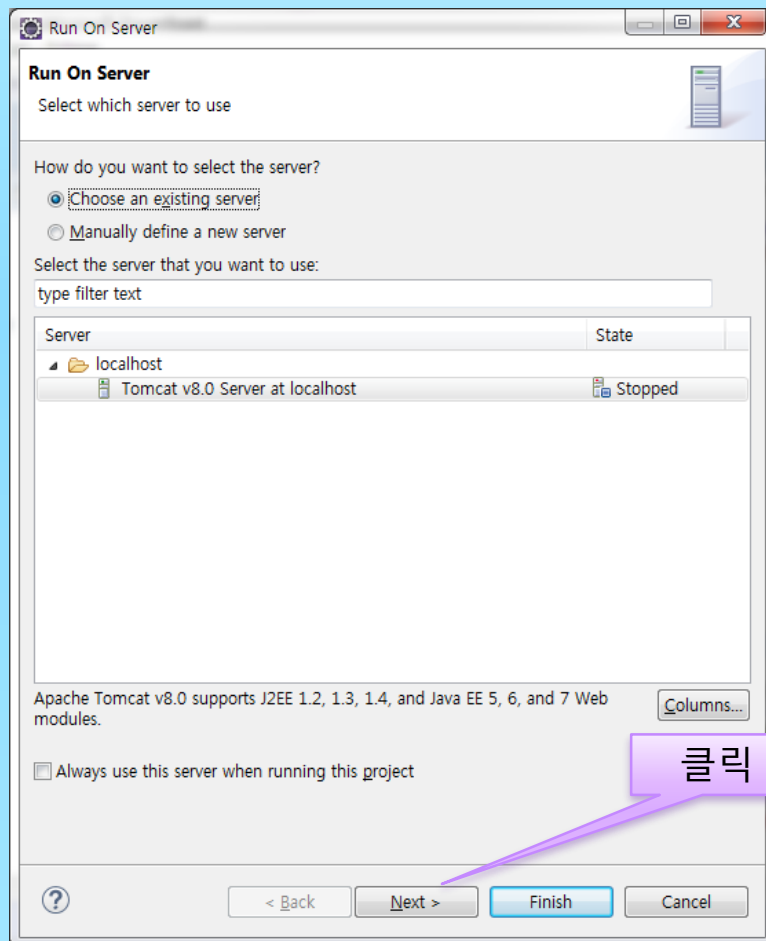
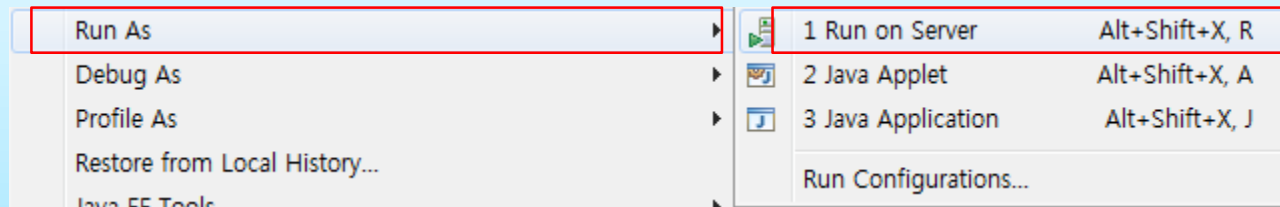
Search for messages. Accepts Java regexes. Prefix with pid:, app:, tag: or text: to limit scope.						
L...	Time	PID	TID	Application	Tag	Text
I	05-20 06:38:56.333	1174	1174	com.example.andr...	서버에서 ...	{"List":[{"msg3":"3번째 메시지!","msg1":"Hello!","msg2":"메시지2!"}]}
I	05-20 06:38:56.333	1174	1174	com.example.andr...	JSON을 분...	Hello!
I	05-20 06:38:56.333	1174	1174	com.example.andr...	JSON을 분...	메시지2!
I	05-20 06:38:56.333	1174	1174	com.example.andr...	JSON을 분...	3번째 메시지!

점검 및 확인 사항

디렉토리 구조

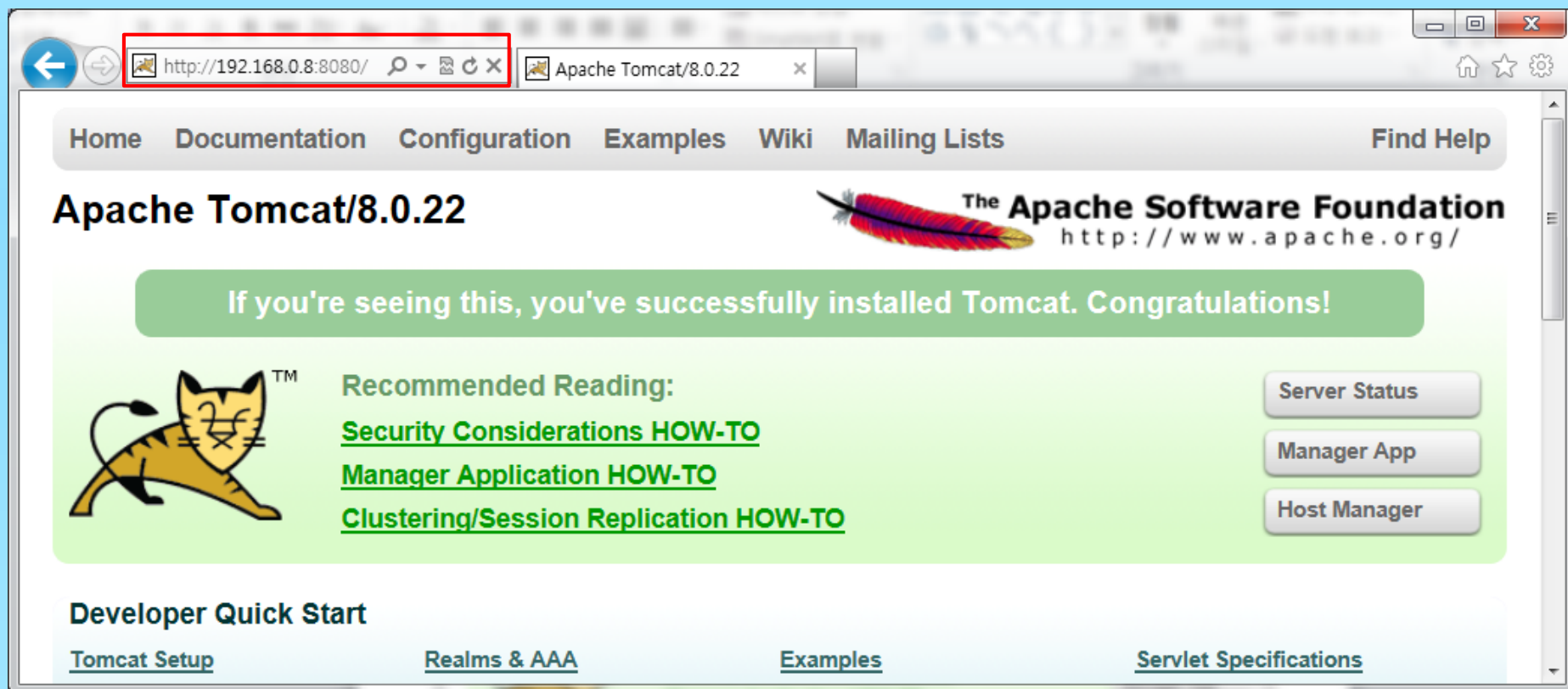


실행





톰캣에서 사용하는 포트와 이클립스에서 사용하는 포트를 다르게 설정하여 사용
톰캣(8080), 이클립스(8090)



이클립스에서 톰캣 정보 확인

Tomcat v8.0 Server at localhost webjson.jsp Apache Tomcat/8.0.22 - Error report

Host name: localhost

Runtime Environment: Apache Tomcat v8.0

Configuration path: /Servers/Tomcat v8.0 Server at loca Browse...

[Open launch configuration](#)

Server Locations

Specify the server path (i.e. catalina.base) and deploy path. Server must be published with no modules present to make changes.

- ☒ Use workspace metadata (does not modify Tomcat installation)
- ☐ Use Tomcat installation (takes control of Tomcat installation)
- ☐ Use custom location (does not modify Tomcat installation)

Server path: .metadata\plugins\org.eclipse.wst.server.c Browse...

[Set deploy path to the default value \(currently set\)](#)

Deploy path: wtpwebapps Browse...

Ports

Modify the server ports.

Port Name	Port Number
Tomcat admin port	8006
HTTP/1.1	8090
AJP/1.3	8010

MIME Mappings

Server Options

Enter settings for the server.

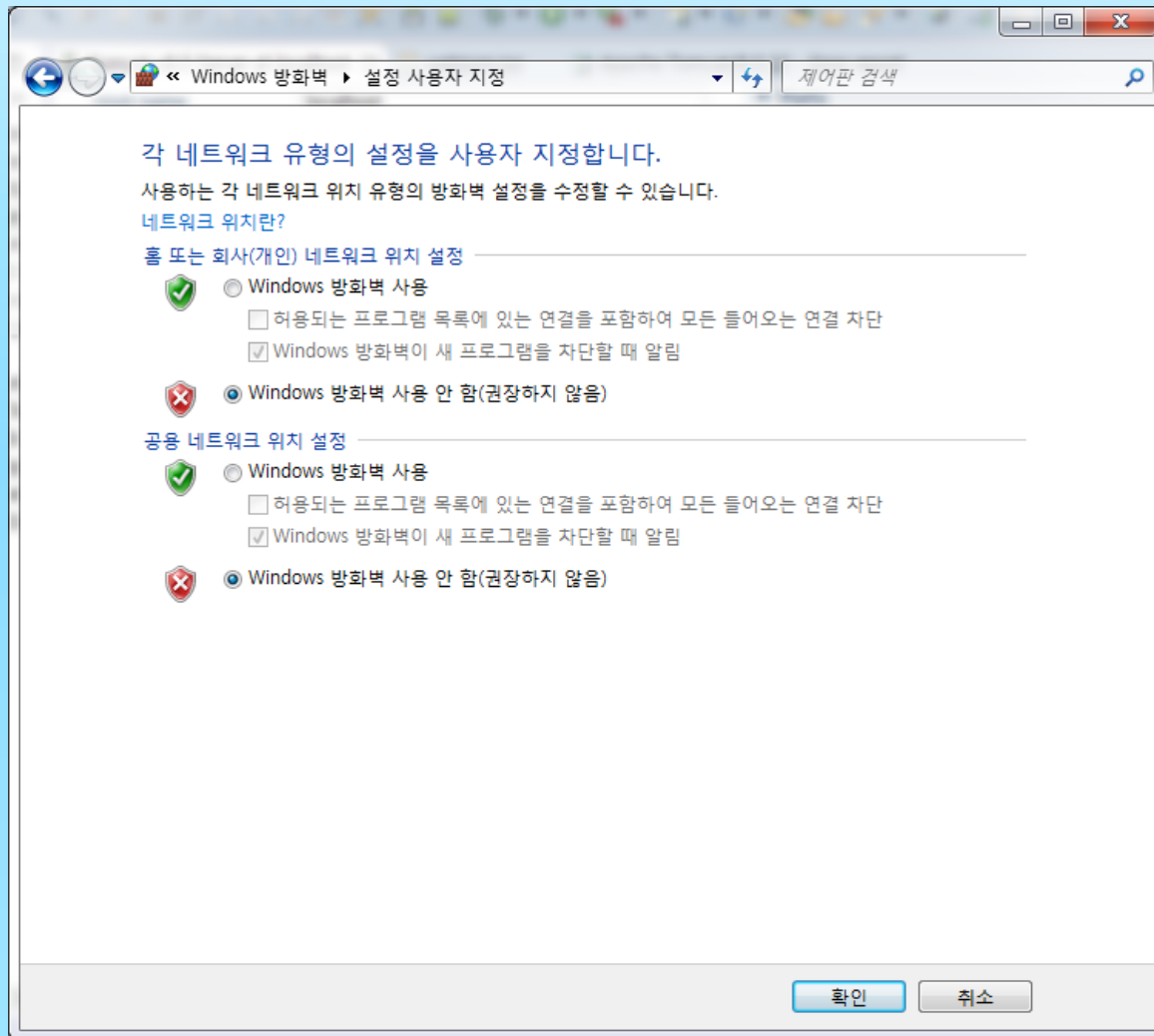
- ☐ Serve modules without publishing
- ☒ Publish module contexts to separate XML files
- ☐ Modules auto reload by default
- ☐ Enable security
- ☐ Enable Tomcat debug logging (not supported by this Tomcat version)

Overview Modules

Markers Properties Servers Data Source Explorer Snippets Problems Console LogCat

Tomcat v8.0 Server at localhost [Started, Synchronized]

더블 클릭하면 위에 화면이 열림



<http://192.168.0.8:8090/jsonWebserver/webjson.jsp>

