

Lab 6

SOAP based RPC Programming

Lab 6. SOAP based RPC Programming


- Setting Development Environments
- Greeting Web Service Server with IntelliJ
- Greeting Web Service Client with IntelliJ
- Archiving Book Rental Shop Web Service as WAR and Deploy with Tomcat
- Packet Analysis

Setting Development Environments


Apache Tomcat

- Apache Tomcat, often referred to as Tomcat Server, is an open-source Java Servlet Container developed by the Apache Software Foundation (ASF).
- Tomcat implements several Java EE specifications including Java Servlet and JavaServer Pages (JSP), and provides a HTTP web server environment

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/8.0.8  The Apache Software Foundation
http://www.apache.org/

If you're seeing this, you've successfully installed Tomcat. Congratulations!

 **Recommended Reading:**
[Security Considerations HOW-TO](#)
[Manager Application HOW-TO](#)
[Clustering/Session Replication HOW-TO](#)

Server Status
Manager App
Host Manager

Developer Quick Start
[Tomcat Setup](#) [Realms & AAA](#) [Examples](#) [Servlet Specifications](#)
[First Web Application](#) [JDBC DataSources](#) [Tomcat Versions](#)

Managing Tomcat
 For security, access to the [manager webapp](#) is restricted. Users are defined in:
 \$CATALINA_HOME/conf/tomcat-users.xml
 In Tomcat 8.0 access to the manager application is split between different users.
[Read more...](#)
Release Notes
Changelog
Migration Guide
Security Notices

Documentation
[Tomcat 8.0 Documentation](#)
[Tomcat 8.0 Configuration](#)
[Tomcat Wiki](#)
 Find additional important configuration information in:
 \$CATALINA_HOME/RUNNING.txt
 Developers may be interested in:
[Tomcat 8.0 Bug Database](#)
[Tomcat 8.0 JavaDocs](#)
[Tomcat 8.0 SVN Repository](#)

Getting Help
[FAQ and Mailing Lists](#)
 The following mailing lists are available:
[tomcat-announce](#)
 Important announcements, releases, security vulnerability notifications. (Low volume).
[tomcat-users](#)
 User support and discussion
[taglibs-user](#)
 User support and discussion for [Apache Taglibs](#)
[tomcat-dev](#)
 Development mailing list, including commit messages

Setup Steps for Tomcat development environment

1. Download & Install Java S/W Development Kit (JDK)
2. Download & Install server (Apache Tomcat 7)
3. Test server (Apache Tomcat 7)
4. Set up development environment
5. Install IntelliJ IDEA

Download & Install JDK 7

- <http://www.oracle.com/technetwork/java/javase/downloads/jdk7u9-downloads-1859576.html>

Download JDK

Community Technologies Training

Java SE Development Kit 7 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

Looking for JavaFX SDK?

JavaFX SDK is now included in the JDK for Windows, Mac OS X, and Linux x86/x64.

See also:

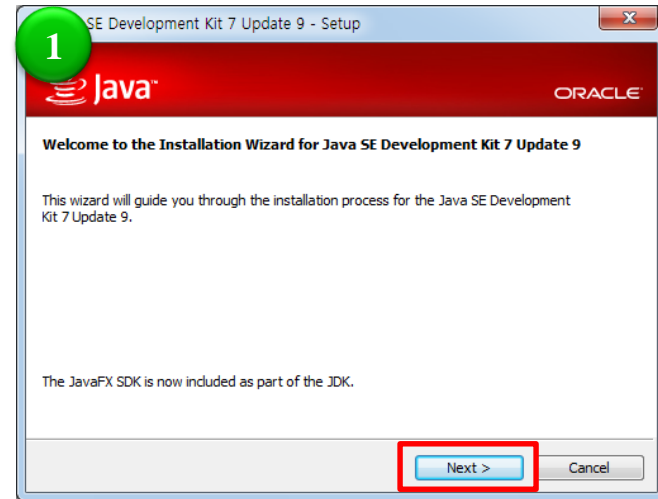
- [Java Developer Newsletter](#) (tick the checkbox under Subscription Center > Oracle Technology News)
- [Java Developer Day hands-on workshops](#) (free) and other events
- [Java Magazine](#)

Java SE Development Kit 7u9

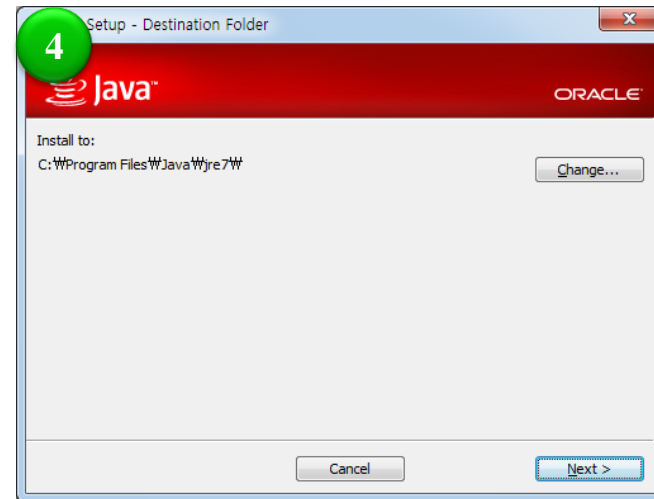
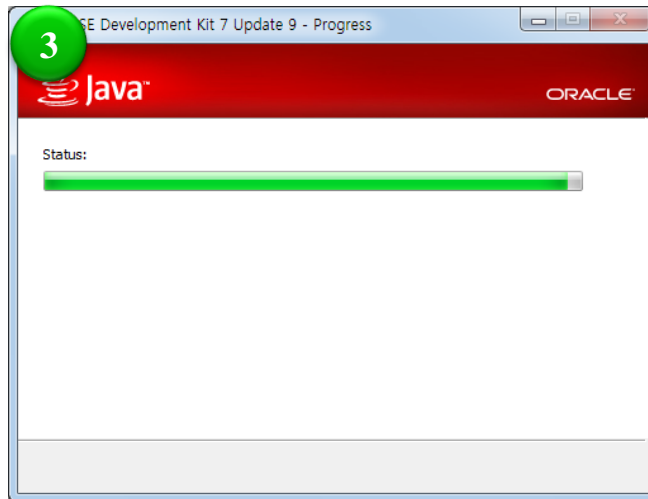
You must accept the **Oracle Binary Code License Agreement for Java SE** to download this software.

☒ Accept License Agreement ☐ Decline License Agreement

Product / File Description	File Size	Download
Linux x86	120.63 MB	jdk-7u9-linux-i586.rpm
Linux x86	92.85 MB	jdk-7u9-linux-i586.tar.gz
Linux x64	118.82 MB	jdk-7u9-linux-x64.rpm
Linux x64	91.59 MB	jdk-7u9-linux-x64.tar.gz
Mac OS X	143.47 MB	jdk-7u9-macosx-x64.dmg
Solaris x86	135.14 MB	jdk-7u9-solaris-i586.tar.Z
Solaris x86	91.51 MB	jdk-7u9-solaris-i586.tar.gz
Solaris SPARC	135.7 MB	jdk-7u9-solaris-sparc.tar.Z
Solaris SPARC	95.15 MB	jdk-7u9-solaris-sparc.tar.gz
Solaris SPARC 64-bit	22.8 MB	jdk-7u9-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	17.51 MB	jdk-7u9-solaris-sparcv9.tar.gz
Solaris x64	22.48 MB	jdk-7u9-solaris-x64.tar.Z
Solaris x64	14.94 MB	jdk-7u9-solaris-x64.tar.gz
Windows x86	88.35 MB	jdk-7u9-windows-i586.exe
Windows x64	90.03 MB	jdk-7u9-windows-x64.exe



Install JDK 7



Install Web Server (Apache Tomcat)

- What is Tomcat?
 - Tomcat is an open source server from the Apache Software Foundation. It is a Web application server, which means that it comes ready to support programming using JavaServer Pages (JSPs) and servlets.¹

Download Apache Tomcat

- <http://tomcat.apache.org>
- Download :
32-bit/64-bit Windows Service Installer

The screenshot shows the Apache Tomcat website layout. On the left is a sidebar with navigation links. In the center is the main content area. On the right is a sidebar with additional links. Blue arrows indicate the navigation path from the 'Download' link in the sidebar to the 'Tomcat 7.0' link in the main content area, and then to the '32-bit/64-bit Windows Service Installer' link in the 'Binary Distributions' section.

Apache Tomcat

- [Home](#)
- [Taglibs](#)
- [Maven Plugin](#)

Download

- Which version?
- **Tomcat 7.0**
- [Tomcat 6.0](#)
- [Tomcat 5.5](#)
- [Tomcat Connectors](#)
- [Tomcat Native](#)
- [Archives](#)

Documentation

- [Tomcat 7.0](#)
- [Tomcat 6.0](#)
- [Tomcat 5.5](#)
- [Tomcat Connectors](#)
- [Tomcat Native](#)
- [Wiki](#)
- [Migration Guide](#)

Mirrors

You are currently using <http://apache.mirror.cdnetworks.com/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are *backup* mirrors (at the end of the mirrors list) that should be available.

Other mirrors: [Change](#)

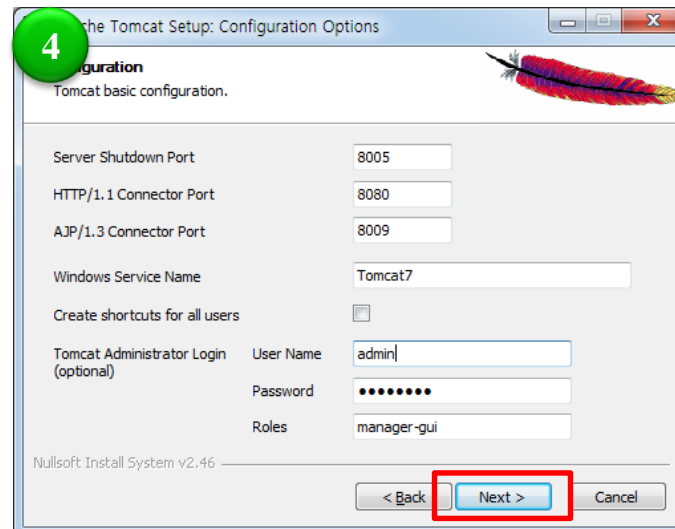
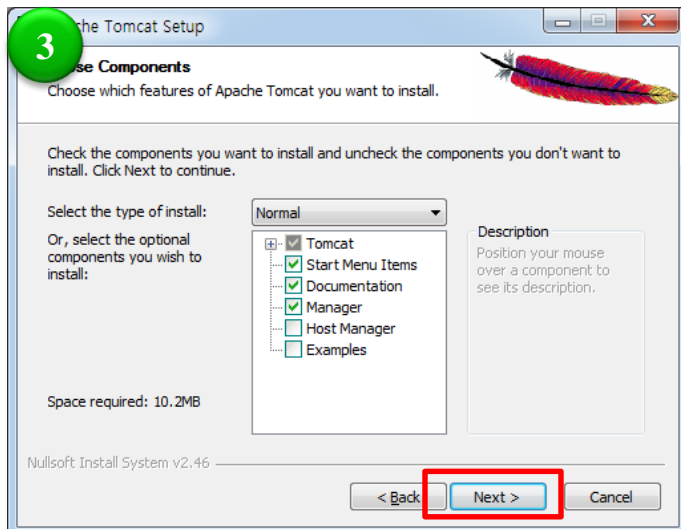
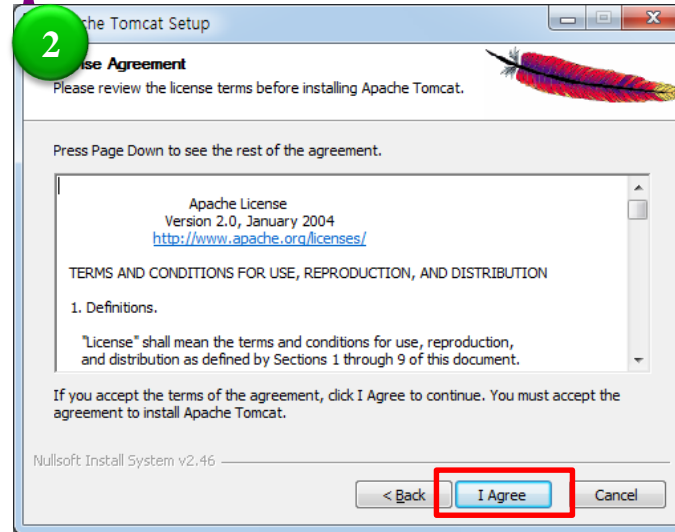
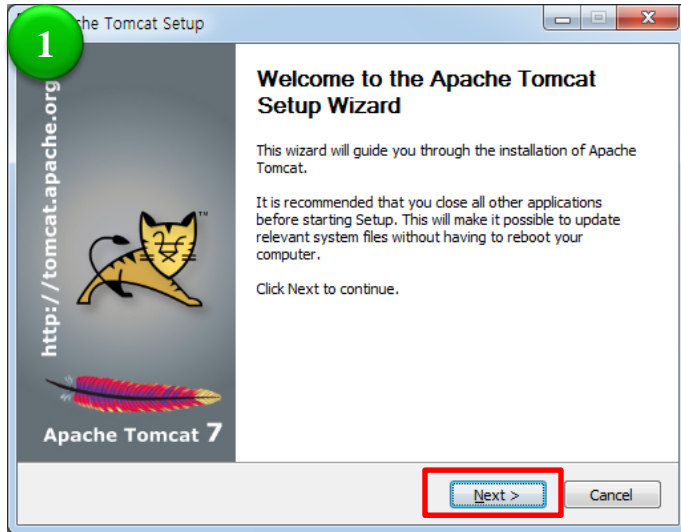
7.0.32

Please see the [README](#) file for packaging information. It explains what every distribution contains.

Binary Distributions

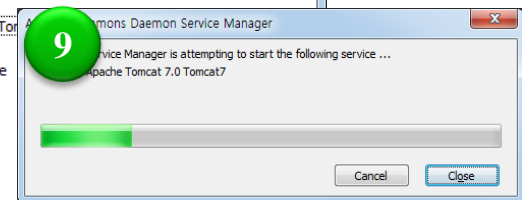
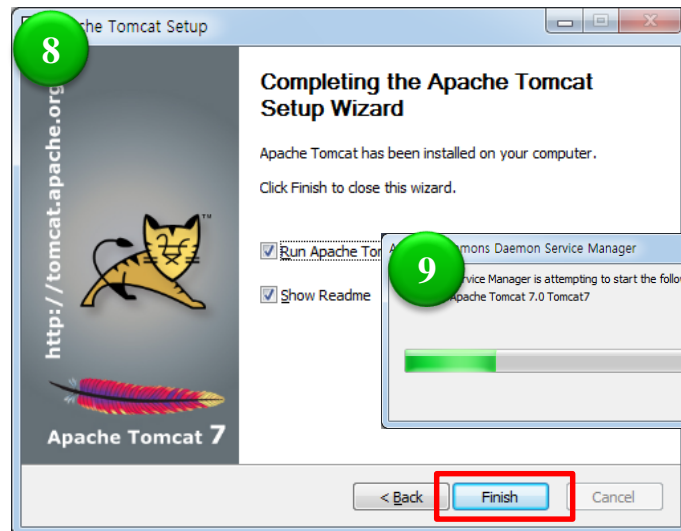
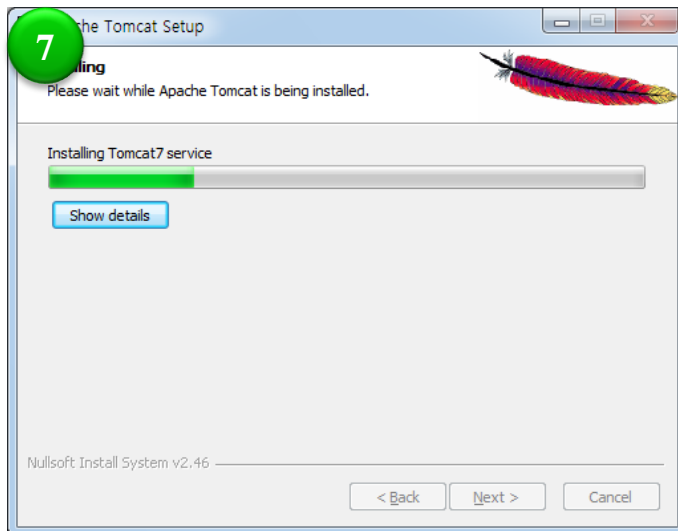
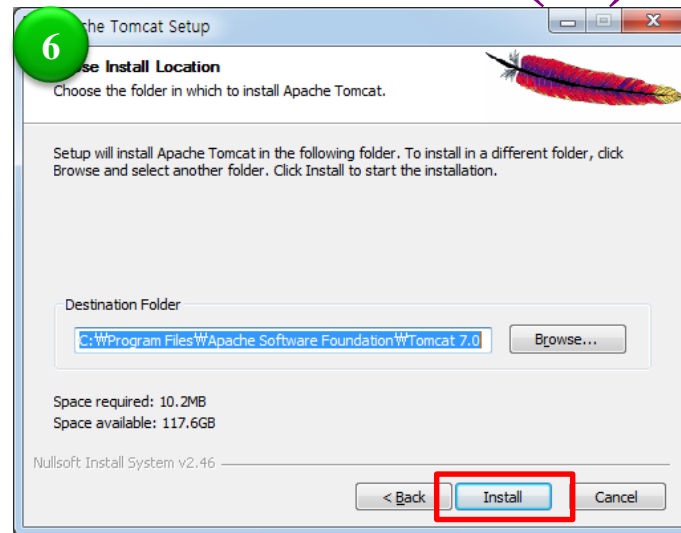
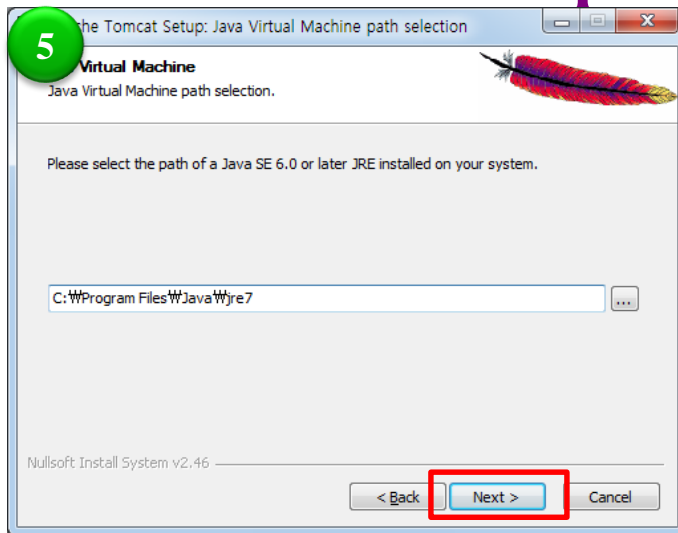
- Core:
 - [zip \(pgp, md5\)](#)
 - [tar.gz \(pgp, md5\)](#)
 - [32-bit Windows zip \(pgp, md5\)](#)
 - [64-bit Windows zip \(pgp, md5\)](#)
 - [64-bit Linux Windows zip \(pgp, md5\)](#)
 - **32-bit/64-bit Windows Service Installer (pgp, md5)**

Install Apache Tomcat (1)



Enter User
Name &
Password

Install Apache Tomcat (2)



Install Apache Tomcat in Ubuntu

- `sudo apt-get install openjdk-7-jre-headless`
- `sudo apt-get install tomcat7`
- `service tomcat7 start`
- `netstat -ntl`

```
root@ubuntu:/usr# netstat -ntl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp6       0      0 :::8080                 :::*                     LISTEN
tcp6       0      0 :::1:631                :::*                     LISTEN
tcp6       0      0 127.0.0.1:8005          :::*                     LISTEN
```

Install Apache Tomcat in Mac

- `brew install cask java`
- `brew install tomcat`
- `{TOMCAT HOME}/bin/startup.sh`


OR

- Download tar.gz file and extract it
- `sudo rm -f /Library/Tomcat`
- `sudo ln -s /usr/local/apache-tomcat-7.0.90 /Library/Tomcat`
- `sudo chown -R <user_ID> /Library/Tomcat`
- `sudo chmod +x /Library/Tomcat/bin/*.sh`
- - Start tomcat
- `/Library/Tomcat/bin/startup.sh`
- - Shut down tomcat
- `/Library/Tomcat/bin/shutdown.sh`


Management Web Page

- <http://localhost:8080/> or [http://\[Host IP Address\]:8080/](http://[Host IP Address]:8080/)

[Home](#)
[Documentation](#)
[Configuration](#)
[Examples](#)
[Wiki](#)
[Mailing Lists](#)
[Find Help](#)

Apache Tomcat/7.0.32

<http://www.apache.org/>

If you're seeing this, you've successfully installed Tomcat. Congratulations!



Recommended Reading:
[Security Considerations HOW-TO](#)
[Manager Application HOW-TO](#)
[Clustering/Session Replication HOW-TO](#)

[Server Status](#)
[Manager App](#)
[Host Manager](#)

Developer Quick Start

[Tomcat Setup](#)
[Realms & AAA](#)
[Examples](#)
[Servlet Specifications](#)
[First Web Application](#)
[JDBC DataSources](#)
[Tomcat Versions](#)

Managing Tomcat
 For security, access to the [manager webapp](#) is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```


 In Tomcat 7.0 access to the manager application is split between different users.
[Read more...](#)
[Release Notes](#)
[Changelog](#)
[Migration Guide](#)
[Security Notices](#)

Documentation
[Tomcat 7.0 Documentation](#)
[Tomcat 7.0 Configuration](#)
[Tomcat Wiki](#)
 Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```


 Developers may be interested in:
[Tomcat 7.0 Bug Database](#)
[Tomcat 7.0 JavaDocs](#)
[Tomcat 7.0 SVN Repository](#)

Getting Help
FAQ and Mailing Lists
 The following mailing lists are available:

announce@tomcat.apache.org
 Important announcements, releases, security vulnerability notifications. (Low volume).

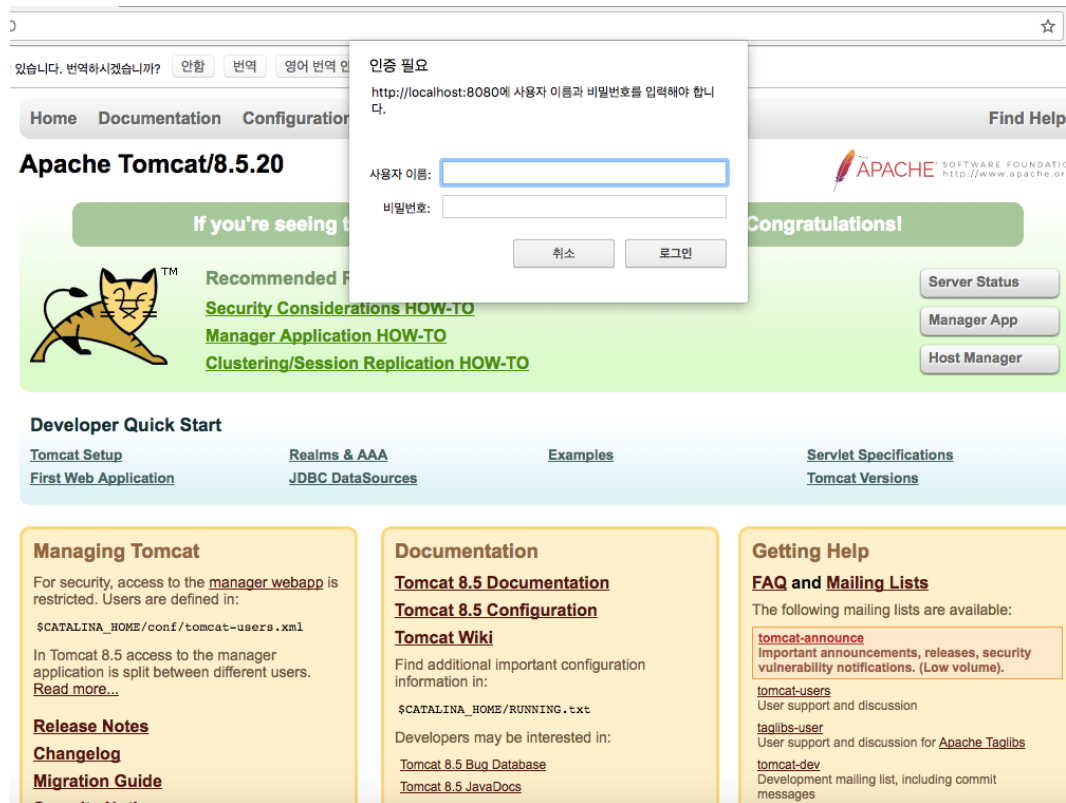
users@tomcat.apache.org
 User support and discussion

taglibs-user@tomcat.apache.org
 User support and discussion for [Apache Taglibs](#)

dev@tomcat.apache.org
 Development mailing list, including commit messages

Authentication Problem

- User account should be created!



Create tomcat users

- Edit the *[/\[TOMCAT HOME\]/conf/tomcat-users.xml](#)*
 - Location of TOMCAT HOME
 - Ubuntu: /etc/tomcat{version}/
 - Mac: /usr/local/Cellar/tomcat/{version}/libexec
- Add role named ‘manager-gui’
- Add user with your username and password with roles ‘manage-gui’
 - This user now can control GUI manager

```
<!--
<role rolename="tomcat"/>
<role rolename="role1"/>
<user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
<user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
<user username="role1" password="<must-be-changed>" roles="role1"/>
-->
<role rolename="manager-gui"/>
<user username="admin" password="1" roles="manager-gui"/>
</tomcat-users>
```


Manage Services

- We can life cycle of service (start /stop / restart / terminate)
 - There is already existing services including docs, examples, etc...



Tomcat Web Application Manager

Message:	OK
----------	----

Manager					
List Applications	HTML Manager Help	Manager Help	Server Status		

Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/> <input type="button" value="Expire sessions"/> with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/> <input type="button" value="Expire sessions"/> with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/> <input type="button" value="Expire sessions"/> with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/> <input type="button" value="Expire sessions"/> with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/> <input type="button" value="Expire sessions"/> with idle ≥ 30 minutes

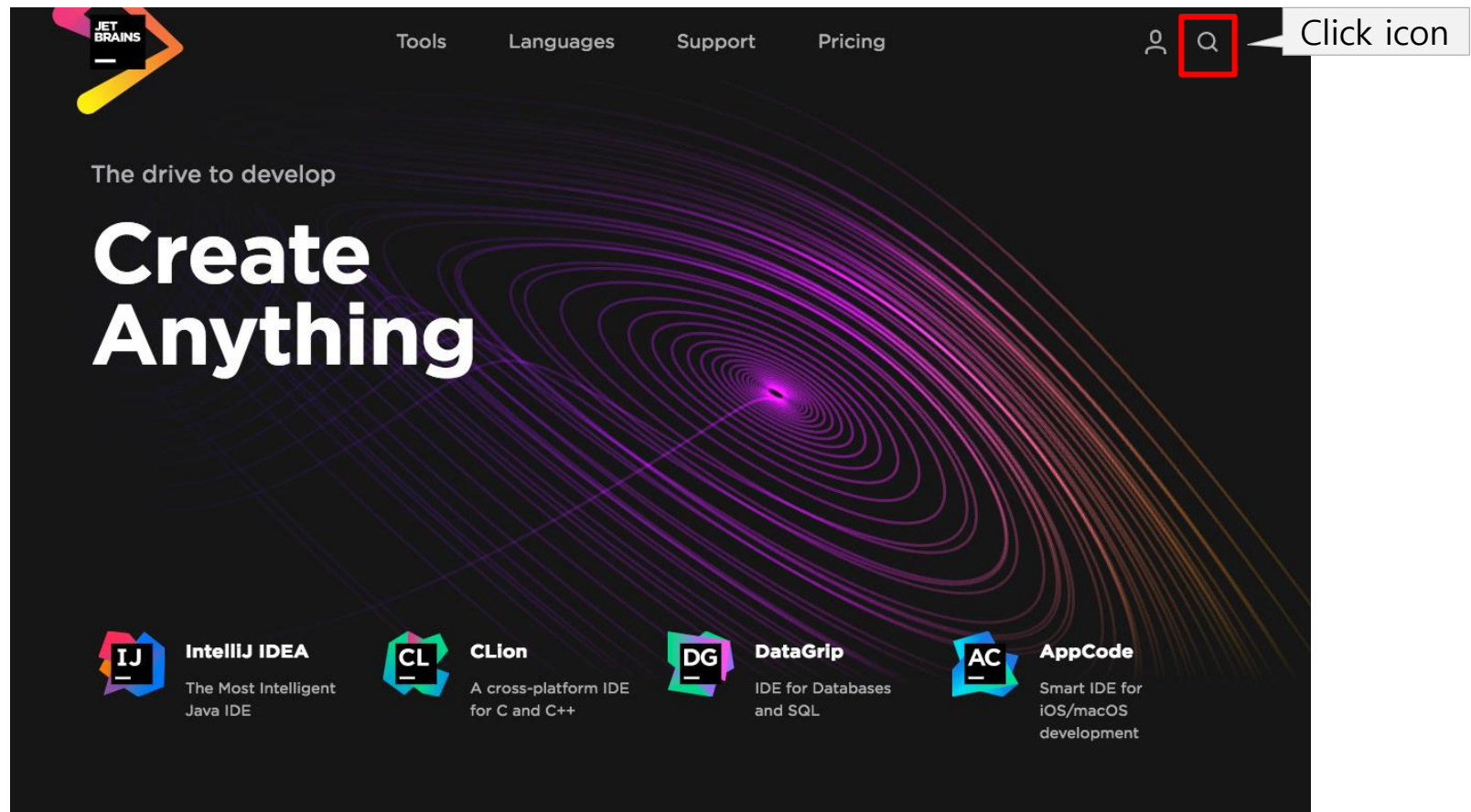
Deploy	
Deploy directory or WAR file located on server	
Context Path (required):	<input type="text"/>
XML Configuration file URL:	<input type="text"/>
WAR or Directory URL:	<input type="text"/>
<input type="button" value="Deploy"/>	

Tomcat Server Test Example

- Chat example
<http://localhost:8080/examples/websocket/chat.xhtml>
- Snake example
<http://localhost:8080/examples/websocket/snake.xhtml>
- Draw example
<http://localhost:8080/examples/websocket/drawboard.xhtml>
- Together
<http://143.248.xxx.xxx:8080/examples/websocket/chat.xhtml>




Install IntelliJ IDEA ver. Ultimate

1. Access a page <https://www.jetbrains.com/>
2. Click account icon



3. Create an account with your KAIST email

Welcome to JetBrains Account

-  **Access your purchases**
and view your order history
-  **Identify expired and outdated licenses,**
order new licenses and upgrades
-  **Manage your company licenses**
and distribute them to end users

Sign in with existing account

Email or Username

Password

[Sign In](#) [Forgot password?](#)

Not registered yet?

Create JetBrains Account

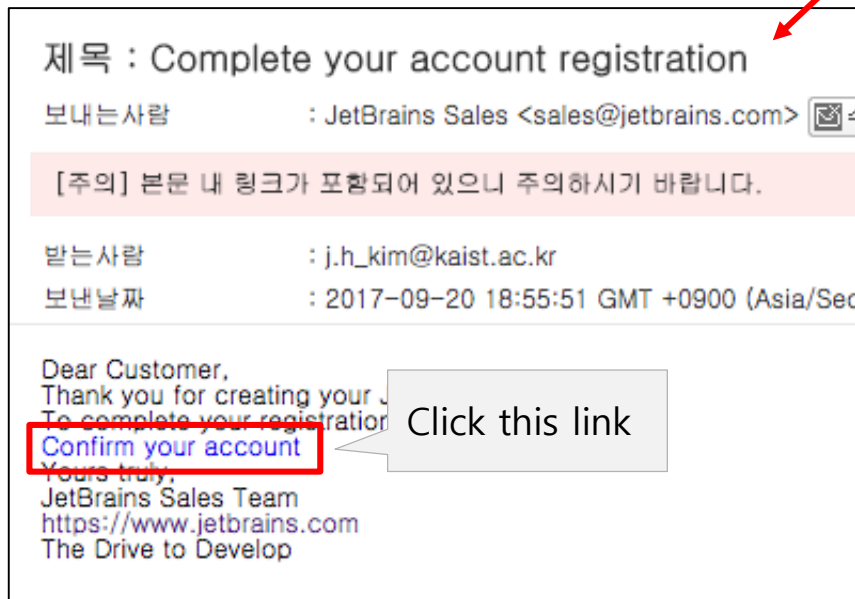
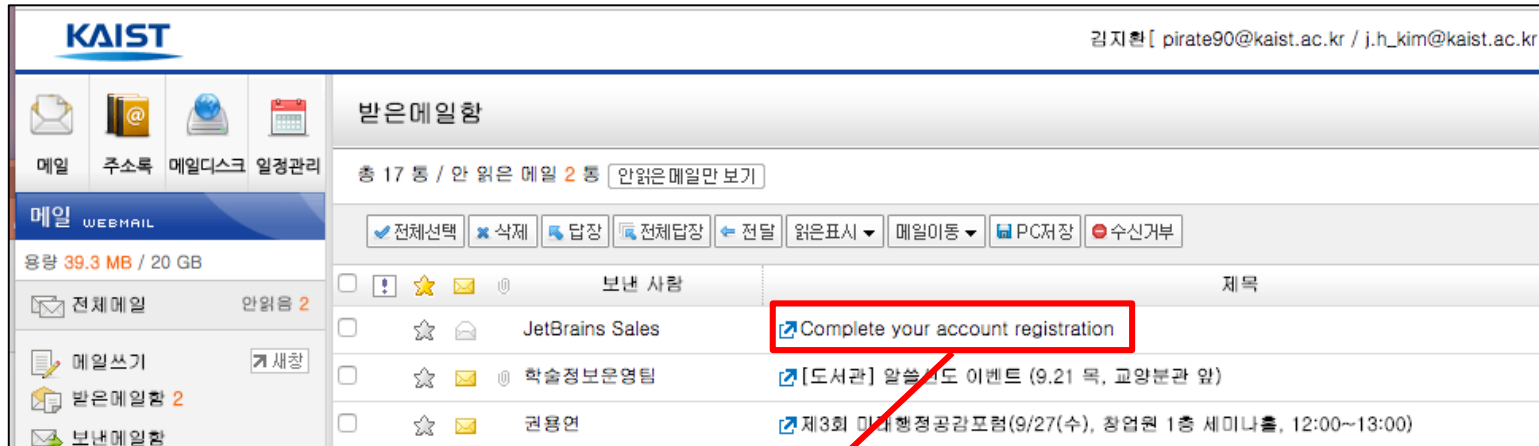
[Sign Up](#)

Insert your kaist mail address. Click "Sign Up" icon.

Thank you for registering your JetBrains Account!

Please follow the instructions we have just emailed to you at j.h_kim@kaist.ac.kr.

4. Go your KAIST email and Confirm your account



5. Insert your information for registration

Welcome to JetBrains Account!

Please complete the registration form below.

Email Address

First Name

Last Name

Username

Please make sure you choose a strong password as your account will have access to your purchases:

Password

Repeat Password

☒ I have read and I accept the [JetBrains Privacy Policy](#)

Insert your
information
and Click
"Submit" icon

6. Apply for free student or teacher license

No Available Licenses

We found no JetBrains product licenses associated with your JetBrains Account. You can:

- Purchase product license(s) at <https://www.jetbrains.com>
- [Link your past purchases](#) to your account
- Contact the person who manages commercial licenses and request an invitation to use them
- [Apply for a free student or teacher license](#)

Click this link

Your JetBrains Account is a single interaction point for activating JetBrains products and accessing the following services:

- [JetBrains Account website](#) (you are here)
- [Products Support](#)
- [Product Blogs](#)
- [Plugin Repository for .NET products](#) (e.g. ReSharper)
- [Plugin Repository for other IDEs](#) (e.g. IntelliJ IDEA, WebStorm and so on)

All Products Pack

Get access to all desktop products including IntelliJ IDEA Ultimate, ReSharper Ultimate and other IDEs. All you need to apply is to be a student and have access to your student email address or a valid ISIC card.

Find out more in [FAQ](#) below.

APPLY NOW

for all products at once!

Click this icon

6. Apply for free student or teacher license

JetBrains Products for Learning

Apply with: **UNIVERSITY EMAIL ADDRESS** | ISIC/ITIC MEMBERSHIP | OFFICIAL DOCUMENT

Status: ☒ I'm a student
☐ I'm a teacher

Name: First name Last name

Our software will be registered to your real name.

Email address:

Your valid university email address, e.g. john.smith@mit.edu.
I confirm that the email address provided above belongs to me.

Country: Korea, Republic of

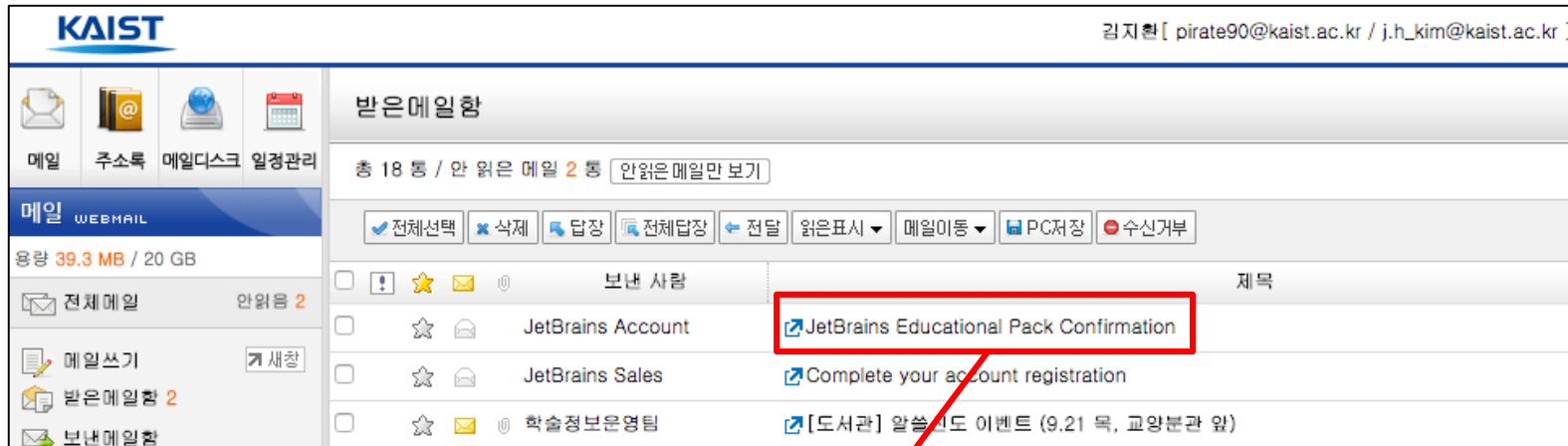
☐ I have read and I accept the [JetBrains](#)

APPLY FOR FREE PRODUCTS

Insert your kaist email address

Insert your information and Click icon

7. Go your KAIST email and Confirm education pack



Click this link

Done your
get licenses

JetBrains Products for Learning

Congrats! You've been approved!

You are now entitled to use JetBrains products for free. Start learning with the instructions we have just emailed you at j.h_kim@kaist.ac.kr.

For more details you can also check out the [JetBrains Student program](#).

8. Access a page

<https://www.jetbrains.com/idea/download/#section=windows>

9. download & install the Ultimate version of IntelliJ

Download IntelliJ IDEA

Windows macOS Linux Check your environment

Ultimate
Web, mobile and enterprise development

DOWNLOAD .EXE
Free trial

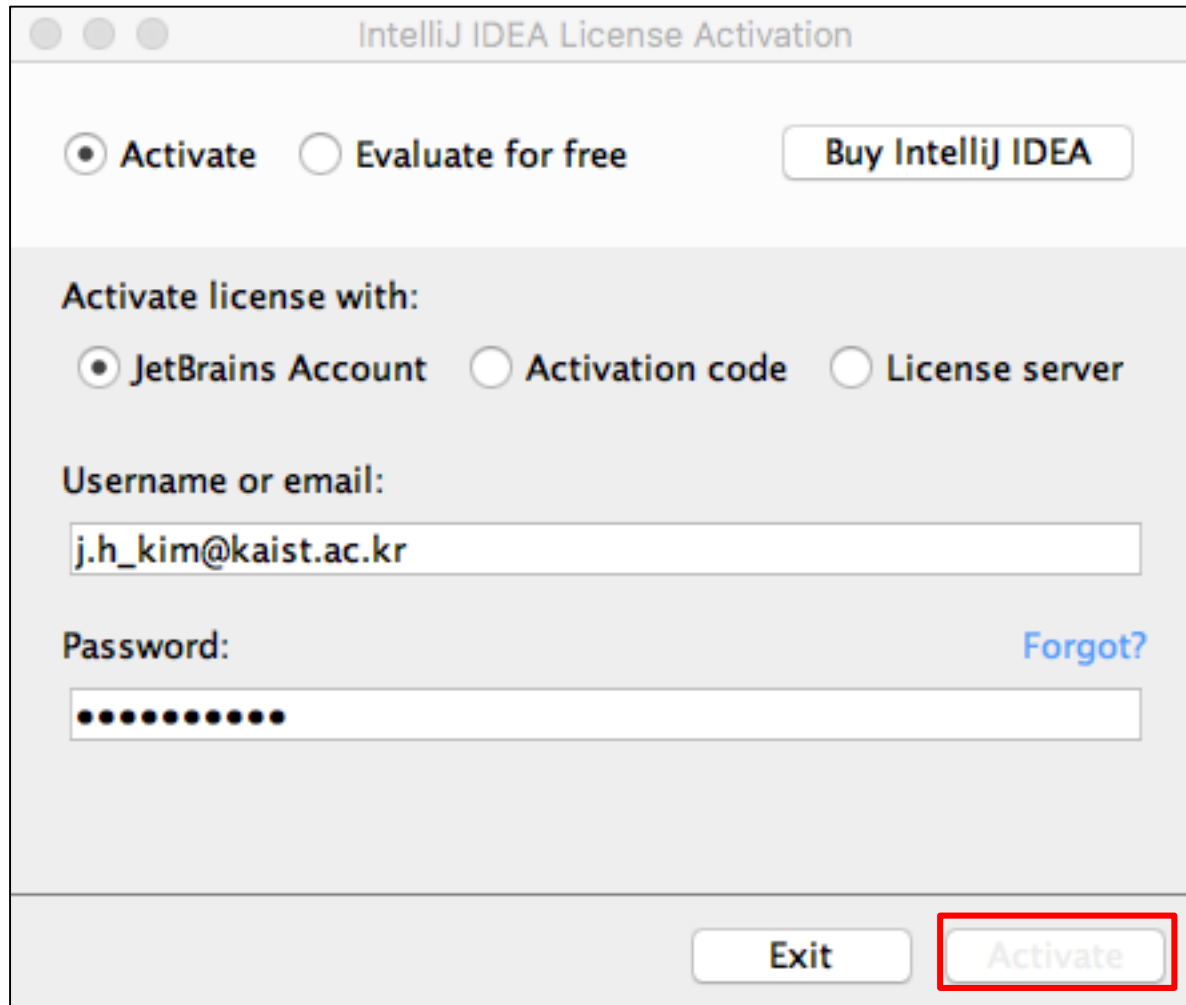
Community
Java, Groovy, Scala and Android development

DOWNLOAD .EXE
Free, open-source

Version: 2017.2.4
Build: 172.4155.36
Released: September 12, 2017
[Release notes](#)

[System requirements](#)
[Installation Instructions](#)
[Previous versions](#)

9. Execute IntelliJ IDEA and activate your license



The image shows the IntelliJ IDEA License Activation dialog box. It has a title bar with three window control buttons and the text "IntelliJ IDEA License Activation". Inside, there are two radio buttons: "Activate" (selected) and "Evaluate for free". To the right is a button labeled "Buy IntelliJ IDEA". Below this, the text "Activate license with:" is followed by three radio buttons: "JetBrains Account" (selected), "Activation code", and "License server". Underneath, there is a label "Username or email:" followed by a text input field containing "j.h_kim@kaist.ac.kr". Below that is a label "Password:" followed by a password input field with ten dots. To the right of the password field is a blue link labeled "Forgot?". At the bottom, there are two buttons: "Exit" and "Activate". The "Activate" button is highlighted with a red rectangular border.

IntelliJ IDEA License Activation

☒ Activate ☐ Evaluate for free [Buy IntelliJ IDEA](#)

Activate license with:

☒ JetBrains Account ☐ Activation code ☐ License server

Username or email:

Password: [Forgot?](#)

[Exit](#) [Activate](#)

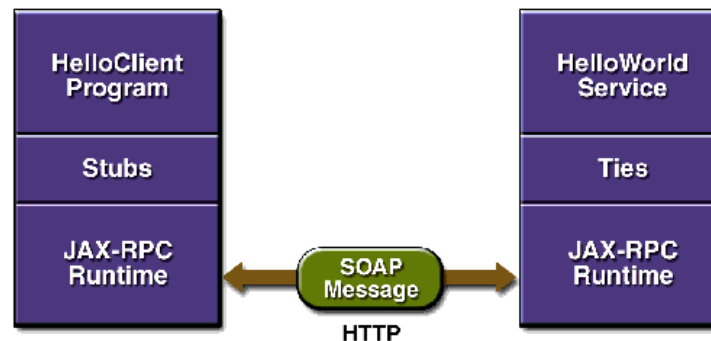
10. Congratulations! IDE Installation is done!



Greeting Web Service Server with IntelliJ

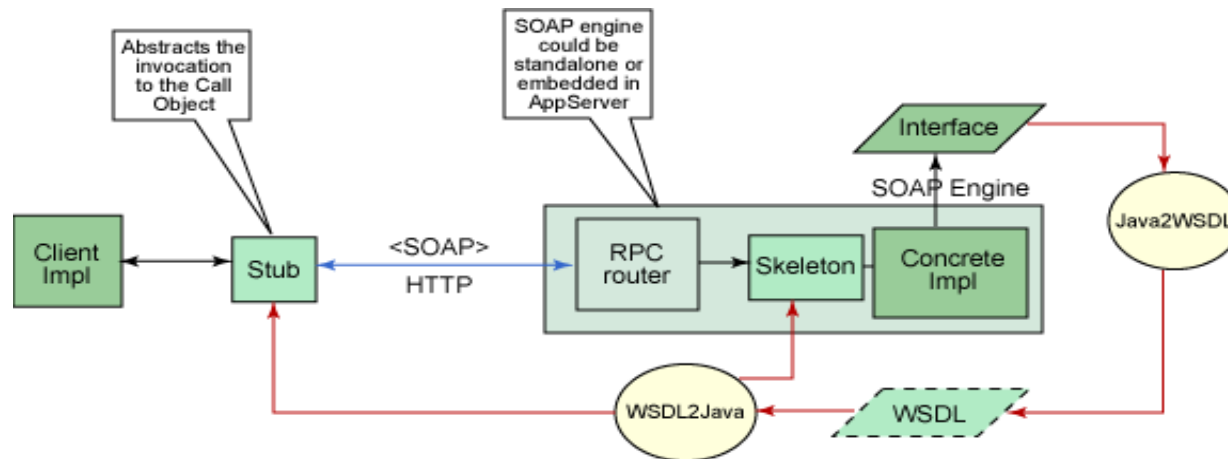
JAX-RPC and JAX-WS

- Java **API** for **XML**-based **RPC** (JAX-RPC) allows to invoke a Java-based **Web service** with WSDL description.
- It can be seen as **Java RMI**s over **Web services**.
- JAX-RPC 2.0 was renamed JAX-WS 2.0 (Java API for XML Web Services) and JAX-RPC is now deprecated
- RPCServer
 - Has the procedure that is called by clients
- RPCClient
 - Calls the procedure which is located RPCServer
- Interfacing
 - XML-based WSDL(Web Service Description Language) is used as an IDL(interface Description Language)



JAX-RPC and JAX-WS

1. A Java program executes a method on a stub (local object representing the remote service)
2. The stub executes routines in the JAX-RPC Runtime System (RS)
3. The RS converts the remote method invocation into a SOAP message
4. The RS transmits the message as an HTTP request

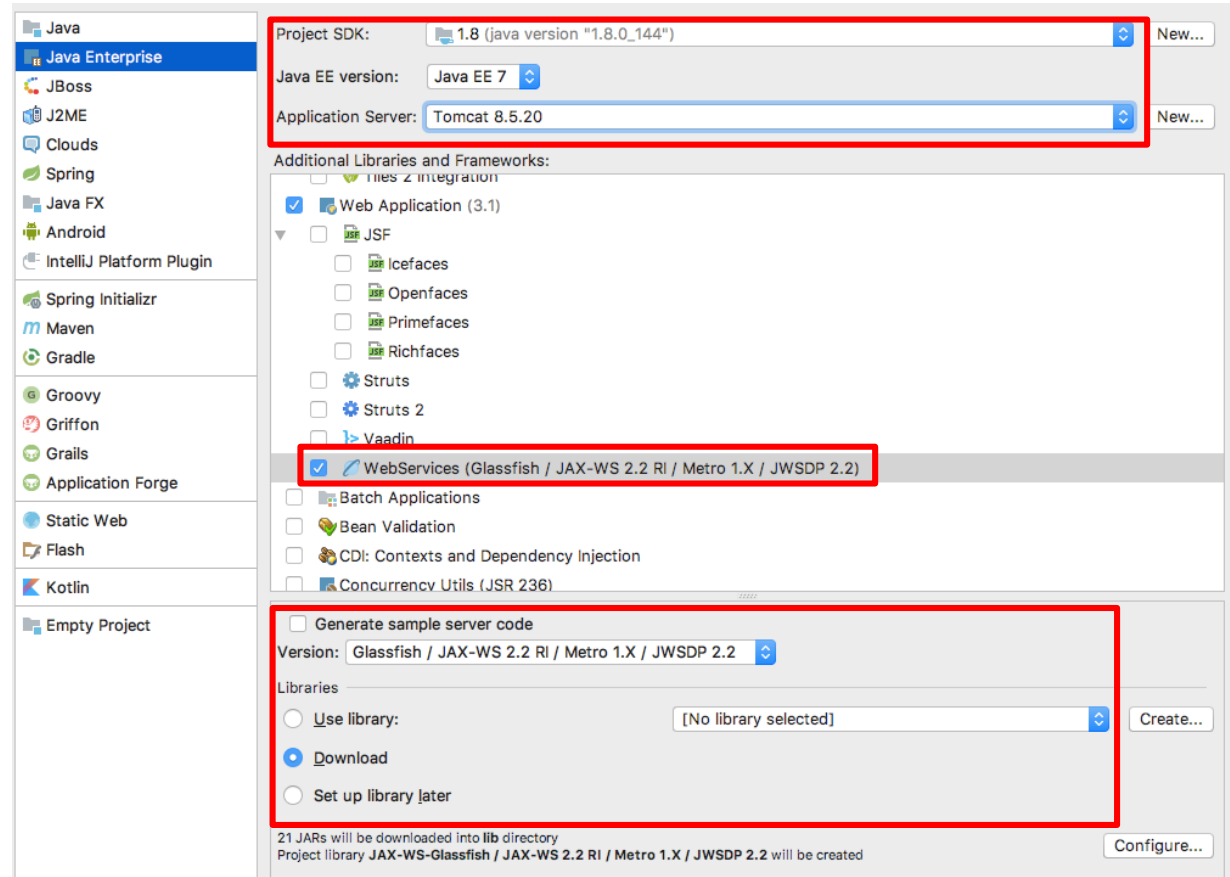


Step to build Web Service Greeting Server

- When the client accesses the web server, the greeting web service displays the "Hello, ee614" string data set by the server on the client screen.
- 1) Create Web Service Project and Project Setting
- 2) Interface Description with JAX-WS Annotation
- 3) Add service endpoint
- 4) Build Service
- 5) Deploy service with IDE
- 6) Check Web Service description with XML

1) Create Web Service Project

- Java Enterprise → Web Application → Web Services
- Select Application Server as 'Tomcat'
- Disable 'Generate sample server code'
- Select 'Glassfish / JAX-WS 2.2 RI / ...'



1) Project Settings

- File → Project Structure → Problems → [Fix] → Add 'JAX-WS-Glassfish / ...' to the artifact



The screenshot shows the 'Project Structure' dialog in IntelliJ IDEA. On the left, the 'Problems' tab is selected, indicated by a blue bar and a circled '1'. The main area displays a list of problems. The first problem is: '1. Artifact **RPC:war exploded**: library 'JAX-WS-Glassfish / JAX-WS 2.2 RI / Metro 1.X / JWSDP 2.2' required for module 'RPC' is missing from the artifact **[Fix]**'. Below this problem, a list of suggested fixes is shown in a white box with a blue header. The fixes are: 'Add 'JAX-WS-Glassfish / JAX-WS 2.2 RI / Metro 1.X / JWSDP 2.2' to the artifact', 'Change 'JAX-WS-Glassfish / JAX-WS 2.2 RI / Metro 1.X / JWSDP 2.2' scope to 'Test'', and 'Change 'JAX-WS-Glassfish / JAX-WS 2.2 RI / Metro 1.X / JWSDP 2.2' scope to 'Provided''.

Project Settings

- Project
- Modules
- Libraries
- Facets
- Artifacts

Platform Settings

- SDKs
- Global Libraries

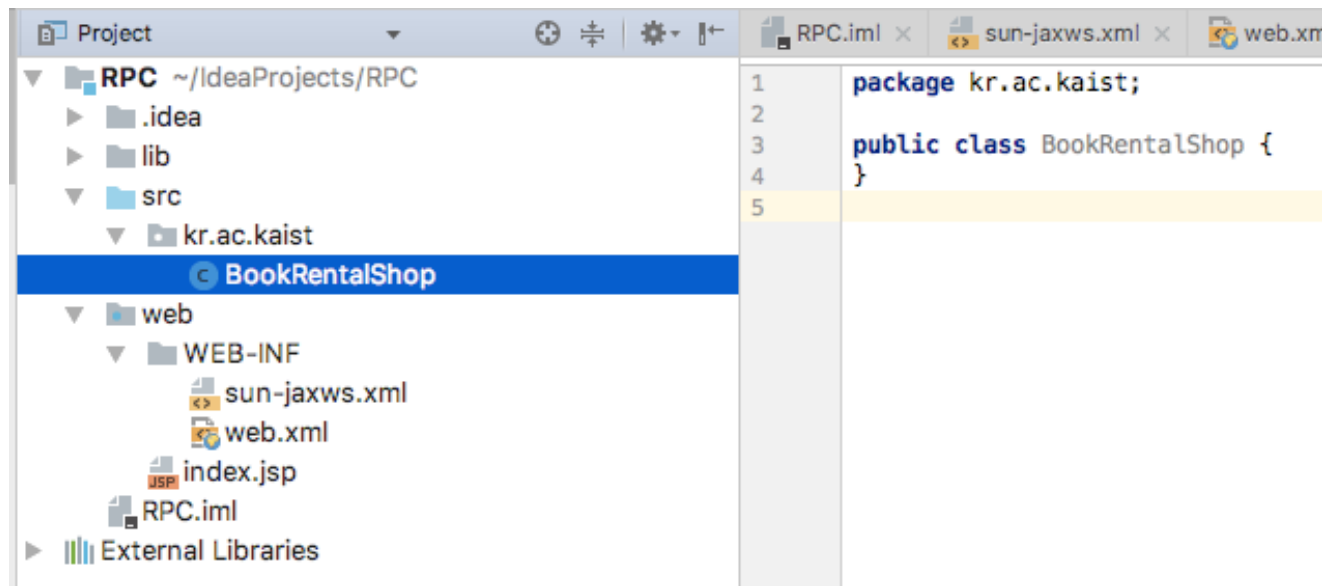
Problems 1

1. Artifact **RPC:war exploded**: library 'JAX-WS-Glassfish / JAX-WS 2.2 RI / Metro 1.X / JWSDP 2.2' required for module 'RPC' is missing from the artifact **[Fix]**

- Add 'JAX-WS-Glassfish / JAX-WS 2.2 RI / Metro 1.X / JWSDP 2.2' to the artifact
- Change 'JAX-WS-Glassfish / JAX-WS 2.2 RI / Metro 1.X / JWSDP 2.2' scope to 'Test'
- Change 'JAX-WS-Glassfish / JAX-WS 2.2 RI / Metro 1.X / JWSDP 2.2' scope to 'Provided'

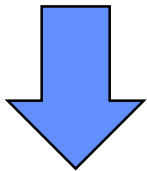
2) RPC Server Programming

- Under 'src' directory, create Package named 'kr.ac.kaist'
- Under package 'kr.ac.kaist', create Java Class named 'BookRentalShop'



2) Interface Description with JAX-WS Annotation

```
public class BookRentalShop {  
  
    public String greeting(String username) {  
        System.out.println("User " + username + " Entered!");  
        return "Hello, " + username;  
    }  
}
```



Code at Lab Materials
– Lab6/src/BookRentalShop0.java

@WebService() JAX-WS Annotation API for declaring this class as a Web Service Port

```
public class BookRentalShop {  
  
    @WebMethod JAX-WS Annotation API for declaring this method as a Web Service operation  
    public String greeting(String username) {  
        System.out.println("User " + username + " Entered!");  
        return "Hello, " + username;  
    }  
}
```

3) Add service endpoint

- web → WEB-INF → sun-jaxws.xml

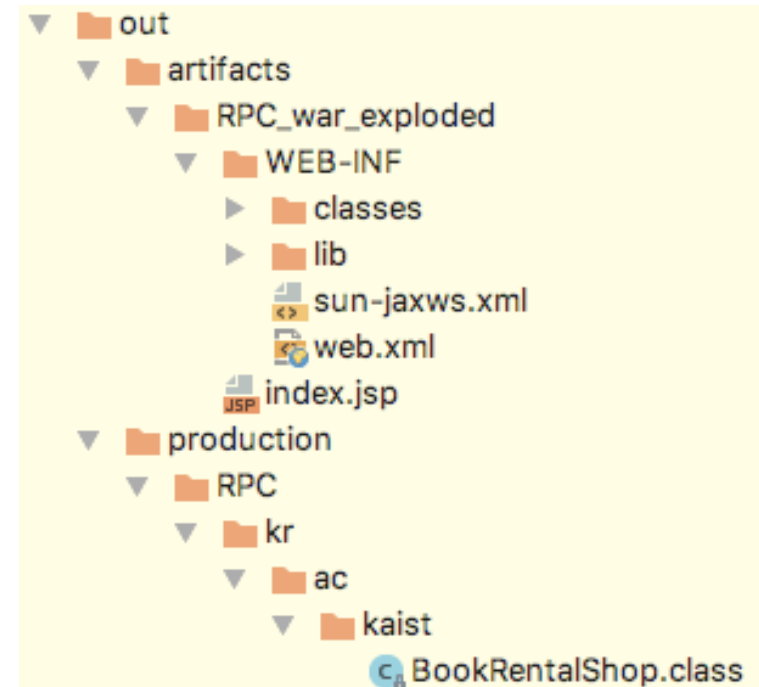
```
<?xml version="1.0" encoding="UTF-8"?>

<endpoints xmlns='http://java.sun.com/xml/ns/jax-ws/ri/runtime' version='2.0'>
  <endpoint
    name='BookRentalShop'
    implementation='kr.ac.kaist.BookRentalShop'
    url-pattern='/services/BookRentalShop' />
</endpoints>
```

Code at Lab Materials
– Lab6/src/sun-jaxws.xml

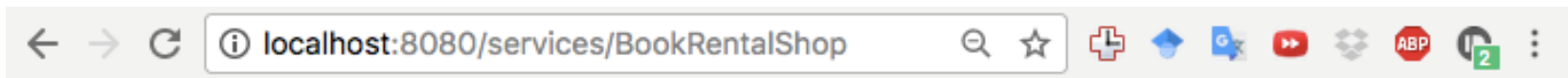
4) Build Service

- [Compile JAVA Class]
Build → Build Project
- [Build JAX-WS] Build → Build Artifacts → war exploded → Build
- Check out directory 'out'



5) Deploy service with IDE

- Run → Run 'Tomcat'
- Access a page
'<http://localhost:8080/services/BookRentalShop>'
with your web browser



웹 서비스

끝점	정보
서비스 이름: { http://kaist.ac.kr/ }BookRentalShopService	주소: http://localhost:8080/services/BookRentalShop
포트 이름: { http://kaist.ac.kr/ }BookRentalShopPort	WSDL: http://localhost:8080/services/BookRentalShop?wsdl
	구현 클래스: kr.ac.kaist.BookRentalShop

6) Cheke Web Service description with XML (WSDL, Web Services Description Language)

- With WSDL, the client can figure out
 - 1) Specification of available service
 - 2) Its input/output interface

← → ↻ ⓘ localhost:8080/services/BookRentalShop?wsdl 🔍 ☆ ⛶ ⚙️ 📄 📱 📧 📧 📧 📧 📧 📧

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!--
  Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2.7-b01 svn-revision#{$svn.Last.
-->
<!--
  Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2.7-b01 svn-revision#{$svn.Last.
-->
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
  xmlns:wsp="http://www.w3.org/ns/ws-policy" xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://kaist.ac.kr/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://kaist.ac.kr/" name="BookRentalShopService">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://kaist.ac.kr/" schemaLocation="http://localhost:8080/services/BookRentalShop?
      xsd=1"/>
    </xsd:schema>
  </types>
  <message name="greeting">
    <part name="parameters" element="tns:greeting"/>
  </message>
  <message name="greetingResponse">
    <part name="parameters" element="tns:greetingResponse"/>
  </message>
  <portType name="BookRentalShop">
    <operation name="greeting">
      <input wsam:Action="http://kaist.ac.kr/BookRentalShop/greetingRequest" message="tns:greeting"/>
      <output wsam:Action="http://kaist.ac.kr/BookRentalShop/greetingResponse" message="tns:greetingResponse"/>
    </operation>
  </portType>
  <binding name="BookRentalShopPortBinding" type="tns:BookRentalShop">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="greeting">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="BookRentalShopService">
    <port name="BookRentalShopPort" binding="tns:BookRentalShopPortBinding">
      <soap:address location="http://localhost:8080/services/BookRentalShop"/>
    </port>
  </service>
</definitions>
```


WSDL: Message, Port and Operation

```
▼ <message name="greeting">
  <part name="parameters" element="tns:greeting"/>
</message>
▼ <message name="greetingResponse">
  <part name="parameters" element="tns:greetingResponse"/>
</message>
▼ <portType name="BookRentalShop">
  ▼ <operation name="greeting">
    <input wsam:Action="http://kaist.ac.kr/BookRentalShop/greetingRequest"
      message="tns:greeting"/>
    <output wsam:Action="http://kaist.ac.kr/BookRentalShop/greetingResponse"
      message="tns:greetingResponse"/>
  </operation>
</portType>
```



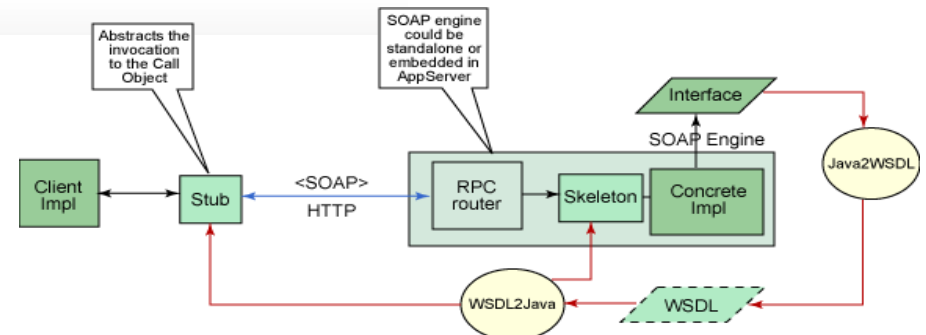
WSDL: SOAP Binding and Service

- To utilize ‘SOAP Envelopment’

```

<binding name="BookRentalShopPortBinding" type="tns:BookRentalShop">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
  <operation name="greeting">
    <soap:operation soapAction="" />
    <input>
      <soap:body use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>
</binding>
<service name="BookRentalShopService">
  <port name="BookRentalShopPort" binding="tns:BookRentalShopPortBinding">
    <soap:address location="http://localhost:8080/services/BookRentalShop" />
  </port>
</service>

```



Greeting Web Service Client with IntelliJ

1. Modify java configuration.

- cd [JAVA_HOME]/jre/lib/
- vi jaxp.properties
- insert “javax.xml.accessExternalSchema = all”

1

```
SeongHwanui-MacBook-Pro:lib jihwankim$ cd /Library/Java/JavaVirtualMachines/  
jdk1.8.0_111.jdk/Contents/Home/jre/lib/
```

2

```
SeongHwanui-MacBook-Pro:lib jihwankim$ vi jaxp.properties
```

3

```
javax.xml.accessExternalSchema = all
```

```
~
```

```
~
```

```
~
```

```
~
```

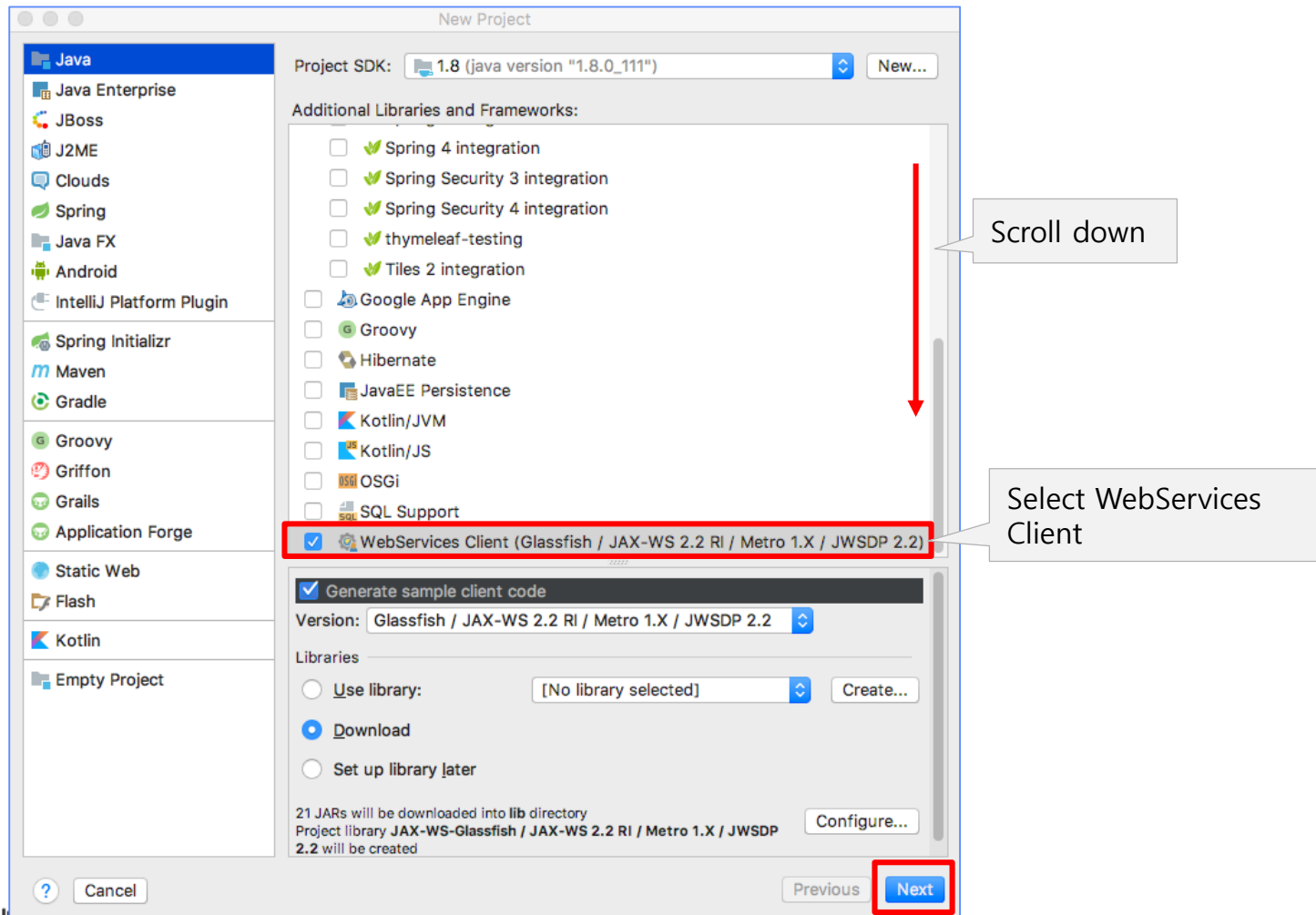
```
~
```

```
~
```

```
~
```

```
~
```

2. Create project for Client program. (Disable 'Generate sample client code')



3. After project indexing is complete, insert Web service wsdl url and Package prefix.

Generate Java Code From Wsdl

Web service wsdl url

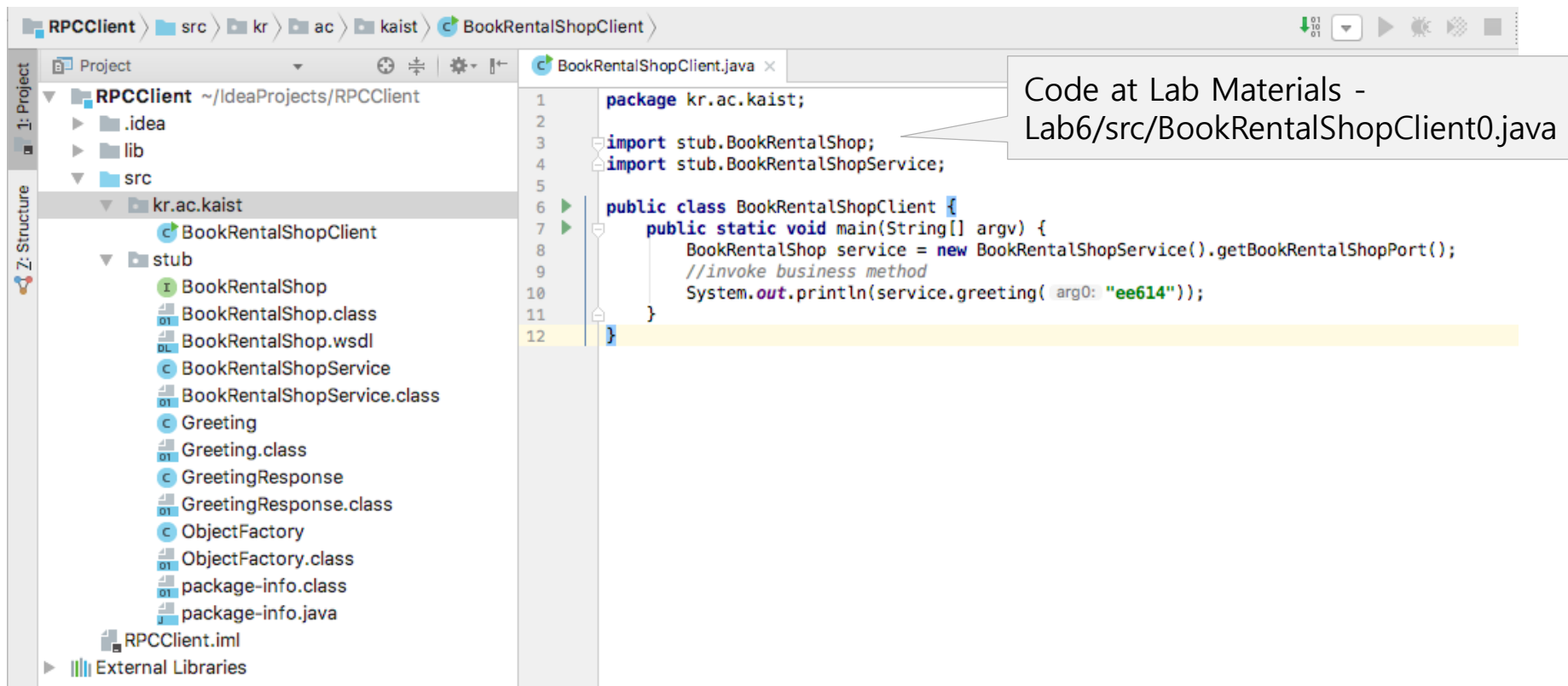
Output path

Package prefix

☒ Allow extensions

4. Create Package and Class.

- Under 'src' directory, create Package named 'kr.ac.kaist'
- Under package 'kr.ac.kaist', create Java Class named 'BookRentalShopClient'



The screenshot shows an IDE with the project structure on the left and the code for `BookRentalShopClient.java` on the right. The project structure shows a package `kr.ac.kaist` under the `src` directory, containing the `BookRentalShopClient` class. The code in the editor is as follows:

```
1 package kr.ac.kaist;
2
3 import stub.BookRentalShop;
4 import stub.BookRentalShopService;
5
6 public class BookRentalShopClient {
7     public static void main(String[] argv) {
8         BookRentalShop service = new BookRentalShopService().getBookRentalShopPort();
9         //invoke business method
10        System.out.println(service.greeting( argv0: "ee614"));
11    }
12 }
```

A callout box points to the import statements, stating: "Code at Lab Materials - Lab6/src/BookRentalShopClient0.java".

KAIST Korea Advanced Institute of
Science and Technology

Auto-generated Interface w/ reference to WSDL (Skeleton)

- Check out src/stub/BookRentalShop

```
@WebService(name = "BookRentalShop", targetNamespace = "http://kaist.ac.kr/")
@XmlSeeAlso({
    ObjectFactory.class
})
public interface BookRentalShop {

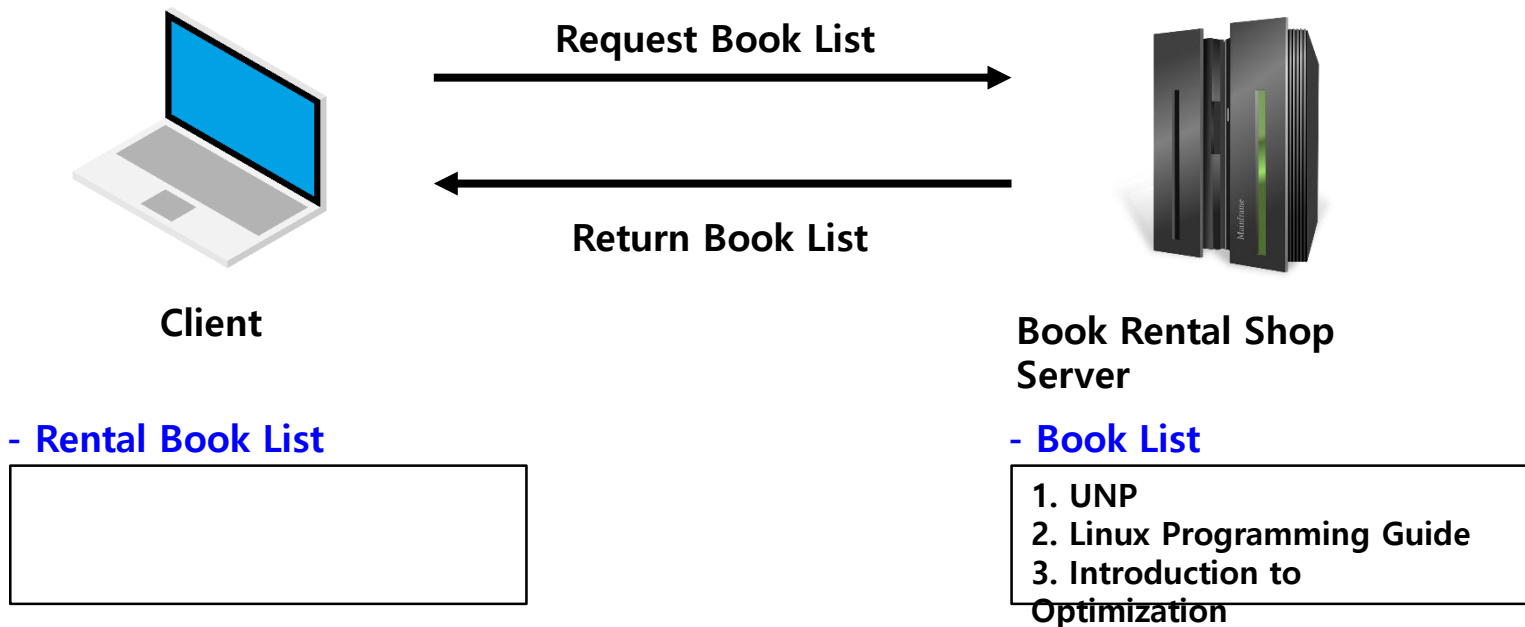
    /**
     *
     * @param arg0
     * @return
     * returns java.lang.String
     */
    @WebMethod
    @WebResult(targetNamespace = "")
    @RequestWrapper(localName = "greeting", targetNamespace = "http://kaist.ac.kr/", className = "stub.Greeting")
    @ResponseWrapper(localName = "greetingResponse", targetNamespace = "http://kaist.ac.kr/", className = "stub.GreetingResponse")
    @Action(input = "http://kaist.ac.kr/BookRentalShop/greetingRequest", output = "http://kaist.ac.kr/BookRentalShop/greetingResponse")
    public String greeting(
        @WebParam(name = "arg0", targetNamespace = "")
        String arg0);
}
```

Archiving BookRentalShop Web Service as WAR and Deploy with Tomcat

WAR (file format) (Web application ARchive)
: File format used to package Java Web applications

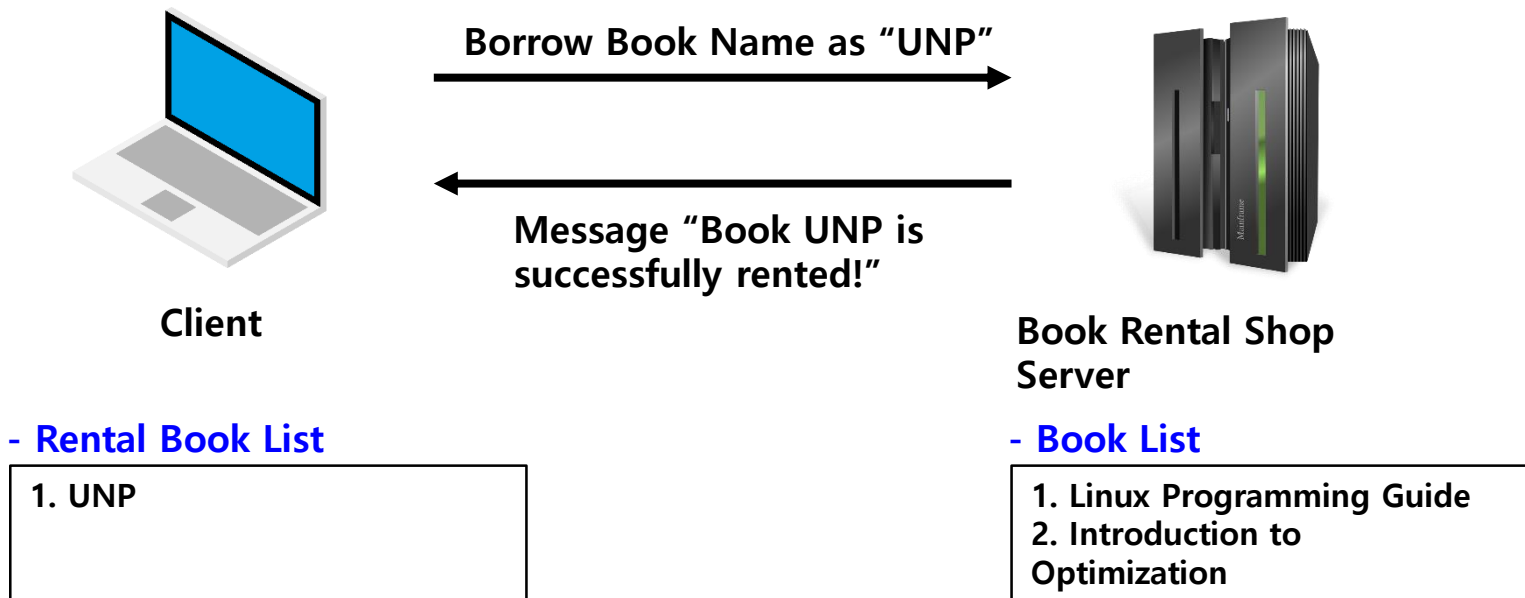
Book Rental Shop Web Service (1)

- The book rental shop web service provides the function to **request a list of books**, borrow books, and return books.



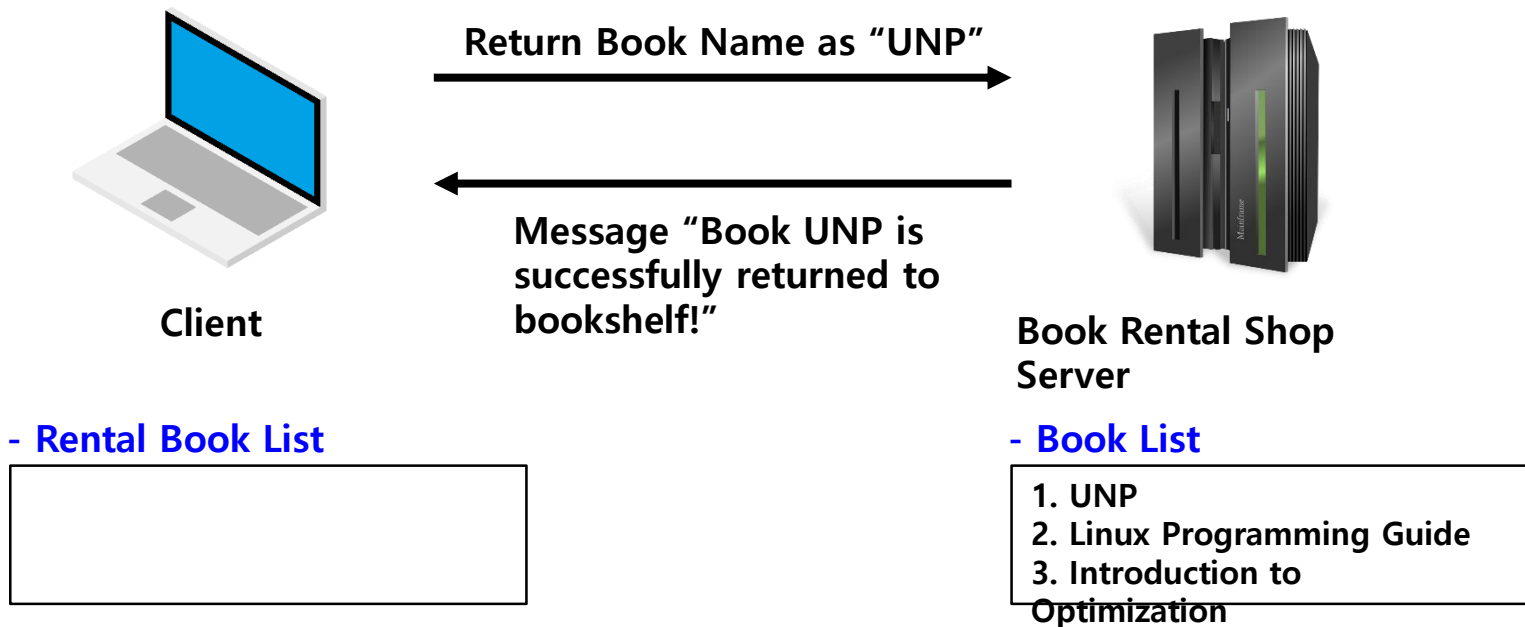
Book Rental Shop Web Service (2)

- The book rental shop web service provides the function to request a list of books, **borrow books**, and return books.



Book Rental Shop Web Service (3)

- The book rental shop web service provides the function to request a list of books, borrow books, and **return books**.

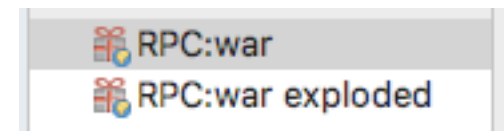
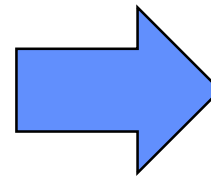
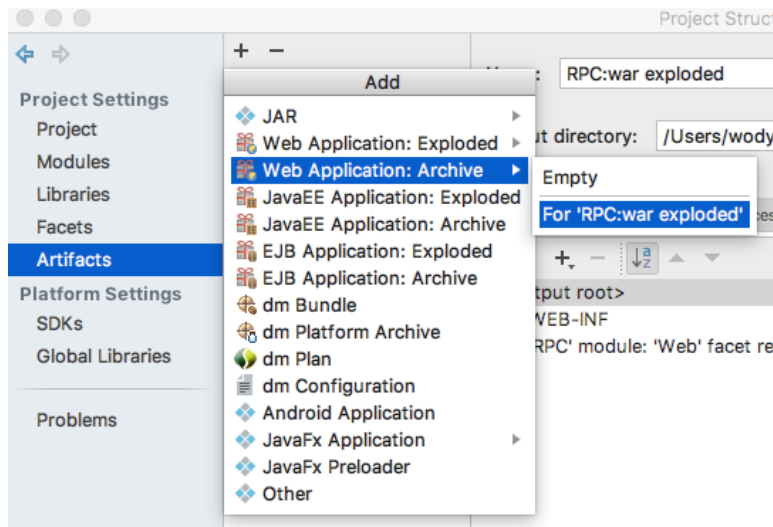


Steps for Archiving BookRentalShop Web Service as WAR and Deploy with Tomcat

- In this practice, we will register the book rental shop web service on the Tomcat server and use it as a client.
- 1) Add Artifacts Rule
- 2) Build WAR File
- 3) Startup Tomcat
- 4) Deploy service
- 5) Access New Service URL and Update Client
- 6) Check created folders and files

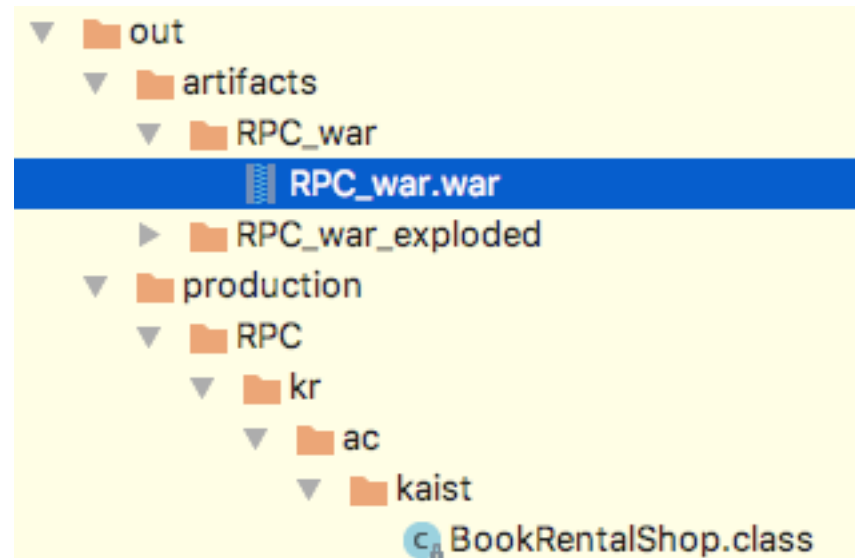
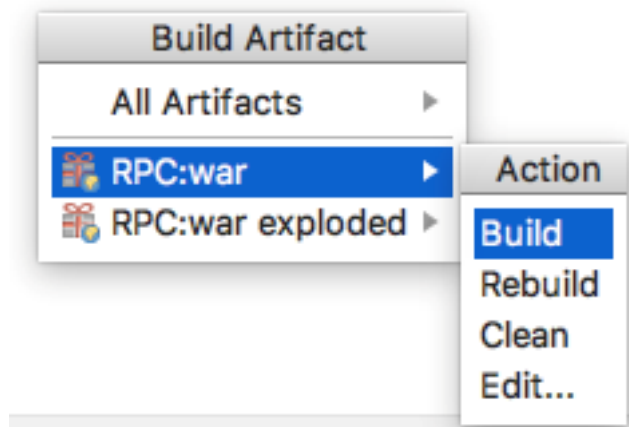
Add Artifacts Rule

- File → Project Structure → Artifacts → + → Web Application: Archive → For 'war exploded'



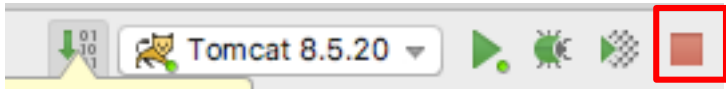
Build WAR File

- Build → Build Artifacts → war → Build
- Or build All Artifacts
- WAR file will be created in artifacts dir



Startup Tomcat

- Start up tomcat server (startup.sh)
 - Before start up tomcat server, we should turn off tomcat instance launched by IntelliJ



```
SeongHwanKimui-MacBook-Air:Tomcat wody34$ bin/startup.sh
Using CATALINA_BASE:   /Library/Tomcat
Using CATALINA_HOME:   /Library/Tomcat
Using CATALINA_TMPDIR: /Library/Tomcat/temp
Using JRE_HOME:        /Library/Java/JavaVirtualMachines/jdk1.8.0_144.jdk/Contents/Home
Using CLASSPATH:       /Library/Tomcat/bin/bootstrap.jar:/Library/Tomcat/bin/tomcat-juli.jar
Tomcat started.
SeongHwanKimui-MacBook-Air:Tomcat wody34$
```



Tomcat Web Application Manager

Message: OK

Manager

List Applications

HTML Manager Help

Manager Help

Server Status

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	<div>StartStopReloadUndeploy</div> <div>Expire sessions with idle ≥ 30 minutes</div>
/docs	None specified	Tomcat Documentation	true	0	<div>StartStopReloadUndeploy</div> <div>Expire sessions with idle ≥ 30 minutes</div>
					<div>StartStopReloadUndeploy</div>

Deploy service

- In panel 'WAR file to deploy', select built war file to deploy

Deploy

Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file URL:

WAR or Directory URL:

Deploy

WAR file to deploy

Select WAR file to upload RPC_war.war

Deploy

Sample file at Lab Materials
– Lab6/src/RPC_war.war

Deploy service

- In application panel, new web application named 'RPC_war' is created and is in start status

Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/RPC_war	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy Service

- When you are unavailable to access Tomcat in GUI-Manager
 - Move war file into [TOMCAT_HOME]/webapps directory

```
[SeongHwanKimui-MacBook-Air:Tomcat wody34$ ls -al
total 0
drwxr-xr-x   9 wody34  admin   306   9 19 11:36 .
drwxr-xr-x  11 wody34  admin   374   9 14 10:08 ..
drwxr-x---  17 wody34  admin   578   9 14 10:08 bin
drwx----- 13 wody34  admin   442   9 20 16:54 conf
drwxr-x---  27 wody34  admin   918   8  3 06:36 lib
drwxr-x---  21 wody34  admin   714   9 20 21:00 logs
drwxr-x---   3 wody34  admin   102   9 19 11:42 temp
drwxr-x---   9 wody34  admin   306   9 20 21:05 webapps
drwxr-x---   3 wody34  admin   102   9 14 10:09 work
```

```
[SeongHwanKimui-MacBook-Air:Tomcat wody34$ cp /Users/wody34/IdeaProjects/RPC/out/
artifacts/RPC_war/RPC_war.war webapps/
_
```

New Service URL

- `http://localhost:8080/RPC_war/services/BookRentalShop`

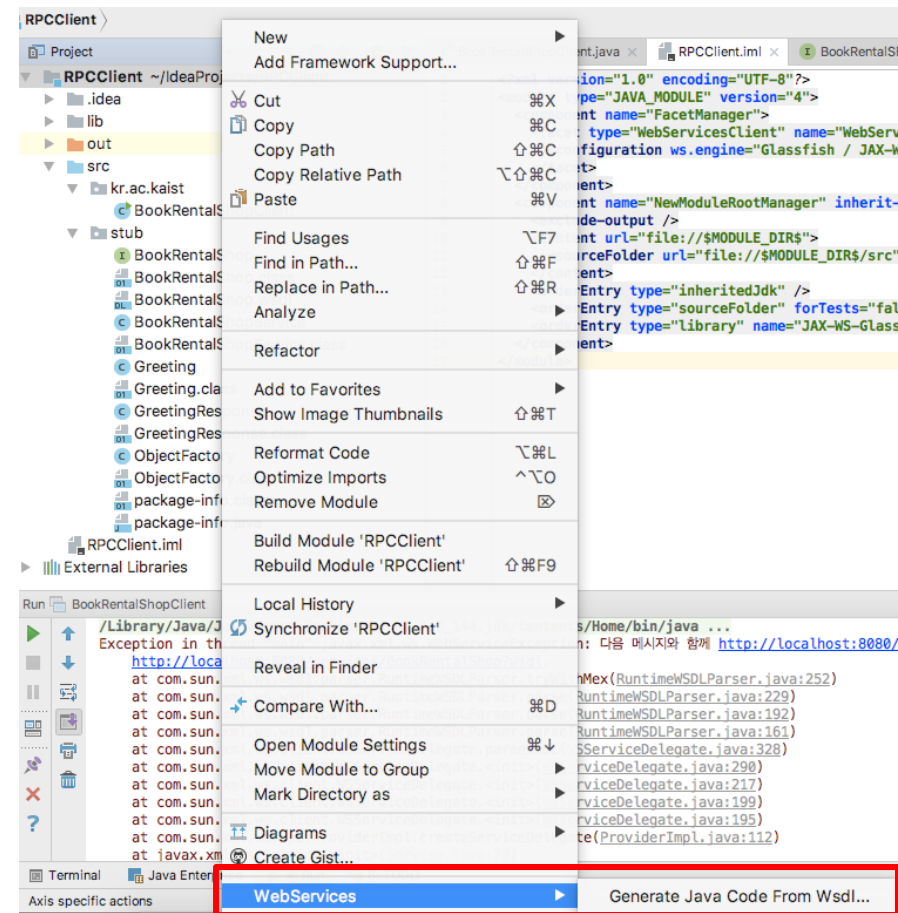
← → ↻ ⓘ localhost:8080/RPC_war/services/BookRentalShop

웹 서비스

끝점	정보
서비스 이름: {http://kaist.ac.kr}BookRentalShopService 포트 이름: {http://kaist.ac.kr}BookRentalShopPort	주소: <code>http://localhost:8080/RPC_war/services/BookRentalShop</code> WSDL: http://localhost:8080/RPC_war/services/BookRentalShop?wsdl 구현 클래스: <code>kr.ac.kaist.BookRentalShop</code>

Client update

- Previous client is no longer available
 - Service URL is changed!
- Right click on Project directory and Generate Java Code from WSDL again with new URL after clean up stub directory
- The procedure is also needed when service is changed (new method, class, etc...)



Update Server and Client with complete code and execution

- Book Store Scenario is completed!

Code at Lab Materials

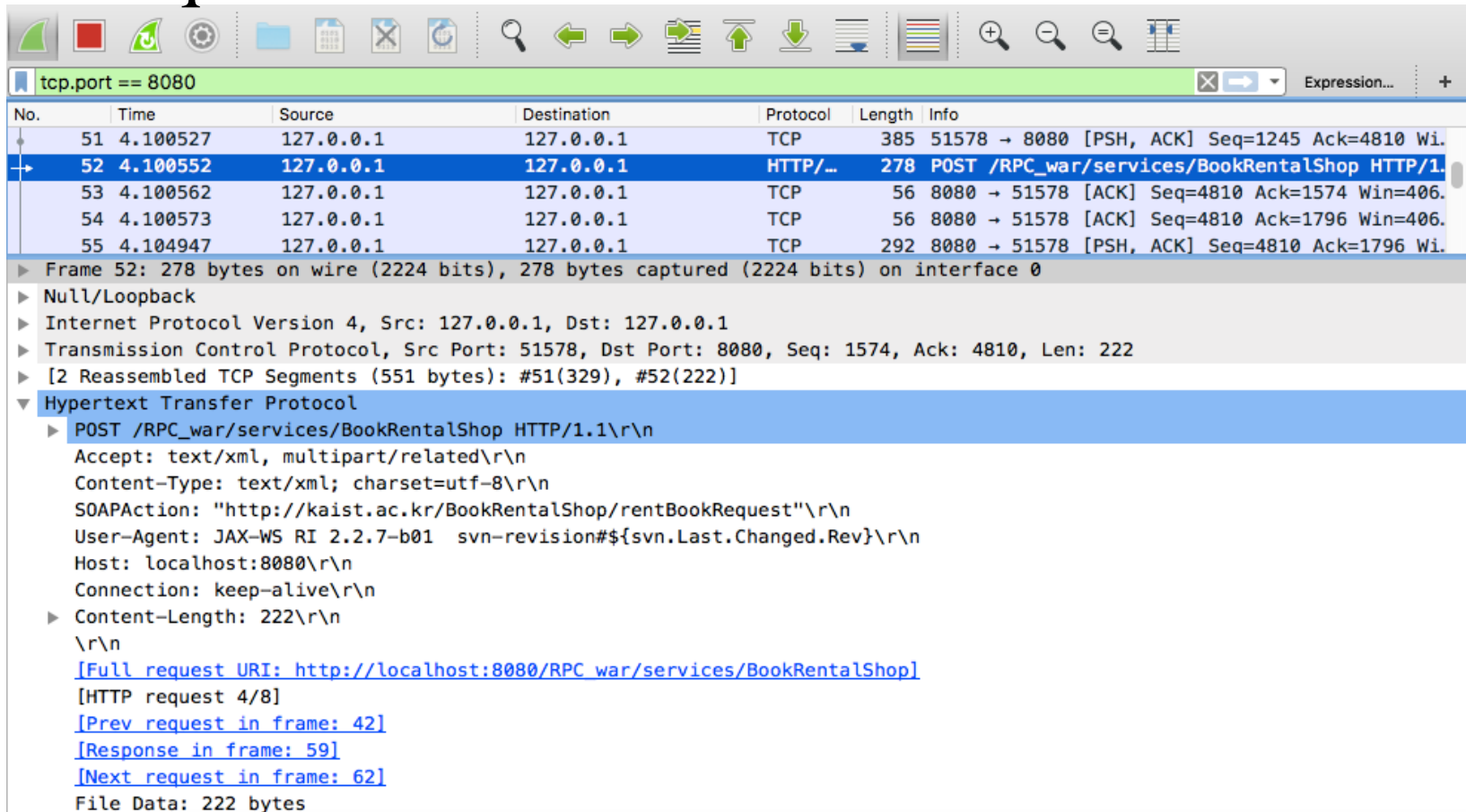
- Lab6/src/BookRentalShop1.java
- Lab6/src/BookRentalShopClient1.java

```
Hello, shkim
Retreived Book List:
- UNP
- Linux Programming Guide
- Introduction to Optimization
There are no book named Machine Learning
Book UNP is successfully rented!
Retreived Book List:
- Linux Programming Guide
- Introduction to Optimization
I have no book named Software Engineering
Book UNP is successfully returned to bookshelf!
Retreived Book List:
- Linux Programming Guide
- Introduction to Optimization
- UNP
```

Packet Analysis

Protocol of SOAP based RPC

- Request is sent over HTTP Protocol



The image shows a Wireshark packet capture interface. The top toolbar includes various icons for file operations, navigation, and analysis. Below the toolbar, a green filter bar displays the expression `tcp.port == 8080`. The main packet list pane shows several packets, with packet 52 selected. The details pane for packet 52 is expanded, showing the following information:

- Frame 52:** 278 bytes on wire (2224 bits), 278 bytes captured (2224 bits) on interface 0
- Null/Loopback**
- Internet Protocol Version 4, Src:** 127.0.0.1, **Dst:** 127.0.0.1
- Transmission Control Protocol, Src Port:** 51578, **Dst Port:** 8080, **Seq:** 1574, **Ack:** 4810, **Len:** 222
- [2 Reassembled TCP Segments (551 bytes): #51(329), #52(222)]**
- Hypertext Transfer Protocol**
 - POST /RPC_war/services/BookRentalShop HTTP/1.1\r\n**
 - Accept:** text/xml, multipart/related\r\n
 - Content-Type:** text/xml; charset=utf-8\r\n
 - SOAPAction:** "http://kaist.ac.kr/BookRentalShop/rentBookRequest"\r\n
 - User-Agent:** JAX-WS RI 2.2.7-b01 svn-revision#{\$svn.Last.Changed.Rev}\r\n
 - Host:** localhost:8080\r\n
 - Connection:** keep-alive\r\n
 - Content-Length:** 222\r\n
 - \r\n**

At the bottom of the details pane, there are several links:

- [\[Full request URI: http://localhost:8080/RPC_war/services/BookRentalShop\]](#)
- [\[HTTP request 4/8\]](#)
- [\[Prev request in frame: 42\]](#)
- [\[Response in frame: 59\]](#)
- [\[Next request in frame: 62\]](#)

The status bar at the bottom indicates **File Data: 222 bytes**.

Protocol of SOAP based RPC

- Message is packaged with SOAP Envelopment

52	4.100552	127.0.0.1	127.0.0.1	HTTP/...	278
53	4.100562	127.0.0.1	127.0.0.1	TCP	56
54	4.100573	127.0.0.1	127.0.0.1	TCP	56
55	4.104947	127.0.0.1	127.0.0.1	TCP	292

▶ Content-Length: 222\r\n

\r\n

[Full request URI: http://localhost:8080/RPC_war/services/BookRentalShop]

[HTTP request 4/8]

[Prev request in frame: 42]

[Response in frame: 59]

[Next request in frame: 62]

File Data: 222 bytes

▼ eXtensible Markup Language

▶ <?xml

▼ <S:Envelope

xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"

▼ <S:Body>

▼ <ns2:rentBook

xmlns:ns2="http://kaist.ac.kr/"

▼ <arg0>

Machine Learning

</arg0>

</ns2:rentBook>

</S:Body>

</S:Envelope>

49	4.098229	127.0.0.1	127.0.0.1	HTTP/...	61
50	4.098245	127.0.0.1	127.0.0.1	TCP	56
51	4.100527	127.0.0.1	127.0.0.1	TCP	385

▶ HTTP chunked response

File Data: 320 bytes

▼ eXtensible Markup Language

▶ <?xml

▼ <S:Envelope

xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"

▼ <S:Body>

▼ <ns2:getBookListResponse

xmlns:ns2="http://kaist.ac.kr/"

▼ <return>

Linux Programming Guide

</return>

▼ <return>

Introduction to Optimization

</return>

▼ <return>

UNP

</return>

</ns2:getBookListResponse>

</S:Body>

</S:Envelope>

Lab 6. SOAP based RPC Programming

- Read text materials and test practices
- Use your notebook PC to examine context and produce the source code when you finish the successful practice.
- Objectives
 - Understand the process of SOAP and WSDL
 - Learn the how to analyze network packet using wireshark
 - Understand packet structure of SOAP based RPC