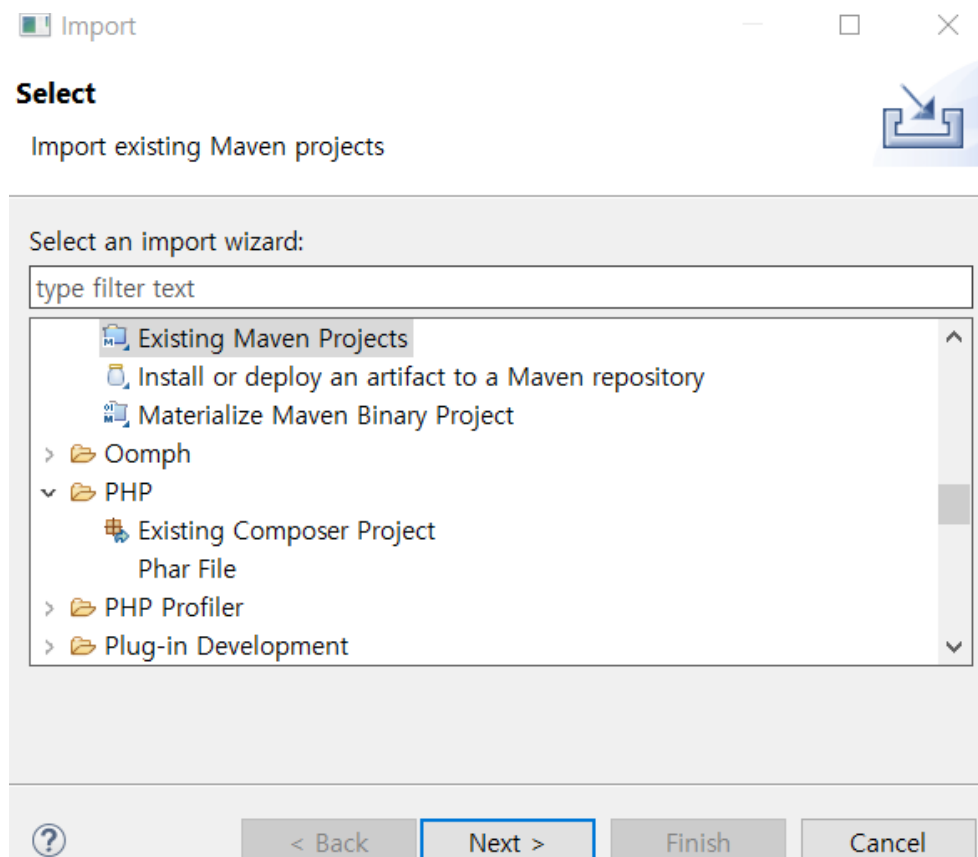


## 개발환경 03

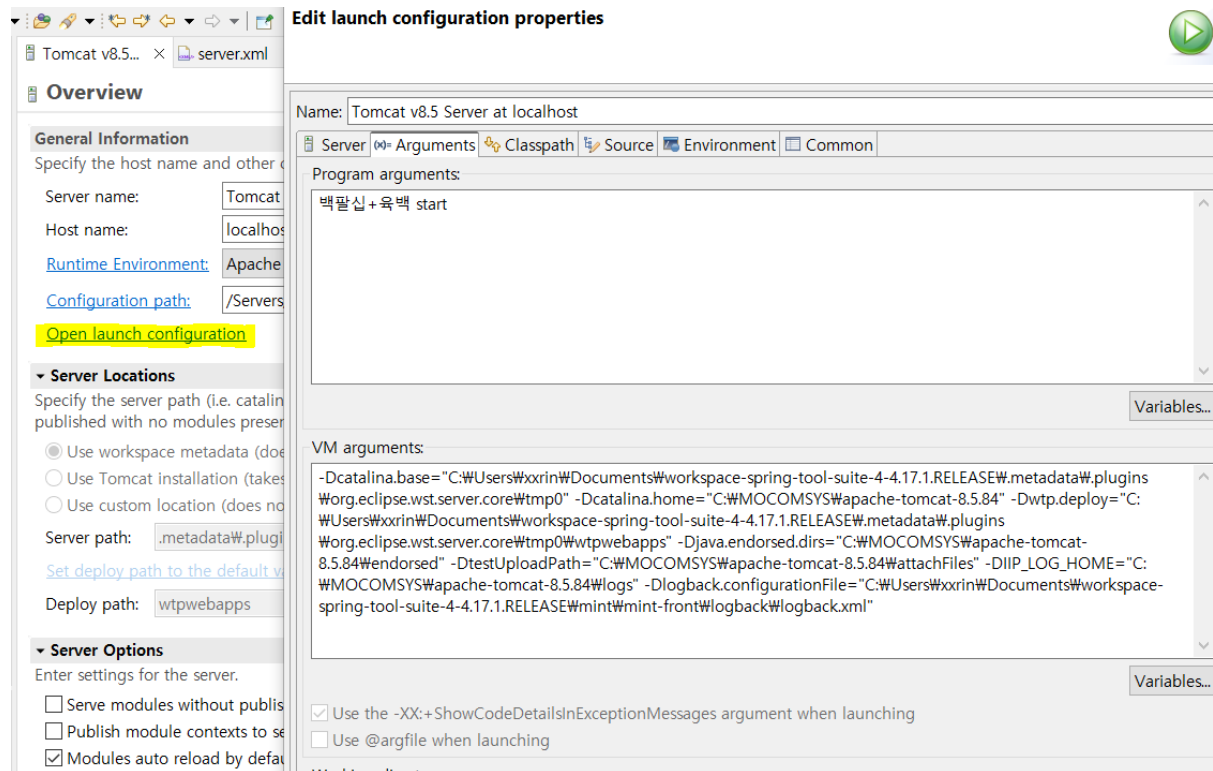
자바 홈과 패스를 잡아줬다면 개발 툴에서의 환경도 잡아줘야 한다. 이클립스를 그냥 설치했을 시 자바 17버전이 자동으로 설치가 된다. 이 때 이클립스 내에서 내가 애플리케이션을 실행할 때 몇 버전으로 진행할지에 대해서 세팅해주면 된다.



Svn 설치된 뒤 있다고 생각하고 checkout을 해준 뒤 convert to maven을 통해 메이븐 프로젝트를 암시해 준다. 그 후 프로젝트 우클릭 → import → Existing Maven Projects → next 후 선택 후 Finish를 통해 프로젝트를 바깥으로 꺼내준다.



## VM ARGUMENTS



open launch configuration을 클릭하고 arguments에서 vmarg를 설정할 수 있다. 애플리케이션의 실행 환경을 조성하기 위한 것인데 자세한 설명은 아래에 있다.

애플리케이션의 실행 환경을 구성하고 조정하기 위해 사용된다.

Dcatalina.base="/Users/whoana/DEV/workspace/.metadata/.plugins/org.eclipse.wst.server.core/tmp1"  
→ 여러 개를 띄우기 위한 base

-Dcatalina.home="/Users/whoana/DEV/apps/apache-tomcat-8.5.56"

-Dwtp.deploy="C:\MOCOMSYS\apache-tomcat-8.5.84\webapps"  
//실행하면 이 안에 파일들이 생김

-Djava.endorsed.dirs="/Users/whoana/DEV/apps/apache-tomcat-8.5.56/endorsed"  
//-- end of eclipse auto config 여기까지 이클립스가 자동으로 잡아주는 것

Dstage="local"  
-DIIP\_LOG\_HOME="C:\MOCOMSYS\apache-tomcat-8.5.84\logs"  
// 프로그램 내부로 나는 지금 로컬에서 개발하고 있다... -D  
//프로그램에서는 GETPROPERTIES로 값을 꺼낼 수 있음 GETSTAGE를 주면 LOCAL이란 값이 나온다  
//필수적인 요소 로그를 어디에 쓸지 PATH 설정

-DtestUploadPath="/Users/whoana/DEV/apps/apache-tomcat-8.5.56/upload"

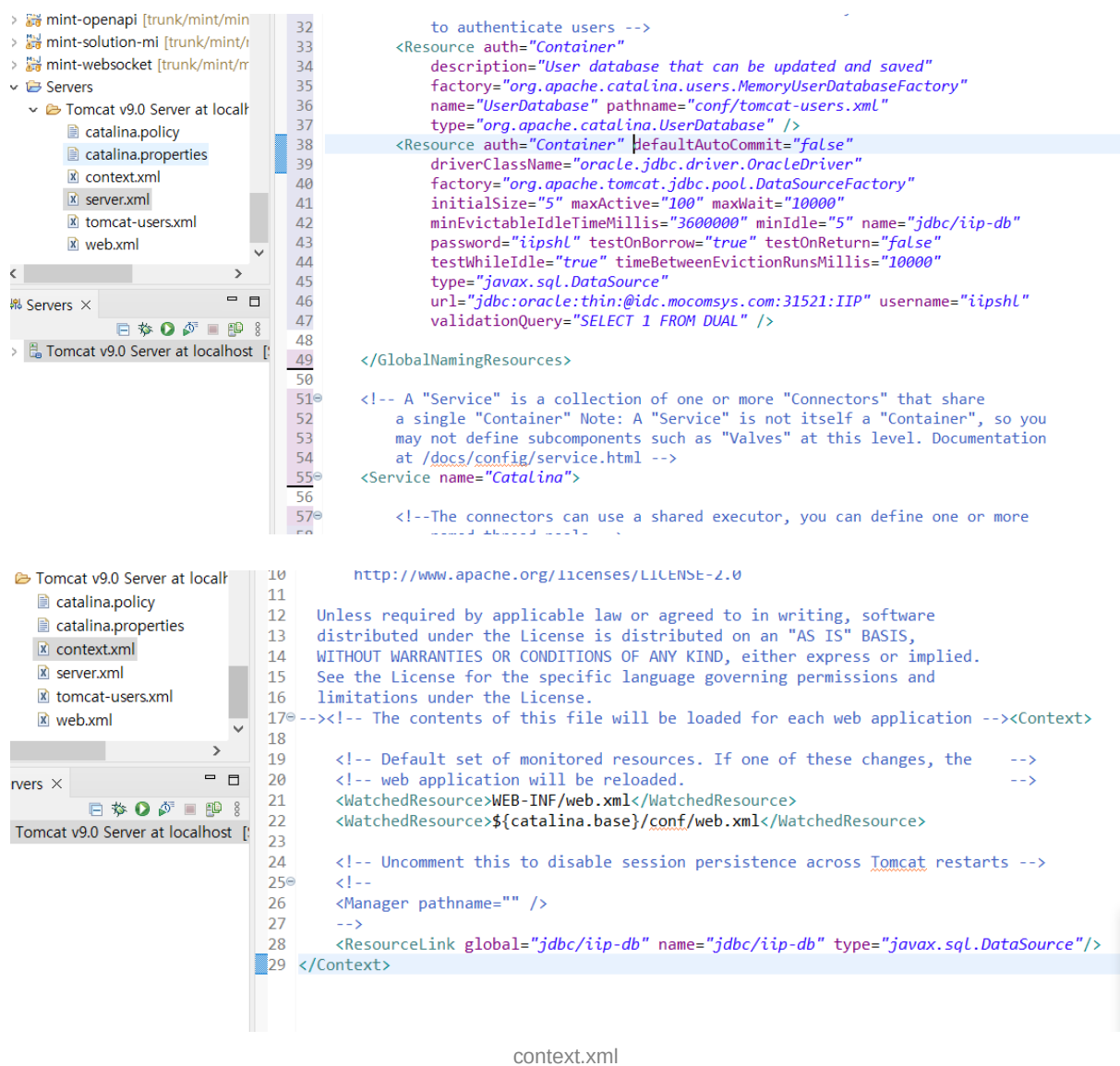
실제DB에 있는 걸 무시하고 내 로컬 저장소에 파일을 올리겠다는 뜻 그냥업로드경로설정이다

-Dlogback.configurationFile="C:\Users\xxrin\Documents\workspace-spring-tool-suite-4-4.17.1.RELEASE\mint\mint-front\logback\logback.xml"

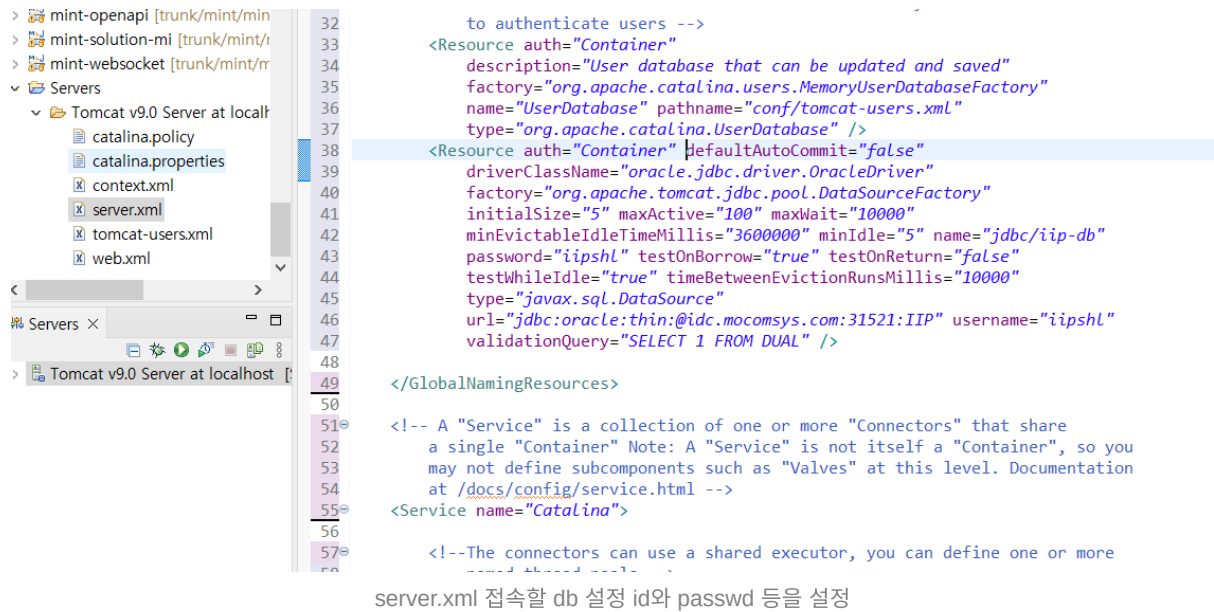
로그를 이곳에 쓰겠다는 의미이다.

## db 설정

db 설정은 필수적인 것임. 설정하지 않으면 접속이 안됨. ( 톰캣 서버에 ojdbc 라이브리는 꼭 설치해야 한다)



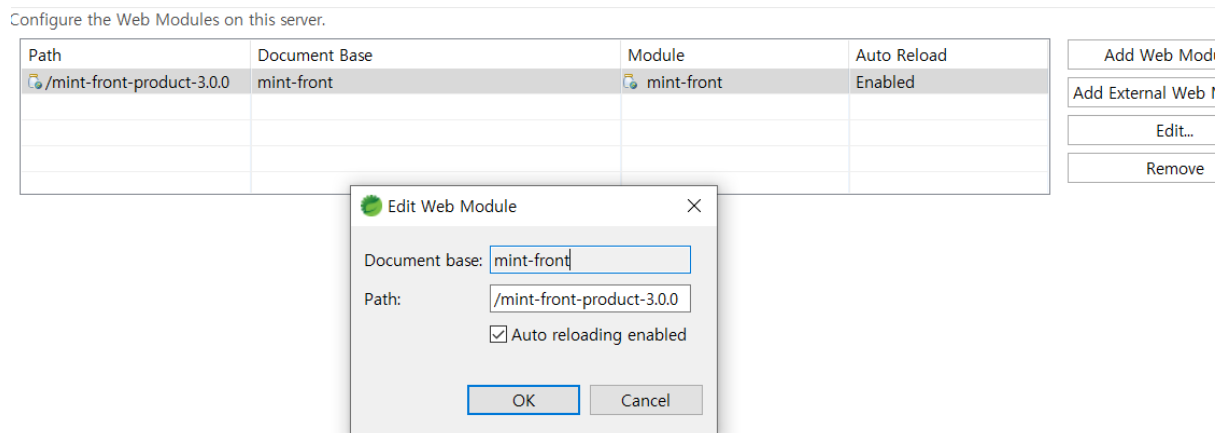
```
32 to authenticate users -->
33 <Resource auth="Container"
34 description="User database that can be updated and saved"
35 factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
36 name="UserDatabase" pathname="conf/tomcat-users.xml"
37 type="org.apache.catalina.UserDatabase" />
38 <Resource auth="Container" DefaultAutoCommit="false"
39 driverClassName="oracle.jdbc.driver.OracleDriver"
40 factory="org.apache.tomcat.jdbc.pool.DataSourceFactory"
41 initialSize="5" maxActive="100" maxWait="10000"
42 minEvictableIdleTimeMillis="3600000" minIdle="5" name="jdbc/iip-db"
43 password="iipshl" testOnBorrow="true" testOnReturn="false"
44 testWhileIdle="true" timeBetweenEvictionRunsMillis="10000"
45 type="javax.sql.DataSource"
46 url="jdbc:oracle:thin:@idc.mocomsys.com:31521:IIP" username="iipshl"
47 validationQuery="SELECT 1 FROM DUAL" />
48
49 </GlobalNamingResources>
50
51 <!-- A "Service" is a collection of one or more "Connectors" that share
52 a single "Container" Note: A "Service" is not itself a "Container", so you
53 may not define subcomponents such as "Valves" at this level. Documentation
54 at /docs/config/service.html -->
55 <Service name="Catalina">
56
57 <!--The connectors can use a shared executor, you can define one or more
58 shared thread pools-->
59
60 <!--
61 http://www.apache.org/licenses/LICENSE-2.0
62 Unless required by applicable law or agreed to in writing, software
63 distributed under the license is distributed on an "AS IS" BASIS,
64 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
65 See the license for the specific language governing permissions and
66 limitations under the license.
67 -->
68 <!-- The contents of this file will be loaded for each web application -->
69 <Context>
70
71 <!-- Default set of monitored resources. If one of these changes, the
72 web application will be reloaded. -->
73 <WatchedResource>WEB-INF/web.xml</WatchedResource>
74 <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>
75
76 <!-- Uncomment this to disable session persistence across Tomcat restarts -->
77 <!--
78 <Manager pathname="" />
79 -->
80 <ResourceLink global="jdbc/iip-db" name="jdbc/iip-db" type="javax.sql.DataSource"/>
81 </Context>
```



SERVICES - MODULES에서 패스를 수정한다

PATH는 /MINT

DOCUMENT BASE는 MINT-FRONT



path 설정

## Web Modules

### Web Modules

Configure the Web Modules on this server.

Path	Document Base	Module	Auto Reload	
/mint	mint-front	mint-front	Enabled	

Add Web Moc  
Add External Web  
Edit...  
Remove

프로젝트 빌드 시 사용할 톰캣 서버

### General Information

Specify the host name and...

Server name:

Host name:

[Runtime Environment:](#)

[Configuration path:](#)

[Open launch configuration](#)

### Server Locations

Specify the server path (i.e. ... published with no modules)

☒ Use workspace metadata

☐ Use Tomcat installation

☐ Use custom location (de...

Server path:

[Set deploy path to the def...](#)

Deploy path:

### Server Options

Enter settings for the server.

☐ Serve modules without publishing

### Edit Server Runtime Environment

#### Tomcat Server

Specify the installation directory

Name:

Tomcat installation directory:  [Browse...](#)

☐ apache-tomcat-8.5.78 [Download and Install...](#)

JRE:  [Installed JREs...](#)

[?](#) [< Back](#) [Next >](#) [Finish](#) [Cancel](#)

띄워져 있는 톰캣 서버를 우클릭하고 clean을 해주면 프로젝트 재 업로드를 해줌

이클립스 안에서 빌드를 하고 또 같은 서버를 사용하며 다른 프로젝트를 다른 위치에서 실행하고 싶다. 이게 바로 docubase를 의미한다.

이 때 이클립스 안에서 실행하면서 만들어진 conf안 파일들과 또 다른 위치에서 실행하는 conf는 다르다. 여러 개의 애플리케이션을 같은 서버에서 돌린다는 것을 의미한다.

1. 톰캣 서버가 실제로 다큐베이스를 통해서 설치되어있는 카탈리나를 가지고 소스를 어디에 배포할 지만 달리 해서 배포를 하는 것이기 때문에 설정도 다 틀려지는 것이다.
2. 서버를 하나 만들 때마다 servers가 생기는 것처럼 그 세팅이 전부 바뀌는 것이다.
3. log파일 등을 설정하기 위한 arguments 또한 다른 것이다.
4. 그러나 똑같은 배포 파일을 넣으면 복사를 해오니 설정은 똑같다. 이 말은 이클립스 내부에서 war파일을 생성한 뒤 실행할 다른 위치에 넣으면 그것은 복사해오는 것이니 설정이 같아지는 것이다.( db 설정 arguments 등등이)
5. 이 때 이클립스에서 한 것이 아니라 톰캣에 한 것이 톰캣 자체에 프로젝트를 배포한 것이 된다는 말이다.

환경 세팅을 위한 설정 파일들 캡처

```

-vm
C:/Users/xxrin/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933/jre/bin
-vmargs

```

java - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

‡ well as revocation information such as CRLs and signed OCSP Responses.

‡ This is in addition to the jdk.certpath.disabledAlgorithms property above.

‡

‡ See the specification of "jdk.certpath.disabledAlgorithms" for the

‡ syntax of the disabled algorithm string.

‡

‡ Note: The algorithm restrictions do not apply to trust anchors or

‡ self-signed certificates.

‡

‡ Note: This property is currently used by the JDK Reference implementation.

‡ It is not guaranteed to be examined and used by other implementations.

‡

‡ Example:

‡ jdk.tls.disabledAlgorithms=MD5, SSLv3, DSA, RSA keySize < 2048, W

‡ rsa\_pkcs1\_sha1, secp224r1

‡ jdk.tls.disabledAlgorithms=SSLv3, TLSv1, TLSv1.1, RC4, DES, MD5withRSA, W

‡ DH keySize < 1024, EC keySize < 224, 3DES\_EDE\_CBC, anon, NULL

hosts - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
# 102.54.94.97 rhino.acme.com # source server
# 38.25.63.10 x.acme.com # x client host

# localhost name resolution is handled within DNS itself.
# 127.0.0.1 localhost
# ::1 localhost

10.10.1.30 defect.mocomsys.com
# Added by Docker Desktop
10.10.8.79 host.docker.internal
10.10.8.79 gateway.docker.internal
# To allow the same kube context to work on the host and the container:
```

svn연결을 위한 localhost 세팅 system32에서 설정할 수 있음

## 디버깅

브레이크 포인트를 찍고

```
1802 * @since version 1.0(2015.07)
1803 */
1804 @RequestMapping(
1805     value="/login",
1806     params={"method=POST", "isTest=true"},
1807     method=RequestMethod.POST,
1808     headers="content-type=application/json")
1809 public @ResponseBody ComMessage<LoginInfo, BasicInfo> testLogin(
1810     HttpSession httpSession,
1811     @RequestBody ComMessage<LoginInfo, BasicInfo> comMessage,
1812     Locale locale,
1813     HttpServletRequest request
1814 ) throws Exception {
1815
1816     //-----
1817     //여기 비즈니스 코드만 작성해라.
1818     //-----
1819     {
1820
1821         String errorCd = "";
1822         String errorMsg = "";
```

디버그 모드를 실행하면 들어오는 값과 메소드 들을 확인할 수 있다.

back.xmlCommonContro...Index.htmldo.jspindex.jsplogin.jspCommonContro... ×%2

```

5      {
6          if (logger.isDebugEnabled()) LogUtil.prefix(debugMsg, "loginInfo 필드 체크");
7          loginInfo = comMessage.getRequestObject();
8          FieldCheckUtil.check("CommonController.login", loginInfo, messageSource, locale);
9          if (logger.isDebugEnabled()) {
10             LogUtil.prefix(debugMsg, "loginInfo 필드 체크 OK");
11             try {
12                 String loginInfoJsonStr = (Util.toJsonString(loginInfo));
13                 debugMsg.append(LogUtil.bar(LogUtil.prefix(loginInfoJsonStr)));
14             } catch (Exception e) {
15                 //[아시이나 보안점검 지적사항 반영] :: 빈공백의 catch 문을 사용하지 말것
16                 logger.error("[무시해도되는에러] : " + e.getMessage());
17             }
18         }
19     }
20
21     //-----
22     //Session에서 기존 로그인 정보 GET
23     //-----
24     {
25         if (logger.isDebugEnabled()) LogUtil.prefix(debugMsg, "Session에서 기존 로그인 정보 GET");
26         user = (User) httpSession.getAttribute("user");
27     }
28
29 }

```

VariablesBreakpointsExpressions

Name	Value
> frontSecurityService	FrontSecurityService (k
> loginService	SHLLLoginService (id=2
> messageSource	ReloadableResourceBur
> sapList	null
> securityService	SecurityService\$\$Enhan
> servletContext	ApplicationContextFace
> ssoLogin	false
> ssoLoginValue	null
> httpSession	StandardSessionFacade
> session	StandardSession (id=2
> accessCount	null
> attributes	ConcurrentHashMap<k
> authType	null
> creationTime	1677460238822
> expiring	false
> facade	StandardSessionFacade

<Choose a previously entered expression>

Logger[pep.per.mint.front.security.Decryp