

하나씩 쉽게 따라 해보는 IBM WebSphere Application Server(WAS) v7 – 12

이정운 (juwlee@kr.ibm.com)

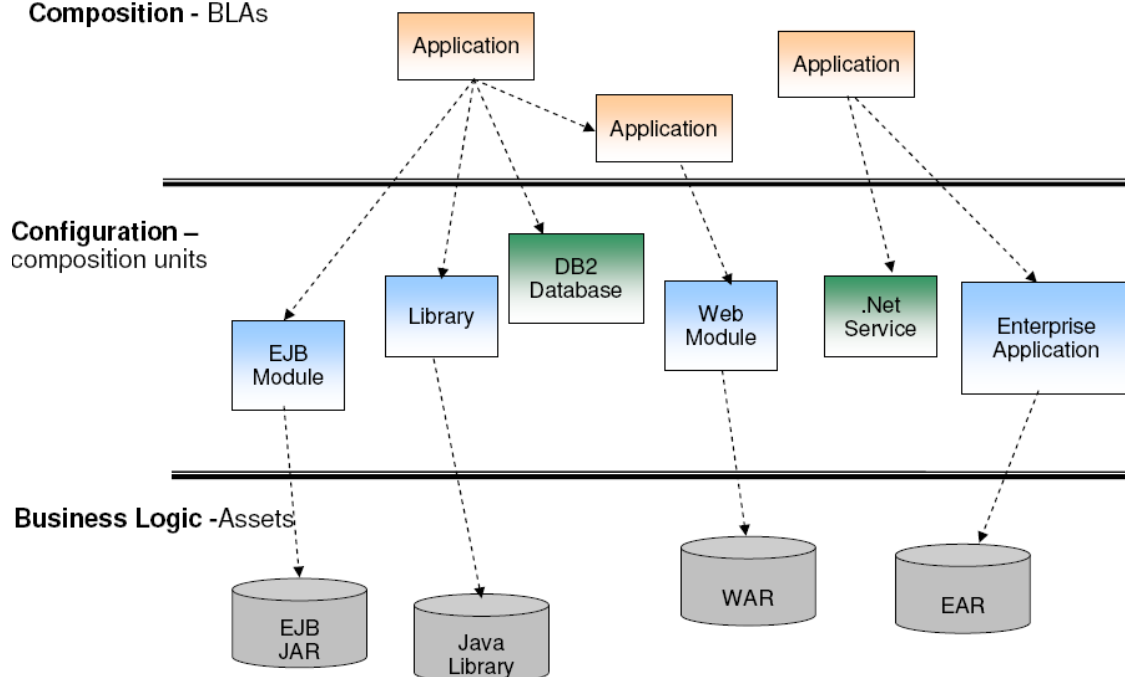
하나씩 쉽게 따라 해보는 IBM WAS v7 그 열 두번째 이야기를 시작합니다. 열 두번째 이야기는 실제 운영환경에서 많이 도움이 될 수 있을만한 엔터프라이즈 어플리케이션(EAR) 간의 Session을 공유해서 사용하는 방법입니다. 사실적으로 Session을 공유해서 여러가지 용도(예를 들어 SSO(Single-Sign On) 기능)로 많이 사용되고 있지만 J2EE 스펙상에서 명시된 Session 공유가 가능한 Scope 은 본래 하나의 동적 웹어플리케이션(WAR) 안에서만 가능합니다. 하지만, 이는 너무 제한적인 기능이라 IBM 에서는 자체적으로 WAS v6.1 에서 EAR 안의 여러 WAR 간의 Session 을 허용하며, WAS v7 에서는 여러 EAR 안의 WAR 끼리 Session 공유를 할 수 있는 기술을 제공하고 있습니다.

(단, 여러 EAR 이라고는 하지만 당연히도 이 EAR 이 전부 분산된 WAS 에 각각 따로 위치하고 있으면 Session 공유를 허용하지 않습니다. - WAS 가 다르면 JVM이 다르고 이는 memory reference 방식으로 접근할 수 없기 때문에 당연한 이야기라고 생각합니다.)

추가적인 확장 기능으로 WAS v7 에서 여러 EAR 간의 WAR 에서 Session 공유를 허용하면서 조건이 있는데 그것은 바로 BLA(Business Level Application) 이라는 관리모델로 묶여있어야 한다는 것입니다. 기존의 어플리케이션은 J2EE 패키징 모델로서 강한 의존성이 있으며 J2EE 프로그래밍 모델에 따라 여러 가지 제약사항들이 있었습니다. 이를 개선하기 위해서 WebSphere 에서 사용하는 것이 Business 입장에서 생각하고 보게 만들어진 관리 모델인 BLA 입니다. 이에 대한 자세한 내용은 하단의 그림을 참조하기 바라며 추가적으로 WAS v7 기본강좌 - step 5 Business Level Application 문서를 참조하시기 바라겠습니다.

(http://freeman98.hosting.paran.com/xe/?mid=was_info_re&page=2&document_srl=762&listStyle=&cpage=)

Composition - BLAs

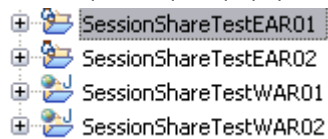


지금까지는 여러 EAR 안의 WAR 간의 Session 공유의 기본조건에 대해서 살펴보았고 이제 실제로 샘플 어플리케이션을 이용해서 Session 을 공유하는 것을 보여드리는 강좌를 진행하도록 하겠습니다.

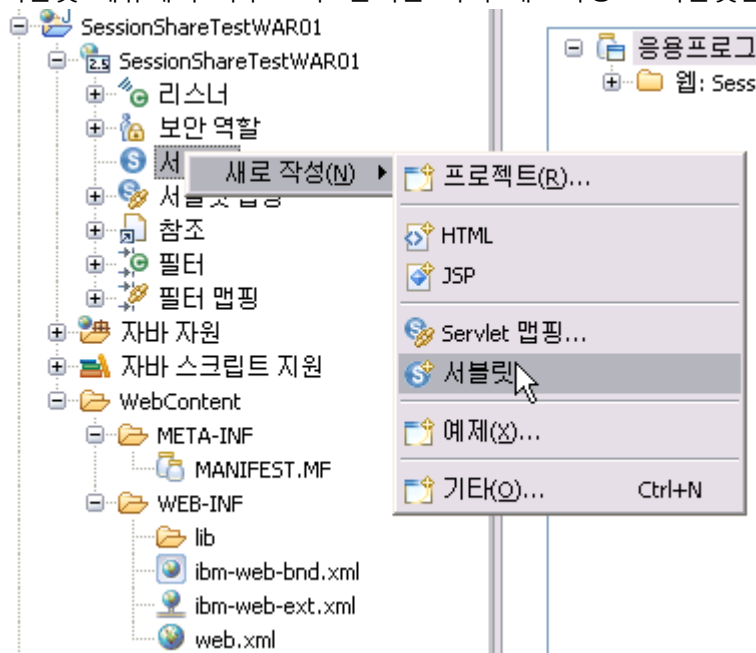
Part 1. Session 공유를 위한 어플리케이션 개발

샘플을 보고자 한다고 하였으니 이제 Session 공유를 위한 어플리케이션을 작성해보도록 하겠습니다. 어플리케이션을 개발하기 위해서 RAD v7.5 를 실행시킨 다음에 샘플로 사용할 엔터프라이즈 어플리케이션 프로젝트 두개와 이에 연결된 동적 웹 어플리케이션 프로젝트 두 개를 하단의 그림처럼 작성합니다.

(본 강좌에서는 하단의 그림처럼 SessionShareTestEAR01 이라는 엔터프라이즈 어플리케이션 프로젝트에 SessionShareTestWAR01 이라는 동적 웹 어플리케이션 프로젝트를 연결하였고, 두 번째로 SessionShareTestEAR02 라는 엔터프라이즈 어플리케이션 프로젝트에 SessionShareTestWAR02 라는 동적 웹 어플리케이션 프로젝트를 연결하여 사용하였습니다.)



이제 하나의 동적 웹 프로젝트에 가서 실제 간단한 서비스를 보여줄 서블릿을 작성하기 위하여 서블릿 메뉴에서 마우스 우 클릭을 하여 새로작성 > 서블릿을 선택합니다.



이렇게 되면 서블릿 작성 마법사가 나타나는데 원하는 패키지와 서블릿 클래스 이름을 입력한 후 다음을 입력합니다.

서블릿 작성

서블릿 작성
클래스 파일 위치를 지정합니다.

웹 프로젝트: SessionShareTestWAR01

소스 폴더: WSessionShareTestWAR01Wsrc 찾아보기...

자바 패키지: com.ibm.juwlee.session 찾아보기...

클래스 이름: SessionShareServlet01

수퍼클래스: javax.servlet.http.HttpServlet 찾아보기...

☐ 기존 서블릿 클래스 또는 JSP 사용

클래스 이름: SessionShareServlet01 찾아보기...

? < 이전(B) 다음(N) > 완료(F) 취소

다음을 입력하면 URL 매핑이 나오는데 URL 매핑만 기억하고 완료를 눌러 서블릿 작성을 완료합니다.

서블릿 작성

서블릿 배치 디스크립터 특정 정보를 입력하십시오.

이름: SessionShareServlet01

설명:

초기화 매개변수:

이름	값	설명

추가...
편집...
제거

URL 매핑:

/SessionShareServlet01

추가...
편집...
제거

? < 이전(B) 다음(N) > 완료(F) 취소

서블릿 작성을 완료하면 하단의 그림과 같이 기본 Servlet 틀에 맞추어서 Method 들이 자동으로 생성되어 나옵니다. 실제 서비스가 진행되는 doGet() 메소드만 내용을 채워주면 Servlet 완성은 끝이 납니다.

```

application.xml  SessionShareServlet01.java
package com.ibm.juulee.session;

import java.io.IOException;

/**
 * Servlet implementation class SessionShareServlet01
 */
public class SessionShareServlet01 extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public SessionShareServlet01() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
    }
}

```

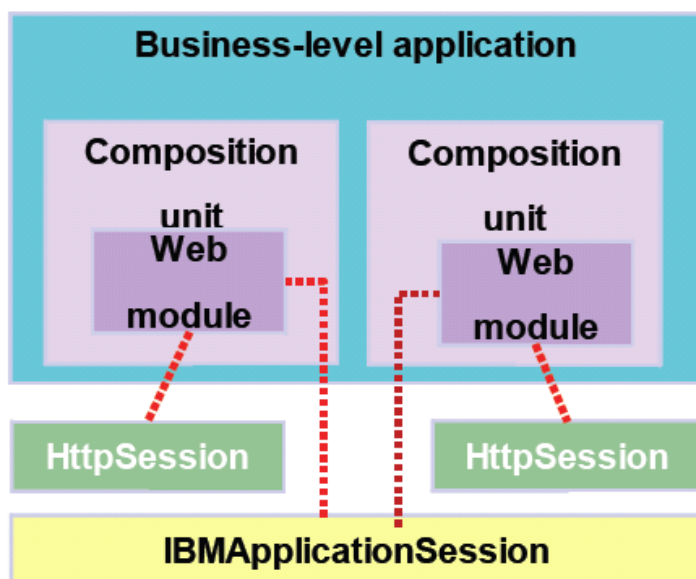
여기가 가장 핵심 포인트입니다. HttpSession 을 가지고 와서 IBM에서 정의한 IBMSession 으로 Casting 한 후, 여기서 IBMApplicationSession 이라는 객체를 생성해 냅니다.

```

//Session 가지고 오기
HttpSession session = request.getSession();
IBMApplicationSession appSession = ((IBMSession)session).getIBMApplicationSession();

```

이렇게 IBMApplicationSession 이라는 객체를 가지고 오면 이를 통해서 하단의 그림처럼 여러 EAR 간의 공유가 가능해지는 것입니다.



이 때, 이 IBMApplicatoinSession 은 생성만 이렇게 하고 나머지 기능은 Session 과 동일하게 사

용하면 됩니다. 테스트를 위해서 간단한 Session 설정 부분과 출력 부분을 추가하고 Servlet 작성을 하단처럼 마무리 합니다.

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    //Session 가지고 오기
    HttpSession session = request.getSession();
    IBMApplicationSession appSession = ((IBMSession)session).getIBMApplicationSession();

    //Session 설정
    appSession.setAttribute("test01", "share01");

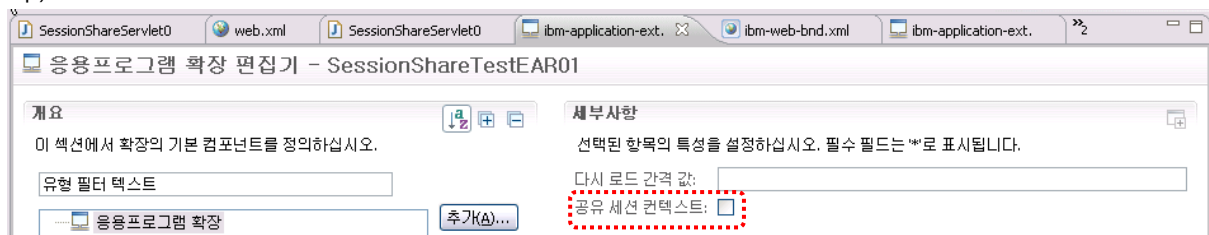
    //샘플 출력
    PrintWriter out = response.getWriter();
    out.println("Session out test01 : " + appSession.getAttribute("test01"));
    out.println("Session out test02 : " + appSession.getAttribute("test02"));
}
```

제가 앞단에 두 개의 동적 웹 프로젝트를 만들라고 하였는데 남은 하나의 동적 웹프로젝트에서도 동일하게 Servlet 을 만들고 하단에서 처럼 Session 설정 값만 구분되게 변경해 주면 됩니다.

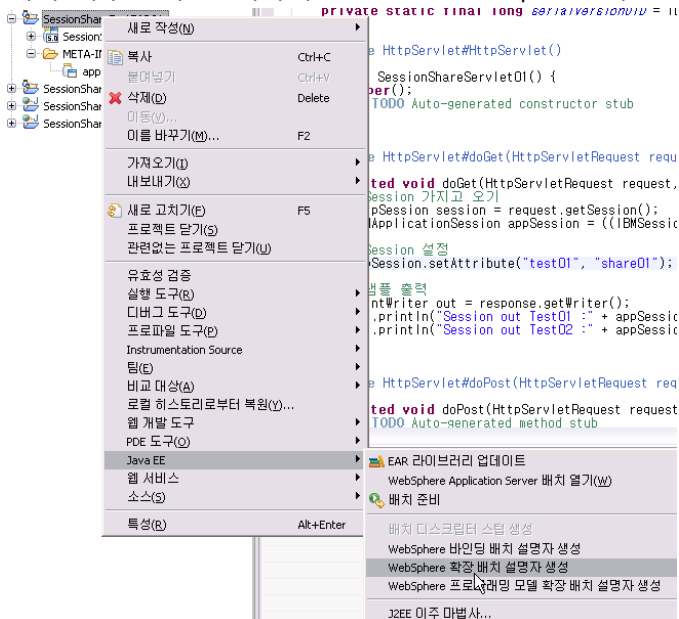
```
//Session 설정
appSession.setAttribute("test02", "share02");
```

이렇게 만 하면 필요한 기본적인 개발은 모두 완료 되었습니다....

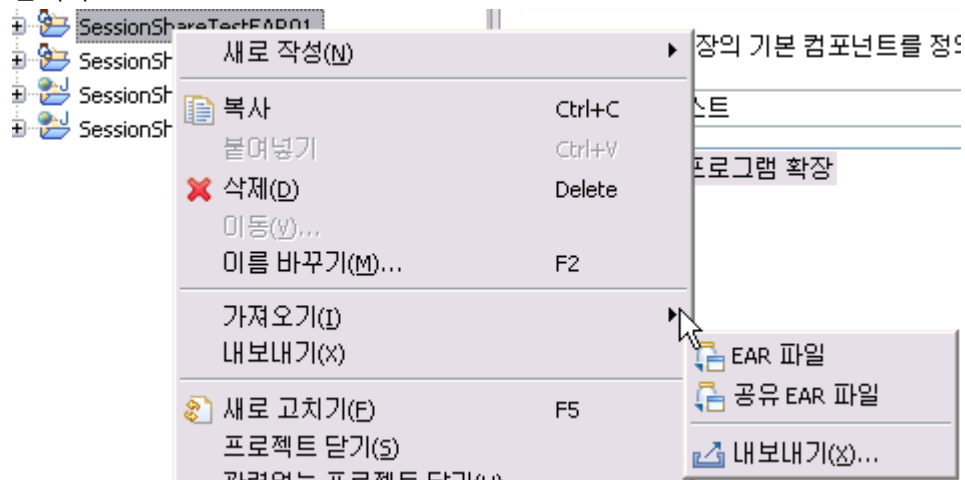
참고적으로 EAR 확장 설정 파일에 있는 공유 세션 컨텍스트는 반드시 체크되지 말아야 합니다. 즉, 하단의 그림처럼 비어 있어야 합니다. (체크하고 실행하면 NullPointerException 오류 발생합니다.)



추가적인 참고사항으로 EAR 확장 설정 파일이 안보인다면 엔터프라이즈 어플리케이션 프로젝트에서 마우스 우 클릭하여 Java EE > WebSphere 확장 배치 설명자 생성을 클릭하면 생성됩니다.



기본적인 어플리케이션 개발이 완료되어 있으므로 해당 EAR 에서 마우스 우 클릭하여 내보내기 > EAR 파일을 선택하여 두 개의 엔터프라이즈 어플리케이션 프로젝트를 각각 EAR 파일로 반출합니다.

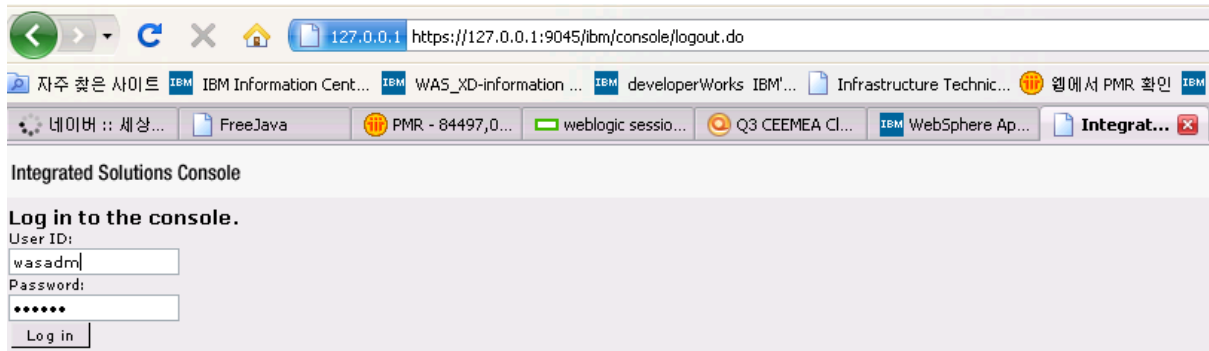


SessionShareTestEAR01.ear	5KB	V3Zip .EAR File	2009-12-04 오전 1...
SessionShareTestEAR02.ear	4KB	V3Zip .EAR File	2009-12-03 오후 7:54

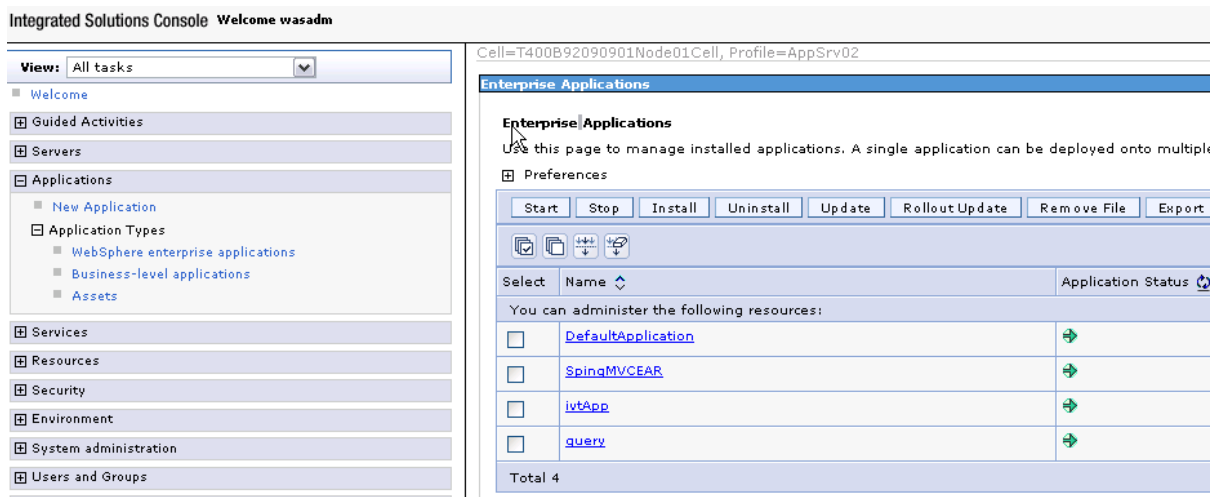
Part 2. WAS에 어플리케이션 설치 및 배포

이전 파트까지 여러 EAR 간의 Session 공유를 위한 간단한 Servlet 어플리케이션을 작성해 봤습니다. 간단하게 요약해드리자면 그냥 HttpSession 객체를 받아서 사용하던 부분을 IBMSession 으로 casting 한 후 IBMApplicatoinSession 객체를 받아오면 끝...많이 어렵지 않죠?...그럼 이제 실제 공유가 되는지 확인하기 위하여 WAS에 어플리케이션 설치 및 배포하는 단계를 진행하도록 하겠습니다.

WAS 에 어플리케이션을 배포하기 위해서는 우선 브라우저를 이용해서 WAS 관리콘솔에 접속하도록 하겠습니다. (사실적으로 배포 관련 부분은 이전 강좌에서도 많이 설명했기 때문에 최대한 간단히 설명하도록 하겠습니다. 이해가 잘 안되신 분들은 이전 강좌의 설명을 참조하세요.)

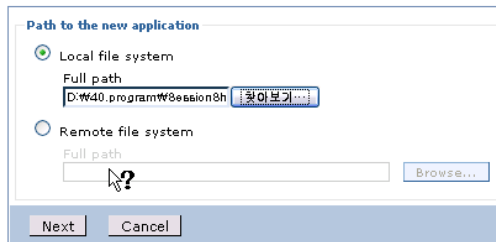


관리콘솔에서 Applications > Websphere enterprise applications 메뉴를 선택하면 배포된 어플리케이션 리스트를 확인하실 수 있는데 거기서 Install 버튼을 클릭하면 어플리케이션 설치 및 배포단계로 들어갈 수 있습니다.

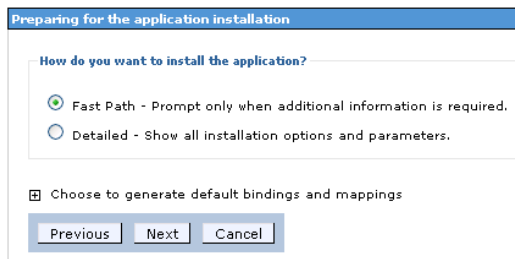


경로에는 이전 파트에서 개발하고 RAD 에서 추출한 EAR 파일을 선택하고 다음을 누릅니다.

Specify the EAR, WAR, JAR, or SAR module to upload and install.

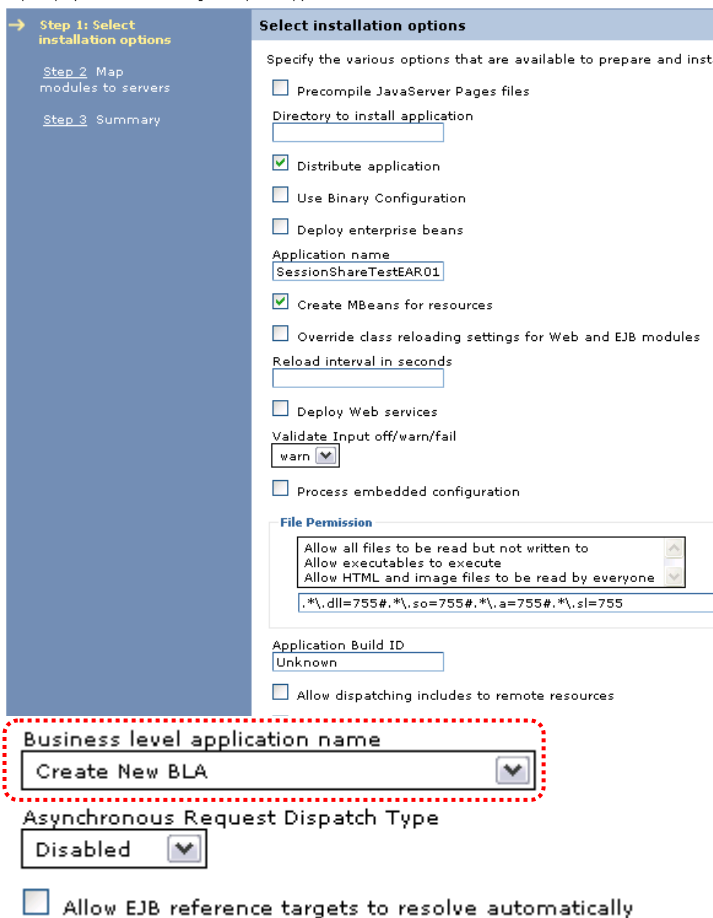


불필요한 내용을 입력하지 않고 설치 단계를 빠르게 진행하기 위해 Fast Path 를 선택합니다.



설치 마법사 step 1 이 실행되는데 변경하실 건 없고 다만 중요한 것은 BLA name 입력에 Create New BLA 로 설정하셔야 합니다. 이렇게 해야지만 지금 만드는 어플리케이션 이름으로 BLA 를 생성할 수 있습니다.

Specify options for installing enterprise applications and modules.



어플리케이션과 서버 맵핑설정을 확인하시고 다음을 누릅니다.

Specify options for installing enterprise applications and modules.

[Step 1 Select installation options](#)
[Step 2: Map modules to servers](#)
[Step 3 Summary](#)

Map modules to servers

Specify targets such as application servers or clusters of application servers where you want to install the module. The module can be installed on the same application server or dispersed among several application servers. Also, specify the plug-in configuration file (plugin-cfg.xml) for each Web server is generated through.

Clusters and servers:

Select	Module	URI	Server
<input type="checkbox"/>	SessionShareTestWAR01	SessionShareTestWAR01.war,WEB-INF/web.xml	WebSphere:cell=T400B92090901Node01Cell,node=T400B92090901Node02,server=server1

마지막으로 요약정보를 확인하시고 이상이 없으면 완료버튼을 클릭합니다.

Specify options for installing enterprise applications and modules.

[Step 1 Select installation options](#)
[Step 2 Map modules to servers](#)
[Step 3: Summary](#)

Summary

Summary of installation options

Options	Values
Precompile JavaServer Pages files	No
Directory to install application	
Distribute application	Yes
Use Binary Configuration	No
Deploy enterprise beans	No
Application name	SessionShareTestEAR01
Create MBeans for resources	Yes
Override class reloading settings for Web and EJB modules	No
Reload interval in seconds	
Deploy Web services	No
Validate Input off/warn/fail	warn
Process embedded configuration	No
File Permission	*,*.dll=755#,*\,.so=755#,*\,.a=755#,*\,.sl=755
Application Build ID	Unknown
Allow dispatching includes to remote resources	No
Allow servicing includes from remote resources	No
Business level application name	
Asynchronous Request Dispatch Type	Disabled
Allow EJB reference targets to resolve automatically	No
Cell/Node/Server	Click here

Installing...

If there are enterprise beans in the application, the EJB deployment process can take several minutes. Do not

Check the SystemOut.log on the deployment manager or server where the application is deployed for specific information about the deployment process.

ADMA5016: Installation of SessionShareTestEAR01 started.

ADMA5067: Resource validation for application SessionShareTestEAR01 completed successfully.

ADMA5058: Application and module versions are validated with versions of deployment targets.

ADMA5005: The application SessionShareTestEAR01 is configured in the WebSphere Application Server repository.

ADMA5053: The library references for the installed optional package are created.

ADMA5005: The application SessionShareTestEAR01 is configured in the WebSphere Application Server repository.

ADMA5001: The application binaries are saved in C:\IBM\WebSphere\AppServer\profiles\AppSrv02\wstemp\794994399\work\SessionShareTestEAR01.ear

ADMA5005: The application SessionShareTestEAR01 is configured in the WebSphere Application Server repository.

SECJ0400: Successfully updated the application SessionShareTestEAR01 with the appContextIDForSecurity information.

ADMA5005: The application SessionShareTestEAR01 is configured in the WebSphere Application Server repository.

ADMA5113: Activation plan created successfully.

ADMA5011: The cleanup of the temp directory for application SessionShareTestEAR01 is complete.

ADMA5013: Application SessionShareTestEAR01 installed successfully.

Application SessionShareTestEAR01 installed successfully.

To start the application, first save changes to the master configuration.

Changes have been made to your local configuration. You can:

- [Save](#) directly to the master configuration.
- [Review](#) changes before saving or discarding.

To work with installed applications, click the "Manage Applications" link.

Manage Applications

Enterprise Applications

⊕ Preferences

나머지 다른 어플리케이션도 위와 동일한 방법으로 설치 및 배포 하시면 되는데, 단, 설치 마법사 step 1의 BLA name 설정 부분을 하단과 같이 이전에 설치한 어플리케이션의 BLA 와 동일한 이름으로 지정을 해주어야 합니다.

Business level application name
WebSphere:blaname=SessionShareTestEAR01 ▼

Asynchronous Request Dispatch Type
Disabled ▼

☐ Allow EJB reference targets to resolve automatically

두 개의 어플리케이션 설치가 모두 완료되면 하단 처럼 하나의 BLA Application 과 두 개의 Enterprise Application 이 설치된 것을 확인할 수 있습니다.

Integrated Solutions Console Welcome wasadm

Cell=T400B92090901Node01Cell, Profile=AppSrv02

Business-level applications

Use this page to manage business-level applications. A business-level application is a configuration that represents an application needs to run. Artifacts typically include Java(TM) Platform, Enterprise Edition (Java EE) applications, files, or other business-level applications.

Preferences

Start Stop New Delete

Select	Name	Description	Status
You can administer the following resources:			
<input type="checkbox"/>	DefaultApplication		➡
<input type="checkbox"/>	SessionShareTestEAR01		✖
<input type="checkbox"/>	SpingMVCEAR		➡
<input type="checkbox"/>	ivtApp		➡
<input type="checkbox"/>	query		➡
Total 5			

Enterprise Applications

Use this page to manage installed applications. A single application can be deployed onto multiple servers.

Preferences

Start Stop Install Uninstall Update Rollout Update Remove File Export Export DDL Export File

Select	Name	Application Status
You can administer the following resources:		
<input type="checkbox"/>	DefaultApplication	➡
<input type="checkbox"/>	SessionShareTestEAR01	✖
<input type="checkbox"/>	SessionShareTestEAR02	✖
<input type="checkbox"/>	SpingMVCEAR	➡
<input type="checkbox"/>	ivtApp	➡
<input type="checkbox"/>	query	➡
Total 6		

Business-level applications

Use this page to manage business-level applications. A business-level application is a configuration that represents application needs to run. Artifacts typically include Java(TM) Platform, Enterprise Edition (Java EE) applications or files, or other business-level applications.

+ Preferences

☐ Messages
i SessionShareTestEAR01 started successfully.

Start Stop New Delete

Select	Name ↕	Description ↕	Status ↻
You can administer the following resources:			
<input type="checkbox"/>	DefaultApplication		
<input type="checkbox"/>	SessionShareTestEAR01		
<input type="checkbox"/>	SpingMVCEAR		
<input type="checkbox"/>	jvtApp		
<input type="checkbox"/>	query		

Total 5

Business-level applications > SessionShareTestEAR01




General Properties

Name

Description

Deployed assets

☒ ☐

Select	Name	Description	Type	Status 
<input type="checkbox"/>	SessionShareTestEAR01		Java EE	
<input type="checkbox"/>	SessionShareTestEAR02		Java EE	

테스트를 진행하기전에 관리콘솔의 Applications > WebSphere enterprise applicatoins > Application name > Context Root For Web Modules 메뉴를 클릭하여 하단과 같이 배포한 어플리케이션의 Context Root 를 미리 확인하여 둡니다.

[Business-level applications](#) > [SessionShareTestEAR01](#) > [SessionShareTestEAR01](#) > Context Root For Web Modules

Context Root For Web Modules

Context root defined in the deployment descriptor can be edited.

Web module	URI	Context Root
SessionShareTestWAR01	SessionShareTestWAR01.war,WEB-INF/web.xml	SessionShareTestWAR01

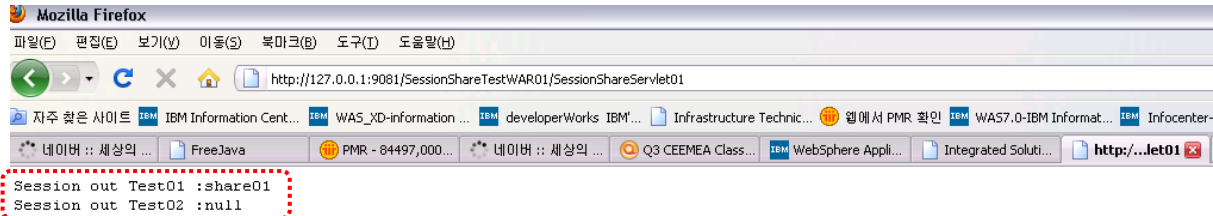
OK

Cancel

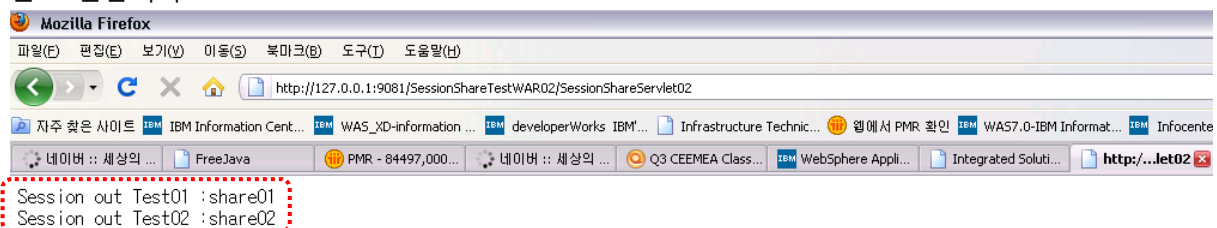
Part 3. 여러 EAR 간의 Session 공유 테스트

어플리케이션을 설치하고 배포하는 작업까지 마무리 하였습니다. 이제 브라우저 화면을 통해서 실제로 확인하는 작업만이 남았는데요. 하단처럼 직접 1번 어플리케이션의 Servlet 을 호출합니다.

(<http://127.0.0.1:9081>(서버 주소:포트)/SessionShareTestWAR01(Context Root)/SessionShareServlet01(Servlet 명))



위의 화면에서 보시는 것처럼 Test01 에 대한 Session 값이 정상적으로 출력되고 Test02 에 대한 Session 값이 null 로 나오는 것을 확인할 수 있습니다. 다음으로 두 번째 어플리케이션의 Servlet 을 호출합니다.



위의 화면에서 확인 가능한 것 처럼 1번 어플리케이션에서 생성한 Test01 에 대한 Session 값과 두 번째 어플리케이션에서 생성한 Test02 에 대한 Session 값을 둘 다 확인할 수 있습니다. 즉, 다른 EAR 에 위치한 1번 어플리케이션과 두번째 어플리케이션이 서로 Session 을 공유한다는 것을 확인할 수 있습니다.

여기 까지가 간단하지만 실제 업무에서 굉장히 많이 활용되어질 여러 EAR 어플리케이션 간의 Session 공유에 대한 샘플어플리케이션 개발 및 확인이었습니다. 하나씩 쉽게 따라 해보는 IBM WAS v7 그 열 두번째 이야기 - 엔터프라이즈 어플리케이션(EAR) 간의 Session 공유는 여기서 마무리 하도록 하겠습니다.

여기서 추가적으로 이야기 드릴 것이 있는데 이번 열 두번째 이야기를 기점으로 초보자를 위한 1부의 이야기를 마무리 하려고 합니다. 그렇다고 전체적으로 하나씩 쉽게 따라 해보는 IBM WAS v7 강좌가 마무리 되는 것은 아니고 초보자를 위한 1부 강좌가 마무리 되는 것입니다. 다음 강좌부터는 2부로서 초보를 넘어서 실제 기업환경에서 많이 사용되는 WAS ND 를 가지고 강좌를 이어가도록 하겠습니다. 많은 격려 부탁드립니다 이번 강좌는 여기서 마무리 하겠습니다. 이만~~~~~^^&;

참고 1) IBM WebSphere Application Server v7.0 InfoCenter

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.multiplatform.doc/info/welcome_nd.html

참고 2) IBM WebSphere Application Server v7.0 InfoCenter

-> Assembling so that session data can be shared

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/tprs_sharing_data.html?resultof=%22%73%65%73%73%69%6f%6e%22%20%22%73%68%61%72%65%22%20

※이 자료의 저작권은 작성자에게 있으며 유포는 자유로이 허용되나 상업적으로 이용은 금합니다.