목 차

오즈 Push 서버 개요 2

오즈 Push 서버 파일 구성 4

오즈 Push 서버 실행 5

오즈 Push 서버 설정 17

오즈 Push 서버 콘솔 명령어 21

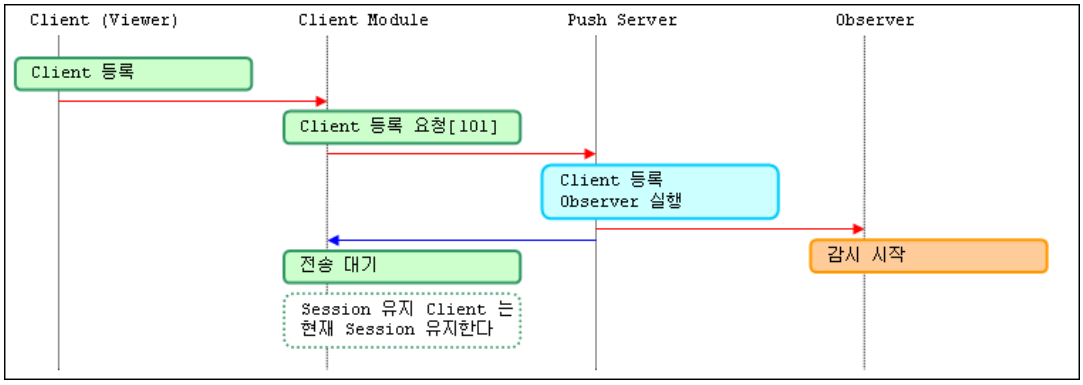
오즈 Push 서버와 오즈 애플리케이션 연동 23

오즈 Push 서버 개요

연결 흐름도

오즈 Push 서버는 서버에 등록한 Observer가 데이터의 변환을 감지하여 클라이언트에 데이터를 전송하는 서버로, 클라이언트의 요청이 없어도 데이터의 변경 상태를 실시간으로 전달하는 서버입니다.

클라이언트와 오즈 Push 서버의 연결 흐름도는 다음과 같습니다.



- 최초 클라이언트 모듈의 요청을 접수하고, Observer를 실행 및 관리합니다.
- 뷰어와의 Session은 최초 클라이언트의 요청 소켓을 유지하는 방법과 연결 정보를 관리하는 방법을 제공합니다.
- Thread 개수를 최소화하기 위하여 클라이언트로 데이터 전송이 필요한 시점에 전송 Thread를 생성하고 전송 완료 후에 자동 소멸합니다.
- Observer ID를 생성하는 로직과 데이터 변경 체크를 위한 로직은 사용자가 직접 개발합니다.
- Observer 목록 및 해당 클래스/로그 정보/Session interval 등을 pushserver.xml 파일로 관리합니다.
- Log4j를 이용한 로깅 기능을 제공합니다.

※ 참고사항 : 오즈 Push 서버에 많은 수의 클라이언트가 등록될 경우 성능 저하가 발생할 수 있습니다.

Observer

- 대상 데이터의 변화 여부를 감지하고, 변화가 감지될 경우 결과를 전송할 클라이언트 목록을 생성하여 Queue에 넣습니다.
- 전송할 클라이언트를 결정한 후 sdm 또는 사용자가 정의한 메시지를 클라이언트로 전송합니다.
- ObserverID와 데이터 변화 감지 로직은 Interface로 제공하고, 사용자가 개발합니다.

※ 참고사항 : Observer 인터페이스 구현시 "_ObserverImpl"을 클래스 이름으로 사용할 수 없습니다.

Client Module

- Push 서버와의 통신을 담당합니다.

오즈 Push 서버 파일 구성

디렉토리	파일명	설명
bin	ozpushcntrl.bat	윈도우(.bat) 또는 유닉스(.sh)용 오즈 Push 서버 컨트롤 파일
	ozpushcntrl.sh	
	ozpushstart.bat	윈도우(.bat) 또는 유닉스(.sh)용 오즈 Push 서버를 실행시키는 파일
	ozpushstart.sh	
	ozpushstop.bat	윈도우(.bat) 또는 유닉스(.sh)용 오즈 Push 서버를 종료시키는 파일
	ozpushstop.sh	
	OZServicePS.exe	오즈 서비스 실행 파일
	OZPushServerInstallService.bat	NT 서비스 환경 설정 및 서비스 등록 파일
	OZPushServerUninstallService.bat	등록된 오즈 Push 서버를 NT 서비스에서 제거하는 파일
	pushserver.ico	오즈 Push 서버 아이콘 파일
conf	pushserver.properties	오즈 Push 서버 기본 설정 파일
	pushserver.xml	오즈 Push 서버의 로그, session, transmission, Observer를 설정
lib	crimson.jar	XML 크림손 파서 라이브러리 파일
	log4j.jar	로그 사용에 대한 라이브러리 파일
	ozpush60.jar	오즈 Push 서버 라이브러리 파일
	ozsfw60.jar	오즈 서버 라이브러리 파일
	OZServiceBootstrap.jar	NT 서비스 관련 파일
	OZUSLClientSession.jar	USL 관련 라이브러리 파일
	parser.jar	XML 파서 클래스 라이브러리 파일
	sample.jar	Observer 샘플 파일
	xalan.jar	XML Xerces 파서 클래스 라이브러리 파일
	xerces.jar	XML Xerces 파서 클래스 라이브러리 파일
	xerceslmpl.jar	XML Xerces 파서 클래스 라이브러리 파일
	xml-apis.jar	XML Xerces 파서 클래스 라이브러리 파일
license	ozlicense.xml	오즈 Push 서버 라이센스 파일
logs	pushserver.log	오즈 Push 서버 로그 파일

오즈 Push 서버 실행

오즈 Push 서버는 데몬 형태로 실행하거나 윈도우 서비스에 등록하여 서비스 형태로 실행할 수 있습니다.

라이센스 설정

오즈 Push 서버 라이센스는 ozlicense.xml 파일 형태로 발급되며 이 파일을 오즈 Push 서버가 설치된 디렉토리의 License 폴더에 복사하여 설정합니다. 각 라이센스 파일에는 라이센스 유효기간, 서버 IP 주소, 운영 체제 종류 등이 설정되어 있습니다. 오즈 Push 서버를 운영하던 중 IP 주소 및 운영 체제가 변경될 경우에는 라이센스 파일을 재발급 받아야 합니다.

오즈 Push 서버 환경 설정

- **ozpushcntrl.bat**

  JRE 또는 오즈 Push 서버를 운영하기 위한 각종 클래스 라이브러리를 설정합니다. 기본적으로 Windows 환경의 경우 자체 JVM을 내장하여 이를 이용하고 있으며, UNIX 환경의 경우 ozpushcntrl.sh 파일에 JRE Path를 직접 설정하여야 합니다.

```
@echo off
rem -----------------------------------------------------
rem
rem          OZ Push Server Contorl(OZA SHELL)
rem
rem  This script file written and tested under Windows 2000
rem
rem  User should set following variables:
rem
rem  JAVA_HOME      - Determines the version of Java used to start
rem                 OZ Server. This variable must point to the
rem                 root directory of a JDK or JRE installation.
rem                 for an up-to-date list of supported JVMs your platform.
rem  MAX_MEMORY     - Maximum memory used by OZ Push Server. (The unit is mega-byte.)
```

```
rem  MIN_MEMORY    - Minimum memory used by OZ Push Server. (The unit is mega-byte.)
rem
rem ------------------------------------------------------

rem  ----------------------------------
rem   Set user-defined variables
rem  ----------------------------------

set JAVA_HOME=..\..\JRE
set MAX_MEMORY=256
set MIN_MEMORY=256
set JAVA_OPTION=

set OZPUSHSERVER_HOME=..
set OZLIBALL=%OZPUSHSERVER_HOME%\lib

rem ----------------------------------
rem       OZ Main Class
rem ----------------------------------
set OZLIB=%OZLIBALL%\ozpush60.jar;%OZPUSHSERVER_HOME%\conf\;

rem ----------------------------------
rem       OZ Server Class
rem ----------------------------------
set OZLIB=%OZLIB%;%OZLIBALL%\ozsfw60.jar

rem ----------------------------------
rem       Library for Logging
rem ----------------------------------
set OZLIB=%OZLIB%;%OZLIBALL%\log4j.jar

rem ----------------------------------
rem       Library for XML
rem ----------------------------------
set OZLIB=%OZLIB%;%OZLIBALL%\xalan.jar
set OZLIB=%OZLIB%;%OZLIBALL%\xercesImpl.jar
set OZLIB=%OZLIB%;%OZLIBALL%\xml-apis.jar
set OZLIB=%OZLIB%;%OZLIBALL%\parser.jar
set OZLIB=%OZLIB%;%OZLIBALL%\xerces.jar
set OZLIB=%OZLIB%;%OZLIBALL%\crimson.jar

rem ----------------------------------
rem      USL Jar for USL
rem ----------------------------------
```

```
set OZLIB=%OZLIB%;%OZLIBALL%\OZUSLClientSession.jar

rem ------------------------------------
rem       NT Service Class
rem ------------------------------------
set OZLIB=%OZLIB%;%OZLIBALL%OZServiceBootstrap.jar

rem ------------------------------------
rem       User Library
rem ------------------------------------
rem set OZLIB=%OZLIB%;%OZLIBALL%\sample.jar

echo %OZLIB%

if not "%1" == "-start" goto notStart
set JAVA_OPTION= %JAVA_OPTION% -Xms%MIN_MEMORY%m -Xmx%MAX_MEMORY%m
:notStart
%JAVA_HOME%\bin\java %JAVA_OPTION% -cp %OZLIB% oz.push.server.Console %1 %2 %3 %4
```

※ 참고사항 : MAX_MEMORY, MIN_MEMORY는 자바 옵션으로 JVM이 사용할 수 있는 최대/최소 메모리를 설정하며, 그 외 기타 자바 옵션은 JAVA_OPTION에서 설정할 수 있습니다. 해당 운영체제의 JAVA_OPTION에서 자세한 옵션 사항을 참조하여 서버 성능에 도움이 되는 옵션 사용을 권장합니다.

■ ozpushcntrl.sh

JRE 또는 오즈 Push 서버를 운영하기 위한 각종 클래스 라이브러리를 설정합니다. 기본적으로 Windows 환경의 경우 자체 JVM을 내장하여 이를 이용하고 있으며, UNIX 환경의 경우 ozpushcntrl.sh 파일에 JRE Path를 직접 설정하여야 합니다.

```
#!/bin/sh
#-----------------------------------------------------
#
#          OZ Push Contorl.
#
# This script file written and tested under HP-UX
#
# User should set following variables:
#
# JAVA_HOME     - Determines the version of Java used to start
#                OZ Server. This variable must point to the
#                root directory of a JDK or JRE installation.
```

```
#               for an up-to-date list of supported JVMs your platform.
# MAX_MEMORY    - Maximum memory used by OZ Server. (The unit is mega-byte.)
# MIN_MEMORY    - Minimum memory used by OZ Server. (The unit is mega-byte.)
# JAVA_OPTIONS  - On some platforms, certain java command-line options are
#               recommanded when running OZ Server.
#
#-----------------------------------------------------



#  ------------------------------------
#   Set user-defined variables
#  ------------------------------------

JAVA_HOME=../../jre
MAX_MEMORY=256
MIN_MEMORY=128
JAVA_OPTION=-hotspot

OZPUSHSERVER_HOME=..
OZLIB=.:$OZPUSHSERVER_HOME/conf
OZLIB=$OZLIB:$OZPUSHSERVER_HOME/lib/ozpush60.jar

#  ------------------------------------
#       OZ Server Library
#  ------------------------------------
OZLIB=$OZLIB:$OZPUSHSERVER_HOME/lib/ozsfw60.jar

#  ------------------------------------
#       Library for Logging
#  ------------------------------------
OZLIB=$OZLIB:$OZPUSHSERVER_HOME/lib/log4j.jar

# ------------------------------------
#      Library for XML
# ------------------------------------
OZLIB=$OZLIB:$OZPUSHSERVER_HOME/lib/xalan.jar
OZLIB=$OZLIB:$OZPUSHSERVER_HOME/lib/xercesImpl.jar
OZLIB=$OZLIB:$OZPUSHSERVER_HOME/lib/xml-apis.jar
OZLIB=$OZLIB:$OZPUSHSERVER_HOME/lib/parser.jar
OZLIB=$OZLIB:$OZPUSHSERVER_HOME/lib/xerces.jar
OZLIB=$OZLIB:$OZPUSHSERVER_HOME/lib/crimson.jar

# ------------------------------------
#     USL Jar for USL
```

```
# ----------------------------------
set OZLIB=%OZLIB%;%OZLIBALL%\OZUSLClientSession.jar

# ----------------------------------
#       User Library
# ----------------------------------
# set OZLIB=%OZLIB%;%OZLIBALL%\sample.jar

if [ "$1" != "-start" ]; then
  JAVA_OPTION=
else
  JAVA_OPTION="$JAVA_OPTION -ms$MIN_MEMORY"m
  JAVA_OPTION="$JAVA_OPTION -mx$MAX_MEMORY"m
fi

echo ""
echo $OZLIB
echo ""

$JAVA_HOME/bin/java $JAVA_OPTION -classpath $OZLIB oz.push.server.Console $1 $2 $3 $4
```

※ 참고사항 : MAX_MEMORY, MIN_MEMORY는 자바 옵션으로 JVM이 사용할 수 있는 최대/최소 메모리를 설정하며, 그 외 기타 자바 옵션은 JAVA_OPTION에서 설정할 수 있습니다. 해당 운영체제의 JAVA_OPTION에서 자세한 옵션 사항을 참조하여 서버 성능에 도움이 되는 옵션 사용을 권장합니다.

데몬 형태로 오즈 Push 서버 실행

윈도우 환경에서 오즈 Push 서버를 실행시키기 위해서는 오즈 Push 서버의 bin 폴더 아래에 위치한 ozpushstart.bat 파일을 실행하고, 오즈 Push 서버를 종료하기 위해서는 ozpushstop.bat 파일을 실행합니다.

UNIX 환경에서 오즈 서버를 실행시키기 위해서는 오즈 Push 서버의 bin 폴더 아래에 위치한 ozpushstart.sh 파일을 실행하고, 오즈 Push 서버를 종료하기 위해서는 ozpushstop.sh 파일을 실행합니다.

- **ozpushstart.bat/ozpushstart.sh**

  ozpushstart.bat 파일과 ozpushstart.sh 파일은 ozpushcntrl.bat 파일과 ozpushcntrl.sh 파일를 실행시켜 오즈 Push 서버를 실행시키는 파일입니다.

- ozpushcntrl.bat

```
@echo off
rem -----------------------------------------------------
rem
rem          OZ Push Server Start-up File
rem
rem This batch file written and tested under Windows 2000
rem check your path indicate <JAVA_HOME>\bin
rem
rem -----------------------------------------------------

ozpushcntrl.bat -start %1
```

- ozpushcntrl.sh

```
# -----------------------------------------------------
#
#           OZ Push Server Start-up File
#
#  This batch file written and tested under UNIX(HP-UX) System
#  check your path indicate <JAVA_HOME>/bin
#
#  -----------------------------------------------------

./ozpushcntrl.sh -start $1
```

윈도우 서비스 형태로 오즈 Push 서버 실행

오즈 Push 서버를 시스템에서 자동으로 실행하고 정지시키기 위해서는 윈도우 서비스로 등록해야 합니다.

■ 환경 설정

서버 설치 시 기본적으로 설치되는 디렉토리에 반드시 CLASSPATH가 등록되어 있어야 서비스 설치 및 구동이 됩니다.

NT 서비스에 등록할 서비스 이름과 서비스 등록 시 필요한 속성은 {오즈 Push 서버 홈 디렉토리}\bin\OZPushServerInstallService.bat 파일에 설정하며, 서비스 제거 관련 속성은 {오즈 Push 서버 홈 디렉토리}\bin\ OZPushServerUnInstallService.bat 파일에 설정합니다.

- OZPushServerInstallService.bat

  윈도우 서비스에 오즈 Push 서버를 등록하는 파일로 오즈 Push 서버가 구동되는데 필요한 JRE 경로 및 클래스 경로 등을 설정합니다.

```
@echo off

rem ---------------------------------------------------------------------
rem OZPushServer Service Install Register Batch File
rem ---------------------------------------------------------------------

rem ---------------------------------------------------------------------
rem       JRE Path
rem
rem ex) set JRE_PATH=C:\Program Files\Forcs\OZ Xstudio 6.0\jre\bin\hotspot\jvm.dll
rem ---------------------------------------------------------------------
set JRE_PATH=C:\Program Files\Forcs\OZ Xstudio 6.0\jre\bin\hotspot\jvm.dll


rem ---------------------------------------------------------------------
rem       OZ PushServer Home
rem
rem ex) set OZPUSHSERVER_HOME_BIN=C:\Program Files\Forcs\OZ Xstudio 6.0\OZ Push Server
6.0\bin
rem ---------------------------------------------------------------------
set OZPUSHSERVER_HOME=..
set OZPUSHSERVER_HOME_BIN=C:\Program Files\Forcs\OZ Xstudio 6.0\OZ Push Server 6.0\bin

rem JAVA option delemter is # or ;
set JAVA_OPTION=

set MAX_MEMORY=256
set MIN_MEMORY=128


if not "%OZPUSHSERVER_HOME%" == "" goto start
echo Unable to determine the value of OZSERVER_HOME.
goto eof

if not "%OZPUSHSERVER_HOME_BIN%" == "" goto start
echo Unable to determine the absolute path value of OZSERVER_HOME_BIN.
goto eof

if not "%JRE_PATH%" == "" goto start
```

```
echo Unable to determine the absolute path value of JRE_PATH
goto eof

:start

set OZLIBALL=%OZPUSHSERVER_HOME%\lib

rem -----------------------------------
rem       OZ Main Class
rem -----------------------------------
set OZLIB=%OZLIBALL%\ozpush60.jar;%OZPUSHSERVER_HOME%\conf\


rem -----------------------------------
rem       OZ NT Service BootStrap
rem -----------------------------------
set OZLIB=%OZLIB%;%OZLIBALL%\OZServiceBootstrap.jar


rem -----------------------------------
rem       OZ Server Class
rem -----------------------------------
set OZLIB=%OZLIB%;%OZLIBALL%\ozsfw60.jar

rem -----------------------------------
rem       Library for Logging
rem -----------------------------------
set OZLIB=%OZLIB%;%OZLIBALL%\log4j.jar

rem -----------------------------------
rem       Library for XML
rem -----------------------------------
set OZLIB=%OZLIB%;%OZLIBALL%\xalan.jar
set OZLIB=%OZLIB%;%OZLIBALL%\xercesImpl.jar
set OZLIB=%OZLIB%;%OZLIBALL%\xml-apis.jar
set OZLIB=%OZLIB%;%OZLIBALL%\parser.jar
set OZLIB=%OZLIB%;%OZLIBALL%\xerces.jar
set OZLIB=%OZLIB%;%OZLIBALL%\crimson.jar

rem -----------------------------------
rem      USL Jar for USL
rem -----------------------------------
set OZLIB=%OZLIB%;%OZLIBALL%\OZUSLClientSession.jar
```

```
rem ----------------------------------
rem       User Library
rem ----------------------------------
set OZLIB=%OZLIB%;%OZLIBALL%\sample.jar

echo %OZLIB%


OZServicePS.exe        //IS//OZPushServer        --DisplayName="OZPushServer"        --
Description="OZPushServer"           --StartPath="%OZPUSHSERVER_HOME_BIN%"           --
StopPath="%OZPUSHSERVER_HOME_BIN%"    --Classpath="%OZLIB%"    --Jvm="%JRE_PATH%"    --
JavaOptions="%JAVA_OPTION%"     --JvmMs="%MIN_MEMORY%"     --JvmMx="%MAX_MEMORY%"     --
StartMode=jvm     --StopMode=jvm         --StartClass=oz.server.OZServiceBootstrap     --
StartParams=OZPushServerStart        --StopClass=oz.server.OZServiceBootstrap        --
StopParams=OZPushServerStop --LogPath="%OZPUSHSERVER_HOME_BIN%" --LogLevel=DEBUG


:eof
```

옵션	필수 여부	설명	
--Classpath	예	오즈 Push 서버를 구동하기 위한 모든 클래스 패스	
--Description	아니오	서비스 이름 설명 (최대 1024 바이트까지 입력할 수 있습니다.)	
--DisplayName	아니오	서비스에 표시될 이름 (기본 값 : ServiceName) ※ 참고사항 : 필수 설정 옵션은 아니나 해당 옵션을 설정하기를 권장합니다.	
--Jvm	예	서비스가 구동될 자바의 jvm.dll 파일의 절대 경로 (예 : C:\j2sdk1.4.2_12\jre\bin\client\jvm.dll)	
--JvmMs	아니오	Initial Memory (단위 : MB)	
--JvmMx	아니오	최대 메모리 (단위 : MB)	
--JvmOptions	아니오	Jvm에 필요한 옵션 (기본 값 : -Xrs)	
--JvmSs	아니오	쓰레드 스택 크기 (단위 : KB)	
--LogLevel	아니오	로그 레벨	
		ERROR	log level 중 낮은 단계이며, 실제로 에러 상황에서만 에러 메시지를 log에 기록함

		WARN	"ERROR" level에서의 log 출력과 함께 에러는 발생하나 프로그램 동작에 크게 영향을 미치지 않는 에러에 대해서 log 기록합니다.
		INFO	"ERROR" level에서의 log 출력과 함께 서버의 트랜잭션 정보를 log에 기록함 (기본 값)
		DEBUG	"ERROR", "INFO" level에서의 log 출력과 함께 기타 전반적인 서버 동작 정보를 log에 기록함
--LogPath	아니오	서비스 관련 로그 생성 경로	
--LogPrefix	아니오	서비스 로그 파일 접두어 (기본 값 : OZService) 예) 접두어가 "OZ"일 경우 "OZ_yyyyMMdd.log"로 파일이 생성됩니다.	
--StartClass	예	서비스를 시작하는 클래스 이름	
		oz.server.OZServiceBootstrap	오즈 Push 서버의 서비스를 시작하는 클래스
--StartMode	예	서비스 시작 모드	
		jvm	Jvm을 사용하여 서비스 시작
--StartParams	예	서비스를 시작하기 위한 패러미터	
		OZPushServerStart	오즈 Push 서버 서비스
--StartPath	예	오즈 서비스를 구동할 작업 경로	
--StopClass	예	서비스를 종료하는 클래스명	
--StopParams	예	서비스를 종료하기 위한 아규먼트	
--StopPath	예	오즈 서비스를 종료할 작업 경로	
--StdOutput	아니오	표준 출력 관련 로그 파일명 예)--StdOutput="stdout.log"	
--StdError	아니오	표준 에러 관련 로그 파일명 예)--StdOutput="stderr.log"	
--Startup	아니오	서비스 시작 유형	
		Manual	서비스 시작 유형을 수동으로 설정합니다. (기본 값)
		Auto	서비스 시작 유형을 자동으로 설정합니다.

- OZPushServerUnInstallService.bat

  윈도우 서비스에 등록된 오즈 Push 서버를 제거하는 파일입니다.

```
@echo off

rem -----------------------------------------------------------------------
rem OZPushServer Service UnInstall Register Batch File
rem -----------------------------------------------------------------------


OZServicePS.exe //DS//OZPushServer
```

■ 서비스 등록

{오즈 Push 서버 홈 디렉토리}\bin\OZPushServerInstallService.bat 파일을 실행합니다.

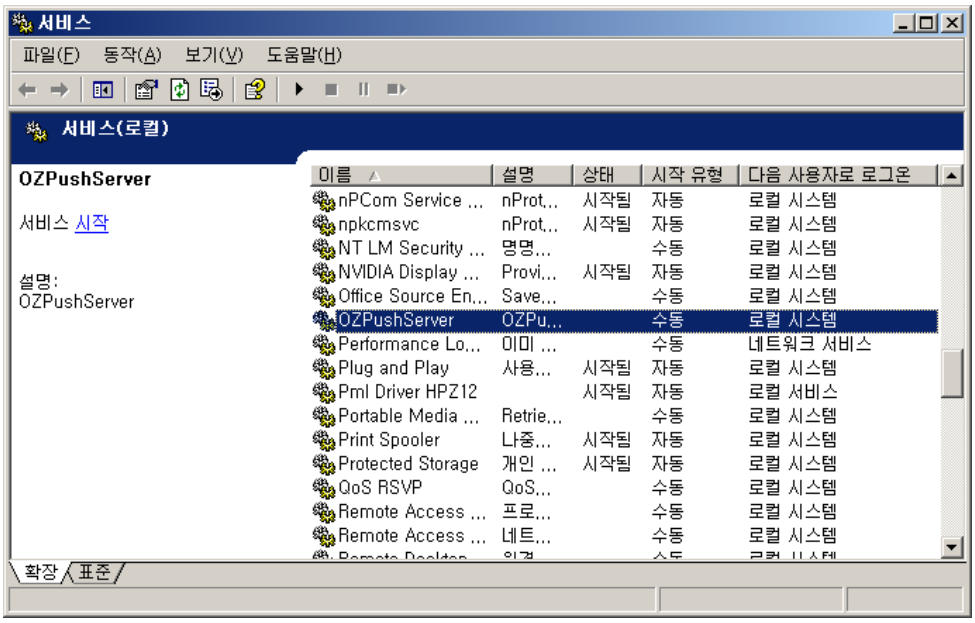
■ 서비스 제거

{오즈 Push 서버 홈 디렉토리}\bin\OZPushServerUnInstallService.bat 파일을 실행합니다.

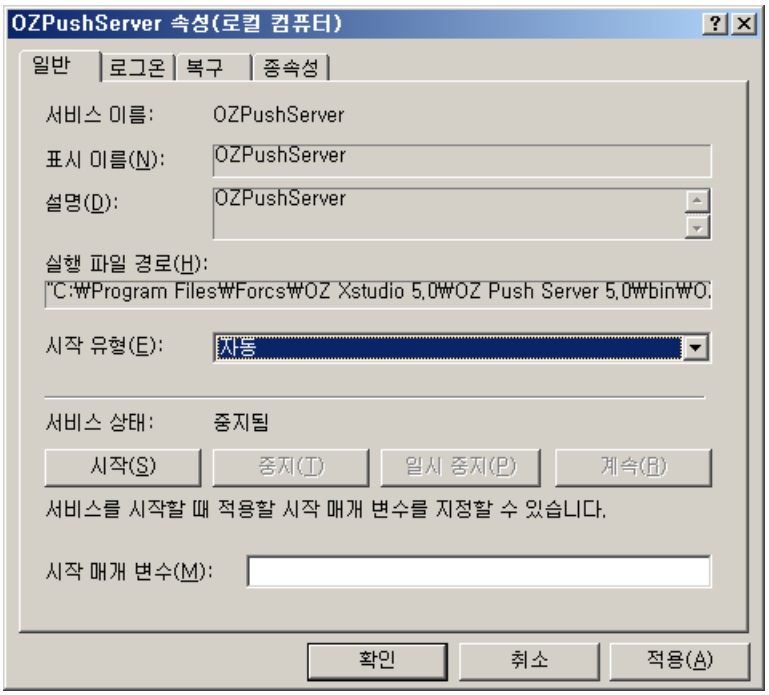
■ 서비스 등록 확인

윈도우 제어판의 관리 도구를 선택한 후 서비스를 클릭하여 나타나는 서비스 다이얼로그에서 오즈 Push 서버가 등록된 것을 확인할 수 있습니다.

OZPushServer를 선택하여 "서비스 시작" 아이콘을 클릭하면 수동으로 오즈 Push 서버를 실행할 수 있습니다.



서버 실행 방식을 수동에서 자동으로 바꾸려면 OZPushServer의 등록 정보 다이얼로그에서 "시작 유형" 옵션 값을 "자동"으로 변경하시기 바랍니다.



오즈 Push 서버를 중지시키기 위해서는 OZPushServer 서비스를 선택한 후 "서비스 중지" 아이콘을 클릭합니다.

오즈 Push 서버 설정

pushserver.properties

오즈 Push 서버의 홈 디렉토리를 설정합니다.

```
#-------------------------------------------
# Main configuration of OZ Push Server
#-------------------------------------------

path.oz.push.home=..\
```

옵션	설명
path.oz.push.home	오즈 Push 서버 홈 디렉토리 경로를 설정합니다. (상대/절대 경로 설정 가능)

pushserver.xml

오즈 Push 서버의 로그, session, transmission, Observer 등의 정보를 설정합니다.

```
<?xml version="1.0" encoding="EUC-KR"?>
<ozpushserver port="8088">
  <log priority="DEBUG" path="../logs/">
    <logger type="Console" aync="false"/>
    <!--
       DailyRollingFile - Use logger
       <logger  type="DailyRollingFile"  filename="pushserver.log"  append="true"
immediateFlush="true" datepattern="yyyy-MM-dd" />
    -->
    <logger     type="RollingFile"     filename="pushserver.log"     append="true"
immediateFlush="true" maxbackupindex="5" maxFileSize="10KB"/>
  </log>
  <session interval='5000'/>
  <transmission multi='false'/>
  <observers>
<!--
    registration of user-defined observer
    <observer name="OB1" classname="sample.observer.OB1" checkInterval="5000"/>
    <observer name="OB2" classname="sample.observer.OB2" checkInterval="5000"/>
-->
```

```
_w  </observers>
</ozpushserver>
```

Element	옵션	설명	
ozpushserver	port	오즈 Push 서버의 포트를 설정합니다. (기본 값 : 8088)	
log	priority	로그 레벨을 설정합니다. (기본 값 : DEBUG)	
		FATAL	로그 레벨 중 가장 높은 단계이며, 제품에 치명적인 오류가 발생했을 경우 에러 메시지를 기록합니다.
		ERROR	로그 레벨 중 낮은 단계이며, 실제로 에러 상황에서만 에러 메시지를 기록합니다.
		INFO	"ERROR" 레벨에 해당하는 로그 내용과 함께 서버의 트랜잭션 정보를 기록합니다.
		DEBUG	"ERROR", "INFO" 레벨에 해당하는 로그 내용과 함께 기타 전반적인 서버 동작 정보를 기록합니다.
	path	로그 파일이 생성될 디렉토리를 설정합니다. (기본 값 : %OZPushHome%\log)	
logger	type	로그 타입 즉, 로그의 출력 형태를 설정합니다. (기본 값 : Console) ※ 참고사항 : 로그 타입은 여러 개로 설정할 수 있으며, 이 때 Console과 RollingFile 형태 또는 Console과 DaillyRollingFile 형태로만 설정할 수 있습니다.	
		Console	로그를 콘솔로 출력합니다.
		RollingFile	로그를 기본 형태의 로그 파일로 출력합니다. ex) pushserver.log, pushserver.log.1, pushserver.log.2, ...
		DaillyRollingFile	로그 파일로 출력하되 로그 파일 이름을 날짜 패턴(DatePattern)에 설정한대로 생성합니다. ex) pushserver.log, pushserver.log.날짜패턴, pushserver.log.날짜패턴, ...

	async	로그 타입이 Console일 경우 버퍼링 여부를 설정합니다. (기본 값 : false)
	filename	로그 타입이 RollingFile 또는 DaillyRollingFile일 경우 로그 파일의 기본 값을 설정합니다. (기본 값 : pushserver.log)
	append	로그 타입이 RollingFile 또는 DaillyRollingFile일 경우 서버가 재구동될 때 기존의 로그 파일에 로그를 추가할지 여부를 설정합니다. (기본 값 : true)
	immediateFlush	로그 타입이 RollingFile 또는 DaillyRollingFile일 경우 버퍼에 있는 로그를 바로 기록할 것인지를 설정합니다. (기본 값 : false)
	datepattern	로그 타입이 DaillyRollingFile일 경우 날짜별로 로그 파일을 생성할 때 생성할 날짜 패턴을 설정합니다. 날짜 패턴은 아래 "DatePattern" 부분을 참조하시기 바랍니다. (기본 값 : YYYY-MM-DD)
	maxBackupIndex	로그 타입이 RollingFile일 경우 Backup될 로그 파일의 최대 개수를 정의합니다. 생성된 로그 파일이 최대 파일 개수를 초과할 경우에는 가장 오래된 파일이 삭제되고 새로운 파일이 생깁니다. (기본 값 : 5)
	maxFileSize	로그 타입이 RollingFile일 경우 로그 파일의 최대 사이즈를 설정합니다. (기본 값 : 1MB)
session	interval	세션 체크 간격을 밀리세컨드 단위로 설정합니다. (기본 값 : 5000)
transmission	multi	데이터 전송 시 새로운 스레드를 생성하여 전송할지 여부를 설정합니다. (기본 값 : false)
Observer	name	Observer의 이름을 설정합니다.
	classname	Observer의 클래스 이름을 설정합니다.
	checkInterval	데이터 변화 감지 체크 간격을 밀리세컨드 단위로 설정합니다. (기본 값 : 5000)

- DatePattern

날짜 패턴	로그 파일 생성 시간	예
yyyy-MM	매달 초	6월의 마지막 밤 12시에 pushserver.log는 pushserver.log.2005-06으로 바뀌며 이후의 로그는 다시 pushserver.log에 기록됩니다.

yyyy-ww	매주 초 (매주 초는 지역에 따라 다를 수 있음)	주초가 일요일이며, 1월 첫째 주 토요일 밤 12시에 pushserver.log는 pushserver.log.2005-01로 바뀌며 이후의 로그는 다시 pushserver.log에 기록됩니다.
yyyy-MM-dd	매일 자정 (기본 값)	7월 1일 자정에 pushserver.log는 pushserver.log.2005-07-01로 바뀌며 이후의 로그는 pushserver.log에 기록됩니다.
yyyy-MM-dd-HH	매 시 정각	7월 1일 10시 정각에 pushserver.log는 pushserver.log.2005-07-01-09로 바뀌며 이후의 로그는 pushserver.log에 기록됩니다.

오즈 Push 서버 콘솔 명령어

- **-help**

  콘솔 명령어의 사용법을 출력합니다.

  예) ozpushcntrl -help

- **-obinfo**

  해당 ObserverID의 정보를 출력합니다.

  출력 형식은 다음과 같습니다.

  ObserverID | ObserverName | ObserverClassName | 등록된 client 의 개수

  예) ozpushcntrl -obinfo ip:127.0.0.1 port:8088 id:ObserverID

- **-oblist**

  요청한 ip 와 포트에 있는 모든 Observer 정보를 출력합니다.

  출력 형식은 다음과 같습니다.

  ObserverID | ObserverName | ObserverClassName | 등록된 client 의 개수

  예) ozpushcntrl -oblist ip:127.0.0.1 port:8088

- **-obrunlist**

  현재 동작 중인 모든 Observer 의 정보를 출력합니다.

  출력 형식은 다음과 같습니다.

  ObserverID | ObserverName | ObserverClassName | 등록된 client 의 개수

  예) ozpushcntrl -obrunlist ip:127.0.0.1 port:8088

- **-obstop**

  ObserverID를 가지는 Observer의 구동을 중지시킵니다.

  예) ozpushcntrl -obstop ip:127.0.0.1 port:8088 id:ObserverID

- **-stop**

  OZ Push 서버를 중지합니다

  예) ozpushcntrl -stop ip:127.0.0.1 port:8088

- **-usage**

  서버 구동 시 설정할 수 있는 옵션의 사용법을 보여줍니다.

  예) ozpushcntrl –usage

- **-version**

  오즈 Push 서버의 버전 정보를 확인합니다.

  예) ozpushcntrl –version –ip:127.0.0.1 –port:8080

오즈 Push 서버와 오즈 애플리케이션 연동

관련 메소드

오즈 Push 서버와 연동하기 위해 관련된 오즈 애플리케이션 메소드는 다음과 같습니다.

■ DataModule.StartPush

Prototype	bool StartPush(string ObserverIP, integer ObserverPort, string ObserverName)	
Definition	해당 데이터 모듈을 오즈 Push 서버에 등록하고 등록 성공 여부를 반환합니다. ※ 참고사항 : 해당 데이터 모듈에 추가되어 있는 데이터셋과 연동된 컴포넌트의 SynchronizeToDataSet 속성 값을 true로 설정하여야 변경된 데이터 내용이 실시간으로 변경됩니다.	
Argument	*ObserverIP*	오즈 Push 서버 IP
	ObserverPort	오즈 Push 서버 Port
	ObserverName	오즈 Push 서버에 등록할 Observer 이름

■ DataModule.EndPush

Prototype	bool EndPush()
Definition	해당 데이터 모듈을 오즈 Push 서버에서 삭제하고 삭제 성공 여부를 반환합니다.

Observer 구현 시 주의사항

개발자가 Observer를 클래스로 구현 시 다음 두 가지 사항을 주의하시기 바랍니다.

1) 등록한 데이터 모듈에 추가되어있는 특정 데이터셋의 데이터가 변경될 때마다 해당 데이터 모듈에 있는 모든 데이터셋의 내용이 오즈 애플리케이션 뷰어로 전송됩니다. 그러므로 대용량 데이터의 경우 전체 데이터 모듈이 바인딩되는 시간을 염두해 두고 구현하시기 바랍니다. 데이터가 변경되어 오즈 Push 서버에서 오즈 애플리케이션 뷰어로 변경된 내용을 전송하는 시간이 데이터 모듈이 바인딩되는 시간보다 짧을 경우 해당 모듈의 바인딩이 완료되지않아

데이터를 전송하지 않습니다.

2) ODI의 데이터셋 숨김, 사용 안 함 여부와 관계없이 등록한 데이터 모듈의 데이터가 변경될 때마다 해당 데이터 모듈이 다시 바인딩됩니다.

환경 설정

오즈 애플리케이션과 오즈 Push 서버를 연동하기 전에 오즈 Push 서버에 다음과 같은 환경 설정이 필요합니다.

- 사용할 Observer를 구현한 클래스를 생성합니다.
  (각 사이트의 개발자가 직접 개발합니다. 본 매뉴얼에서는 두 개("OB1", "OB2")의 Observer가 구현된 pushsample.jar 파일을 사용합니다.)

- pushsample.jar 파일을 Push Server가 설치된 경로의 /lib 폴더 아래에 위치시킵니다.

- Push Server가 설치된 경로의 /bin 폴더 아래 ozpushcntrl.bat 파일에 아래와 같이 사용자 라이브러리를 설정합니다.

```
rem -----------------------------------
rem       User Library
rem -----------------------------------
set OZLIB=%OZLIB%;%OZLIBALL%\pushsample.jar
```

- Push Server가 설치된 경로의 /conf 폴더 아래 pushserver.xml 파일에 아래와 같이 Observer를 설정합니다.

```
<?xml version="1.0" encoding="EUC-KR"?>
<ozpushserver port="8088">
  <log priority="DEBUG" path="../logs/">
    <logger type="Console" aync="false" />
    <!--
       DailyRollingFile - Use logger
       <logger   type="DailyRollingFile"   filename="logfile.log"   append="true"
immediateFlush="true" datepattern="yyyy-MM-dd" />
    -->
    <logger     type="RollingFile"     filename="PushServer.log"     append="true"
immediateFlush="true" maxbackupindex="5" maxFileSize="10KB" />
  </log>
  <session interval='5000'/>
```

```
  <transmission multi='false'/>
  <observers>
    <observer name="OB1" classname="sample.observer.OB1" checkInterval="5000" />
    <observer name="OB2" classname="sample.observer.OB2" checkInterval="5000"/>
  </observers>
</ozpushserver>
```

적용 예

➢ 오즈 서버와 오즈 Push 서버를 구동합니다.

※ 참고사항 : 오즈 서버를 구동하는 방법은 "오즈 엔터프라이즈 서버 관리자 매뉴얼"을, 오즈 애플리케이션 프로젝트를 디자인하는 방법은 "오즈 애플리케이션 디자이너 사용자 가이드"를, 오즈 애플리케이션 프로젝트 파일을 호출하는 태그 작성 방법은 "오즈 애플리케이션 뷰어 사용자 가이드"를 참고하시기 바랍니다.

➢ 오즈 애플리케이션 디자이너를 실행하여 새로운 프로젝트에 ODI 파일과 테이블을 추가한 후 테이블 마법사에서 화면에 표시할 필드를 설정합니다.

➢ 보드에 버튼 두 개를 추가한 후 버튼의 "OnClick" 이벤트에 아래와 같이 스크립트를 추가합니다.

- Button1(StartPush)

```
//Push Server에 등록할 데이터 모듈과 Push Server의 정보를 설정합니다.
var dm = _GetDataManager.GetDataModule("ozcar");
//Push Server의 IP, Port, Observer Name을 설정합니다.
var push = dm.StartPush("127.0.0.1","8088","OB1");
```

- Button2(EndPush)

```
// Push Server의 목록에서 삭제할 데이터 모듈을 설정합니다
var dm = _GetDataManager.GetDataModule("ozcar");
var push = dm.EndPush();
```

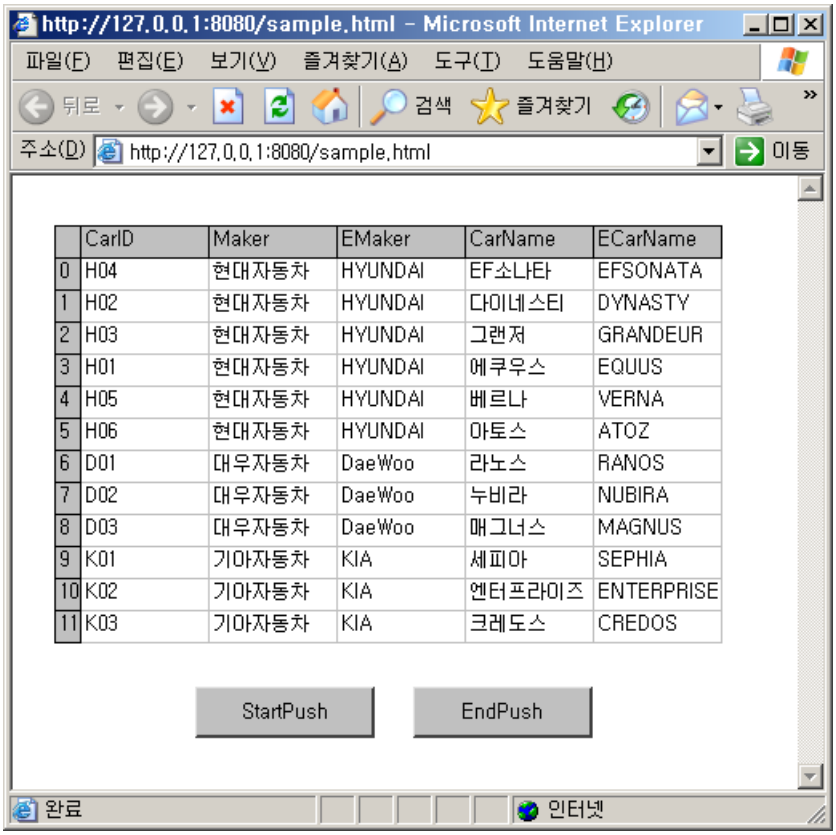
➢ 오즈 서버에 해당 프로젝트를 ozsample.oza로 저장한 후 프로젝트 파일 호출 태그를 아래와 같이 작성합니다.

- sample.html

```
<html>
<body>
<div id = "RunOZViewer">
```

```
   <object id = "OZApplicationViewer" CLASSID = "CLSID:907A00F3-7390-4EF2-
931B-360546587804" width = "100%" height = "100%">
   <param name = "repository_agent.type" value = "FROM_OZSERVER">
   <param    name    =    "repository_agent.ozserver.servlet"    value    =
"http://127.0.0.1:8080/oz/server">
   <param name = "repository_agent.user" value = "admin">
   <param name = "repository_agent.pass" value = "admin">
   <param name = "ozadoc.path" value = "ozsample.oza">
   </object>
</div>
</body>
</html>
```

➢ 웹 브라우저에서 sample.html 파일을 실행하여 애플리케이션 프로젝트 파일을 호출한 후 [StartPush] 버튼을 클릭하여 오즈 Push 서버와 연동합니다.

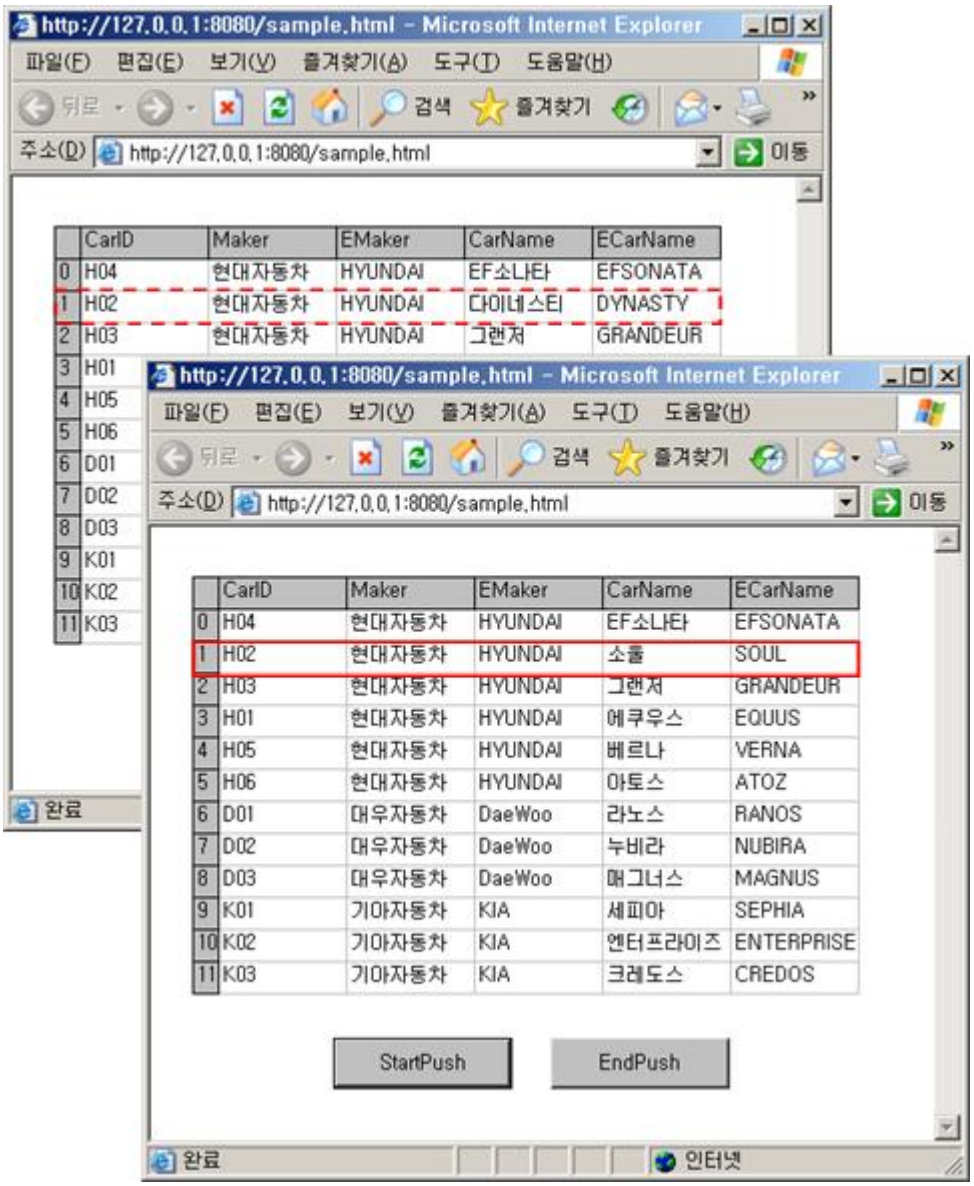


	CarID	Maker	EMaker	CarName	ECarName
0	H04	현대자동차	HYUNDAI	EF소나타	EFSONATA
1	H02	현대자동차	HYUNDAI	다이네스티	DYNASTY
2	H03	현대자동차	HYUNDAI	그랜저	GRANDEUR
3	H01	현대자동차	HYUNDAI	에쿠우스	EQUUS
4	H05	현대자동차	HYUNDAI	베르나	VERNA
5	H06	현대자동차	HYUNDAI	아토스	ATOZ
6	D01	대우자동차	DaeWoo	라노스	RANOS
7	D02	대우자동차	DaeWoo	누비라	NUBIRA
8	D03	대우자동차	DaeWoo	매그너스	MAGNUS
9	K01	기아자동차	KIA	세피아	SEPHIA
10	K02	기아자동차	KIA	엔터프라이즈	ENTERPRISE
11	K03	기아자동차	KIA	크레도스	CREDOS

➢ 데이터 값이 변경되면 애플리케이션에서 데이터를 새로고침하지 않아도 변경된 데이터가 화면에 표시되는지 확인합니다.

※ 참고사항 : 외부 프로그램 또는 데이터베이스 클라이언트 프로그램에서 직접 변경한 후

확인하시기 바랍니다.


http://127.0.0.1:8080/sample.html - Microsoft Internet Explorer
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
뒤로 검색 즐겨찾기
주소(D) http://127.0.0.1:8080/sample.html
이동
CarID Maker EMaker CarName ECarName
0 H04 현대자동차 HYUNDAI EF소나타 EFSONATA
1 H02 현대자동차 HYUNDAI 다이너스티 DYNASTY
2 H03 현대자동차 HYUNDAI 그랜저 GRANDEUR
3 H01
4 H05
5 H06
6 D01
7 D02
8 D03
9 K01
10 K02
11 K03
완료
http://127.0.0.1:8080/sample.html - Microsoft Internet Explorer
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
뒤로 검색 즐겨찾기
주소(D) http://127.0.0.1:8080/sample.html
이동
CarID Maker EMaker CarName ECarName
0 H04 현대자동차 HYUNDAI EF소나타 EFSONATA
1 H02 현대자동차 HYUNDAI 소울 SOUL
2 H03 현대자동차 HYUNDAI 그랜저 GRANDEUR
3 H01 현대자동차 HYUNDAI 에쿠우스 EQUUS
4 H05 현대자동차 HYUNDAI 베르나 VERNA
5 H06 현대자동차 HYUNDAI 아토스 ATOZ
6 D01 대우자동차 DaeWoo 라노스 RANOS
7 D02 대우자동차 DaeWoo 누비라 NUBIRA
8 D03 대우자동차 DaeWoo 매그너스 MAGNUS
9 K01 기아자동차 KIA 세피아 SEPHIA
10 K02 기아자동차 KIA 엔터프라이즈 ENTERPRISE
11 K03 기아자동차 KIA 크레도스 CREDOS
StartPush
EndPush
완료
인터넷