

PHP 개발자를 위한 이클립스

이클립스에서 PHP 개발하기

IDE의 묘미는 디버깅이다. 텍스트 에디터에서 지나친 에러를 잡아내어 알려준다. 이클립스에서 PHP 프로그래밍을 할 때도 이 같은 묘미가 느껴진다. 지난 시간에 이어 PHP 실행 관련 추가 내용과 디버깅을 위한 설정, 웹서버 연동 등 IDE로 활용하기 위한 방법을 알아본다.

연재 순서

1회 | 2007. 1 | PHP 프로그래밍을 위한 이클립스
2회 | 2007. 2 | 이클립스에서 PHP 개발하기

연재 가이드

운영체제 | 윈도우, 리눅스, 유닉스, 매킨토시 등 자바 가능한 GUI 플랫폼
개발도구 | 자바SE 1.5 이상, 이클립스 3.1이상, PHP, DBG, XAMPP
기초지식 | PHP 프로그래밍
응용분야 | PHP 웹 애플리케이션 개발

홍영환 xenonix@gmail.com | 컴퓨터를 처음으로 사주셨던 부모님, 리눅스 CD를 건네주었던 친구, 오픈소스를 알게 해준 후배, 옆에서 한결같은 모습으로 바라보는 여자 친구에게 항상 고마운 마음을 갖고 있다. 테더블즈에서 개발자로 근무하는 창년으로 정보와 기술은 공유 또는 공개 되어야 한다고 믿는다.

스
텝
바
이
스
텝
5

PHP 실행

환경변수를 지정하여 실행하기

하나의 웹 게시판이 있다고 가정하자. 웹 게시판 목록에서 게시물을 선택하면 웹 브라우저는 웹 서버로 선택한 게시물 번호를 전송한다. PHP 프로그램은 웹 서버를 통해 입력받은 게시물 번호를 데이터베이스에서 검색한 뒤 찾은 정보를 HTML 형식으로 웹 브라우저에 전송한다. 게시판 뿐만 아니라 웹 기반 프로그램은 이런 방식으로 데이터를 교환한다. 이클립스에서 환경변수를 지정하여 웹 브라우저를 통해 데이터를 입력하는 것과 동일하게 프로그램을 실행시킬 수 있다.

QUERY_STRING은 GET 방식으로 입력받은 값을 갖는 환경 변수다. 이 환경변수의 값은 PHP에서 \$_REQUEST, \$_GET, \$_POST 등으로도 가져올 수 있다. QUERY_STRING을 이클립스에서 입력하려면 메뉴 'Run->Run' 창을 띄우면 된다(〈화면 2〉 참조). PHP DBG Script에서 실행 할 프로그램을 선택하고, 〈화면 2〉 오른쪽 영역의 탭 메뉴 중 4번째 Environment를 선택한다. CGI-Vars를 클릭하면 환경변수 선택 창이 열린다(〈화면 3〉 참조). QUERY_STRING()을 선택하고 OK를 클릭하면 환경 변수 지정 목록에 QUERY_STRING이 추가 된다. 목록에서 QUERY_STRING을 선택하고 Edit 버튼을 클릭한다. Value에



필자 메모

이클립스를 실행하는 운영체제가 한글 윈도우라면 이클립스의 기본 인코딩은 MS949이다. 자바의 경우 UTF-8을 많이 사용하지만 PHP는 여전히 EUC-KR을 주로 사용하므로 한글 윈도우 환경이면 EUC-KR로 작성된 소스 코드를 편집하는데 문제가 없다. 그런데 영문 윈도우는 이클립스의 기본 인코딩이 Cp1252다. Cp1252는 Western Alphabet 인코딩이라 한글이 제대로 표현되지 않는 문제가 있다. 한글 인코딩이 가능하게 하는데 두 가지 방법이 있다. 윈도우 제어판 'Regional Options(국가 및 언어 옵션)'에서 Locale을 Korea로 변경하거나 이클립스 메뉴 'Window->Preferences'에서 'General->Workspace'의 Text file encoding을 Other로 변경하고 'MS949'를 직접 입력하는 방법이다. 이전 버전의 이클립스는 인코딩 설정 항목이 'General->Editors'에 있다.



〈화면 1〉 인코딩 설정 화면

name=value&name=value와 같은 형식으로 원하는 값을 입력하면 웹 브라우저에서 입력받는 것처럼 \$_GET 값을 얻을 수 있다(화면 4 참조). 이 외에도 원하는 임의 값을 지정할 수 있다. [항목 1]은 <화면 2>의 버튼에 대한 설명이다.



<화면 2> 환경 변수 지정 목록 화면

[항목 1] 환경 변수 지정 목록 버튼 설명

- New : 새로운 환경 변수를 만든다.
- CGI-Vars : 웹 서버를 통해 입력 받는 기본적인 환경 변수를 지정한다.
- Select : 이클립스에서 제공하는 시스템의 환경 변수를 선택하여 지정한다.



<화면 3>
환경 변수 선택 화면



<화면 4> QUERY_STRING 환경 변수에 값을 지정하는 화면

실행 결과를 파일로 저장하기

실행한 결과에 HTML이 포함된 경우 콘솔 창에서 결과를 확인하기가 어렵다. 이럴 땐 이클립스에서 PHP 프로그램을 실행하여 얻은 결과를 파일로 저장하고 확인할 수 있다. 메뉴 'Run->Run' 창을 띄운다. 실행할 프로그램을 선택하고 탭 메뉴의 Common을 선택한다. Standard Input and Output에서 File을 선택하고 결과를 저장할 파일명을 입력한다. Append 옵션은 파일을 덮어쓰지 않고 결과를 이어서 저장한다(화면 5 참조).



<화면 5> Standard Input and Output 설정 화면

디버깅

프로그램의 오류를 찾아 수정하는 것을 디버그(debug)라고 하며, 디버그 하는 작업을 디버깅이라고 한다. 웹 프로그램은 스크립트라는 특성상 소스 코드를 읽어가며 분석하여 디버깅을 한다. 이 방법은 소스 코드가 복잡해질수록 디버깅이 어려워지는 단점이 있다.

디버깅을 위한 준비

PHPEclipse는 DBG라는 PHP 디버거의 도움을 받아 이클립스에서 PHP 프로그램을 디버깅할 수 있는 환경을 제공한다. DBG는 <http://dd.cron.ru/dbg/>에서 상용 버전과 무료 버전 두 가지로 배포되고 있다. <표 1>은 DBG 버전에 대한 PHP 버전 설명이다.

DBG 버전	지원하는 PHP 버전
DBG 3.1.6(상용)	4.3.0 ~ 4.4.4 5.0.0 ~ 5.1.6
DBG 2.15.1(무료)	4.0.6 ~ 4.4.2 5.0.0 ~ 5.1.2

<표 1> DBG 버전 별 PHP 지원 버전

최신 무료 버전인 DBG 2.15.1은 64비트 환경을 지원하지만, 소스 코드 형태로만 제공되어 직접 컴파일을 해야 한다. 여기서는 바이너리 형태로 제공되는 무료 버전인 DBG 2.13.1을 설치한다. <표 1>을 참고하여 설치된 PHP 버전에 맞는 DBG를 설치하자.

필자의 개발 환경

- 윈도우XP
- 이클립스 3.2.1
- PHPEclipse 1.1.8
- PHP 5.1.2(Windows binary), <http://kr.php.net/releases.php>
- DBG 2.13.1(DBG 2.13.1 dbg modules) <http://dd.cron.ru/dbg/downloads.php>

DBG 설치

내려 받은 DBG 파일의 압축을 푼다. 폴더를 열면 i686 폴더 안에 php_dbg.dll-x.x.x라는 파일이 있는데 뒤에 숫자는 PHP 버전을 의미한다. 설치된 PHP 버전에 맞는 DBG를 찾아 PHP가

설치된(ext 또는 extension) 폴더에 복사한 후 php_dbg.dll로 파일 명을 변경한다. PHP에 DBG 모듈을 추가해야 한다. 만약 php.ini 파일이 없다면 PHP 폴더에 php.ini-dist 파일 또는 php.ini-recommended 파일 중 하나를 php.ini으로 파일 명을 변경하면 된다.

php.ini 파일을 열어 implicit_flush를 찾아 값을 On으로 변경하고 extension_dir의 값을 PHP Extension 폴더의 경로로 변경한다(〈리스트 1〉 참조). PHP 4는 확장 모듈의 폴더가 extension으로 되어 있다. 그리고 php.ini 마지막에 〈리스트 2〉의 내용을 추가한다. implicit_flush 옵션은 print()와 echo(), HTML을 만났을 때 flush() 함수를 호출하여 PHP의 출력 버퍼 내용을 전송하도록 한다.

〈리스트 1〉 php.ini 변경 내용

```
implicit_flush = Off
==> implicit_flush = On
extension_dir = "./"
==> extension_dir = "./ext/"
```

〈리스트 2〉 php.ini 추가 내용

```
extension = php_dbg.dll
debugger.enabled = on
debugger.profiler_enabled = on
debugger.JIT_host = clienthost
debugger.JIT_port = 7869
```

이제 DBG가 제대로 등록되었는지 확인해야 한다. 이클립스를 실행하고 파일을 하나 생성해 〈리스트 3〉의 코드를 입력한다. 실행한 다음 콘솔 영역에 〈화면 6〉처럼 DBG 등록정보가 나타나면 정상적으로 설치가 완료되었다.

〈리스트 3〉 phpinfo() 소스 코드 입력

```
< ?php
phpinfo();
?>
```



〈화면 6〉 phpinfo() 결과에 DBG 정보가 출력된 화면

디버깅을 위한 파일 생성

디버깅을 위해 debugging.php라는 파일을 생성하고 〈리스트 4〉의 소스 코드를 입력한다. 이 소스 코드는 매우 간단하지만 디버깅을 통해 실행 과정에서 변수 값의 변화를 보여준다.

〈리스트 4〉 Debugging.php 파일 소스코드

```
<?php

$strName = "Eclipse";
$intCount = 1000;

$arrInfo['name'] = $strName;
$arrInfo['count'] = $intCount;

$arrResult = $arrInfo;

echo "End...";

exit();
?>
```

Debug Perspective로 전환

Debugging.php 파일을 생성하였다면 메뉴 'Window -> Open Perspective'에서 Debug를 선택하여 디버깅을 위해 Debug Perspective로 전환한다. Debug Perspective는 PHP Perspective와 다르게 구성되어 있다. 〈화면 7〉은 Debug Perspective로 전환한 화면이고 [항목 2]는 Debug Perspective에 대한 설명이다.



〈화면 7〉 Debug Perspective로 전환한 화면

〈항목 2〉 Debug Perspective 화면 설명

- Debug : 디버그 프로세스를 보여준다.
- Variables : 실행 과정의 PHP 변수에 대한 정보를 보여준다.
- Breakpoints : Breakpoint로 지정된 위치 정보를 보여준다.

디버그 설정

여기까지 진행되었다면 이제 디버깅을 위한 디버그 설정을 해야 한다. 이 과정은 PHP 프로그램을 실행(Run)하는 과정과 비슷하다. 메뉴 'Run->Debug'를 선택하면 디버그 설정 창이 열린다. 화면 왼쪽 'PHP DBG Script'를 마우스 오른쪽 버튼으로 클릭하고 New를 선택한다. <화면 8>과 같은 화면이 나타난다. 항목이 이미 채워져 있을 것이다. Name 항목은 설정에 대한 고유 이름이니 원하는 이름으로 변경한다.

그리고 탭 메뉴의 3번째 Environment를 이동한다. <화면 9>와 같이 PHP 인터프리터를 선택하는 화면이 나타난다. 이미 지정했던 PHP 인터프리터가 선택돼 있으면, DBG는 PHP 인터프리터가 CGI 상태로 구동돼야 동작한다. 만약 PHP 인터프리터가 콘솔 구동과일인 php-win.exe(PHP 4의 경우 cli/php.exe)가 선택되어 있다면, CGI 구동방식인 php-cgi.exe로 변경한다. PHP 4의 경우엔 PHP 폴더 바로 아래에 있는 php.exe를 선택한다. Apply를 클릭하여 저장하고 Debug를 실행해 보자. <화면 10>은 Debug를 실행했을 때 Debug 영역에 나타난 결과이다. 프로그램이 정상적으로 종료되었음을 의미한다.



<화면 8>
Debug 설정 화면



<화면 9>
Debug 설정 창의 Environment 화면



<화면 10>
Debug를 실행했을 때
Debug 영역 화면

Breakpoint 지정

프로그램을 디버깅하기 위해서는 소스 코드의 어디서부터 디버깅할 것인지 Breakpoint를 지정해야 한다. Breakpoint는 프로그램이 실행되는 과정에서 디버깅을 위해 실행을 일시 중지하는 위치를 말한다. 디버깅이 되지 않을 경우는 Breakpoint를 지정하지 않아서다. 프로그램이 멈추지 않고 끝까지 실행됐기 때문이다. Breakpoint를 지정하려면 소스 코드를 편집할 수 있는 Edit 영역의 왼쪽 모퉁이를 마우스 왼쪽 버튼으로 두 번 클릭하거나 마우스 오른쪽 버튼을 클릭하여 'Toggle PHP Breakpoint'를 선택하면 된다. Breakpoint가 지정되면 동그란 점이 표시된다. <화면 11>은 Breakpoint를 지정한 화면이다. Breakpoint는 하나 이상 지정할 수 있으며 지정된 Breakpoint는 지정 방법과 동일하게 해제할 수 있다.



<화면 11>
(리스트 4)의 소스 코드
에 Breakpoint를 지정한
화면

디버깅하기

Debug를 실행해 보자. 이전과는 다른 결과가 각 영역에 나타난다. <화면 12>의 Debug 영역을 보면 프로그램이 3번째 줄 Breakpoint에서 일시 정지되어 있다는 것을 알 수 있다. Variables 영역에는 이 시점에서 PHP 프로그램이 갖고 있는 상수와 변수에 대한 정보가 나열되어 있다. Variables 영역의 항목을 펼쳐보면 웹 서버 상수, PHP 상수, 프로그램에서 선언한 변수 등이 있다(<화면 13> 참조). 이 상태에서 프로그램을 한 줄씩 진행하면서 디버깅 하면 된다. <표 2>는 Debug 영역의 버튼에 대한 설명이다. 동일한 디버그 실행을 반복하려면 <화면 14>의 버튼을 클릭하여 실행한다.



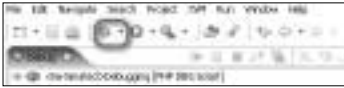
<화면 12> Debug 실행 후 Debug 영역 화면



<화면 13> Variables 영역
화면

	Resume	Breakpoint가 있는 위치까지 프로그램을 진행한다.
	Terminate	프로그램의 실행을 중단한다.
	Step Into	멈춰 있는 다음 줄로 진행한다. 만약 함수가 있다면 함수 안으로 이동한다.
	Step Over	멈춰 있는 다음 줄로 진행한다. 함수 안으로 이동하지는 않는다.
	Step Return	함수 실행을 완료하고 함수 밖으로 나온다.

<표 2> Debug 버튼 설명



〈화면 14〉 Debug 실행 버튼

웹 서버와 연동

PHP 프로그램은 웹 프로그램 특성상 웹 서버에서 실행해야 제대로 동작하고, 그 결과를 볼 수 있는 경우가 대부분이다. 물론 이클립스에서 PHP 프로그램을 만들고 실행하는 단계까지는 문제가 없다. 실행 후 콘솔 영역에 출력되는 복잡한 HTML, 전혀 알 수 없는 결과 값이 나타나면 허탈할 뿐이다. 그래서 여전히 작성한 소스 코드를 웹 서버에 전송한 후 실행시키는 개발 과정을 되풀이 하는 것이다. PHPEclipse는 작성한 프로그램을 웹 서버와 연동하여 실행할 수 있다. 지난 호에서 PHP Browser 영역에 대한 설명을 미루며 기능을 비활성화 했었다. PHP Browser 영역을 통해 이클립스에서 웹 서버의 결과를 볼 수 있다.

준비하기

PHP가 실행되는 웹 서버가 필요하다. 여기서는 가장 많이 사용하는 웹 서버 아파치를 기반으로 무료인 XAMPP을 사용한다. XAMPP는 아파치 웹 서버, MySQL 서버, FTP 서버, 메일서버 등을 쉽게 설치할 수 있도록 만들어진 패키지 프로그램이다.

XAMPP는 <http://www.apachefriends.org/en/xampp.html>에서 받을 수 있다. XAMPP의 최신 버전은 1.5.5이며 이 안에는 PHP 5.2.0과 PHP 4.4.4 버전이 들어 있다. 그러므로 우리가 설치했던 DBG 2.13.1이 지원하는 PHP 5.1.2와 PHP 4.4.2로 구성된 XAMPP 1.5.2를 설치한다. XAMPP의 이전 버전은 http://sourceforge.net/project/showfiles.php?group_id=61776&package_id=89552에서 내려 받을 수 있다. 여기서는 xampp-win32-1.5.2-installer.exe를 사용한다. [항목 3]은 XAMPP 1.5.2 패키지 내용이다.

[항목 3] XAMPP 1.5.2

- Apache 2.2.0
- MySQL 5.0.20
- PHP 5.1.2 + PHP 4.4.2-pl1 + PEAR
- eAccelerator 0.9.4-rc1 (not stabil with PHP 5.1.2)
- PHP-Switch win32 1.0 (von Apachefriends, man nehme die "php-switch.bat")
- XAMPP Control Version 2.2 from www.nat32.com
- XAMPP Security 1.0
- SQLite 2.8.15
- OpenSSL 0.9.8a

- phpMyAdmin 2.8.0.3
- ADOdb 4.80
- Mercury Mail Transport System v4.01b
- FileZilla FTP Server 0.9.14a
- Webalizer 2.01-10
- Zend Optimizer 3.0.0

설치하기

내려 받은 XAMPP 파일을 실행하여 설치한다. XAMPP는 설치과정부터 한글을 지원하므로 어렵지 않게 설치할 수 있다. 설치가 완료되면 Apache, MySQL, FTP를 서비스에 대한 등록여부를 묻는다(〈화면 15〉 참조). 서비스에 등록하게 되면, 아파치와 MySQL이 제대로 실행되지 않는 경우가 발생할 수 있으니 등록하지 않도록 한다. 그런데 Filezilla FTP 서버는 서비스에 등록해야 하므로 XAMPP 컨트롤 패널에서 Svc를 선택하여 등록한다. 〈화면 16〉은 XAMPP 컨트롤 패널 화면이다. XAMPP 컨트롤 패널에서 아파치를 Start한 후 웹 브라우저에서 <http://localhost>로 접속한다. 〈화면 17〉과 같은 화면이 나오면 정상적으로 설치된 것이다.



〈화면 15〉 서비스 등록 묻음 화면



〈화면 16〉 XAMPP 컨트롤 패널 화면



〈화면 17〉 XAMPP이 설치된 화면

XAMPP는 기본적으로 C:\Program Files\xampp에 설치된다. xampp 폴더 안에는 각 서버의 실행 파일과 설정 파일이 존재한다. 이클립스와 연동하기 위해서는 우선 아파치 웹 서버의

설정파일(기본 위치 : C:\xampp\apache\conf\httpd.conf)을 열고 DocumentRoot의 값을 이클립스의 Workspace 폴더 위치로 변경한다. 그리고 <Directory> </directory> 값도 변경해야 한다. <리스트 5>는 httpd.conf 안에서 변경할 내용이다.

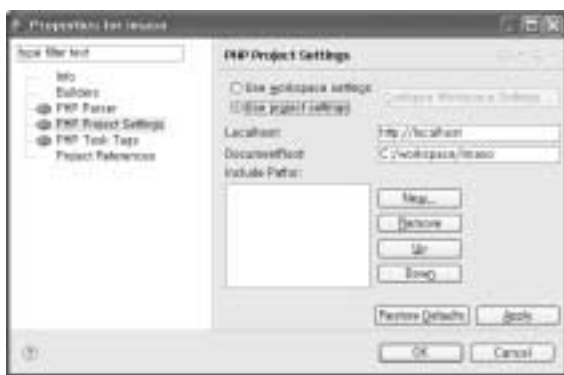
```

<리스트 5> XAMPP 아파치 웹 서버의 httpd.conf 변경 내용

DocumentRoot "C:/Program Files/xampp/htdocs"
=> DocumentRoot "C:/workspace" (이클립스에서 지정한 workspace 경로)
<Directory "C:/Program Files/xampp/htdocs">
=> <Directory "C:/workspace"> (이클립스에서 지정한 workspace 경로)
    
```

이클립스 설정하기

이클립스로 돌아와 메뉴 'Window->Preferences' 창에서 'PHEclipse Web Development->Browser Preview Defaults'를 선택하여 두 개의 항목을 활성화한다. 만약 기본 workspace가 아닌 별도의 경로에서 작업하려 한다면 웹 서버의 Document Root를 해당 폴더 경로로 지정한다. 그리고 이클립스에서 Navigator 영역의 프로젝트 이름을 마우스 오른쪽 버튼으로 클릭하여 Properties 창을 열고 PHP Project Settings에서 'Use project settings'로 변경한다. Localhost에는 웹 서버의 URL을 입력하고 DocumentRoot에 원하는 작업 경로를 지정하면 된다. <화면 18>은 프로젝트에 대한 workspace와 웹 서버 URL을 변경하는 화면이다. 여기까지 완료하고 편집 화면으로 돌아가면 <화면 19>처럼 PHP Browser에 결과가 나타난다.



<화면 18> 프로젝트 workspace 설정 화면



<화면 19> PHP Browser에 결과가 출력된 화면

웹서버와 DBG 연동하기

이클립스의 PHP Browser는 내장된 웹 브라우저이다. 이 웹 브라우저를 통해 웹 서버에서 PHP 프로그램이 실행된 결과를 볼 수 있다. 이클립스의 원격 디버그 기능을 이용하면 웹 서버에서 실행되는 PHP 프로그램을 디버깅할 수 있다. XAMPP 1.5.2를 설치한 이유도 DBG를 연동하기 위해서다.

웹 서버에 DBG를 설치하는 방법은 DBG 설치와 비슷하지만 약간의 차이가 있다. php_dbg.dll_5.1.2 파일을 XAMPP의 php/ext 폴더에 복사한 후 php_dbg.dll로 파일명을 변경한다. ext와 extension 두 개의 폴더가 있는데, PHP 5에서는 ext 폴더를 사용한다. XAMPP의 apache/bin 폴더에 있는 php.ini를 열고 앞에서 설명한 <리스트 1, 2>처럼 변경한다. 아마 extension_dir은 경로가 올바르게 지정되어 있을 것이다. 그런데 여기서 주의할 것이 하나 있다.

DBG는 eAccelerator나 Zend Optimizer가 활성화 되어 있는 경우 사용할 수 없다. XAMPP의 PHP는 기본적으로 Zend Optimizer가 활성화되어 있으므로, php.ini에서 [Zend] 항목을 찾아 모두 주석처리 한다. <화면 20>은 php.ini에서 Zend Optimizer를 비활성화 한 화면이다. php.ini를 저장하고 다시 아파치 웹 서버를 시작한다. phpinfo()를 실행해 DBG가 정상적으로 설치되었는지 확인한다(<화면 21> 참조).



<화면 20> php.ini에서 Zend Optimizer를 비활성화 한 화면



<화면 21> DBG가 설치된 화면

342쪽에서 계속