Июль / Август '04 № 5

PHPInside

электронный журнал для веб-разработчиков

phpCONF 2004

Третья международная конференция

«Современные технологии эффективной разработки веб-приложений с использованием PHP»

Редактируй это!

Содержание

Обратная связь

Всем привет.

Несмотря на то, что это лето в некоторых местах нашей необъятной родины (я говорю про планету Земля) подкачало и выдалось прохладным, оно продолжается. В связи с сезоном отпусков мы выпускаем этот номер как июльский и как августовский одновременно, а начиная со следующего – сентябрьского выпуска, все снова войдет в свою колею, и мы надеемся видеть по одному номеру в месяц без задержек и опозданий.

В связи с тем, что не все наши читатели смогли узнать о задержке выхода этого номера журнала, о которой сообщалось на форумах phpclub.ru, мы решили создать информационный сайт http://phpinside.net, который будет объединять информацию о журнале и сообщать официальные новости о новых номерах. Однако, стоит заметить, что обсуждения номеров, объявления новых выпусков, да и сами новые выпуски по-прежнему будут тесно связаны с одним из самых популярных русскоязычных PHP-сообществ – phpclub.ru. Если вы хотите всегда быть в курсе событий, посещайте сайт редакции журнала phpinside.net и форумы phpclub.ru (конкретные ссылки есть на сайте редакции).

В этот раз мы снова выступаем в роли информационного спонсора конференции по PHP в Москве, которая обещает стать одним из самых заметных событий года в мире отечественного веб-девелопмента.

И как всегда, не забывайте, что журнал делается благодаря вам! Присылайте ваши авторские статьи и переводы на адрес nw@phpinside.net. Мы ответим на все ваши вопросы.

Анонс: конференция по PHP!

23-24 сентября PHPClub совместно с интернет-агентством WebProfy проводят 3-ю международную конференцию «Современные технологии эффективной разработки веб-приложений с использованием PHP».

http://phpconf.ru

Место проведения: Москва, м. Сокол, ул. Дубосековская, д. 8, ДК МАИ.

В программе конференции:

- PHP5: новые возможности и рекомендации к переходу на новую версию.
- Вопросы безопасности.
- Хостинг PHP-приложений.
- Разработка модулей (расширений) PHP на примере memcache.
- TDD – экстремальное программирование в PHP.
- Работа с графикой.
- Поиск на сайте средствами php, mysql и ispell: выбор между возможностями, качеством и производительностью.
- Работа с шлюзами и системами оплаты (кредитки, WebMoney).
- Интеграция информационной системы предприятия (на базе 1С) с веб-сайтом и PHP-приложениями.
- Секреты PostgreSQL.
- CMF как инструмент freelance-разработки.

Организаторы:

- PHPClub (http://phpclub.ru) – это сообщество веб-разработчиков, которое существует уже более 5 лет. Задачи клуба – популяризация языка PHP и повышение качества проектов, написанных на этом языке.
- Интернет-агентство WebProfy (http://webprofy.ru) – компания, специализирующаяся на предоставлении полного спектра интернет-услуг: проектирование, программирование, дизайн, продвижение и реклама в сети интернет.

За более подробной информации обращайтесь в интернет-агентство WebProfy тел. (095) 744-31-18

Официальный сайт конференции: http://phpconf.ru

BHOST.RU

Bhost.ru — динамично развивающийся хостинг-провайдер, оказывающий услуги профессионального хостинга. Сайты пользователей размещаются в датацентре, находящимся в Москве и имеющем высокоскоростные каналы связи как с основными российскими, так и с заграничными провайдерами на серверах, имеющих сертификат Минсвязи. Высокое качество оказываемых услуг подтверждает лицензия Минсвязи № 26313 и полученное разрешение на эксплуатацию узла связи.

Каждому разработчику сайтов мы стараемся предоставить, помимо всех стандартных возможностей, оптимальные и наиболее удобные условия, как технические, так и организационные. При размещении нескольких сайтов предоставляются скидки на хостинг. Для размещения проектов, требующих нестандартных условий или программного обеспечения, может использоваться выделенный сервер, полностью или частично администрируемый разработчиком.

Опыт оказания нами услуг хостинга подтверждает целесообразность использования PHP при создании динамических сайтов, в связи с чем наша компания и стала спонсором третьей международной конференции «Современные технологии эффективной разработки веб-приложений с использованием PHP».

Предложения для спонсоров:

Спонсор конференции:

- Участие в конференции одного представителя компании.
- Размещение плаката в зале конференции с логотипом и названием организации спонсора (плакат предоставляется спонсором).
- Включение логотипа организации в информацию о конференции в электронных СМИ:
  - Phpclub.ru (http://phpclub.ru) – посещаемость проекта – более 50000 уникальных посетителей в месяц.
  - Размещение информации в выпуске, посвященном конференции, в электронном журнале PHP Inside (http://phpinside.net). Аналогичный выпуск после весенней конференции был скачан уже более 7000 раз.
  - Phpconf.ru (http://phpconf.ru) – официальный сайт конференции.
- Размещение рекламной информации (1 лист А4) об организации в сборнике материалов конференции.

Докладчик по теме «Хостинг проектов на PHP»:

- Участие в конференции одного представителя компании.
- Размещение плаката в зале конференции с логотипом и названием организации спонсора (плакат предоставляется спонсором).

- Включение логотипа организации в информацию о конференции в электронных СМИ:
    - Phpclub.ru (http://phpclub.ru) - посещаемость проекта - более 50000 уникальных посетителей в месяц.
    - Размещение информации в выпуске, посвященном конференции, в электронном журнале PHP Inside (http://phpinside.net). Аналогичный выпуск после весенней конференции был скачан уже более 7000 раз.
    - Phpconf.ru (http://phpconf.ru) - официальный сайт конференции.
- Размещение рекламной информации (1 лист А4) об организации в сборнике материалов конференции.
- Размещение информации о докладчике, в программе конференции, а также в сопроводительных материалах к конференции.

Тема доклада может быть изменена, например: «Хостинг проектов с большой нагрузкой, кластеризация, настройка PHP, безопасность». Доклад не должен носить рекламный характер.

Внимание: в программе конференции запланирован только один доклад на эту тему.

Проведения анкетирования среди участников конференции:

Мы готовы предоставить вам возможность провести анкетирование участников нашей конференции.

Всем участникам конференции будут предоставлены опросные листы. По завершении конференции, среди тех, кто сдал заполненные анкеты, будет проведена викторина. Призы для викторины предоставляются вашей компанией.

Если Вы заинтересованы в проведении опроса, вам необходимо заранее связаться с оргкомитетом конференции и уточнить условия и форму проведения анкетирования.

За более подробной информацией обращайтесь:

Телефон: +7 (095) 744-31-18

http://phpconf.ru

В фокусе

Редактируй это!

Автор:
Петр Елагин [AlienZzzz]

Когда у тебя за спиной более одного сайта, написанного на PHP, невольно начинаешь задумываться, есть ли какие-нибудь другие редакторы для PHP, кроме обычного блокнота, в котором, по сути, не реализованы даже простая подсветка синтаксиса и нахождение парных скобок?

Эта статья представляет собой обзор редакторов кода для PHP на платформе Windows и ставит перед собой задачу помочь разработчикам в нахождении оптимального варианта.

Прежде всего, для успешного сравнения, попытаемся представить себе идеальный редактор и выделить его ключевые функции и особенности. Вот они:

Функции редактирования

- подсветка синтаксиса;
    - настройка подсветки синтаксиса;
- продвинутые функции;
    - запоминание позиции курсора при выходе;
    - запоминание списка открытых файлов при выходе;
    - контроль изменений файлов внешними программами;
    - закладки;
    - поиск, замена текста;
    - нахождение парных структур;
    - мапирование команд на клавиши;
- настройка;
    - клавиш;
    - меню;
- одновременная поддержка различных кодировок.

Интеграция с PHP и пользовательскими функциями

- наличие стилиста (отвечает за расстановку отступов и структуру скрипта);
    - настройка стилиста;
- подключение к словарю функций;
- встроенная справка по PHP или интеграция с официальной документацией;

- показ и навигация по структуре классов и функций с учетом включений файлов (includes);
- автозаполнение функций, классов, переменных и.т.д;
- управление проектами.

Внешняя интеграция

- работа с FTP;
  - настройка;
  - мапирование локальных файлов и файлов на FTP;
- управление взаимодействием с базой данных (например, MySQL);
  - настройка;
  - получение ResultSet – набора данных, результата определенного sql-запроса.

Как я уже говорил, в данную статью вошли обзоры редакторов под операционную систему Windows, и, замечу, что сюда не входит рассмотрение и сравнение отладчиков кода. Также необходимо отметить, что данная статья является мнением автора и его попыткой помочь разработчикам и сэкономить их время.

Теперь, когда мы немного определились с тем, что будем искать в редакторах кода, нам осталось только подобрать эти самые редакторы и применительно к ним сделать обзорный анализ.

Для начала нам нужно получить список программ из какого-нибудь источника. Для этого вполне подойдет сайт http://www.thelinuxconsultancy.co.uk/phpeditors/ (пусть вас не смущает его название), указанный в списке полезных ссылок на http://php.net.

В фильтре поисковой строки сайта выберем Windows, и, нажав «фильтр», получим список из 91 редактора. Конечно же, всего номера журнала не хватит на обзор всего этого списка, поэтому попробуем рассмотреть только те программы, которые делают основной упор на PHP или являются одними из самых популярных среди php-разработчиков. Каждый выбранный редактор мы будем тем или иным образом сравнивать с нашим списком функционала «идеального редактора», и в конечном итоге получим некий рейтинг редакторов, который читатель сможет составить для себя сам.

Вначале последовательно сделаем обзор из полученного списка, а в конце рассмотрим таких «монстров жанра», как Dreamviewer, Maguma Studio и Zend Studio. Скриншоты интерфейса всех программ можно найти в приложении к журналу. Итак, вперед!

Каждый выбранный редактор мы будем тем или иным образом сравнивать с нашим списком функционала «идеального редактора», и в конечном итоге получим некий рейтинг редакторов, который читатель сможет составить для себя сам

Название: Antechinus PHP Editor

Сайт: http://www.c-point.com/apedit.htm

Последняя версия: 2.2

Статус: платный, стоит $35.

Размер в дистрибутиве: 5.7 Мб

Размер на диске: 7.3 Мб

Оригинальное описание:

Antechinus PHP Editor v2.2 allows you to create, edit and run PHP scripts from the integrated environment. PHP Editor works on Windows 95, 98, 2000 Me, 2000 Pro, NT, XP, and Server 2003.

Скриншот:

[См. в приложении к журналу – AntechinusPHPEditor.png]

Обзор:

В программе присутствует подсветка синтаксиса, которая, однако, в нашем тесте не поняла переноса строки, чем немного ухудшила читаемость кода. В остальном подсветка, вроде, не плоха, но еще не хватает подсветки пользовательских функций.

Настройка редактора ничем не отличается от других, но, тем не менее, как-то скудно: настраиваются только внешний вид (шрифт, цвета текста, цвет фона) и количество пробелов в табуляции.

Открытые файлы программа запоминает, но не помнит положение курсора на момент закрытия, что немного усложняет процесс редакции. На параллельное изменение файлов в другой программе редактор никак не реагирует. Присутствуют закладки, но они просты – никак не нумеруются и не хранят комментариев. Перемещение по закладком происходит строго по нажатию клавиши F2. Поиск и замена реализована, как у стандартного блокнота, однако, мне пришлось вызвать диалог замены из меню, стандартная комбинация клавиш Ctrl + H никак не реагировала на это.

Отсутствует стилист и автозаполнение. Подсказка по функции всплывает только после полного написания ее наименования. Справка по PHP есть, но она не контекстная. Также я обнаружил, что программа определяет структуру классов, но не предоставляет навигации по ней и не «видит» включений других файлов. Нет даже намека на работу с FTP и базами данных.

Вывод:

Редактор, в котором существует базовый набор функций, но и те не всегда хорошо реализованы.

Название: Arisesoft Winsyntax

Сайт: http://www.winsyntax.com/

Последняя версия: 2.0

Статус: freeware

Размер в дистрибутиве: 453 кб

Размер на диске: 545 кб

Оригинальное описание:

Arisesoft Winsyntax – is the small and free PHP-code editor, with the fast syntax highlighting engine and the context help for any PHP keywords for comfortable coding.

Скриншот:

[См. в приложении к журналу – ArisesoftWinsyntax.png]

Обзор:

В редакторе существует простая подсветка синтаксиса, но она никак не настраивается. Программа не запоминает открытых файлов и позиции курсора при выходе (для восстановления при открытии). Закладки отсутствуют.

Положительными моментами является то, что редактор отслеживает параллельное изменение файла внешними программами и имеет неплохой поиск и замену. Присутствует даже некая функция XSearch, с помощью которой можно осуществлять поиск с учетом директорий и типов файлов. К плюсам можно отнести и умение находить и подсвечивать парные структуры (например, фигурные скобки).

Отсутствует стилист и минимальная возможность настраивать цветовую гамму интерфейса. В возможности пользователя входит только изменение шрифта. Зато легко можно редактировать файлы в различных кодировках, правда, кодировку необходимо указывать в настройках перед открытием файла.

Все функции PHP программа делает полужирными в написании. Список этих функций расширяем и находится в файле \winsyntax\Parsers\php.wsp. В программе нет автозаполнения и возможности работать с FTP и базами данных. Нет так же и возможности вести проекты, правда, присутствует неплохой навигатор по дискам компьютера, который в некоторых случаях может заменить менеджер проектов.

Вывод:

Простой, легкий редактор, в котором можно что-то быстро написать, и им можно заменить привычный блокнот.

Название: Code-Genie

Сайт: http://www.code-genie.com/cgenie.html

Последняя версия: 4.05.18

Статус: Commercial (trial-version)

Размер в дистрибутиве: 732/552 KBytes

Размер на диске: 1200 кб

Оригинальное описание:

Code-Genie is a text-editor designed for programmers and веб-developers, but others can also use it as a replacement of Notepad. Features syntax-highlighing for HTML, PHP, CSS, JavaScript, Java, C++ etc.; keyword-lookup; color printing; auto-complete.

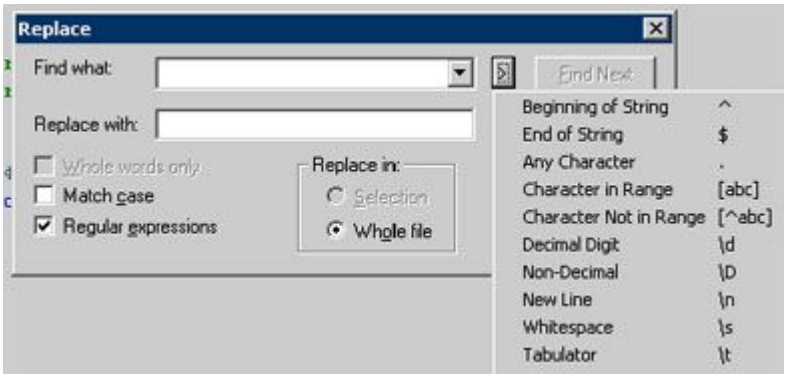
Скриншот:

[См. в приложении к журналу – CodeGenie.png]

Обзор:

Подсветка синтаксиса просто так не настраивается, надо скачать еще один дополнительный файл по подсветке. При закрытии программы запоминается позиция курсора и список открытых файлов. Если файл открыт в редакторе, то его поменять внешней программой нельзя (делается файл .lock). Закладки есть, но очень скудные (нет названий и номеров), и редактор их не запоминает при выходе.

Для поиска и замены реализована возможность писать регулярные выражения, что очень удобно:



Настраивается редактор изменением записей в файле конфигурации. Поддерживаются различные кодировки, хотя при тестировании я так и не смог нормально прочитать файл в KOI-8. Подключение к словарю функций имеется, при нажатии на Ctrl+Space, редактор даже воспринимает пользовательские функции, но довольно странно, после некоторых экспериментов было установлено, что он понимает только те функции, которые были продекларированы до позиции курсора. Контекстная справка подключается дополнительно после установки программы. Есть функция автозаполнения, но редактор не «видит» включенных файлов.

Дополнительно:

Большой плюс – то, что в нем есть HEX-редактор, но здесь мы рассматриваем только PHP-редакторы.

Вывод:

Удобный и простенький редактор кода, лучше, чем WinSyntax, но только если вы работаете в одной кодировке и если вам нужно иногда работать с HEX-кодом.

Название: CodeCharge Studio

Сайт: http://www.codecharge.com

Последняя версия: 2.3

Статус: Commercial, Evaluation available

Размер в дистрибутиве: 24.1 Мб

Размер на диске: 33.1 Мб

Оригинальное описание:

CodeCharge Studio builds on CodeCharge 2.0's features adding more powerful controls, visual editors and provides even more control and power. CodeCharge Studio features HTML and code editors, pre-built drag-and-drop components to visually assemble pages, powerful wizards to generate applications with a few clicks.

Скриншот:

[См. в приложении к журналу – CodeChargeStudio.png]

Обзор:

После определенной работы с творением от YesSoftware понимаешь, что продукт сделан хорошо, но не подходит для редакции. Точку зрения обосную списком функций, которые в нем отсутствуют:

- подсветка не настраивается и полностью отсутствует для строк, чисел и пользовательских функций;
- при открытии и закрытии он не запоминает позицию курсора, и, конечно же, список открытых файлов. Кроме того, он не запомнил моего проекта и попросил показать, где тот находится при следующем запуске (в настройках параметра «открыть последний проект» нет);
- при изменении файлов внешней программой, редактор никак не отреагировал;
- нет настройки закладок и быстрых клавиш;
- правая клавиша мыши, по которой мы привыкли видеть выпадающее меню – Popup, никак не реагирует. Т.е. Popup-меню нет;
- открыть файл в другой кодировке он так и не сумел, хотя я везде проставил (например, в свойствах проекта), что файлы у меня в Koi8-R;
- интеграция с PHP отсутствует: ни справки, ни автозаполнения нет;
- стилист отсутствует, есть, правда, какая-то клавиша «Generate Code», но ее смысл в этом редакторе интуитивно не ясен;
- управление проектами есть, но не совсем ясно, как добавлять в проект файлы, потому как программа предлагает только создать (не открыть или добавить) новый файл и добавить его к проекту. Я обошел эту ситуацию «сбоку»: создал файл, потом переписал свой файл поверх этого.

Серьезным плюсом программы можно назвать управление соединениями с FTP и базой данных.

Вывод:

Я рисую формы и устанавливаю соединение с базой данных, потом нажимаю «Generate Code», и программа сама делает код. Тип визарда по всему сайту, поэтому в первую очередь этот редактор будет интересен тем, кто привык к подобным рисователям форм, но определенно сказать могу, что писать на нем код просто невозможно, т.е. возможность есть, но проще использовать что-то другое.

Название: Davor's PHP Constructor

Сайт:http://www.pleskina.com

Последняя версия: 2.3

Статус: Shareware

Размер в дистрибутиве: 1 Мб

Размер на диске: 2.4 Мб

Описание:

Davor's PHP Constructor is compact and fast Windows IDE for PHP development with : Free-style windowed interface with configurable desktop and multiple edit windows, Advanced multipaged editor with syntax Highlighting for several programming languages – PHP, HTML, CSS, XML, JS, SQL and ASP,Project management,PHP syntax checking and script interpretation,Project and/or script/page Run/Preview,Advanced code analysis (PHP, JS and HTML),Basic HTML WYSIWYG editor,FTP-based file synchronization,NEW! (in development) Internal PHP Script Debugger! No need for external debuggers.

Скриншот:

[См. в приложении к журналу – Davor'sPHPConstructor.png]

Обзор:

Подсветка синтаксиса и ее настройка никаких нареканий не вызвала. Реализовано запоминание списка открытых файлов и позиции курсора на момент закрытия программы. Редактор справляется с поиском и заменой, а также с обнаружением парных конструкций.

Нет закладок, поэтому если у тебя 1000 строчек кода, навигация не очень удобна. Настроек клавишей и меню нет. И опять проблема с разными кодировками – редактор не захотел правильно загружать файл в другой кодировке. Хочу сказать, что очень мало редакторов понимают кодировку на лету. Интеграция с PHP выполнена только на уровне выпадающего списка функций, справки и автозаполнения нет.

В программе отсутствует стилист, однако, есть инспектор кода. Правда, как показала практика, пока открыт инспектор кода, невозможно работать в редакторе. Программа «видит» подключенные файлы, но не распознает их структуру.

В пробной версии программы отсутствует работа с FTP и базами данных, но в полной версии поддержка публикации в интернете обещана.

Вывод:

Неплохой легкий редактор с приятным и интуитивно понятным интерфейсом, с поддержкой FTP и проектов. Минус в том, что нет поддержки закладок и кодировок, а также очень скудно сделана интеграция с PHP (автозаполнение, помощь, пользовательские функции).

Название: DzSoft PHP Editor

Сайт: http://www.dzsoft.com/dzphp.htm

Последняя версия: 2.0

Статус: Shareware $39

Размер в дистрибутиве: 1.7 Мб

Размер на диске: 3.4 Мб

Оригинальное описание:

DzSoft PHP Editor is a handy and powerful tool for writing and testing PHP and HTML pages. With its deceptive simplicity, the interface of DzSoft PHP Editor is comfortable both for beginners and experienced programmers, making PHP development easy and productive

Скриншот:

[См. в приложении к журналу – DzSoftPHPEditor.png]

Обзор:

В этом редакторе особо стоит отметить подсветку, которая настраивается не только по элементам, но и по различным схемам. Стоит отметить, что программа умеет подсвечивать, например, отдельно «Float» или «String String». Запомнить список файлов и позицию курсора редактор не захотел, и на изменение файла извне никак не отреагировал.

Закладки есть, и по клавишам. Удобно, потому как каждой закладке присваивается свой номер. Поиск и замена с регулярными выражениями, что, несомненно, является плюсом для редактора.

Программа находит парные скобки, но цветом их никак не выделяет. Настроить команды на разные клавиши в этом редакторе невозможно, можно только задать тип настройки:

В редакторе отсутствует автозаполнение, однако можно подключить внешнюю справку в формате CHM, работа с которой будет контекстной.

Приятно удивила возможность открывать и редактировать файлы прямо по FTP. С базами данных программа не работает и периодически выдает ошибки «Access violation».

Вывод:

Хороший редактор пхп кода, если учесть встроенную возможность работы с FTP и включения файла справки, но отсутствие автозаполнения и баги играют не лучшую роль.

Название: HAPedit

Сайт: http://hapedit.free.fr/index.php

Последняя версия: 3.1

Статус: Freeware

Размер в дистрибутиве: 1.3 Мб

Размер на диске: 2.7 Мб

Оригинальное описание:

HAPedit is an acronym for Html Asp Php editor; a win32 text-mode editor useful for all developpers of dynamic web pages. Its main features are: syntax highlighting for html/php, html/asp, html, JavaScript, CSS and SQL; page preview in browser; project manager; php code «compilation»; edit html tags; code completion; preview images; SQL console; FTP Manager; Code Explorer and more.

Скриншот:

[См. в приложении к журналу – HAPedit.png]

Обзор:

Редактор успешно справился с подсветкой переноса строк. Настройка подсветки очень гибкая и позволяет не только задавать цвета для различных операторов и конструкций, но и составлять пользовательские цвета.

При запуске программа прекрасно вспомнила список открытых файлов и последнюю позицию курсора, а когда внешним редактором был изменен текст файла, HAPedit любезно предложил перегрузить этот файл для соблюдения актуальности. Отлично реализованы поиск и замена, так как есть возможность не только использовать регулярные выражения, но и применять фильтры к результатам поиска (это нововведение очень удобно, если надо найти не все выражения, а по определенной логике).

Общую картину немного омрачило отсутствие закладок и неполная поддержка поиска парных структур. Редактор нашел парные скобки, но отказался искать второй экземпляр кавычек. Настройки клавиш нет, есть пункт в меню «Open config file», но про клавиши там ничего не сказано.

Вообще, настроить можно только панель инструментов (toolbar). Поддержка разных кодировок отсутствует.

Складывается мнение, что программисты большинства редакторов понятия не имеют о других кодировках, потому как пока из всех рассмотренных редакторов открыть файл в кодировке, отличной от Windows-1251, смог только один – Winsyntax.

Программа неплохо интегрируется с PHP. В файле php.lng описываются все функции и вносится краткое описание каждой из них. Справка по PHP настраивается (путь до файла справки) в меню, но контекстный хелп вызвать так и не удалось (пункт меню не активен). Автозаполнение есть, но реализовано не полностью, так как редактор прекрасно показывал встроенные функции, но вот пользовательские показывал только те, которые были определены до курсора.

Функции, находящиеся во включенных файлах классов, программа не понимает, и они не работают по автозаполнению. В один проект могут подключаться файлы только из одной папки проекта. Есть возможность работы с FTP, однако прямое редактирование удаленных файлов невозможно. Для каждого проекта сохраняются отдельные настройки FTP. Не реализована работа с базами данных.

Хочется отметить, что в редактор добавлена возможность Plugging (например, модуль работы с FTP, это плагин), а также возможность обновления программы. Есть возможность просмотра времени загрузки сайта, что очень удобно при больших проектах, потому как разработчик самостоятельно определяет скорость связи.

Вывод:

Этот редактор оставил очень хорошие впечатления, и если не считать недоделанного автозаполнения, открытия файлов непосредственно из FTP (например, было бы удобно добавлять проект, как FTP) и поддержки различных кодировок, то можно смело ставить ему 5+ .

Название: PHP Expert Editor

Сайт: http://www.ankord.com/ru/index.html

Последняя версия: 3.2

Статус: Для пользователей стран бывшего СССР бесплатная регистрация

Размер в дистрибутиве: 2.6 Мб

Размер на диске: 3 Мб

Оригинальное описание:

PHP Expert Editor – удобный в использовании PHP-редактор, разработанный специально для PHP-мастеров.

PHP Expert Editor имеет встроенный HTTP сервер и отладчик для запуска и отладки PHP скриптов (вы можете использовать любой внешний HTTP сервер), проверка синтаксиса PHP, встроенный браузер, FTP-клиент, файловый менеджер, настраиваемые шаблоны кода, три режима подсветки кода (PHP & HTML, HTML only, PHP only), подсветка Java Script и CSS фйлов, функции быстрой навигации в PHP-коде, и многое другое.

Скриншот:

[См. в приложении к журналу – PHPExpertEditor.png]

Обзор:

После первого запуска этого редактора я подумал, что он очень напоминает DzSoft PHP Editor, хотя после некоторого времени использования, стало ясно, что схожесть эта только внешняя. Различия начинаются уже в том, что имеется русская версия программы. В PHP Expert Editor отлично реализована настройка подсветки. Также редактор запоминает позицию курсора и список открытых файлов, а на изменение файла в другой программе он отреагировал адекватно. Закладки реализованы по клавишам и номерам. Активизируются по нажатию комбинации клавиш ALT и номера закладки. В поиске и замене есть возможность использовать регулярные выражения, что только добавляет плюсов этому редактору.

Сложнее всего реализовать не конкретную кодировку, а принципиальный механизм поддержки разных кодировок

Настройки клавиш нет, но практически все операции уже мапированы на клавиши. Нет поддержки различных кодировок, но вот что сообщили мне разработчики данного редактора на мой запрос о кодировках: «Сложнее всего реализовать не конкретную кодировку, а принципиальный механизм поддержки разных кодировок. KOI-8R появится первой – это однозначно».

Теперь насчет интеграции кода. Стилист отсутствует, хотя это было бы очень удобно, судя по нашему обзору, он пока не присутствовал ни в одном из рассмотренных на данный момент редакторов. Подключение к словарю функций выполнено прекрасно.

Контекстная справка тоже, при нажатии на Ctrl+ F1 открывается справка по конкретной функции. Автозаполнение также присутствует, однако, оно понимает только встроенные функции и те функции, которые продекларированы в текущем файле, но это обещают исправить в следующих версиях. Автозаполнение не действует на переменные.

Управление FTP, файлами и проектами в PHP Expert Editor заслуживает особого внимания.

Все функции выведены в отдельный Bar, в виде которого, кстати, реализована и навигация по классу.

Не хватает только работы с базами данных, но я думаю, что это поправят в последующих версиях.

Вывод:

Пока единственным недостатком этого редактора является немного недоработанное автозаполнение и невозможность работы с разными кодировками и базой данных, а в остальном редактор заслуживает награды.

Название: Maguma Workbench

Сайт: http://www.maguma.com/

Последняя версия: 2.0.5

Статус: Commercial

Размер в дистрибутиве: 15.6 Мб

Размер на диске: 33 Мб

Оригинальное описание:

Maguma Workbench is built on the philosophy that a craftmans' workbench is the base on which all activities are accomplished, hence the name. A workbench is flexible, expandable, and easy to modify for whatever task is at hand. Maguma Workbench is designed to function just this way. Workbench is a new breed of IDE (Intergrated Development Enviroment) that assists the user in crafting their web applications or websites more effectively.

Скриншот:

[См. в приложении к журналу – MagumaWorkbench.png]

Обзор:

Подсветка, как и в других редакторов такого уровня, реализована неплохо. Программа справилась и с подсветкой строки, содержащей переносы. Настройка редактора есть в меню, но не очень расширенная, авторы программы могли бы и потрудиться (все-таки платная). Запоминания списка файлов и позиции курсора отсутствует, но на изменение файла из другой программы редактор отреагировал.

Присутствуют закладки (даже в двух видах), а поиск и замена реализованы с поддержкой регулярных выражений. Парные скобки определяются, причем размечаются разным цветом (различается скобка пустой функции и скобка функции с параметрами).

Большинство команд мапировано на клавиши, но настроек клавиш нет. Поддержка разных кодировок отсутствует.

Подключение к словарю функций выполнено сносно, не хватает только описания функции и аргументов. Есть встроенная и контекстная справка по PHP. Проекты в этом редакторе называются WorkSpace. Все происходит как обычно: добавляются файлы и директории, но есть один момент – добавлять можно только локальные файлы. Возможность открывать файлы с FTP есть, но добавлять их в проект нельзя. С базами данных не взаимодействует.

Вывод:

Обычный редактор с рядом существенных недостатков, таких, как неправильная работа автозаполнения, недоработанное управление проектами, отсутствие запоминания позиции курсора и списка открытых файлов.

Название: Dreamweaver

Сайт: http://www.macromedia.com/software/dreamweaver/

Последняя версия: MX 2004

Статус: Commercial

Размер в дистрибутиве: 104 Мб

Размер на диске: 132 Мб

Оригинальное описание:

Macromedia Dreamweaver MX 2004 is an easy, powerful, and open authoring tool that every member of the development team can use to quickly build robust websites and Internet applications. It provides rich, powerful CSS support and lets you work within one environment to easily create and manage any professional website, whether it's built using HTML, XHTML, XML, web services, ColdFusion, ASP.NET, ASP, JSP, or PHP

Скриншот:

[См. в приложении к журналу – Dreamweaver.png]

Обзор:

Традиционно начнем с рассмотрения подсветки и ее настройки. Продукты фирмы Macromedia всегда радовали пользователей функциональностью и простотой в использовании. Подсветка и ее настройка реализованы превосходно. В программе существует очень большое количество типов подсветки. Настройка цвета сопровождается функцией предпросмотра.

Редактор успешно справился с запоминанием списка открытых файлов, но вот с курсором подкачал – он не запомнил позицию. Контроль параллельных изменений файла в другой программе присутствует. Закладок нет, но зато прекрасно реализованы Поиск/Замена. Помимо обычного поиска есть еще так называемый «FindAll», где результаты поиска сводятся в таблицу (очень удобная возможность).

Нахождения или подсветки парных структур нет, что очень не удобно в повседневной работе, зато есть настройка всех команд на клавиши – большая редкость в нашем обзоре. Есть поддержка различных кодировок, однако принцип ее работы не совсем ясен. Эксперименты показали, что редактор правильно определяет кодировку не в зависимости от настроек, а от того, указана ли кодировка в тегах HTML.

Стилиста нет. Подключение к словарю функций есть – находится в файле «\Configuration\Content\Reference\PHP\Reference.xml». Есть автозаполнение по встроенным функциям, однако нет распознавания включенных файлов и кода в них. Нет контекстной справки по PHP и нет навигации по структуре кода.

В редакторе реализовано управление проектами – сайтами. Есть также поддержка FTP. Она заключается в указании данных фтп-сервера, на который будет производиться выкладка файлов. Можно настроить редактор таким образом, что при сохранении редактируемого файла он будет отправляться на FTP. Когда я писал статью, я пользовался многими редакторами, лично для меня очень важно, чтобы редактор мог работать напрямую по FTP, поэтому локальная копия движка моего сайта не изменялась. И при тесте Dreamviewer сохранил мою локальную копию поверх той копии, которая была на сайте. Он даже не посмотрел на дату изменения! Будте внимательны!

Из плюсов работы с FTP могу выделить то, что редактор прекрасно синхронизирует локальную директорию и удаленный сайт (при создании директории на локальном диске он создает ее на FTP).

Работа с базой данных возможна, но подключиться удалось только к локальной базе – редактор даже предложил мне написать код по работе с базой данных (есть настройка по ResultSet, например).

Вывод:

Этот редактор скорее подходит для рисования дизайна, нежели для кодирования. Есть обширная настройка и по клавишам и по подсветке, но отсутствие контекстной справки и недоработанная связь, управление базами данных и отсутствие инспектора кода делают его не лучшим выбором среди редакторов кода.

Название: PHPEdit

Сайт: http://www.waterproof.fr/

Последняя версия: 1.0.1.59

Статус: Commercial

Размер в дистрибутиве: 24.5 Мб

Размер на диске: 67.5 Мб

Оригинальное описание:

PHPEdit is the best IDE (Integrated Developement Environment) under windows to work with PHP. It offer a bunch of tools which allow you to work easier and faster on your eash day work. Here is a short overview of its features: Syntax Highlighting ,Code Hint, Code Insight , Integrated PHP debugger ,Help Generator ,Customizable shortcuts ,More than 100 scriptable commands ,Keyboard templates, Todo report generator ,QuickMarks ,Plugins.

Скриншот:

[См. в приложении к журналу – PHPEdit.png]

Обзор:

Настройка подсветки синтаксиса в этом редакторе более насыщена, чем у других: помимо привычного выбора типов расцветки и цветов, еще присутствуют такие параметры, как «Кодировка» и Эффект изменения регистра. Эффект изменения регистра, кстати, очень удобен. Например, можно заставить редактор писать все зарезервированные слова в нижнем регистре.

Сама подсветка синтаксиса хорошо реализована – он без проблем определил, что строковая переменная находится не на одной строчке, а на нескольких. Списка открытых файлов программа не помнит, т.е. при открытии ее заново она просто открывает новый файл с названием untitled.php. При параллельном изменении файла в другой программе редактор попросил этот файл перезагрузить.

Есть закладки по номерам, и еще хочу отметить очень интересную, как мне кажется, идею. В PHPEdit можно расставить пометки (красные треугольники) в любом месте текста , и перемещаться по ним, если есть необходимость. Это очень интересная идея, потому как в остальных редакторах, можно поставить закладку только на всю строчку, а здесь – в определенном месте на строчке. К сожалению, они не запоминаются при выходе.

Поиск и замена реализованы с возможностью написания регулярных выражений. Парные структуры подсвечиваются, и также по ним можно перемещаться с помощью клавиш. Заметьте, что все команды мапируются на клавиши, для этого есть даже целый интерфейс, в котором мы можем написать не одну команду, а группу команд, и добавить для этого клавишу.

Это, по-моему, единственный редактор кода, в котором реализован стилист. Он называется Code Beautifier и помогает правильно расставить структуру документа, сделать пробелы там, где надо и.т.д. Он имеет настройку, в которой галочками можно выделить те действия, которые вам нужны.

Подключение к словарю функций реализовано, но, к сожалению, редактор не определил, что подключены дополнительные классы и не показал мне их список. Есть также встроенная и контекстная справка по PHP.

Автозаполнение действует всегда по нажатию Ctll+Space, причем показываются не только функции, но и переменные. Управление проектами выполнено в виде отдельного дополнения, в нем вы можете настроить FTP-доступ и другие полезные параметры.

Возможно, вас также заинтересует реализация доступа к системе контроля версий. Она будет очень удобна, если проект разрабатывает не один человек, а группа. Работа с FTP выведена в проект, т.е. при создании проекта вы можете создать FTP-соединение.

Вывод:

Превосходный редактор PHP-кода с такими прекрасными возможностями, как встроенный стилист, мапирование всех команд на клавиши, добавление шаблонов быстрого набора кода, и т.д.

Он все-таки имеет ряд минусов, а именно: он забывает список открытых файлов, неправильно определяет кодировки и не сохраняет закладки в файлах.

И теперь, как я и обещал, я расскажу вам про очень хороший редактор PHP-кода, это даже не редактор, это целая среда.

Название: Zend Studio

Сайт: http://www.zend.com/store/products/zend-studio.php

Последняя версия: 3.0.5

Статус: Commercial

Размер в дистрибутиве: 31 Мб

Размер на диске: 101 Мб

Оригинальное описание:

Zend Studio™ – The Professional PHP Development Environment.

Скриншот:

[См. в приложении к журналу – ZendStudio.png]

Обзор:

Как видно по скриншоту (смотрите в приложении к журналу), этот редактор мультиязычный и поддерживает русский язык. Для этого не надо скачивать что-то дополнительно.

Подсветка прекрасно реализована, в том числе и настройка. В окне настройки поддерживаются профайлы настроек цвета и есть клавиша «Применить», нажав на которую можно сразу посмотреть, как изменится расцветка вашего кода, правда, не хватает еще выделения тегов HTML в строках.

Запоминание позиции курсора и списка файлов при выходе, контроль над изменением из других программ есть. Что не менее важно, редактор запоминает позиции закладок в файлах! Потому что, когда у тебя классы выползают за 1000 строчек кода, очень сложно делать навигацию по классу, поэтому я делаю закладки на ключевые моменты кода. Когда приходится выходить и входить в редактор, закладки нужно расставлять заново**.** Перемещение по закладкам осуществляется с помощью клавиши F2, но не по номерам, зато есть список всех закладок с возможностью заполнения комментария к каждой закладке**.**

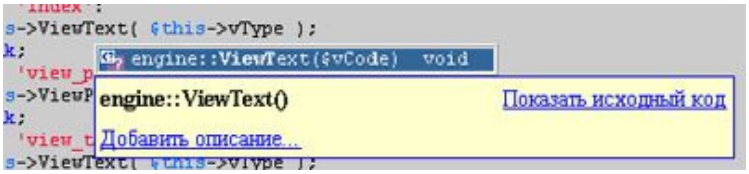
Поиск и замена реализованы, как положено, с регулярными выражениями, жаль только, что нет возможности вывести результаты поиска в отдельное окно.

Парные структуры редактор ищет и показывает, причем не только скобки, но и кавычки. В меню есть настройка горячих клавиш, поэтому вы можете легко поменять неудобную для вас комбинацию на другую.

Отдельно хочу сказать на счет поддержки разных кодировок. Создатели этого редактора пошли простым путем и сделали пункт «Кодировка» при открытии файла. Поэтому с кодировками у меня проблем не было (проблем не было только с этим редактором!).

Интеграция с PHP у Zend Studio великолепная (еще бы!). Стилиста нет, но есть команда автоматического выравнивания кода. Редактор на лету подсвечивает ошибочный код, что очень удобно. Также есть анализатор кода, который покажет, где у вас есть ошибки и какие объявленные переменные вы ни разу не использовали.

Все встроенные функции и функции пользователей показываются во всплывающих подсказках, включая и те, что подключены в отдельных файлах, причем к каждой функции можно добавить свое описание.



Еще нужно отметить, что показываются не только функции, но и переменные, причем очень умно. Если у меня будет конструкция типа такой:

```
$Eng-> …
```

то в выпадающем списке будут только функции этого класса, но если написать:

```
$this->…
```

то высветятся функции только текущего класса. Если этот класс наследовался от другого класса, то высветятся и функции родителя.

Что касается управления проектами, то только можно похвалить создателей этого замечательного редактора. Работа с файлами, FTP и проектами сведена в одно окошко, причем необходимо заметить, что добавить к проекту можно любой файл, вплоть до FTP, что очень удобно.

Вывод:

По моему мнению, это лучший редактор кода. Я написал далеко не обо всех достоинствах это замечательного редактора, и если вы начнете в нем работать, думаю, сами найдете их.

Если говорить о минусах, то не хватает поддержки работы с базой данных и налицо долгая загрузка (редактор написан на Java, возможно, поэтому грузится не быстро).

В начале статьи, я ставил перед собой цель помочь читателю и сэкономить его время при выборе редактора PHP-кода. Однако именно за читателем остается окончательное решение, и я советую присмотреться к следующим редакторам:

PHPEdit от Waterproof – этот редактор очень хорош своими новаторскими идеями и прекрасно справляется с функциями набора кода, и если вы работаете в одной кодировке и вам не надо работать с FTP на прямую, то я считаю, это ваш выбор.

Macromedia Dreamviewer – редактор для тех, кто любит работать не только с написанием кода, но и рисованием дизайна.

Arisesoft Winsyntax – это выбор тех, кому надо быстро что-то поправить, он легко грузится и правильно определяет все кодировки, а возможность открывать файлы непосредственно с FTP делает его серьезным конкурентом редакторам его класса.

PHP Expert Editor – очень хороший редактор. Второе место. Если, как заявлено авторами этой программы, в следующей версии они поправят несколько недоработок, то у него есть все шансы занять первое место.

Zend Studio – как я уже говорил, первое место занял этот редактор. Не буду еще раз пересказывать его достоинства и преимущества, просто вам самим надо взять и попробовать его в деле.

Идеи

Инъективные SQL-атаки – вы в безопасности?

Автор:
Митчел Харпер
Адаптация и перевод:
Сергей Корнильев
Александр Ширяев

База данных является сердцем многих веб-приложений: она содержит данные жизненно необходимые для веб-сайтов и приложений. В ней может содержаться такая засекреченная информация как, счета, платежи, инвентарь и т.д. Посредством сочетания БД и скриптовых языков для Web мы как разработчики можем выпускать сайты, которые позволяют удовлетворять потребности клиента, оплачивать счета, а главное – продвигать нашу деятельность.

Но что происходит, когда вы обнаруживаете, что ваша важная информация не в безопасности? Что происходит, когда вы узнаете, что только что был обнаружен новый баг в системе безопасности? Вероятнее всего, вы ставите заплаты или обновляете сервер БД до новой версии, без багов. Дыры в безопасности и патчи к ним всегда находятся и для БД, и для языков программирования, но я готов спорить, что 9 из 10 среди вас никогда не слышали об инъективных SQL-атаках.

В этой статье я попытаюсь пролить свет на эти недокументированные атаки и объясню, что такое инъективная SQL-атака, и как вы можете обезопасить от них вашу компанию. В конце этой статьи вам будут предложены ситуации, в которых инъективная SQL-атака могла бы позволить неавторизованной личности проникнуть в вашу систему, а также вы научитесь исправлять существующий код с тем, чтобы не допустить этого.

Что такое инъективная SQL-атака?

Как вы уже знаете, SQL расшифровывается как Structured Query Language (структурированный язык запросов). Этот язык в своем проявлении имеет различные диалекты, среди которых самым популярным является диалект, основанный на SQL-92 ANSI стандарте. SQL-запрос содержит в себе одну или более SQL-команду, такую как SELECT, UPDATE или INSERT. В запросах, содержащих команду SELECT, обычно содержится оператор, определяющий, какую информацию возвратит запрос. К примеру:

```
SELECT * FROM Users WHERE userName = 'justin';
```

Содержащийся в вышеприведенном запросе оператор WHERE username = 'justin' означает, что мы хотим получить только те записи из таблицы Users, в которых поле userName тождественно строке Justin.

Именно этот тип запросов делает язык SQL таким популярным и гибким… именно он делает возможными инъективные SQL-атаки. Как видно из названия, инъективная SQL-атака делает «инъекцию», манипулирует SQL-кодом. Путем добавления непредусмотренного SQL-кода в запрос можно манипулировать БД самым непредвиденным путем.

Самым распространенным способом авторизовать пользователя на сайте является использование HTML-форм, посредством которых он может ввести свой логин и пароль. Допустим, у нас есть следующая HTML-форма:

```
<form name="frmLogin" action="login.php" method="post">
Username: <input type="text" name="userName">
Password: <input type="text" name="password">
<input type="submit">
</form>
```

Когда форма утверждена, содержание полей username и password передаются скрипту login.php и становятся доступными через массив $_POST. Наиболее простым способом авторизовать пользователя было бы создать SQL-запрос и в нем проверить по БД достоверность пользовательских данных. Мы могли бы создать скрипт наподобие того, что приведен в листинге 1.

```
<?
$userName1 = $_POST["userName"];
$password1 = $_POST["password"];

$hostname = "localhost";
$dbName = "myDB";
$username = "sa";
$passdb = "";
$linkid=mssql_connect($hostname, $username, $passdb);
        mssql_select_db($dbName, $linkid);

$query = "select * from users where userName = '$userName1' and userPass
= '$password1'";
$result = mssql_query($query, $linkid);
$number = Mssql_NUM_ROWS($result);

IF ($number == 0)
echo "Bad Credentials";
else echo "Logged In";

?>
```

Листинг 1.

В вышеприведенном примере пользователь видит сообщение «Logged In», если данные совпадают, и «Bad Credentials» в противном случае. Прежде, чем мы продолжим, давай создадим БД, к которой был выполнен запрос.

Также давайте создадим таблицу users и внесем в нее парочку примитивных записей.

```
create database myDB;

use myDB;

create table users
(
userId integer not null primary key auto_increment,
userName varchar(50) not null,
userPass varchar(20) not null
)

insert into users values(NULL,'john', 'doe')
insert into users values(NULL,'admin', 'wwz04ff')
insert into users values(NULL,'fsmith', 'mypassword')
```

Итак, если я введу логин john и пароль doe, тогда будет выведена надпись «Logged In». Сам запрос будет выглядеть приблизительно так:

```
select * from users where userName='john' and userPass='doe'
```

В этом запросе нет ничего опасного... так ли это? Может, на первый взгляд и нет, но, что произойдет если я введу логин john и пароль ‘ or 1=1 --'?

Результирующий запрос теперь будет выглядеть так:

```
select * from users where userName='john' and userPass=''   or 1=1'
```

В этом примере я акцентировал внимание на логине и пароле таким образом, чтобы они легче воспринимались, но по сути происходит то, что теперь запрос проверяет все записи на равенство логина значению john. Вместо того, чтобы проверять на равенство пароля, он проверяет на условное равенство 1=1. Это означает, что если поле пароля пусто ИЛИ 1 равно 1 (что так и есть), то в БД была найдена верная запись.

Итак, с помощью login.php скрипта, который мы только что создали, одна запись будет возвращена, и текст «Logged In» будет выведен. Мы бы могли пойти дальше и сделать то же с полем логина таким образом:

```
Username: ' or 1=1
Password: [Empty]
```

что привело бы к исполнению следующего запроса к таблице users:

```
select count(*) from users where userName='' or 1=1 --' and
userPass=''
```

Примеры инъективных атак

Форсирование логина через HTML-форму в случае, который мы только что наблюдали – типичный пример инъективной SQL-атаки, и чуть позже мы рассмотрим, как можно это предотвратить.

Но сначала я хотел бы привести парочку примеров того, как подобные атаки проводятся. Продолжим рассмотрение нашей формы с логином и паролем.

Пример №1

У Microsoft SQL Server свой SQL диалект, который называется Transact SQL, или сокращенно TSQL. Мы можем по-разному воспользоваться средствами TSQL, чтобы показать, как работают инъективные SQL-атаки. Взгляните на следующий запрос, который основывается на таблице users, созданной нами на предыдущей странице:

```
select userName from users where userName='' having 1=1
```

Если вы разбираетесь в SQL, то, несомненно, увидите, в чем заключается ошибка. Без труда осуществляем этот запрос к нашей БД через страницу login.php с использованием следующих данных:

```
Username: ' having 1=1 ---
Password: [Anything]
```

Когда я, заполнив данные, произвожу запрос для осуществления авторизации, SQL запрос принуждает PHP обругать нас следующим сообщением об ошибке:

```
Microsoft OLE DB Provider for SQL Server (0x80040E14)
Столбец users.userName является недействительным в списке select,
ибо он не принадлежит к  собирательной функции, и в запросе нет
оператора GROUP BY.  /login.php, line 16
```

Получается, что это сообщение об ошибке говорит неавторизованному пользователю имя того столбца из БД, по которому мы пытались авторизовать логин: users.userName. Используя это имя, мы можем использовать оператор LIKE, чтобы залогиниться с такими данными:

```
Username: ' or users.userName like 'a%' ---
Password: [Anything]
```

Итак, еще разок: это инъектированный SQL-запрос к нашей таблице users.

```
select userName from users where userName='' or
users.userName like 'a%' --' and userPass=''
```

Когда мы создавали таблицу users, мы также создали пользователя, чье имя было admin, а пароль wwz04ff. Вход под этими логином и паролем осуществляется посредством SQL оператора like через сравнение логина. Запрос отыскивает в поле userName первой записи, в которой значение этого же поля начинается с a. В данном случае это admin:

```
Logged In As admin
```

Пример №2

SQL Server, наряду с другими БД, разграничивает запросы с разделителями. Суть разделителей в том, чтобы сгруппировать несколько запросов в один и выполнить последовательно. К примеру:

```
select 1; select 1+2; select 1+3;
```

...вернет три набора записей. Первый будет содержать значение 1, второй – значение 3 и третий – значение 4 и т.д. Итак, если мы авторизованы со следующими данными:

```
Username: ' or 1=1; drop table users; --
Password: [Anything]
```

то запрос будет осуществлен в двух частях. Во-первых, будет произведена выборка поля userName по всем записям из таблицы users. Во-вторых, будет удалена таблица users, так что когда мы в следующий раз попробуем залогиниться, то увидим следующее сообщение об ошибке:

```
Microsoft OLE DB Provider for SQL Server (0x80040E37)
Invalid object name 'users'.
/login.php, line 16
```

Пример №3

И последний пример, относящийся к нашей форме логина, в котором используется специфика TSQL-команд и расширенных хранимых процедур. Большое количество веб-сайтов использует default system account (sa) user (стандартная система пользовательских аккаунтов) для авторизации в SQL Server из своих PHP-скриптов или приложений. По умолчанию этот пользователь имеет доступ ко всем командам и может удалять, переименовывать и добавлять базы данных, таблицы и хранимые процедуры.

Одна из самых мощных команд SQL Server – SHUTDOWN WITH NOWAIT, которая выключает SQL Server, непосредственно останавливая сервис в Windows. После выполнения этой команды, для того, чтобы перезапустить SQL Server, вам понадобится менеджер сервиса SQL или что-то в этом духе.

Еще раз, эта команда может быть использована в нашем примере с авторизацией:

```
Username: '; shutdown with nowait; --
Password: [Anything]
```

Это заставит наш скрипт login.php выполнить следующий запрос:

```
select userName from users where userName='';
shutdown with nowait; --' and userPass=''
```

Если пользователь установлен как default (sa) (см. выше) или просто имеет соответствующие привилегии, то SQL Server выключится, и для дальнейшего функционирования потребует включения.

SQL Server имеет несколько хранимых процедур, которые по существу являются DLL-модулями, основанными на специфичном C++. Они могут содержать мощный C/C++ код для манипуляции сервером, чтения директорий и регистров, удаления файлов, запуска командной строки и т.д. Все расширенные хранимые процедуры существуют в составе главной БД и начинаются с «xp_».

Существуют несколько расширенных хранимых процедур, которые могут причинить непоправимый вред системе. Одну из таких процедур мы можем реализовать с использованием нашей формы авторизации, включив ее в строку логина:

```
Username: '; exec master..xp_xxx; --
Password: [Anything]
```

Все, что нам надо сделать, так это выбрать процедуру и заменить ее именем xp_xxx в вышеприведенном коде. К примеру, если на той же машине, что и SQL Server, установлен IIS (что характерно для одного-двух #########), то мы можем перезапустить его, использовав xp_cmdshell процедуру (которая выполняет команду в строке как системную), и сбросить. Все, что нам для этого нужно, так это следующие данные в нашем getlogin.php:

```
Username: '; exec master..xp_cmdshell 'iisreset'; --
Password: [Anything]
```

Это выполнит следующий запрос к SQL Server:

```
select userName from users where userName='';
exec master..xp_cmdshell 'iisreset'; --' and userPass=''
```

Не сомневаюсь, вы согласитесь, что таким образом можно причинить немало хлопот, а применение определенных команд может полностью вывести сервер из строя.

Пример №4

Ладно, пора отложить в сторону наш login.php и перейти к другим подобным методам проведения инъективных SQL-атак.

Сколько раз вам приходилось видеть на сайтах URL, подобные этому: www.mysite.com/products.php?productId=2?

Очевидно, 2 – это ID продукта, и множество подобных сайтов по отношению к productId строят запросы следующего типа:

```
Select prodName from products where id = 2
```

Прежде, чем продолжить, условимся, что на нашем SQL Server создана следующая таблица:

```
create table products
(
id int identity(1,1) not null,
prodName varchar(50) not null,
)

insert into products(prodName) values('Pink Hoola Hoop')
insert into products(prodName) values('Green Soccer Ball')
insert into products(prodName) values('Orange Rocking Chair')
```

Также создан скрипт PHP с названием products.php:

```
<?

$hostname = "localhost";
$dbName = "myDB";
$username = "sa";
$passdb = "";
$linkid=mssql_connect($hostname, $username, $passdb);
        mssql_select_db($dbName, $linkid);


$query = "select prodName from products where id = '$prodId'";
$result = mssql_query($query, $linkid);
$number = MSSQL_NUM_ROWS($result);

if ($number != 0)  {
$prodId = mssql_result($result,0,"prodName");
echo "Got product: $prodId";
}
else
echo "No product found";

?>
```

Итак, если мы выполним products.php в браузере по следующему URL: http://localhost/products.php?prodId=1 то увидим в нем такое сообщение: Got product: Pink Hoola Hoop.

Заметьте, что в данном случае products.php возвращает поле из записи, основываясь на имени записи:

```
Echo "Got product: $prodId";
```

Несмотря на то, что такой подход кажется более безопасным, на самом деле это не так, и мы все также можем манипулировать БД, как и в предыдущих примерах.

Обратите внимание, что в этот раз оператор WHERE основан на числовом значении:

```
$query = "select prodName from products where id = '$prodId'";
```

Для корректной работы products.php необходимо, чтобы ID продукта был воспринят, как переменная prodId. И это нам не составит особого труда: http://localhost/products.asp?prodId=0 or 1=1. Исходя из products.php запрос к БД будет выглядеть так:

```
select prodName from products where id = 0 or 1=1
```

Используя немного умения и знаний кодирования URL мы без труда выудим название поля продукта из таблицы: http://localhost/products.php?prodId=0%20having%201=1. За чем последует следующее сообщение об ошибке:

```
Microsoft OLE DB Provider for SQL Server (0x80040E14)

Column 'products.prodName' is invalid in the select
list because it is not contained in an aggregate
function and there is no GROUP BY clause.

/products.php, line 13
```

Теперь мы, зная имя поля продуктов (products.prodName) можем вызвать в браузере следующий URL:
http://localhost/products.php?prodId=0;insert%20into%20products(prodName)%20values(left(@@version,50)).

Вот запрос без запросов URL:
http://localhost/products.php?prodId=0;insert into products(prodName values(left(@@version,50)).

Фактически будет выведено сообщение «No products found», однако также будет произведен запрос INSERT к таблице products, с добавлением первых 50 символов переменной SQL Server @@version (которая содержит детальное описание SQL Server) в качестве новой записи.

В реальной ситуации вы, вероятно, используете эту таблицу для множества записей, но все это потребует аналогичного подхода.

Теперь вывод версии – дело всего одного вызова products.php таким образом: http://localhost/products.php?prodId=(select%20max(id)%20from%20products). Этот запрос выуживает ID последней записи из таблицы products, используя серверную функцию MAX. Получим:

```
Got product Microsoft SQL Server 2000 - 8.00.534 (Intel X86)
```

Такой метод инъективной атаки может осуществить множество целей. Однако, целью этой статьи было дать защиту от нападений, чем мы сейчас и займемся.

Предотвращение инъективных SQL-атак

Если вы разрабатываете свой скрипт внимательно и аккуратно, то без труда избежите инъективной атаки. Существует несколько вещей, с помощью которых мы как проектировщики понизим восприимчивость наших сайтов к атаке. Далее мы рассмотрим некоторые из них.

Ограничение доступа пользователя

Установленную по умолчанию учетную запись (sa) (см. выше) в SQL Server 2000 никогда не следует использовать. Вы должны устанавливать отдельные учетные записи для различных целей.

К примеру, если вы сопровождаете БД, которая позволяет пользователям вашего сайта просматривать и заказывать продукцию, следует настроить отдельного пользователя webUser_public, у которого есть права на SELECT для таблицы products и INSERT только для таблицы orders.

Если вы не пользуетесь расширенными хранимыми процедурами, или у вас есть не используемые триггеры, хранимые процедуры, пользовательские функции и т.д., то удалите их или поместите на изолированный сервер. Самые опасные инъективные атаки проводятся с использованием таких хранимых процедур, как xp_cmdshell и xp_grantlogin, так что их заблаговременное удаление теоретически предупреждает атаку.

Выходные кавычки

Как видно из предыдущих примеров, большинство инъективных атак требуют одиночных кавычек для завершения выражения. Простым использованием функции по конвертации всех одиночных кавычек в двойные вы лишаете подобные атаки шанса на успех.

С использованием PHP дело лишь в создании обычной функции по замене, которая позаботится об одинарных кавычках:

```
<?

function stripQuotes($strWords){
str_replace("'", "''", $strWords);
}

?>
```

В применении к нашему первому примеру функция stripQuotes изменит запрос:

```
select count(*) from users where userName='john' and
userPass='' or 1=1 --'
```

на

```
select count(*) from users where userName='john'' and
userPass=''' or 1=1 --'
```

В сущности атака прервана, так как теперь оператор WHERE требует, чтобы оба поля: userName и userPass – были действительными.

Удаление «преступных» символов/символьных последовательностей

Как было показано в статье, определенные символы и их последовательности, такие как ;, --, select, insert, xp_, могут быть использованы для проведения инъективных атак. Их удаление из пользовательского ввода перед созданием запроса может в дальнейшем уменьшив шанс на успех для атаки.

Как и в случае с одинарной кавычкой, нам понадобится всего одна функция, чтобы решить нашу задачу:

```
<?
function killChars($strWords) {

$badChars = array("select", "drop", ";", "--", "insert",
"delete", "xp_");

for ($i = 0; $i < 7; $i++){
$strWords  = str_replace ($badChars[$i], "", $strWords);
}
}

?>
```

Использование stripQuotes[1] в сочетании с killChars сполна лишает любую атаку шанса на победу. Итак, если у нас был запрос:

```
select prodName from products where id=1; xp_cmdshell 'format
c: /q /yes '; drop database myDB; --
```

, то после обработки stripQuotes и killChars он примет вид:

```
prodName from products where id=1 cmdshell ''format c:
/q /yes '' database myDB
```

..., который абсолютно бесполезен и не выдаст ни одной записи.

Лимитирование длины пользовательского ввода

Нет ничего хорошего в том, если поле ввода в форме рассчитано на 50 символов, а сравнивать вы его будете с полем, содержащим всего 10 символов. Ограничивая до минимума все поля ввода, вы не оставляете места для символов, которые могли бы составить инъективную SQL-атаку.

Если вы принимаете строку запроса с ID продукта, всегда используйте функцию, подобную IsNumeric() для PHP, для проверки типа значения: числовой ли он. Если тип таковым не является, выдайте сообщение об ошибке, либо перенаправьте пользователя на другую страницу, где он сможет выбрать продукт.

1 Стоит заметить, что функции экранирования есть почти в любом API для каждой СУБД, и прежде чем писать свою, рекомендуется изучить документацию (прим. ред.).

Также всегда старайтесь использовать формы с атрибутом метода POST, чтобы знающим людям не пришла идея в связи со знанием переменных, которые можно извлечь из URL.

Резюме

В этой статье мы рассмотрели, что такое инъективная SQL-атака, и как можно их проводить путем вмешательства в формы и URL.

Не всегда возможно предотвратить подобные атаки, однако, надеюсь, теперь вы имеете представления о некоторых из них, и уже разработали план, как с ними бороться на своем сервере.

Хоть мы и рассмотрели все примеры атак на примере Microsoft SQL Server, не забывайте, что среди уязвимых серверов находятся Oracle, MySQL и пр.

ООП и процедурное программирование в PHP

Автор:
Роберт Пик
Перевод:
Владимир Шапиро

В отличие от большинства других языков, PHP предоставляет возможность использовать при программировании как объектно-ориентированный, так и процедурный подходы. Большинство PHP-программистов по умолчанию используют последний, поскольку сами веб-страницы обрабатываются именно в процедурном подходе (один тег, затем другой, затем следующий и т.д.). Код, написанный в процедурном подходе, гораздо проще комбинируется с HTML, и, как следствие, программисты зачастую вырабатывают свой собственный стиль, основанный на таком подходе.

Предполагаемая аудитория

Эта статья рассчитана на начинающих PHP-программистов, которые желают получить представление об объектно-ориентированном и процедурном подходах написания программ.

Предполагается, что читатели владеют основными знаниями в области PHP и классов.

Вступление

«Настоящий гений проявляет себя в умении обрабатывать неточную и противоречивую информацию» (Уинстон Черчиль)

В отличие от большинства других языков, PHP предоставляет возможность использовать при программировании как объектно-ориентированный, так и процедурный подходы. Большинство PHP-программистов по умолчанию используют последний, поскольку сами веб-страницы обрабатываются именно в процедурном подходе (один тег, затем другой, затем следующий и т.д.). Код, написанный в процедурном подходе, гораздо проще комбинируется с HTML и, как следствие, программисты зачастую вырабатывают свой собственный стиль, основанный на таком подходе.

Если вы новичок в PHP, то, скорее всего, единственный способ программирования, которым вы пока пользовались – это процедурное. Тем не менее, если вы уже исследовали ресурсы, посвященные PHP, в поисках новых решений и приемов, вы наверняка сталкивались с сообщениями на тему «распухших объектов» (object bloat) или находили различные руководства по написанию объектно-ориентированного кода на PHP. Возможно, вы скачивали библиотеки, которые использовали классы, учились создавать объекты и вызывать их методы. При этом, вполне возможно, что вы никогда до конца не осознавали почему в том или ином случае имеет смысл применять объектно-ориентированный подход.

У обоих подходов есть свои преимущества и недостатки, заставляющие их приверженцев обмениваться незамысловатыми репликами вида: «Объекты – это плохо!» или «Объекты – это хорошо!». Данная статья не является попыткой склонить читателя в сторону того или иного подхода, а призваны исследовать преимущества и недостатки каждого.

Вот пример процедурного кода, выводящего строку:

```
<?php

print "Hello, world.";

?>
```

Вот пример объектно-ориентированного кода, который делает то же самое:

```
<?php

class helloWorld {
  function myPrint() {
    print "Hello, world.";
  }
}
$myHelloWorld = new helloWorld();
$myHelloWorld->myPrint();

?>
```

Сразу отметим, что эта статья не является попыткой дать определение объектно-ориентированному программированию как программированию с использованием классов. Для получения начальных сведений о принципах ООП рекомендуется воспользоваться Google. На данный момент существует достаточное количество хороших (и немного плохих) статей, посвященных этой теме.

Кто пишет подобный код?

Для того, чтобы понять, каким образом рассматриваемая тема стала причиной многочисленных «религиозных войн», давайте изучим пару экстремальных примеров с обоих сторон. Вначале мы взглянем на «Процедурного Фанатика», а затем на «Объектного Фанатика». Возможно, кто-то из этих двоих покажется знакомым и самому читателю.

«Процедурный Фанатик»

«Процедурный Фанатик» всегда подвергается критике со стороны своего учителя по информатики за плохое использование абстрактного подхода при выполнении своих заданий. Приводимый в оправдание аргумент: «Но ведь оно работает!», – так и не улучшает его отметок. В дальнейшем такой программист избирает в качестве предметной области такие задачи, как написание драйверов, файловых систем и других низкоуровневых программ, где его акцент на скорости и лаконичности приносит ему признание.

Экстремальными примерами «Процедурного Фанатика» являются полное отрицание объектов и критика абстракции как таковой. Такие программисты всегда хотят сделать свой код быстрым, не заботясь особенно о тех, кто будет его читать позднее.

Очень часто они рассматривают процесс программирования как соревнование, а не командную разработку. Они любят участвовать в заковыристых конкурсах по написанию кода. Их любимыми языками, наравне с PHP, являются C и Assembler. В PHP они могут заниматься написанием модулей PECL, способствуя тому, чтобы код был рациональным и эффективным.

«Объектный фанатик»

Объектный фанатик совсем недавно открыл для себя ООП и теперь готов применять его везде! Он не до конца осознает проблему производительности при используемом им подходе. Иногда кажется, что концепции абстрактного дизайна для него гораздо важнее самого кода. Возможно, такой программист имеет хорошие шансы на карьеру менеджера проектов или технического писателя.

Объектные фанатики очень любят упрекать процедурных в том, что без абстракций мир до сих пор программировал бы с помощью нулей и единиц. Они известны также тем, что говорят на псевдокоде. Экстремальным примером таких программистов являются люди, использующие объекты, забывая о производительности и не щадящие элегантности и даже читаемости собственного кода. Их любимыми языками программирования, помимо PHP, являются Java и Smaltalk. В PHP-сообществе они, как правило, занимаются разработками модулей PEAR, создавая хорошо документированный и легко сопровождаемый код.

Эти двое никогда не встретятся

Почему страницы форумов забиты предвзятыми высказываниями? Дело в том, что уникальный опыт, накапливаемый вами в процессе работы над проектом и, как следствие, выработанная им «ваша философия» предопределяют путь поиска новых идей. Как программисты мы должны постоянно остерегаться подобных предубеждений, оставаясь открытыми для изучения новых вещей. Как информатики (computer scientists) мы должны уметь отбрасывать эти предубеждения, чтобы находить лучшие методы решения проблемы в конкретной ситуации.

Каковы ваши цели?

Задумайтесь на мгновенье над тем, какие цели стоят перед вами при написании той или иной PHP программы. Часто эти цели выражаются очень туманно, иногда указываются в задании. Вы можете придерживаться этих целей в любом из своих проектов. Одна из них, за которую особо ратует автор этой статьи – элегантность. Определение элегантности не будет дано здесь, так как эта тема, скорее подходит для отдельной статьи. Есть и другие более абстрактные цели, которые не всегда сопоставимы с теми решениями, которые хороши для проекта – для конкретных задач, стоящих перед вами. Фактически такие цели могут быть в данном случае теми самыми предубеждениями, о которых говорилось ранее.

Абстрактные цели

- Предложить решение, содержащее как можно меньше строчек кода;
- Думать о проблеме на уровне проблемы.

Звучит великолепно, не правда ли? Но что понимается под выражением «как можно меньше строчек кода»? Учитываются ли комментарии? Должны ли мы соединять весь код в одну строчку, разбивая ее точками с запятой? А как насчет фигурных скобок? Некоторые любят оставлять для них отдельную строку. Судя по поставленной цели, все это будет прямыми нарушениями задания.

Как насчет «решения на уровне проблемы»? Означает ли это, что для каждой концепции, используемой в решении, мы должны создавать класс? Или нам необходимо размещать каждую подзадачу в отдельный файл и создавать сложное дерево, отображающее реальную структуру? Вот оно – файл и класс на каждую мысль!

Очевидно, что такие обобщения, возведенные в принцип, становятся нелепыми. Но есть и более тонкие проявления той же самой проблемы. Как часто программисты, работающие в команде, изобретают решение в одну строчку и вставляют его в код без всякого комментария, только лишь для того, чтобы поставить в тупик своих товарищей по команде, которые будут затем этот код сопровождать?

И наоборот, как часто погрязший в бюрократизме главный программист впадает в транс, создавая интерфейсы и классы, которые не только осложняют жизнь его подопечным, реализующим код, но и ограничивают производительность и гибкость самой программы в случае, когда конечному заказчику потребуется дополнительная функциональность. Все это – всего лишь незаметные проявления тех же самых предубеждений, которые подсознательно работают в головах наших «слишком человечных» программистов.

Практические цели

Решением проблемы является такой подход, при котором в первую очередь рассматриваются практические цели, стоящие перед проектом. Каковы же цели конкретного проекта? Вот несколько вариантов:

- Писать это быстро, выпускать (release) часто;
- Заставить это работать настолько быстро, насколько возможно;
- Сделать это простым в сопровождении, распространении и расширении;
- Опубликовать программный интерфейс (API).

Первые две цели больше тяготеют к процедурному подходу, тогда как последние две – к объектно-ориентированному.

Какой подход наиболее правильный?

Теперь давайте попробуем оценить практические преимущества каждого метода.

Процедурный подход

Основным аргументом в пользу процедурного подхода является тот факт, что PHP является интерпретатором. Это означает, что, в отличие от многих других языков, он не компилируется в бинарный код для исполнения, а интерпретируется и исполняется «на лету». Это язык сценариев и каждый сценарий хранится как обычный текст (естественно, за исключением инструментов компилирования Zend).

Еще одним аргументом против использования объектно-ориентированного подхода в PHP4 является тот факт, что подобная функциональность не была полностью продумана в этой и более ранних версиях. Если послушать Расмуса (Расмус Лердорф – автор первых версий PHP), то получается, что поддержка ООП была добавлена уже после окончания основных работ над ядром языка. Из этого следует, что использование объектов в PHP4 не настолько рационально и эффективно, как могло бы быть. Как бы то ни было, эта ситуация изменилась с выходом новой версии PHP5.

С использованием процедурного подхода написаны такие популярные приложения как osCommerce и phpMyAdmin. Они появились достаточно быстро и обладают относительно высокой скоростью работы. Оба проекта сильно привязаны к HTML, который без ограничений комбинируется с PHP.

osCommerce

osCommerce содержит известное количество объектов, но большинство работы осуществляется в процедурном стиле. Автору статьи приходилось не раз править исходный код, добавляя нужную клиентам функциональность (порой настолько нужную, что удивителен сам факт того, что она не была предусмотрена в самом начале). Как бы то ни было, необходимо отметить, что процесс добавления изменений не был тривиален. Большинство процедурного кода требовало изрядное количество времени на его разбор и понимание, поскольку сам код был перемешан с системой шаблонов, а также был спроектирован с использованием различных языковых нотаций.

Теперь это работает и работает быстро на множестве сайтов, занимающихся электронной коммерцией. Также стоит отметить, что у сообщества osCommerce есть форум и некое подобие структуры для разработки новых модулей и расширений. Как следствие, для системы можно найти очень много полезных «плагинов».

phpMyAdmin

phpMyAdmin напрямую использует только один класс: Mimer SQL Validator, который в случае phpMyAdmin зависит от PEAR пакетов Mail_Mime, Net_DIME и SOAP.

Такое решение скорее всего было основано на убеждении: «оно хорошо служит своей цели – давайте его используем». Во всем остальном, похоже, что код приложения написан в процедурном стиле, и вновь мы имеем тесное связывание HTML и PHP.

phpMyAdmin – это инструмент, который используется ежедневно. Более того, для простых модификаций автор статьи поощряет своих клиентов в использовании phpMyAdmin как системы администрирования контента (естественно, при соответственном ограничении их прав). phpMyAdmin работает очень хорошо и, учитывая то, что ему приходится делать, он работает очень быстро. Автор часто расширяет phpMyAdmin для некоторых приложений, делая его системой администрирования, усиливая такие его свойства, как избранные запросы, для того чтобы клиенты могли быстро и просто создавать выборки и редактировать их. Сам же код должен быть достаточно хорошо сопровождаемым, поскольку с каждым выходом новой версии рассматриваемый продукт развивается все в более и более практичный инструмент.

Процедурный подход – заключение

Процедурные части обоих приложений хорошо документированы. В случае osCommerce предоставление определенного каркаса для разработки расширений также способствует поддержке и расширению этого продукта. Однако не существует никакого подобия программного интерфейса (API), позволяющего расширить продукт до чего-либо серьезно отличающегося от оригинала.

Потребуется очень много усилий, чтобы превратить osCommerce в бухгалтерское приложение, и еще больше, чтобы превратить phpMyAdmin в настраиваемую систему администрирования контента. Тем не менее, ту работу, для которой изначально были созданы эти оба продукта, они выполняют очень хорошо.

Объектный подход

Одним из основных аргументов в пользу объектно-ориентированного подхода является расширяемость и модульность полученного кода. Сам по себе факт программирования в объектно-ориентированом стиле не сможет закомментировать код или создать документацию, но он может подвигнуть вас к этому действию. И однажды стремление к расширяемости создаваемого вами кода может привести вас к написанию настоящего программного интерфейса (API) для вашего приложения.

PHP5 обещает сделать процесс объектно-ориентированного программирования гораздо более приятным. С известной долей иронии можно сравнивать новую версию с выходом «Java 2», поскольку она включает в себя ряд таких свойств, как интерфейсы, ссылочная модель объектов и структуры try/catch, уже зарекомендовавших себя в мире Java. Но даже для PHP4 и более ранних версий можно найти примеры удачного применения объектов.

Одним из самых удачных автор статьи считает Smarty – систему шаблонов, позволяющих отделять код PHP от контента. Другим полезным примером является свободно распространяемый генератор PDF, полностью написанный на PHP и получивший название FPDF. Оба программных продукта практикуют подход при котором HTML код отделяется от PHP. Особенно важно отметить тот факт, что обе программы могут быть расширены до других полезных приложений.

Smarty

Smarty является инструментом для создания сложных форм и сайтов, основанных на шаблонах. В большинстве случаев, когда нужно построить прототип системы с полностью настраиваемым внешним видом, – Smarty самый подходящий инструмент. Внешний вид сайта может быть полностью изменен, не затрагивая функциональности, которая стоит за этим. Чтобы сделать процесс смены оформления более простым для дизайнеров, автор создал индивидуальную систему тегов, как расширение системы тегов Smarty. Так, например, при вводе в начале страницы директивы:

```
[navigation horizontal separated by " | "]
```

на сайте появится горизонтальный блок навигации, разделенной между собой символами " ". Поскольку Smarty уже обладает очень развитым механизмом отображения данных, содержащихся в переменных, описанный процесс подразумевает всего лишь отображение сложных тегов Smarty в простые директивы для оформления. Более подробно об этом можно прочесть на http://simpletags.sourceforge.net/ и http://simplequiz.sourceforge.net/.

Поскольку Smarty представляет собой класс и все его методы хорошо документированы, то его функциональность можно легко расширять. Кроме того, Smarty предоставляет уровень отделения области видимости прочих переменных, используемых в PHP, тем, что заставляет передавать шаблоном только те переменные, которые вы собираетесь использовать.

Такого рода «экологический надзор за переменными» позволяет создать надежную рабочую связь между дизайнерами, создающими шаблоны, и программистами, которые предоставляют для них данные.

FPDF

FPDF – отличный инструмент. Если вы разочаровались в постоянно изменяющемся API к pdflib и не хотите платить за хорошо поддерживаемое приложение, если вы не можете управлять расширениями на той версии PHP, с которой работаете (как это бывает в случае хостинга с общими ресурсами), задумайтесь над использованием этого свободно распространяемого, полностью написанного на PHP PDF-генератора.[1].

Этот класс хорошо документирован, включая массу приятных примеров того как создавать и располагать в документе текст и графику. Для уже упомянутого сайта онлайнового обучения автор использовал FPDF для динамического создания сертификатов окончания курсов с использованием шрифтов семейтсва True Type и графики с разрешением 300 dpi. Те дополнительные секунды, требующиеся для вызова класса и исполнения операций с PDF практически не заметны на фоне того, что сам PDF-файл иногда требует нескольких минут для скачивания. Ввиду этого, на низкоскоростных соединениях дополнительное время, затраченное на создание и отправку PDF-файла порой оказывается меньше, чем скачивание статического аналога.

Кроме того, поскольку FPDF основана на объектно-ориентированном подходе, она может быть расширена. Многие методы до конца не реализованы и просто существуют в качестве структуры, которая может быть расширена вами, например, для создания собственных.

Объектный подход – заключение

Как Smarty, так и FPDF предоставляют хорошо документированный интерфейс для расширения их основного класса. Заключение всех методов и данных в класс в данном случае является хорошо продуманным решением. Подобной функциональности можно было бы достичь, используя обычные функции и глобальные переменные, но ее трудно было бы расширить впоследствии. Кроме того, невозможно было бы создавать несколько сущностей с одинаковой функциональностью, как это можно делать с объектами. Такие объекты могут задавать стиль PDF- или HTML-документа, который вы создаете. Таким образом вы можете публиковать одни и те же данные в различных форматах.

Описанные выше приложения являются отличным примером использования объектно-ориентированного подхода для создания полезной библиотеки.

Когда и что применять?

Мы рассмотрели примеры применения робких подходов для разных целей и познакомились с эффективными примерами применения этих подходов в реальных приложениях. Что же делать, если ваши цели требуют обоих подходов?

Общие проверки производительности объектного и процедурного кода дадут вам очень отдаленное представление о том, что можно ожидать от конкретного приложения, создаваемого вами. Вместо подобных проверок выбирайте тот способ оценки, который больше всего вам подходит для оценки действительной конкретного кода на том или ином участке. Проверяйте свои функции на возможность использования ими общих ресурсов. Продумайте необходимость расширения таких функций. Подумайте, насколько написание класса поможет вам сделать ваш код расширяемым.

Кроме всего прочего, распределите ваши цели по степени приоритета. Взвесьте их. Если вы будете использовать практический подход от проекта к проекту для определения эффективности объектно-ориентированного или процедурного кода, вам гарантировано непредвзятое и подкрепленное вашим опытом решение. И помните: используя PHP, у вас всегда есть шанс. Вы всегда сможете комбинировать оба подхода.

Почему оба подхода необходимы?

Возвращаясь к фанатам объектного и процедурного подходов, мы не забудем отметить и их заслуги.

- Поблагодарим «объектных» программистов за практичность и расширяемость Smarty и FPDF
- Поблагодарим «процедурных» программистов за быстроту и надежность функционирующих osCommerce и phpMyAdmin.

На самом деле эти признания идут гораздо дальше – к основам PHP.

Как PECL, так и PEAR уже снискали себе достаточно славы и критики. Оба проекта могут служить хорошей иллюстрацией различий между процедурным и объектно-ориентированным программированием.

PECL предоставляет расширения для PHP, написанные на C, с использованием процедурного подхода и ориентированные на скорость и лаконичность кода. Зачастую это перенос существующих приложений с лицензией LGPL (Lesser General Public License). Таким способом за несколько лет в PHP была добавлена масса интересных функций. Кроме того, не стоит забывать, что и сам PHP написан на C.

PEAR способствовал появлению массы полезных и интересных классов, начиная от создания Excel-таблиц и заканчивая изменением DNS-записей. Используя библиотеки PEAR, вы экономите себе массу времени и можете создавать функциональность, которая способна заставит вас воскликнуть: «Я даже не знал, что PHP может такое!».

Заключение

Автор надеется, что эта статья дала вам возможность по достоинству оценить полезность каждого из подходов программирования и, что более важно, снабдила вас информацией и желанием для более глубокого изучения этой проблемы. Думайте о своих целях, тестируйте реальные примеры и относитесь к радикальным суждениям с известной долей скептицизма. Оба подхода имеют свои преимущества. Поэтому идите и напишите пару строчек хорошего кода!

Статья предоставлена сайтом: http://detail.phpclub.ru

ext/mysqli: Обзор и подготовленные выражения

Авторы:
Зак Грэнт
Георг Рихтер
Перевод:
Илья Гофман [gufy]

Начиная с середины 90х, ext/mysql служило основным мостом между PHP и MySQL. Хотя в нем имелись недостатки и проблемы росли с годами, в общем ext/mysql делал свое дело неплохо и шел в ногу с изменениями как в PHP, так и в MySQL. Однако с появлением PHP 5 и MySQL 4.1 все изменилось – начали образовываться несколько достаточно обширных трещин.

Предполагаемая аудитория

Статья предназначена для читателей, имеющих некоторый опыт использования PHP и MySQL. Она предполагает, что читатель понимает основные принципы работы с базами данных и программирования и может использовать сценарий PHP для отправки запроса серверу MySQL.

Обратите внимание на то, что в конце статьи имеются сноски для разъяснения некоторых утверждений и словарь терминов.

Инструкции по установке PHP и MySQL выходят за рамки данной статьи;

- Для получения информации об установке PHP 5 посетите http://www.php.net/installation;
- Информация о компиляции PHP 5 с поддержкой ext/mysqli доступна по адресу http://www.php.net/mysqli;
- За информацией по установке MySQL 4.1.2 или выше обращайтесь на http://www.mysql.com/doc/en/Installing.html.

Введение

Начиная с середины 90х, ext/mysql служило основным мостом между PHP и MySQL. Хотя в нем имелись недостатки и проблемы росли с годами, в общем, ext/mysql делал свое дело неплохо и шел в ногу с изменениями как в PHP, так и в MySQL.

Однако с появлением PHP 5 и MySQL 4.1 все изменилось – начали образовываться несколько достаточно обширных трещин.

В ext/mysql имелись «достоинства, оказавшиеся недостатками»: в первую очередь это mysql_pconnect()[1] , подключение по умолчанию и автоматическое подключение[2]. Кроме того, проявились несовместимости между функциями ext/mysql и теми, что поддерживались клиентской библиотекой MySQL, на которой основаны и ext/mysql, и ext/mysqli.

В попытке исправить эти расхождения Георг Рихтер создал очередное расширение PHP 5, которое поддерживает новые возможности MySQL 4.1+. Это расширение получило название ext/mysqli, где 'i' заменяет одно из слов: improved(улучшенное), interface(интерфейс), ingenious(изобретательное), incompatible(несовместимое) или incomplete(неполное).[3]

Основные цели

Некоторыми из основных целей создания нового расширения были:

- Простота использования. Код ext/mysql стал очень сложным и беспорядочным. Значительная модернизация функциональности MySQL потребовала возможности подключения и отключения тех или иных частей в зависимости от версии клиентской библиотеки. Другие проблемы требовали изменения функциональности в зависимости от операционной системы.
- Лучшая совместимость. Расширение должно было более аккуратно использовать клиентскую библиотеку MySQL, чтобы будущие усовершенствования библиотеки проще поддерживались в PHP.
- Обратная совместимость. Хотя совместимость между ext/mysql и ext/mysqli не идеальна, были приложены значительные усилия для облегчения портирования приложений с ext/mysql на ext/mysqli.

Основные возможности

Ext/mysqli поддерживает новые возможности, появившиеся в последних версиях MySQL, и предлагает новые функции.

Основные возможности расширения:

- Процедурный интерфейс, очень похожий на интерфейс ext/mysql.
- Объектно-ориентированный интерфейс, который позволяет использовать стиль, более простой и расширяемый, нежели процедурный интерфейс.
- Поддержка нового бинарного протокола MySQL, введенного в версии 4.1. (новый протокол более эффективен, чем старый, и поддерживает более широкий набор возможностей, например, подготовленные выражения).
- Поддержка полного набора возможностей клиентской библиотеки MySQL C, в том числе установки сложных параметров соединения с помощью mysqli_init() и других функций. Кроме того, расширение имеет поддержку дополнительных функций мониторинга, отлова ошибок, управления загрузкой и репликации.

Зачем переходить?

Кроме получения доступа к дополнительному функционалу MySQL 4.1+, зачем же стоит переходить на использование ext/mysqli?

В дополнение к упомянутому функционалу, ext/mysqli имеет несколько существенных преимуществ:

- Заметно большая скорость. Усовершенствования, как в расширении, так и в MySQL, ускорили большинство операций, иногда достигая 40-кратного увеличения производительности по сравнению с ext/mysql.

- Усиленная безопасность. В ранних версиях MySQL RDBMS (см. Словарь терминов в конце статьи – прим. переводчика), существовала возможность отловить хэш слабого пароля в сети и затем воссоздать пароль пользователя. Новая процедура аутентификации гораздо прочнее и повторяет устойчивые к атакам механизмы авторизации таких инструментов как SSH.

Предупреждения и неожиданности

Некоторые аспекты ext/mysqli сильно отличаются от старого расширения. С целью исправления определенных изъянов в дизайне и поведения, склонного к ошибкам, некоторые возможности были убраны:

- Подключение к базе данных по умолчанию. Если вы явно не подключитесь к ней, ext/mysqli не сделает этого за вас.
- Соединение по умолчанию(link). Необходимо явно обращаться к соединению с сервером базы данных, которое вы хотите использовать, если вы работаете с ext/mysqli через процедурный интерфейс, например, mysqli_query($link, $query);

Покажите мне код!

Теперь, когда вы знаете, что изменилось, мы начнем анализировать код, который демонстрирует, как выглядит и работает новое расширение. Весь самостоятельный код, приведенный в этой статье, использует базу данных «world», которая бесплатно доступна на сайте http://www.mysql.com/documentation/index.html.

Базовое использование

Вот простой скрипт, который соединяется с сервером MySQL, посылает запрос серверу с помощью этого соединения, выводит результаты запроса и затем освобождает результирующее множество запроса и закрывает соединение.

```
<?php

/* Подключение к серверу MySQL */
$link = mysqli_connect(
            'localhost',  /* Хост, к которому мы подключаемся */
            'user',       /* Имя пользователя */
            'password',   /* Используемый пароль */
            'world');     /* База данных для запросов по умолчанию */

if (!$link) {
      printf("Невозможно подключиться к базе данных. Код ошибки: %s\n",
mysqli_connect_error());
   exit;
}

/* Посылаем запрос серверу */
if ($result = mysqli_query($link, 'SELECT Name, Population FROM City ORDER BY
Population DESC LIMIT 5')) {
```

Листинг 1 (начало)

```
    print("Очень крупные города:\n");

    /* Выборка результатов запроса */
    while( $row = mysqli_fetch_assoc($result) ){
        printf("%s (%s)\n", $row['Name'], $row['Population']);
    }

    /* Освобождаем используемую память */
    mysqli_free_result($result);
}

/* Закрываем соединение */
mysqli_close($link);
?>
```

Листинг 1 (окончание)

Приведенный сценарий должен вывести что-то вроде:

```
Очень крупные города:

Mumbai (Bombay) (10500000)
Seoul (9981619)
Sao Paulo (9968485)
Shanghai (9696300)
Jakarta (9604900)
```

Как видно из кода, ext/mysqli и ext/mysql могут быть очень похожи. Единственным существенным различием является то, что процедурный стиль ext/mysqli несколько более «многословен».

Заметьте, что без проверки на ошибки приведенный скрипт мог бы дать сбой в любом месте и вывести пользователю мерзкое сообщение об ошибке.

Использование объектно-ориентированного интерфейса

Объектно-ориентированный интерфейс предоставляет немного более лаконичный и менее восприимчивый к ошибкам метод использования ext/mysqli. Код, приведенный ниже, производит те же действия, что и предыдущий, однако, имеются несколько ключевых отличий, на которые стоит обратить внимание:

- Нам не нужно явно задавать соединение, используемое в наших командах. Информация о подключении содержится в наших объектах $mysqli и $result и доступна при вызове соответствующих методов.
- Когда производится выборка из результирующего набора данных запроса с использованием fetch_assoc(), не нужно явно задавать идентификатор используемого результирующего набора. Также, как и информация о подключении, он содержится в объекте $result.

```php
<?php

/* Подключение к серверу MySQL */
$mysqli = new mysqli('localhost', 'user', 'password', 'world');

if (mysqli_connect_errno()) {
   printf("Подключение к серверу MySQL невозможно. Код ошибки: %s\n",
mysqli_connect_error());
   exit;
}

/* Посылаем запрос серверу */
if ($result = $mysqli->query('SELECT Name, Population FROM City ORDER BY
Population DESC LIMIT 5')) {

    print("Очень крупные города:\n");

    /* Выбираем результаты запроса: */
    while( $row = $result->fetch_assoc() ){
        printf("%s (%s)\n", $row['Name'], $row['Population']);
    }

    /* Освобождаем память */
    $result->close();
}

/* Закрываем соединение */
$mysqli->close();
?>
```

Листинг 2.

Подготовленные выражения

Теперь, когда мы разобрали азы использования расширения, рассмотрим несколько новых возможностей.

Подготовленные выражения предоставляют разработчикам возможность создавать запросы, которые являются более безопасными, имеют более высокую производительность и более удобны в написании.

Подготовленные выражения можно использовать двумя способами: с заданными параметрами и с заданными результатами.

С заданными параметрами

Подготовленные выражения с заданными параметрами позволяют создавать шаблоны запросов и хранить их на сервере MySQL. Когда нужно создать запрос, данные, заполняющие шаблон, отправляются серверу MySQL, и полностью сформированный запрос выполняется.

Основной процесс создания и использования подготовленных выражений с заданными параметрами прост.

Создается шаблон запроса и посылается серверу MySQL. Сервер его получает, проверяет его корректность, чтобы убедиться, что он имеет смысл, и сохраняет его в специальном буфере. Затем сервер возвращает идентификатор, который может быть в дальнейшем использован для обращения к подготовленному выражению.

Когда нужно создать запрос, данные, заполняющие шаблон, отправляются серверу MySQL и полностью сформированный запрос выполняется.

В этом процессе заключено несколько очень важных деталей.

Тело шаблона отсылается серверу MySQL только один раз. Для выполнения выражения посылаются только данные, необходимые для заполнения шаблона.

Большая часть работы по проверке и обработке запроса проделывается только один раз, вместо того, чтобы делать это каждый раз.

Кроме того, для запросов, которые содержат небольшое количество данных, расходы сильно уменьшены. Например, если у вас есть запрос типа:

```
INSERT INTO City (ID, Name) VALUES (NULL, 'Calgary');
```

то каждый раз при выполнении запроса нужно отослать лишь около 16 байт вместо обычных 60 или более байт. (Эти приближенные числа включают расходы на все данные вроде идентификатора подготовленного выражения, длины данных запроса – для безопасности бинарных данных – и т.д., но не включают расходы на строку запроса.)

Данные запроса не должны проходить через функции вроде mysql_real_escape_string(), чтобы убедиться, что нет угрозы атаки «SQL-инъекции»[4]. Вместо этого клиент и сервер MySQL работают так, чтобы убедиться, что посланные данные безопасно обработаны при их комбинировании с подготовленным выражением.

Шаблон запроса выглядит как-то так:

```
INSERT INTO City (ID, Name) VALUES (?, ?);
```

Знак '?' можно использовать в большинстве мест, где используются символьные данные, например, запрос может быть переделан из

```
SELECT Name FROM City WHERE Name = 'Calgary';
```

в

```
SELECT Name FROM City WHERE name = ?;
```

В листинге 3 приведен более полный пример, демонстрирующий весь процесс.

```php
<?php
$mysqli = new mysqli('localhost', 'user', 'password', 'world');

/* Проверка соединения */
if (mysqli_connect_errno()) {
    printf("Подключение невозможно: %s\n", mysqli_connect_error());
    exit();
}

$stmt = $mysqli->prepare("INSERT INTO CountryLanguage VALUES (?, ?, ?, ?)");
$stmt->bind_param('sssd', $code, $language, $official, $percent);

$code = 'DEU';
$language = 'Bavarian';
$official = "F";
$percent = 11.2;

/* выполнение подготовленного выражения  */
$stmt->execute();

printf("%d Row inserted.\n", $stmt->affected_rows);

/* Закрытие соединения и выражения*/
$stmt->close();

/* Очистить таблицу CountryLanguage */
$mysqli->query("DELETE FROM CountryLanguage WHERE Language='Bavarian'");
printf("%d Row deleted.\n", $mysqli->affected_rows);

/* Закрыть подключение */
$mysqli->close();
?>
```

Обратите внимание на то, что первым параметром bind_param () является короткая строка. Это строка формата, используемая для определения того, как объявленные параметры должны быть интерпретированы.

В случае вышеприведенного сценария 'sssd' означает, что значения первых трех параметров: $code, $language и $official – будут посланы как строки, а четвертый параметр $percent будет содержать значения типа double с плавающей запятой.

Для каждой заявленной переменной в bind_param(), должна быть своя буква в строке формата, которая означает, как переменная будет отправлена. Например:

```php
$stmt->bind_param('s', $foo);
$stmt->bind_param('si', $foo, $bar);
$stmt->bind_param('sid', $foo, $bar, $baz);
```

Объявление типов обеспечивает то, что расширение mysqli знает, как зашифровать данные для большей эффективности.

Определения типов очень просты: данные в заданных переменных будут обрабатываться как целочисленные, рациональные числа (double) или как строки.

Также имеется специальный тип, позволяющий отправлять блобы (большие бинарные объекты) порциями.

Следующая таблица иллюстрирует типы и возможности использования:

Идентификатор типа	Тип столбца
i	Все INT типы
d	DOUBLE и FLOAT
b	BLOB'ы
s	Остальные типы

С заданными результатами

Подготовленные выражения с объявленными результатами позволяют привязывать переменные PHP-скрипта к значениям полей данных в результирующем множестве запроса.

Процесс объявления таков:

- Создать запрос;
- Попросить сервер MySQL заготовить запрос;
- Привязать переменные PHP к столбцам в заготовке запроса.
- Заставить сервер MySQL выполнить запрос;
- Запросить добавление нового ряда данных в привязанные переменные.

Вот простой фрагмент кода, иллюстрирующий процесс:

```
<?php

$mysqli = new mysqli("localhost", "user", "password", "world");
if (mysqli_connect_errno()) {
    printf("Подключение невозможно: %s\n", mysqli_connect_error());
    exit();
}

/* Подготовленное выражение */
if ($stmt = $mysqli->prepare("SELECT Code, Name FROM Country ORDER BY Name
LIMIT 5")) {
    $stmt->execute();
    /* Привязывание переменных к заготовке */
    $stmt->bind_result($col1, $col2);

    /* Выборка значений */
    while ($stmt->fetch()) {
        printf("%s %s\n", $col1, $col2);
    }
    /* Закрытие оператора $stmt->close();
}
/* Закрытие соединения */
$mysqli->close();

?>
```

Использование заданных параметров и результатов вместе

Вот более полный пример, демонстрирующий использование и заданных параметров, и заданных результатов одновременно:

```
<?php
$mysqli = new mysqli("localhost", "user", "password", "world");

if (mysqli_connect_errno()) {
    printf("Подключение невозможно: %s\n", mysqli_connect_error());
    exit();
}

/* Подготовленное выражение */
if ($stmt = $mysqli->prepare("SELECT Code, Name FROM Country WHERE Code
LIKE ? LIMIT 5")) {

    $stmt->bind_param("s", $code);
    $code = "C%";

    $stmt->execute();

    /* Объявление переменных для заготовленного выражения*/
    $stmt->bind_result($col1, $col2);

    /* Выборка значений */
    while ($stmt->fetch()) {
        printf("%s %s\n", $col1, $col2);
    }

    /* Закрытие выражения */
    $stmt->close();
}
/* Закрытие подключение */
$mysqli->close();

?>
```

Резюме

В этой статье мы привели обзор возможностей и архитектуры ext/mysqli, а также краткое изложение истории его развития. К этому моменту вы должны понимать, как использовать и получать выгоду от подготовленных выражений MySQL, и должны чувствовать удобство использования объектно-ориентированного интерфейса к ext/mysqli.

Словарь терминов

ext/mysql – старое расширение PHP для работы с MySQL. Не поддерживает всех возможностей MySQL версий 4.1 и выше.

ext/mysqli – новое расширение PHP 5 для работы с MySQL. Поддерживает возможности MySQL версий от 3.22 до 5.0.

Клиентская библиотека MySQL – Компонент MySQL RDBMS (MySQL Relational DataBase Management System – Система управления реляционной базой данных MySQL – прим. переводчика), который позволяет программам общаться с RDBMS.

Сервер MySQL – Компонент MySQL RDBMS, который обрабатывает и отвечает на запросы, управляет файловым представлением данных внутри базы и т.д.

[1] – Функция mysql_pconnect() была создана для предоставления механизма уменьшения затрат на установление и разрыв соединений с сервером MySQL. К сожалению, из-за взаимодействия между архитектурами сервера Apache и PHP, большой трафик на сайте, использующем pconnect, мог быстро загрязнить сервер MySQL большим количеством неиспользуемых соединений, которые мешали активным соединениям получать доступ к базе данных.

[2] – Возможности автоматического соединения позволяли определенным вызовам функций автоматически соединяться с базой данных (если правильная информация о соединении находилась в конфигурационном файле php.ini). Возможность соединения по умолчанию работала по следующему принципу. Последнее открытое соединение с базой MySQL становится используемым соединением, если параметр соединения не был явно указан в аргументах функции.

[3] – Это расширение все еще находится в стадии разработки. В то время как набор возможностей ядра должен быть действительно стабильным ни MySQL 4.1, ни PHP 5.0 не имеют стабильных релизов (статья появилась до выхода PHP 5.0.0 – прим. переводчика). Также дополнительный набор возможностей, который не очень аккуратно использует клиентскую библиотеку MySQL, все еще дорабатывается.

[4] – Атаки типа «SQL-инъекции» возможны, когда данные входят в запрос, заставляя его совершать неожиданные и/или злонамеренные действия. Пусть, для примера, дан простой запрос в PHP скрипте типа «DELETE FROM grades WHERE class_name='test_$class'». Атакующий может получить контроль над переменной $class и получить возможность an attacker who can gain control over the value of $class can force unintended deletes to occur by changing the value of $class to something like «oops' or class_name LIKE '%'».

Об авторах

Зак Грэнт (Zak Greant) – профессиональный защитник концепции Open Source, писатель и программист. Он работает в MySQL AB пропагандистом Сообщества.

Георг Рихтер (Georg Richter) – создатель расширения mysqli. Он также поддерживает расширения mysql и ncurses.

Оригинал статьи находится по адресу: http://www.zend.com/php5/articles/php5-mysqli.php
Перевод предоставлен сайтом: http://detail.phpclub.ru

Люди

Интервью: компания «Аист», разработчик CMS «NetCat»

Интервью брал:
nw

Профессиональная система управления сайтами NetCat является одной из ведущих систем управления контетом (CMS, Content Management System) на российском рынке. Первая версия системы была разработана в 1999 году. Согласно исследованию российского рынка CMS, проведенного интернет-изданием Webinform, система NetCat является самым продаваемым универсальным средством управления сайтами в России. Технический директор компании «Аист», разработавшей «NetCat», Петр Филатов, любезно согласился дать интервью для нашего журнала.

nw: Немного истории. Насколько я понял из информации с вашего сайта (http://www.aist.ru/about/history/), вы не сразу стали коммерческой компанией. Изначально это был некий творческий союз. Расскажите немного о том периоде: что объединило двух первых разработчиков? При каких обстоятельствах к ним присоединились еще люди? Что, по вашему мнению, позволило этим людям стать коммерческой организацией, ведь у каждого были свои дела, и их объединяло только хобби?

ПФ: Изначально разработчиков объединяла общая идея создания полезных и интересных интернет-ресурсов (Netinfo.ru, Fanat.ru, Basketball.ru). Это были веб-порталы достаточно серьезного уровня, для которых требовалась мощная программная платформа, но оказалось, что для реализации подобной платформы и работы над проектами двух человек явно недостаточно, поэтому к процессу разработки были привлечены новые люди. Впоследствии созданная программная платформа была переработана в первую версию системы NetCat.

Изначально разработчиков объединяла общая идея создания полезных и интересных интернет-ресурсов (Netinfo.ru, Fanat.ru, Basketball.ru)

Через некоторое время после открытия нескольких веб-проектов было принято решение оказывать услуги по созданию сайтов. Это решение, в основном, было вызвано тем, что порталы являлись некоммерческими, но при этом требовали все больше времени и это время необходимо было как-то окупать. Мы получили несколько предложений по разработке сайтов на нашей системе, что явилось показателем востребованности подобной системы на рынке веб-девелопмента и послужило одним из аргументов в пользу оказания услуг по созданию сайтов. Также при создании порталов нами был накоплен солидный опыт в веб-разработке и это позволило нам сразу заниматься сложными проектами, минуя стадию «ваш сайт за сто долларов».

В 2001 году мы представили коробочный продукт NetCat 2.0. Текущая версия системы – NetCat 2.2.

nw: Почему вы выбрали название NetCat? Это просто красивое словосочетание (в переводе) или оно чем-то обусловлено?

ПФ: Название системы является сокращением двух слов «InterNET CATalog». Любой человек, знакомый с иностранными языками, может легко перевести это словосочетание. Безусловно, в настоящее время функциональность системы гораздо шире, нежели тривиальный интернет-каталог, но это привлекательное сокращение было решено оставить в качестве названия системы. Можно сказать, что так исторически сложилось.

nw: На вашем сайте сказано, что NetCat – это лучшая российская система управления сайтами в своем классе. А какие вообще бывают классы систем управления сайтами? Расскажите подробнее, чем они на ваш взгляд отличаются друг от друга и, если можно, приведите примеры систем каждого класса?

ПФ: NetCat является лучшей российской системой управления сайтами в классе легких и средних систем согласно исследованию интернет-издания WebInform.ru. В статье (http://www.webinform.ru/analyst/681.html), которая является самым актуальным независимым исследованием рынка CMS на текущий момент, представлены основные системы управления сайтами различных классов.

Наша система относится к классу легких и средних систем управления сайтами, поскольку линейка продуктов NetCat охватывает достаточно широкий спектр сайтов, начиная с сайтов-визиток (NetCat Small Business) и заканчивая корпоративными сайтами и сложными порталами (NetCat Standard и Extra).

Функция eval() в движке обеспечивает необходимую гибкость при работе с шаблонами. Гораздо проще реализовать что-то в шаблоне средствами PHP, нежели изучать внутренний макроязык

nw: Почему для NetCat вы выбрали именно PHP и MySQL? У нас на форуме phpclub.ru также прозвучал вопрос, почему в основу шаблонного движка положена именно функция PHP eval()? Или это не так?

ПФ: Связка PHP и MySQL является самой распространенной для отечественных хостингов, а среди веб-разработчиков подавляющее большинство знакомы с данными средствами разработки.

Функция eval() в движке обеспечивает необходимую гибкость при работе с шаблонами. Гораздо проще реализовать что-то в шаблоне средствами PHP, нежели изучать внутренний макроязык.

nw: Некоторые разработчики жалуются на то, что в NetCat не самым лучшим образом представление отделено от логики (т.е. есть куски PHP в шаблонах). В следующих версиях NetCat вы планируете совершенствовать свою систему работы с шаблонами, или, быть может, есть идеи по использованию таких разработок, как Smarty? Почему?

ПФ: Куски PHP в шаблонах не является частью логики, это вспомогательный код для отображения информации, который может как отсутствовать, так и прописываться индивидуально для каждого шаблона или макета дизайна.

Логика описана в ядре системы и отделена от пользовательских настроек.

nw: Какие проблемы, на ваш взгляд, являются самыми трудными при разработке системы управления сайтами? Как вы их решали?

ПФ: Основной проблемой для всех разработчиков CMS являлась и является реализация удобного инструмента как для разработки, так и управления сайтом, с которым сможет работать не только веб-разработчик, но и обычный, рядовой пользователь.

В нашем случае, удобство разработки, в частности, обеспечивает универсальный механизм шаблонов данных, позволяющий создать практически любой тип данных: новости, каталог, фотогалерею и т.д. С точки зрения управления контентом был проработан интуитивно понятный интерфейс, который не требует никаких специальных знаний в области веб-технологий.

nw: Сейчас разработано много различных систем управления сайтами, в том числе и бесплатных. Почему ваши клиенты выбрали именно вас? В чем платные системы (и NetCat в частности) выигрывают у таких известных бесплатных систем, как PHP Nuke, Xoops, Mambo и т.д?

ПФ: Клиенты отдают предпочтение отечественным разработкам, которые обеспечивают русскоязычную документацию, техническую поддержку, оперативность исправления ошибок. Мы гарантируем клиентам и партнерам работоспособность системы, постоянно развиваем наш продукт с учетом новейших достижений в области интернет-технологий.

Практика наших партнеров показывает, что сейчас в большинстве случаев гораздо более выгодно использовать уже готовые решения, имеющие дилерские программы (например, тот же NetCat), чем тратить громадное количество времени на разработку своего продукта

nw: И, наконец, что бы вы могли пожелать читателям нашего журнала, которые разрабатывают (или принимают участие в разработке) своих систем управления сайтами?

ПФ: Главное – оценить востребованность и перспективность своего продукта. Многие разработчики совершают ошибку, создавая системы управления «под себя», соответственно уровень отчуждаемости продукта в этом случае получается крайне мал. Необходимо на этапе проектирования продумать все нюансы, чтобы в итоге получить продукт, который будет удобен не только для вас, но и для ваших клиентов и любых людей, выбравших вашу систему для управления своими сайтами.

Еще одна ошибка, которая часто допускается при разработке CMS – непроработанная финансовая часть проекта. Практика наших партнеров показывает, что сейчас в большинстве случаев гораздо более выгодно использовать уже готовые решения, имеющие дилерские программы (например, тот же NetCat), чем тратить громадное количество времени на разработку своего продукта.

В любом случае, мы желаем читателям вашего журнала достичь успеха в развитии хороших, добротных продуктов и найти свое место на молодом, но активно развивающемся рынке веб-разработок.

nw: Спасибо за интервью и желаем успехов вашей компании!

Интервью: Питер Росомофф

Интервью брал:
nw
Перевод:
nw

Сайт http://phpclasses.org, известный нам и по интервью в одном из прошлых номеров с его основателем Мануелем Лемосом, проводит ежемесячный конкурс Phpclasses Innovation Award, на самый лучший php-класс месяца. Это состязание поддерживается известными журналами о PHP и компанией Zend, которые и предоставляют призы для победителей. Один из победителей этого конкурса, Питер Росомофф (Peter Rosomoff), любезно согласился дать интервью для нашего журнала.

nw: Прежде всего, расскажите, пожалуйста немного о себе: где живете, кем работаете и каково ваше хобби?

PR: Я живу в городе Аламеда, Калифорния. У меня есть собственный бизнес по оказанию услуг в сфере IT-консалтинга, который специализируется на аутсорсинге IT-менеджмента для малого бизнеса. А помимо работы я люблю играть в гольф и сам создавать мебель.

nw: Вы выиграли апрельский кубок Phpclasses Innovation Award с классом BrowseEdit. Расскажите о нем. Для чего он предназначен и что умеет делать?

PR: BrowseEdit решает проблемы, которые я часто встречал во многих приложениях. Я написал его для таких клиентских программ, которые построены вокруг списков, например, учет запасов и тайм-менеджмент.

Я искал способ редактирования информации в таблицах в том виде, в каком она в них представлена, без необходимости создавать отдельное окно для каждой редактируемой записи. Такой подход можно увидеть, например, в офисных приложениях MS Access или MS Excel, но он редко реализовывается в веб-ориентированных приложениях связки PHP+MySQL.

Класс как раз и позволяет редактировать все в одном окне, включая добавление новых строк, удаление, форматирование представления и возможность сортировки таблицы по щелчку на заголовке необходимого столбца.

В качестве дополнительной функциональности реализован постраничный вывод, проверка форм с помощью JavaScript, использование псевдонимов для столбцов, дополнительные столбцы для URL или командных кнопок и многое другое.

Сейчас я использую этот класс практически во всех приложениях, которые я пишу.

nw: Как вы считаете, почему подписчики phpclasses.org выбрали именно ваш класс?

PR: Этот класс является фундаментальным строительным кирпичиком, с появлением которого исчезают такие рутинные вещи, как многочисленные отдельные окна для редактирования, добавления и удаления записей.

После того, как об этом классе узнают, я надеюсь его станут использовать во многих приложениях. Он решает много проблем, экономит время пользователя и сокращает количество обращений к серверу, снижая нагрузку.

nw: Какие еще классы или приложения вы разработали? Какая у вас специализация в веб-девелопменте (электронная коммерция, промышленность, маркетинг и др.)?

PR: Пока я не написал ни одного опубликованного класса, но имею много идей для будущих публикаций. Я специализируюсь в настройке и подгонке клиентских приложений.

nw: Возможно, вы пишете не только на PHP? Почему вы предпочитаете (или нет) именно PHP?

PR: Я пишу код на различных языках, в зависимости от нужд моих клиентов. Я использую PHP, потому что он является интуитивно понятным, четким и надежным.

Я использую PHP потому что он интуитивно понятный, четкий и надежный

nw: Что вы думаете о PHP 5? Это просто новая версия языка или совершенно новый инструмент для веб-разработок?

PR: Я пока еще не исследовал PHP 5, поэтому не смогу прокомментировать ваш вопрос.

nw: Наконец, что вы можете пожелать российскому PHP-сообществу и читателям журнала PHP Inside?

PR: В начале двадцатого века семья моего отца приехала в Соединенные Штаты из России. Иногда мне хочется посетить те места, где мой дед родился и провел свое детство. Я желаю народу России обладать той свободой и процветанием, которого он заслуживает, и еще, чтобы будущее наших детей было светлым. Я считаю, что возможность общения с другими культурами и народами, подаренная нам новыми технологиями, – это огромное чудо, которое может стать инструментом для построения мирного глобального сообщества.

nw: Спасибо за интервью!

Linuxfest 6.0

Автор:
Елена Тесля [Lenka]

С тридцатого июля по первое августа 2004 года, в Боровском районе Калужской области на берегу живописной речки Протвы прошел шестой ежегодный фестиваль Linux. Фестиваль как всегда был посвящен свободному программному обеспечению , операционной системе Linux и сообществу ее пользователей и разработчиков. Среди участников этого мероприятия была и делегация PHP-клуба.

На склоне молодой лес, полный кустов орешника, внизу луг, поросший сочной травой, а за лугом мелкая река Протва, которую в тех местах можно перейти пешком как вдоль, так и поперек. Близость реки освежает уже одной только мыслью о том, что можно в любой момент плюхнуться в воду. Солнце припекает, птички поют, по земле ползают муравьи и скачут кузнечики... И все это природное спокойствие и благость разбавляют более 300 человек, собравшихся на шестой ежегодный фестиваль Linux, которые совершенно невероятным образом заселили не такой уж большой кусок земли всего на несколько дней.

Палаток море. У каждой компании свой костер. И еще один центральный костер, на который ведут указатели «центр. кос.» – по-нашему «центр космоса». Тут вечером собирается основная масса людей: кто-то играет на гитаре, кто-то поет, кто-то ведет неспешную беседу, конечно же, на околокомпьютерные темы, а кое-кто и спорит вовсю.

Вечером начинаются «миграции» – от костра к костру, от палатки к палатке мы ходим и знакомимся с новыми людьми, и новые люди знакомятся с нами. АнТоХа сразу же проникся этим занятием, и мы перезнакомились с множеством народа. Правда, утром при свете дня оказалось, что обилие новой информации и благодушно предложенного пива стерло из памяти большинство имен, но зато лица в памяти остались. И когда мы подошли к реке, про нас сказали: «О, пхп пошло...» – на наших футболках было написано «Phpclub».

Кстати о футболках – их было множество. Разнообразные надписи: «AltLinux», «Debian», «Gentoo», «Linux», множество различных «Free software...», «Linux_fest-6.0», наши «Phpclub» и т.д. – тут же давали понять, какой дистрибутив человек предпочитает или чем занимается. Флаги, развешанные как над палатками, так и на пляже, тоже пестрили разнообразием: английский, белорусский флаги, флаг с надписью «AltLinux», и даже флаг «Феррари», который просто очень нравился присутствующему товарищу, а вовсе не потому, что он имел какое-то отношение к фирме «Феррари».

Волосатые и стриженные, худые и толстые, программисты, администраторы, начальники, провайдеры и пользователи, из Москвы, Санкт-Петербурга, Обнинска, Калуги, Кирова, Гомеля, Балабаново, Боровска, Зеленограда, Жуковского, Ижевска, Минска, Харькова, Сум и еще множества городов – здесь все были своими и все понимали друг друга.

Без перманентного разговора наших соседей о выделенных линиях, ценах и скоростях становилось уже как-то тихо. А когда Тони, держа в руках моток витой пары, предназначенной для конкурса, решил пошутить и прямо в лесу спросил у подошедших ребят, не найдется ли у них обжимных клещей, они, совершенно не удивившись, посетовали: «Ой, а мы не захватили». Да и чему удивляться, если здесь обитали компьютерщики.

Понятно, когда организаторы ходят с рациями – иначе им приходилось бы по много раз подниматься и спускаться с горы. Но когда компания таким образом собирала грибы!..

Windows на ноутбуке все же вызвал подозрение

- Прием-прием. Я нашел лисичку. Что с ней делать?

- Прием. Бери, что с ней еще делать...

- А тут еще есть какой-то гриб...

И так Shapa по рации руководил сбором грибов. Вполне успешно, надо сказать.

Компьютерщики – люди ленивые, и поэтому никто не удивлялся ни рациям, ни ноутбукам, ни еще каким-то прибамбасам. Правда, Windows на ноутбуке все же вызвал подозрение. Однако, предложенная в последний день ироничная тема «Какой Windows лучше: 2000 или XP?» начала развиваться достаточно горячо, пока кто-то не подошел с замечанием, что это офтопик. Зато тут же было рассказано, что windows можно поставить из исходников.

Размышления по поводу дистрибутивов, их преимуществ и сложности установки продолжались постоянно. Под конец фестиваля была выведена безупречная формула создания флейма: достаточно зайти на форум, посвященный Linux, и написать: «Я новичок. Какой Linux мне поставить?» – тема разовьется с неимоверной скоростью. Сначала посоветуют поставить AltLinux или Mandrake, затем кто-то решит, что надо ставить RedHat, затем мелькнет Gentoo, следом – Debian и Slackware, и после этого начнется...

Примерно таким же образом Энайт заводил разговор с любым, кто приходил к нашему костру. Он спрашивал: «А почему вы выбрали именно этот дистрибутив?» – разговор завязывался быстро.

Лишь однажды Энайт отошел от этого принципа, решив поддержать затихший разговор с заглянувшим к нам на огонек соседом, у которого на футболке красовалась надпись «Debian», вопросом: «Скажите, а это правда, что Debian часто виснет?». И сосед задумался.

Общая идея свободного программного обеспечения проникла в сознание настолько, что все старались поделиться с соседями, помочь или угостить. Инициативная группа, обитающая на месте со вторника, сделала место цивилизованным и организовала буквально все до мелочей. Они построили туалеты в лесу, везде установили указатели на основные достопримечательности («вода», «wc», «центр. кос.» и т.д.) – тут стоит вспомнить фразу Энайта: «Без указателей никуда, что бы ни говорил товарищ VS».

Они соорудили скамейки, привоз дров и вывоз мусора, построили деревянную лестницу, сделали деревянный мостик для входа в воду, предоставили возможность приобрести футболки с символикой шестого фестиваля Linux, установили флаг и даже сделали волейбольное поле с сеткой... Удивительно согласовано и смиренно все собирали после себя мусор и выносили его из леса.

Организаторы присматривали, чтобы никто ничего не оставлял, а Ольга выдавала всем мусорные пакеты – они позаботились обо всем.

Вот, кому надо сказать спасибо – инициативной группе. Правда, кроме Воинса, Райдера, Аватара, Ольги и Раорна с его «раорновкой», которые постоянно собирали народ и являлись центральным организаторским началом, было еще много активных людей, однако, всех не перечислишь.

Запомнился Владлен, который подходил исключительно к девушкам, целовал им ручки и представлялся: «Я Владлен». И ничего не говоря более, уходил. Конечно же, я запомнила Сергея из Харькова – моего земляка – он входил в инициативную группу и жил там со вторника. Запомнились соседи-белорусы. Их было много, человек 20, все светлые и высокие. Из них запомнился Максим, который поддерживал с нами связь, все остальные делали все дружно и к нам по одному заходить не рисковали.

Запомнились ребята из Москвы: Николай, Павел и Аркадий. У них был самый настоящий пингвин, который висел на ветке и перенес даже сильный ливень, промокнув до нитки, но все же пройдя это испытание. После этого все заинтересовались пингвинами, и даже (ох, уж эти мне компьютерщики с их любовью все систематизировать) составили классификацию пингвинов: у настоящих пингвинов не бывает желтых лап. У пингвинов линуксовых лапы желтые, чтобы их можно было отличать. Но встречаются и исключения: рассказывали, что кто-то видел синего пингвина (промолчим о состоянии видевшего) и даже двуглавого (еще раз промолчим).

Мы, php-клубовцы, тоже запомнились людям. Особенно после того, как устроили конкурс по перетягиванию той самой витой пары с призом – бочкой пива. Конкурс прошел на ура, витую пару даже умудрились порвать. И выигравшая команда получила бочку пива, которая и была открыта прямо на пляже, реализовывая «мечту студента». Пиво лилось из краника сбоку сильной струей, и желающий попросту подставлял рот. Конечно же, пиво зачастую выливалось, и человек оказывался облитым пивом с головы до ног – вот это и было мечтой студента.

Да и распространенный «спам» о будущей в сентябре конференции по PHP добавил нам известности, особенно Тони, который этот самый спам распространял – на приглашении к участию так и было написано: «spam ;-)» – и это очень обрадовало окружающих. Приглянулась всем и та часть «спама», которая была распечатана в нечитаемой кодировке – угадайте, в какой операционной системе она печаталась.

Прошедшие ливни не только не помешали, но даже сдружили нас всех, придав всему мероприятию экстрима – тем, кто спускался вниз по размякшей от грязи тропинке, пришлось против своей воли поучавствовать в слаломе. Однако, похоже, что это никого не расстраивало.

Уезжая, мы называли всех этих малознакомых людей «нашими». Хотя со многими из них мы так и не познакомились. Но все же все мы «наши». Потому что мы за свободу.

Команда этого выпуска

Авторы и переводчики

- Петр Елагин [AlienZzzz]
- Александр Ширяев [Dallas]
- Сергей Корнильев [Ded Karnilo]
- Владимир Шапиро
- Илья Гофман [gufy]
- Елена Тесля [Lenka]
- Андрей Олищук [nw]

Редакционная коллегия, корректировка

- Александр Смирнов [PHPclub];
- Елена Тесля [Lenka];
- Александр Войцеховский [young];
- Антон Чаплыгин;
- Андрей Олищук [nw], координатор проекта PHP Inside;
- Александр Ширяев [Dallas];
- Дмитрий Попов;
- Александр Ильяшов [Silya];
- http://phpclub.ru

Подготовка обложки, верстка

- Денис Зенькович;
- Антон Чаплыгин;
- Андрей Олищук [nw].

Спасибо всем участникам проекта!

Координаты редакции журнала:

http://phpinside.net

nw@phpinside.net

Обсуждаем номер на

http://phpclub.ru/talk

Скидка для читателей PHP Inside!

Внимание!

Всем читателям нашего журнала, желающим принять участие в 3-ей международной конференции «Современные технологии эффективной разработки веб-приложений с использованием PHP», предоставляется скидка! Зарегестрироваться для участия в конференции по льготной цене 3300 рублей (вместо 3600) можно по адресу http://phpconf.ru/inside/. Внимание! Количество мест ограничено.

Редакция журнала, PHP Club и спонсоры конференции надеются увидеть на конференции и вас. Будет интересно!

Для заметок

(Сюда можно заносить свои примечания)