PHP Wrap Up

June 17 2014

Thomas Beebe
Advanced DataTools Corp
(tom@advancedatatools.com)

Advanced DataTools

Tom Beebe

Tom is a Senior Database Consultant and has been with Advanced DataTools for over 10 years. He has been working with Informix since college and is currently the lead consultant for Networking, Unix System Administration and Web Development. Tom is Project Manager and lead developer on a variety of Web Development projects.

Contact Info:

tom@advancedatatools.com

www.advancedatatools.com

703-256-0267 x 106

Agenda

- Review Of Concepts
- Building A Portal System
- Handling Files
- Common and Useful Functions
- Working with Existing Projects
- Future of the Language

Review Strings

- Single quotes – string literal
- Double quotes – Will parse some variables inside
- . - concatenate strings
  - **“this” . “ that” . $hello;**
- Can perform direct comparisons using == and ===
- Control characters can be added to double quoted strings $x = “Hello\n”;
- Single quoted strings will print out: \n

Review - PDO

- $dbh = new PDO("informix:host=10.19.40.5;service=1516;database=$database;server=testtcp;", $username, $password);
- The connection string needs to be on one line, or without line breaks.
- $dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
- $dbh->setAttribute(PDO::ATTR_CASE, PDO::CASE_LOWER);

Try/Catch

- The try block is executed
- If an exception or error is reached it is stopped and the catch block is executed
- Use this for error handling, it will make you life much much easier
- Can create your own exceptions
- Can manually throw exceptions:
  throw new Exception("A user error has occurred");

Review Classes

- class user_class { }
- $foo = new user_class();
- __construct and __destruct are the built in functions to auto run
- $this is the internal object for use inside of the class definition
- $foo = new stdClass(); – New standard (empty) class

What Is A Portal System

- An online application that has multiple functions
- Has independent logins and often access levels
- Does “something”
- Whatever marketing buzzword folks want to use today

Turning a Web App Into a Portal

- Just add logins
- And user management
- And forgot your password
- Maybe captchas
- Do you need access levels
- Limited parts of the site
- Administrative tools

PHP and Login Systems

create table users (

user_id serial, email varchar(80),

email_confirmed char(1) default 'N',

full_name varchar(80),

last_on datetime year to second,

last_ip char(15),

password char(32),

date_added datetime year to second default current year to second,

active char(1) default 'Y',

recovery_email varchar(80),

recovery_phone varchar(30));

Login Systems – Sign Up Form

- Always confirm the password
- Confirm the email address if it is the primary communication source
- Enforce password length
- Javascript can be helpful to make it user friendly but don't rely on it
- Consider if you want to allow double quotes in names
- Make error messages friendly, this is the first and easiest place to lose a visitor if they get discouraged

Storing Login Information

- Use the database datestamp for logging signups
- Store as much information as you need
- Make sure to get the IP address
  - **$_SERVER['REMOTE_ADDR']**
  - **$_SERVER['HTTP_X_FORWARDED_FOR']**
- If it is sensitive data, store the personal information encrypted
  - **openssl_encrypt**
- Consistently name your serial key field across your database

Passwords

- Never store your passwords in plain text
- Use a hash system
  - **md5**
  - **sha1**
  - **password_hash**
- Salt your password, don't just use a static salt
- define ('SITE_SALT', 'asd0-19j23dq-9wjd');
- $password = 'pass1234';
- $hashed_password = md5($password . SITE_SALT . $user_created_date);
- It is an arms race, if you have a high security site, talk to your security team about approved password management
- A hashed password can not be recovered, however with enough computing power a match can be found

Forgot Your Password

- This can be a very simple form
- Just prompt the user for their login or email address
- Send them an email with a custom URL page with a parameter to a unique key
- That is a one time 'reset the password for this account' looking from a database table
- It will make your life easier, require a token or secondary authentication if you are worried about needing high security

Logging In

- User puts in username and password
- Password is hashed and compared to the one in the database, if they match you let them in
- Store information in a $_SESSION for them
  - **user_id, last_on, full name, etc**
- Use the user_id to verify they are allowed to be on for each page they visit
- Keep their non sensitive data in the $_SESSION so it will be easily retrieved by the server without needing a database lookup
- Make sure if you do, if they edit their information you update the $_SESSION variable values

Permissions

- Access Level Types
  - **Admin Yes/No**
    - Good for simple sites where the only limited spot is the user/site management piece
  - **Access levels 0-10**
    - Good for scaled sites, user management is 10, but adding content is a 5 and 3 is comments. Easy to maintain
  - **Individual areas of permissions**
    - User Admin
    - Content Admin
    - News Admin
    - Global Admin (always have this)
    - This method takes more work to manage and keep consistent but gives a great deal of flexibility to give users custom roles
    - User a permissions lookup table
- If you store access information in the $_SESSION variable you will need to make users log out and back again if you make a change

Handling Files

- Use the php.ini settings to increase the upload file size
  - **file_uploads = On**
  - **upload_tmp_dir =**
  - **upload_max_filesize = 2M**
  - **post_max_size = 8M**

File Uploads

- <form method='post' enctype="multipart/form-data">
- <input type='file' name='new_file'>
  - **$_FILES["new_file"]["name"] – the filename**
  - **$_FILES["new_file"]["type"] - the mimetype**
  - **$_FILES["new_file"]["size"] - the size in bytes of the file**
  - **$_FILES["new_file"]["tmp_name"] - the filename on disk currently**
  - **$_FILES["new_file"]["error"] - the error code, 0 means none**
- Do not trust the type parameter, it can be tricked, if exposing the file to the world make sure to do a verification beforehand.
- move_uploaded_file($tmp_name, "$target_location/$file");

Common and Useful Functions

- Hashing Passwords
  - **md5**
  - **crypt**
  - **sha1**
  - **password_hash**
- trim – (run this on passwords from input)

Useful Functions

- array_diff_key – compares the keys of 2 arrays
- array_flip – swap the keys and values
- array() – define a blank array
- array_values – returns the values of the array
- array_rand – returns random value of array
- array_unique – returns de-duplicated array
- in_array – Does value exist
- shuffle – randomize array
- sizeof – array length
- sort – sorts the array, there are a series of these

Useful Functions

- fopen/fclose
- fread/fwrite
- fgets/fputs
- file_get_contents – reads a file into a variable
- file_put_contents – dumps a string into a file
- fpassthru – outputs file to browser
- parse_ini_file – reads in an ini file easily
- file_exists – easily check if a file exists

Useful Functions

- Imagecreatefromjpeg
- mktime/date – create dates in almost any format
- pspell_check – spell checks the word
- exif_read_data – read exif headers from file
- sem_get – get semaphore by ID
- json_encode – convert an array to json format
- curl_exec – one of the many curl functions
- ldap_bind – Connect to ldap database
- Xmlreader/writer – xml processing

Working With Existing Projects

- Many open source projects are written in PHP
- Every project will be a little different, of course
- A few things with PHP that helps make it easy to work with
  - **Easy class system, often times a system's API will be written as an object you can work with or extend**
  - **Using the curl extension you can make web calls to any external site**
  - **Native JSON, XML and SOAP support allow integration with most systems**

Working With Existing Systems

- Often they will have a module system, that tends to be the easiest way to get started
- PHP code will be in source, so you can go read the existing code if you have access
- Many of them use a PHP template system where code can be used with HTML, this is good if you want to just customize your site with extra functionality
- If working on the core code do one of:
  - **Make sure to either work with the maintainers to get your code in the main system**
  - **Fork the project and maintain your own version**
  - **Work using the extensibility of classes so that future upgrades won't break your code**

Working With Existing Systems

- Make sure to document your code
- If the system uses caching, make sure to take that into account with your code if it is dynamic
- Document your code both in the code as well as in whatever documentation system the project uses
- ~~Most~~ Many projects have coding standards that they publish, try to follow them

Links

- http://www.php.net
- http://framework.zend.com/
- http://www.zend.com/
- http://www.iiug.org/opensource
- http://pecl.php.net/package/PDO_INFORMIX
- http://pecl.php.net/package/PDO_IBM
- http://www.openadmintool.org

Questions?

Fastest DBA 2014

Webcast July 8th.

Information will be posted at

www.advancedatatools.com

Advanced DataTools

PHP Wrap Up

June 17 2014

Thomas Beebe
Advanced DataTools Corp
(tom@advancedatatools.com)

Advanced DataTools