

5.Implement an effective Logging System using file in PHP.

The most important thing is when and how to log errors. You can log PHP errors at your will when working in the dev mode. Either create a PHP error log file or save them as notifications on different channels, as you get the convenience of logging errors per your requirements.

Different Types of PHP Errors

PHP errors occur when something is off-base within the code. They can be as complex as calling an incorrect variable or as simple as missing a semicolon. You must understand the kind of errors you face to solve them effectively.

Warning Error

PHP warning errors alert about a problem that may cause a more critical error in the long run. Warning errors do not break down the code execution and commonly occur when using a file path that doesn't exist.

```
1. <?php
2. echo "Warning error";
3. include("external_file.php");
4. ?
5. >
```

There is no file named "external_file," so the output will display an error message box, and the execution will not break down.

Notice Error

Notice errors are minor; like warning errors, they don't halt code execution. These errors can confuse the system in analyzing if there's an actual mistake or if it's just the standard code. Notice errors usually occur when the script needs access to an undefined variable.

Syntax Error

A syntax error is often caused by parsed, misused, or missing symbols. When the compiler catches the error, it terminates the script.

Parse/syntax errors are caused by:

- Unclosed brackets or quotes
- Missing or extra semicolons or parentheses
- Misspellings

Here's an example of a parse error message:

```
PHP Parse error: syntax error, unexpected '5' (L_NAME), expecting ')' i  
n /home/u802426761/domains/username/public_html/wp-config.php on line 3  
2
```

Fatal Error

A fatal error happens when the code calls the function, but the function itself isn't characterized. Unlike other PHP errors, a fatal error breaks down the execution and sometimes crashes the application as well. There are three types of fatal errors:

- A startup fatal error happens when the framework can't run the code due to a mistake during installation.
- A compile-time fatal error occurs when the developer uses an undefined function, class, or non-existent variable or data.
- Runtime fatal error is similar to a compile-time fatal error, but happens during the program execution.

Here's an example of a PHP fatal error:

```
PHP Fatal error: Call to undefined function get_header() in /var/www/username/public/blog/wp-content/themes/theme/index.php on line 37
```

Another reason for a fatal error is exceeding the execution time:

```
Fatal error: Maximum execution time of 30 seconds exceeded in /home/username/domains/domain.com/public_html/wp-includes/class-phpmailer.php on line 737
```

Enable Error Logging in php.ini

To log errors in PHP, open the `php.ini` file and uncomment/add the following lines of code.

1. `error_reporting = E_ALL & ~E_NOTICE`
- 2.
3. `error_reporting = E_ALL & ~E_NOTICE | E_STRICT`
- 4.
5. `error_reporting = E_COMPILE_ERROR|E_RECOVERABLE_ERROR|E_ERROR|E_WARNING|E_PARSE|E_NOTICE|E_CORE_ERROR|E_CORE_WARNING|E_DEPRECATED|E_DEPRECATED_WARNING`
- 6.
7. `error_reporting = E_ALL & ~E_NOTICE`

If you want to enable PHP error logging in individual files, add this code at the top of the PHP file.

1. `ini_set('display_errors', 1);`
- 2.
3. `ini_set('display_startup_errors', 1);`
- 4.
5. `error_reporting(E_ALL);`

Now, you must enable only one statement to parse the log errors in the php.ini file:

```
display errors = on.
```

Now, you can easily see logged errors in your browser.

How to log effectively (PHP)

```
<?php
```

```
ini_set("display_errors",1);
```

```
ini_set("log_errors",1);
```

```
ini_set("error_log","temp/error_log.php");
```

```
$x=5;
```

```
$y=0;
```

```
if($y==0)
```

```
{
```

```
trigger_error("Cannot divide by zero",E_USER_ERROR);
```

```
}
```

```
try
```

```
{
```

```
$value=$x%$y;
```

```
}
```

```
catch(Exception $e)
```

```
{  
  
throw new Exception("Divide error..");  
  
$message="Exception:". $e->getMessage();  
  
}  
  
?>
```

OUTPUT:

```
[04-Nov-2023 10:48:13 Asia/kolkata] PHP Fatal error:  Cannot divide by zero in C:  
\xampp\htdocs\errorlog\error_log.php on line 10
```