

XDebug - PHP developer's swiss knife

Advanced tool to debug, profile & trace PHP scripts

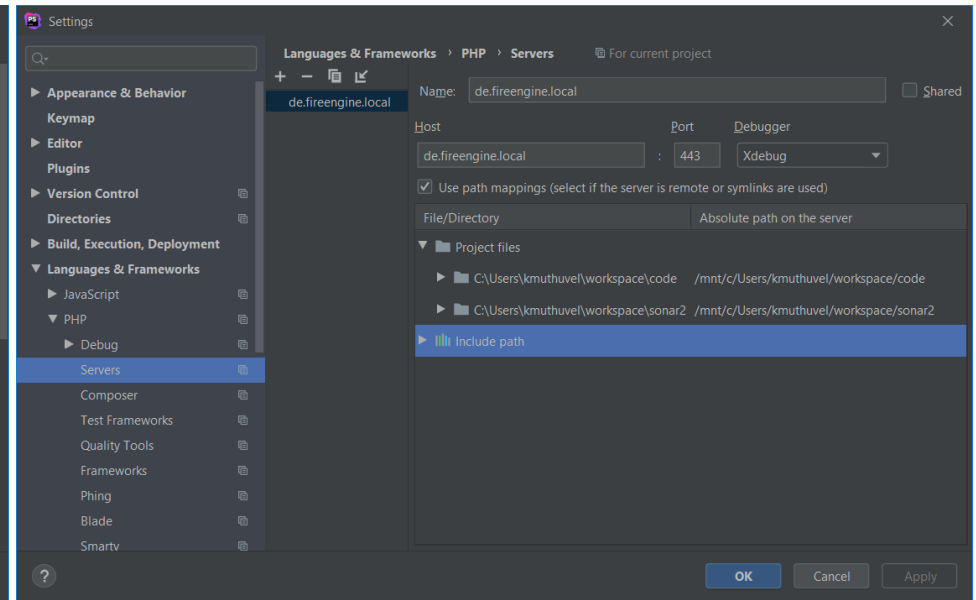
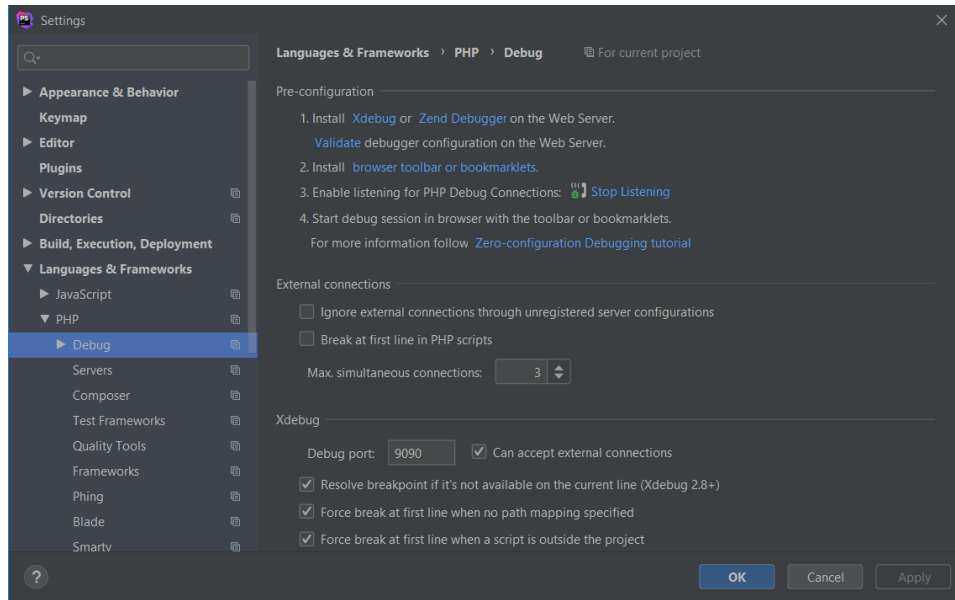
1. Remote debugging
2. Profiling
3. Tracing

Remote debugging

- Ensure php-xdebug installed & loaded. `phpinfo()` from any runnable file
- Configure corresponding `ini` file. Sample below:

```
$bat /etc/php/7.4/mods-available/fire_engine.ini
File: /etc/php/7.4/mods-available/fire_engine.ini
1 max_execution_time = 60
2 memory_limit = 1G
3 error_reporting = E_ALL & ~E_DEPRECATED & ~E_STRICT
4 display_errors = On
5 display_startup_errors = On
6 date.timezone = America/Chicago
7 soap.wsdl_cache_limit = 10
8 opcache.revalidate_path = 1
9 opcache.memory_consumption = 256
10
11 xdebug.profiler_enable=0
12 xdebug.profiler_enable_trigger=1
13 xdebug.profiler_output_dir=/tmp/xdebug.out
14 xdebug.profiler_output_name=callgrind.out.%t
15 xdebug.profiler_append=0
16
17 xdebug.trace_output_dir=/tmp/xdebug.out
18
19 xdebug.remote_enable = 1
20 xdebug.remote_connect_back = 1
21 xdebug.remote_port= 9090
22
23 #vdebug
24 xdebug.remote_handler="dbgp"
25 xdebug.remote_host=localhost
26 xdebug.remote_autostart=0
0 ✓ success ^_^
```

PHPStorm



1. Setup PHPStorm IDE – Configure the port & **mapping**
2. Pass the cookie with help of [bookmarklets](#)

The image shows the PhpStorm IDE interface with two PHP files open. The left pane shows `OrdersPlacedDeReport.php` with a function `patchResultUrlCodes` that constructs an SQL query. The right pane shows `Auth_login.php` with a `login_user` function that queries the database and sets session data.

```
769 * Get mapping array to get order id to result row array index/key.
770 * @param $results
771 * @return array
772 */
773 protected function getResultOrderIdMap($results)
774 {
775     $orderMap = [];
776     array_walk(&$results, function ($row, $key) use (&$orderMap) {
777         $orderMap[$row['order_id']] = $key;
778     });
779
780     return $orderMap;
781 }
782
783 protected function patchResultUrlCodes(array $results, array $orderMap)
784 {
785     $orderIds = array_keys($orderMap);
786     $sql = "SELECT
787         ova.order_id,
788         GROUP_CONCAT(CONCAT(uc.name, '=', ova.value) SEPARATOR '|') as url
```

```
76 LIMIT 1;
77 $query = $this->db->query($sql, [$username]); $sql: "SELECT id, password_has
78 if (!($results = $query->result_array())) { $query: {conn_id => PDO, resul
79     return false;
80 }
81 $user = $results[0]; $results: {[2]}[1] $user: {id => "1191", password_hasi
82
83 if (!\password_verify($password, $user['password_hashed'])) { $password: "P
84     return false;
85 }
86
87 $this->ci->session->set_userdata('user_id', $user['id']); $user: {id => "119
88 xdebug_stop_trace();
89 return $user['id'];
90 }
91
92 function show_login()
93 {
94     $this->ci->load->library('layout', array('layout' => 'layouts/main'));
95     $this->ci->layout->view('auth/auth_login', array (
```

The debugger window at the bottom shows the current execution state:

- Debugger: index.php
- Frames:
 - Auth_login.php:107, Auth_login->login_user()
 - Auth_login.php:69, Auth_login->current_user()
 - Auth_login.php:36, Auth_login->__construct()
- Variables:
 - \$sql = "SELECT id, password_hashed \n\n FROM sonar_users \n\n WHERE username = ? AND effective_date <= CURDATE() AND user..."
 - \$user = {array} [2]
 - \$username = "kesavan"
 - \$this = {Auth_login} [7]
 - \$_COOKIE = {array} [2]

Atom:

The screenshot shows the VS Code editor with a PHP file open. The code is as follows:

```
83 function logout_user()
84 {
85     $this->ci->session->unset_userdata('us
86     redirect();
87 }
88
89 function login_user($username, $password)
90 {
91     $this->db = $this->load->database('database', true);
92     $sql = "SELECT id, password_hashed
93     FROM sonar_users
94     WHERE username = ? AND effective_date <= CURDATE() AND user_status = 1
95     LIMIT 1";
96     $query = $this->db->query($sql, [$username]);
97     if (!$results = $query->result_array()) {
98         return false;
99     }
100     $user = $results[0];
101
102     if (!$password_verify($password, $user['password_hashed'])) {
```

The debugger interface is visible, showing the following panels:

- Debugger:** The debugger is not attached. Buttons for "Attach debugger..." and "Launch debugger..." are present.
- Breakpoints:** Three breakpoints are set at lines 92, 96, and 100.
- Stack Trace:** Shows the current function call: 0 Auth_login->login_user.
- Context:** Shows local variables: \$password = "Password-1", \$query = object[CI_DB_pdo_result], \$results = ?, \$sql = "SELECT id, password_hashed FROM sonar_users WHERE username = ? AND effective_date <= CURDATE() AND user_status = 1 LIMIT 1", \$user = ?, \$username = "kesavan", \$this = object[Auth_login].
- Watches:** A new watch is added: \$myVariable.
- Breakpoints:** Shows the current breakpoints and exceptions.

The status bar at the bottom indicates the current file is application\libraries\Auth_login.php at line 96:3.

VSCode:

The screenshot displays the Visual Studio Code IDE with the following components:

- Menu Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Toolbar:** RUN, Listen for XDebug, launch.json, CommissionReport.php, Auth_login.php, Settings, and various execution and navigation icons.
- WATCH:** Shows a SQL query: `$sql: "SELECT id, password_hashed"`.
- CALL STACK:** Labeled "PAUSED ON BREAKPOINT", showing the call stack from `Auth_login->login_user` down to `{main}`.
- BREAKPOINTS:** Shows two breakpoints set on `Auth_login.php` at lines 97 and 101.
- VARIABLES:** Shows the current state of variables:
 - `$password: "Password-1"`
 - `> $query: CI_DB_pdo_result`
 - `> $results: array(1)`
 - `$sql: "SELECT id, password_hashed"`
 - `$user: uninitialized`
 - `$username: "kesavan"`
 - `> $this: Auth_login`
 - `> Superglobals`
 - `> User defined constants`
- Code Editor:** Shows PHP code for `Auth_login.php`. A breakpoint is set at line 101, which is `$user = $results[0];`. The code includes a `login_user` function that queries a database for user credentials.

vim – My personal choice

Please use plugin - [vdebug](#)

```
Ubuntu 18.04
Auth_login.php 4 Auth_login.php
64 if (isset($_POST[$this->login_post_var]) && isset($_POST[$this->password_post_
var]))
65 {
66     $login = $_POST[$this->login_post_var];
67     $pass = $_POST[$this->password_post_var];
68
69     if ($user_id = $this->login_user($login, $pass))
70     {
71         return $user_id;
72     }
73     else
74     {
75         $this->login_error = "Invalid Username/Password";
76     }
77 }
78 }
79 return false;
80 }
81 }
82
83 function logout_user()
84 {
85     $this->ci->session->unset_userdata('user_id');
86     redirect();
87 }
88
89 function login_user($username, $password)
90 {
91     $this->db = $this->load->database('database', true);
92     xdebug_start_trace();
93     $sql = "SELECT id, password_hashed
94           FROM sonar_users
95           WHERE username = ? AND effective_date <= CURDATE() AND user_status = 1
96           LIMIT 1";
97     $query = $this->db->query($sql, [$username]);
98     if (!$results = $query->result_array()) {
99         return false;
100     }
101     $user = $results[0];
102
103     if (!password_verify($password, $user['password_hashed'])) {
104         return false;
105     }
106
107     $this->ci->session->set_userdata('user_id', $user['id']);
108     xdebug_stop_trace();
109     return $user['id'];
110 }
111
112 function show_login()
113 {
114     $this->ci->load->library('layout', array('layout' => 'layouts/main'));
115     $this->ci->layout->view('auth/auth_login', array (
116         'error' => $this->login_error,
117         'username' => $this->login_post_var,
118         'password' => $this->password_post_var
119     ));
120 }
121 }

Auth_login.php 101,9 Bot DebuggerWatch 1,1 Top
python3 debugger.run()
```

QCacheGrind – Benchmarking the performance

Parts Overview

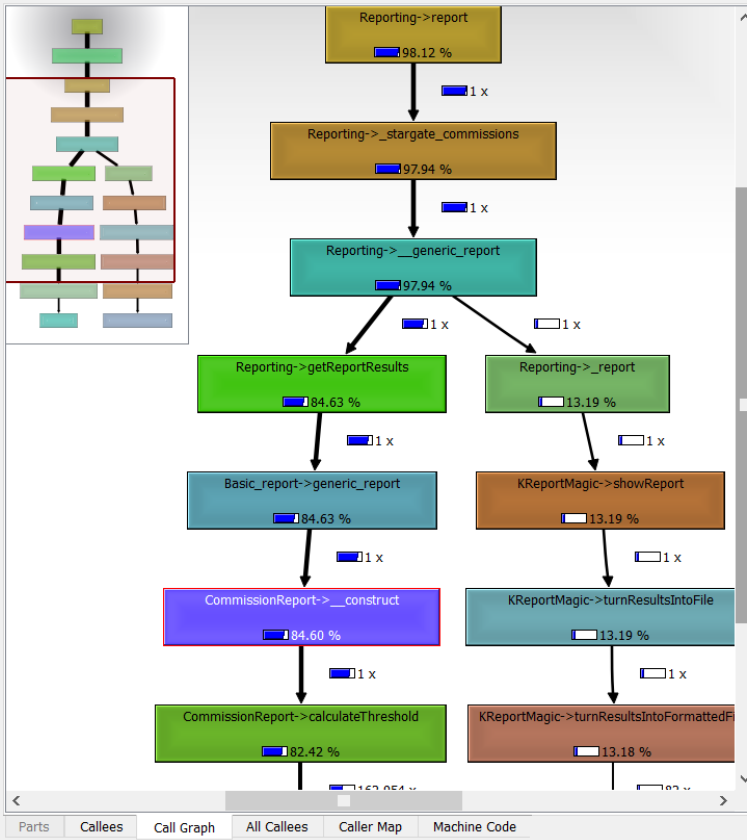
Top Cost Call Stack

Time	Calls	Function
--	--	-- (main)
100.53	1	require_once::/mnt/c/Users/kmuth...
98.66	1	Reporting->report
98.47	1	Reporting->_stargate_commissions
98.47	1	Reporting->_generic_report

Flat Profile

Search: (No Grouping)

Incl.	Self	Called	Function
100.55	0.01	(0)	(main)
100.53	0.08	1	require_once::/mnt/c/Users/kmuth...
98.66	0.00	1	Reporting->report
98.47	0.00	1	Reporting->_stargate_co...
98.47	0.00	1	Reporting->_generic_re...
85.09	0.00	1	Reporting->getReportRe...
85.09	0.01	1	Basic_report->generic_re...
85.06	1.09	2	CommissionReport->_c...
82.87	27.27	1	CommissionReport->cal...
52.16	40.91	162 954	CommissionReport->get...
13.26	0.00	1	Reporting->_report
13.26	0.00	1	KReportMagic->showRe...
13.26	0.00	1	KReportMagic->turnRes...
13.26	0.11	1	KReportMagic->turnRes...
7.88	0.29	203	PhpOffice\PhpSpreadsh...
6.84	0.04	164	PhpOffice\PhpSpreadsh...
6.79	0.97	1 438	PhpOffice\PhpSpreadsh...
6.15	6.15	325 912	php::date
6.11	0.16	176	PhpOffice\PhpSpreadsh...
5.60	5.60	163 108	php::strtotime
4.01	1.55	1 480	PhpOffice\PhpSpreadsh...



(main)

Types Callers All Callers Callee Map Source Code

Incl. Self Distance Calling Callee

99.98	0.08	1	1	require_once::/mnt/c/Users/kmthuve\workspace/sonar2/ap...
98.12	0.00	2	1	Reporting->report (Reporting.php)
97.94	0.00	3	1	Reporting->_stargate_commissions (Reporting.php)
97.94	0.00	4	1	Reporting->_generic_report (Reporting.php)
84.63	0.00	5	1	Reporting->getReportResults (Reporting.php)
84.63	0.01	6	1	Basic_report->generic_report (Basic_report.php)
84.60	1.09	7	1	CommissionReport->_construct (BaseReport.php, ...)
82.42	27.12	8	1	CommissionReport->calculateThreshold (CommissionReport...
51.87	40.68	9	162 954	CommissionReport->getBucketByDate (CommissionReport...
13.19	0.00	5	1	Reporting->_report (Reporting.php)
13.19	0.00	6	1	KReportMagic->showReport (KReportMagic.php)
13.19	0.00	7	1	KReportMagic->turnResultsIntoFile (KReportMagic.php)
13.18	0.11	8	1	KReportMagic->turnResultsIntoFormattedFile (KReportMagi...
7.82	0.28	10	105	PhpOffice\PhpSpreadsheet\Style\Style->applyFromArray (St...
6.81	0.04	9	82	PhpOffice\PhpSpreadsheet\Style\Alignment->setHorizontal...

Update 2022/Jan - Eclipse for PHP dev

1. Install **Eclipse** for PHP dev. It's preloaded with java11 & you no need to worry on JREs
2. Load a project from your workspace and let Eclipse index it first
3. Choose **Break at First Line** at `Window > Preferences > PHP > Debug`
4. Configure Xdebug (`Window > Preferences > PHP > Debug > Debuggers > Xdebug > Configure`)
 - a. Port matching your `/etc/php/7.4/mods-available/fire_engine.ini`
 - b. Listen to remote call. Preferably `Accept remote session(JIT) : prompt`
5. Configure Mapping (`Window > Preferences > PHP > Servers`)
 - a. Add `New` & config the mapping;
 - b. Ensure `Xdebug` picked & `path mapping` is relative. Refer sample below
6. Add some breakpoint on your code journey
7. Trigger request from browser with cookie `'XDEBUG_SESSION'`
 - a. Remote debug session request should prompted; Approve to debug

