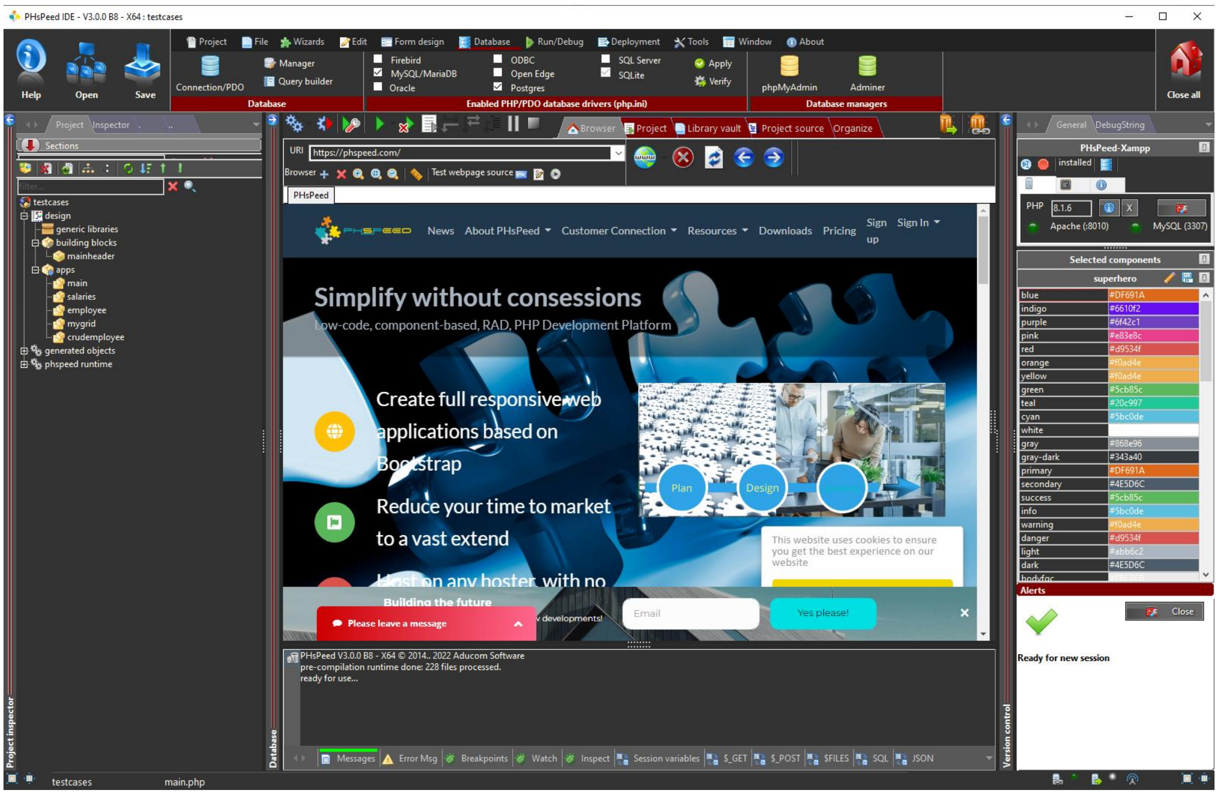
PHsPeed Introduction

Debugging your PHP code

7

PHsPeed: Getting Started



```
PHsPeed V3.0.0 B8 - X64 © 2014.. 2022 Aducom Software
pre-compilation runtime done: 228 files processed.
ready for use...
```

blue	#DF691A
indigo	#6610f2
purple	#6f42c1
pink	#e83e8c
red	#d9534f
orange	#f0ad4e
yellow	#f0ad4e
green	#5cb85c
teal	#20c997
cyan	#5bc0de
white	
gray	#868e96
gray-dark	#343a40
primary	#DF691A
secondary	#4E5D6C
success	#5cb85c
info	#5bc0de
warning	#f0ad4e
danger	#d9534f
light	#abb6c2
dark	#4E5D6C
bodyfgc	[illegible]

Welcome to the world of PHsPeed! This manual is designed to introduce you to our low-code PHP development tool and help you work faster and smarter, aligning with the modern approach to web application development.

In the previous documents, we discussed the fundamental features of the Integrated Development Environment (IDE), created the first basic PHsPeed project demonstrating the 'hello world' example, and showed you how to leverage the application wizards to expedite your development process. You also know by now how to create a form manually and wrote your first PHP code in an Ajax event. We also explained the bootstrap grid system to lay out your forms. You have a basic understanding about events, integrating custom code, and the application logic flow. Now it is time to dive into the possible failures. What if you have issues in the code. How can you debug and trace the error(s)?

Possible issues

This manual aims to provide you with comprehensive insights into the debugging capabilities that PHsPeed offers for PHP applications. With PHsPeed, you can efficiently diagnose and resolve issues within your PHP projects, all without the need for external software. Our integrated development environment (IDE) empowers you to streamline your debugging process, enabling you to locate and address problems right where they occur.

In standard PHP development, encountering errors is a common occurrence. These errors often manifest as distinctive orange message forms, alerting you to the specific issues that have caused your application to cease functioning. Typically, these messages display the line number at which the execution was interrupted. However, merely possessing this information does not always lead to an immediate identification of the underlying problem. While it points you to the point of application termination, it does not necessarily reveal where the issue was introduced.

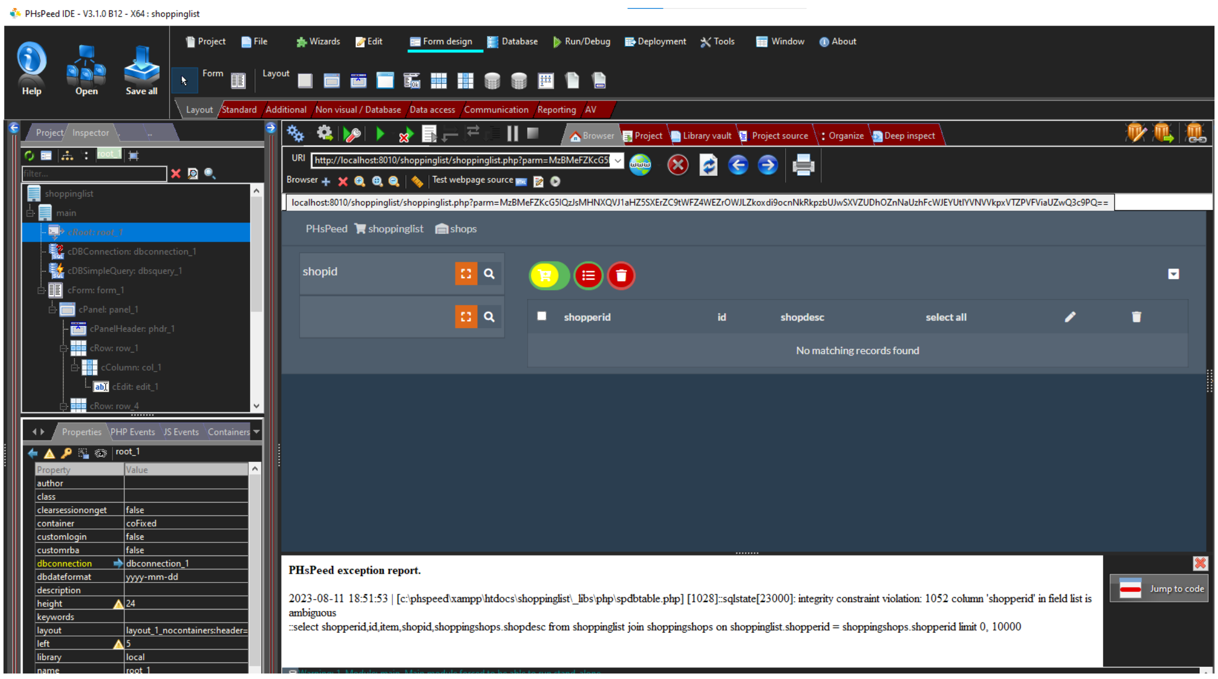
Web applications encompass intricate structures with numerous potential points of failure. Various factors can contribute to malfunctions, ranging from CSS anomalies that result in flawed field rendering, JavaScript glitches leading to application dysfunction, to template errors preventing the rendering of fields altogether. Given this complexity, determining the starting point for debugging can be a perplexing challenge.

This manual provides a comprehensive overview of the debugging methodologies natively supported by PHsPeed. The distinctive feature of our approach is that it eliminates the need for supplementary software tools. All debugging processes are seamlessly integrated into our IDE, streamlining your workflow and enhancing your debugging efficiency.

The subsequent sections of this manual will delve into specific debugging techniques that PHsPeed offers "out of the box". We will illustrate step-by-step instructions on utilizing these tools effectively, ensuring you gain a comprehensive understanding of how to diagnose and rectify issues within your PHP applications.

Debugging PHP.

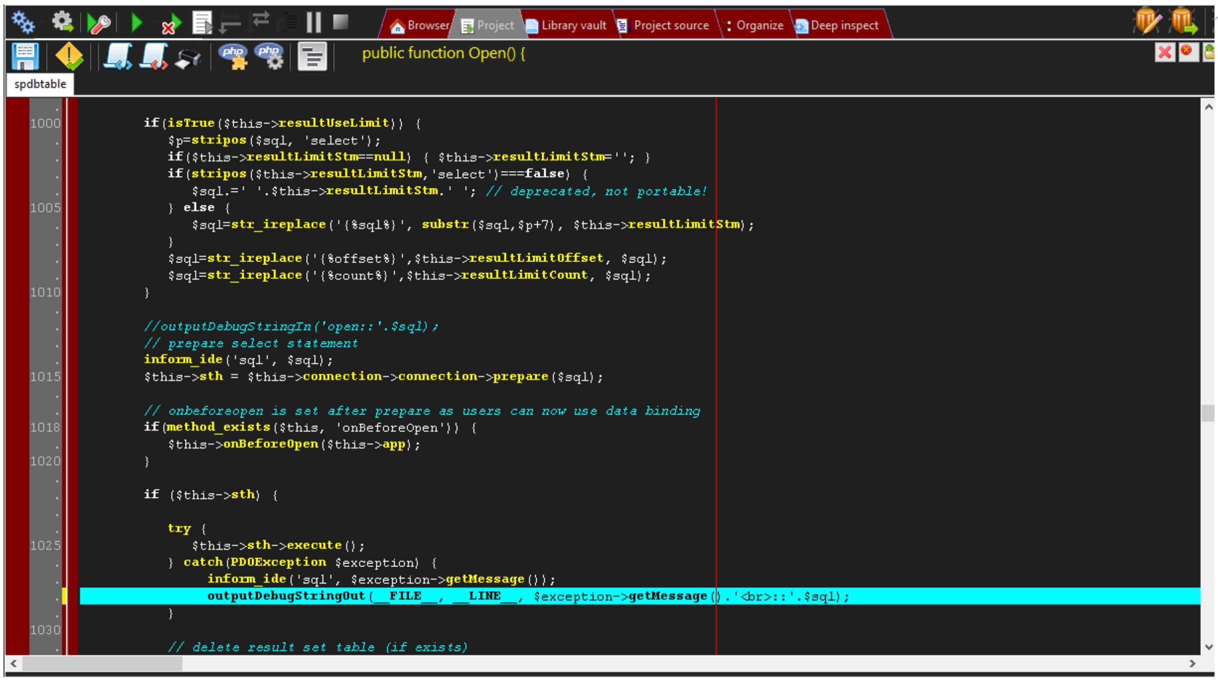
When you generate and run the code, PHsPeed will abort on issues and shows the error with line:



```
PHsPeed exception report.

2023-08-11 18:51:53 | [c:\phspeed\xampp\htdocs\shoppinglist\_libs\php\spdbtable.php] [1028]::sqlstate[23000]: integrity constraint violation: 1052 column 'shopperid' in field list is ambiguous
::select shopperid,id,item,shopid,shoppingshops.shopdesc from shoppinglist join shoppingshops on shoppinglist.shopperid = shoppingshops.shopperid limit 0, 10000
```

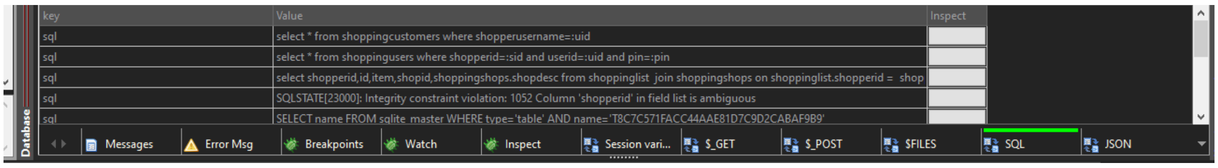
PHsPeed IDE - V3.1.0 B12 - X64 : shoppinglist
Jump to code
No matching records found

Clicking on ‘jump to code’ will open the PHP editor and displays the position where the application failed. This looks like the traditional PHP way. The line where the message is generated does not provide information about the real issue. In this case, it is NOT a PHP failure but a database failure (SQL error).



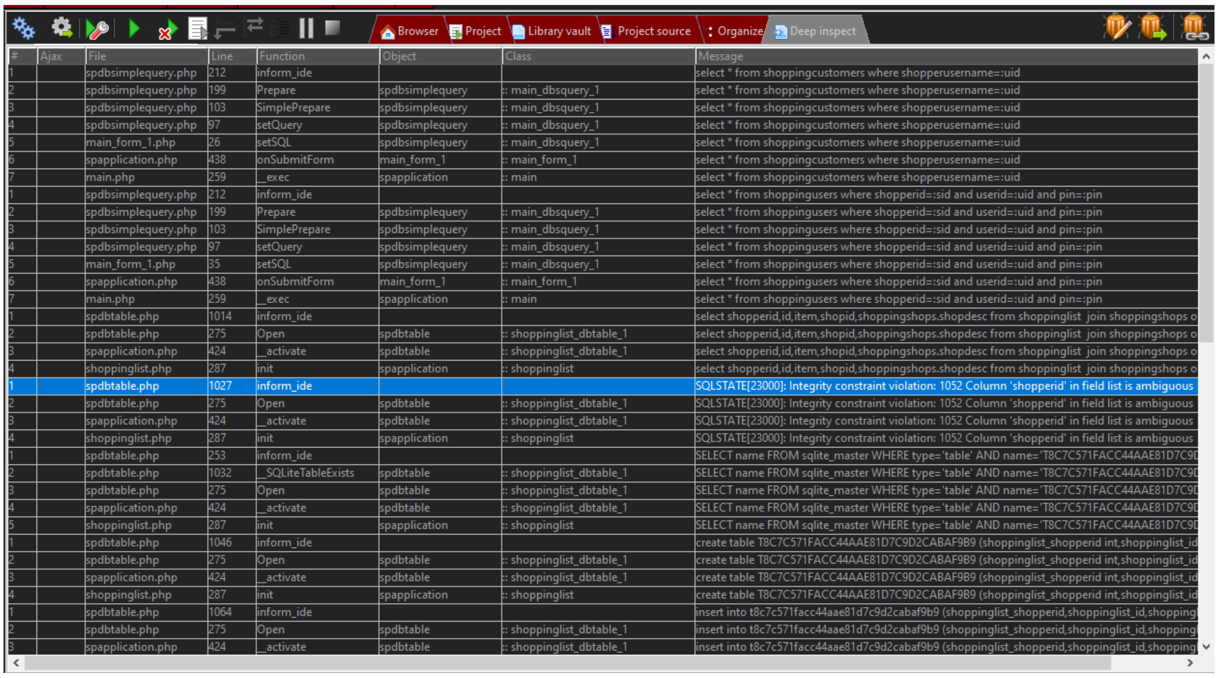
```
        if(isTrue($this->resultUseLimit)) {
           $p=stripos($sql, 'select');
           if($this->resultLimitStm==null) { $this->resultLimitStm=''; }
           if(stripos($this->resultLimitStm,'select')===false) {
              $sql.=' '.$this->resultLimitStm.' '; // deprecated, not portable!
           } else {
              $sql=str_ireplace('{%sql%}', substr($sql,$p+7), $this->resultLimitStm);
           }
           $sql=str_ireplace('{%offset%}',$this->resultLimitOffset, $sql);
           $sql=str_ireplace('{%count%}',$this->resultLimitCount, $sql);
        }

        //outputDebugStringIn('open::'.$sql);
        // prepare select statement
        inform_ide('sql', $sql);
        $this->sth = $this->connection->connection->prepare($sql);

        // onbeforeopen is set after prepare as users can now use data binding
        if(method_exists($this, 'onBeforeOpen')) {
           $this->onBeforeOpen($this->app);
        }

        if ($this->sth) {

           try {
              $this->sth->execute();
           } catch(PDOException $exception) {
                 inform_ide('sql', $exception->getMessage());
                 outputDebugStringOut(__FILE__, __LINE__, $exception->getMessage().'<br>::'.$sql);
           }

           // delete result set table (if exists)
```

In PHsPeed, you can view the history of executed SQL statements. Open the tab SQL to look at the executed statements:



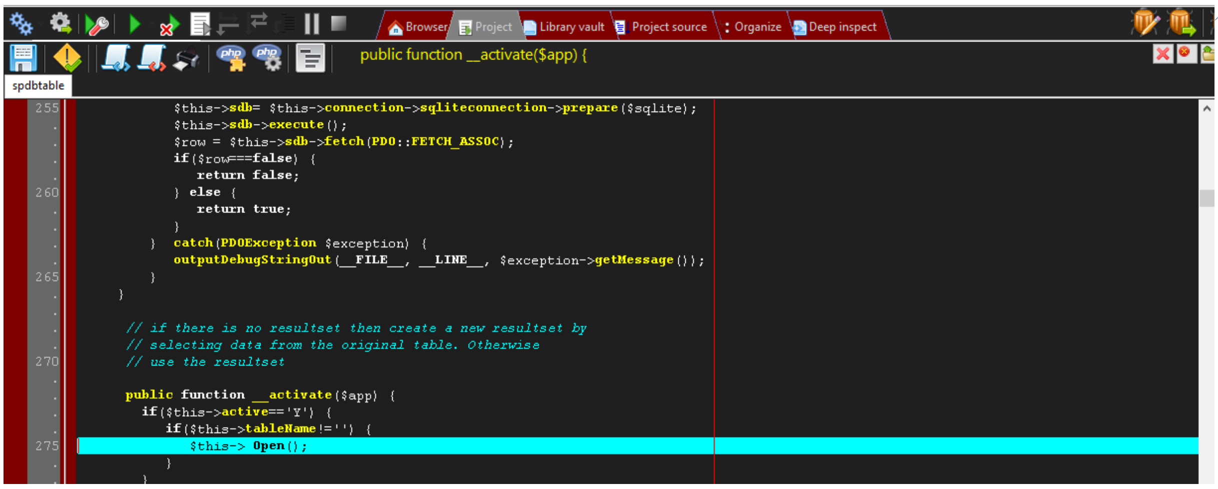
key	Value	Inspect
sql	select * from shoppingcustomers where shopperusername=:uid	
sql	select * from shoppingusers where shopperid=:sid and userid=:uid and pin=:pin	
sql	select shopperid,id,item,shopid,shoppingshops.shopdesc from shoppinglist join shoppingshops on shoppinglist.shopperid = shop	
sql	SQLSTATE[23000]: Integrity constraint violation: 1052 Column 'shopperid' in field list is ambiguous	
sql	SELECT name FROM sqlite_master WHERE type='table' AND name='T8C7C571FACC44AAE81D7C9D2CABAF9B9'	

The ‘SQLSTATE[23000]: Integrity constraint violation: 1052 Column 'shopperid' in field list is ambiguous ‘ is found in the list, lets open the full history by clicking on the button behind.



#	Ajax	File	Line	Function	Object	Class	Message
1		spdbsimplequery.php	212	inform_ide			select * from shoppingcustomers where shopperusername=:uid
2		spdbsimplequery.php	199	Prepare	spdbsimplequery	:: main_dbsquery_1	select * from shoppingcustomers where shopperusername=:uid
3		spdbsimplequery.php	103	SimplePrepare	spdbsimplequery	:: main_dbsquery_1	select * from shoppingcustomers where shopperusername=:uid
4		spdbsimplequery.php	97	setQuery	spdbsimplequery	:: main_dbsquery_1	select * from shoppingcustomers where shopperusername=:uid
5		main_form_1.php	26	setSQL	spdbsimplequery	:: main_dbsquery_1	select * from shoppingcustomers where shopperusername=:uid
6		spapplication.php	438	onSubmitForm	main_form_1	:: main_form_1	select * from shoppingcustomers where shopperusername=:uid
7		main.php	259	__exec	spapplication	:: main	select * from shoppingcustomers where shopperusername=:uid
1		spdbsimplequery.php	212	inform_ide			select * from shoppingusers where shopperid=:sid and userid=:uid and pin=:pin
2		spdbsimplequery.php	199	Prepare	spdbsimplequery	:: main_dbsquery_1	select * from shoppingusers where shopperid=:sid and userid=:uid and pin=:pin
3		spdbsimplequery.php	103	SimplePrepare	spdbsimplequery	:: main_dbsquery_1	select * from shoppingusers where shopperid=:sid and userid=:uid and pin=:pin
4		spdbsimplequery.php	97	setQuery	spdbsimplequery	:: main_dbsquery_1	select * from shoppingusers where shopperid=:sid and userid=:uid and pin=:pin
5		main_form_1.php	35	setSQL	spdbsimplequery	:: main_dbsquery_1	select * from shoppingusers where shopperid=:sid and userid=:uid and pin=:pin
6		spapplication.php	438	onSubmitForm	main_form_1	:: main_form_1	select * from shoppingusers where shopperid=:sid and userid=:uid and pin=:pin
7		main.php	259	__exec	spapplication	:: main	select * from shoppingusers where shopperid=:sid and userid=:uid and pin=:pin
1		spdbtable.php	1014	inform_ide			select shopperid,id,item,shopid,shoppingshops.shopdesc from shoppinglist join shoppingshops o
2		spdbtable.php	275	Open	spdbtable	:: shoppinglist_dbtable_1	select shopperid,id,item,shopid,shoppingshops.shopdesc from shoppinglist join shoppingshops o
3		spapplication.php	424	__activate	spdbtable	:: shoppinglist_dbtable_1	select shopperid,id,item,shopid,shoppingshops.shopdesc from shoppinglist join shoppingshops o
4		shoppinglist.php	287	init	spapplication	:: shoppinglist	select shopperid,id,item,shopid,shoppingshops.shopdesc from shoppinglist join shoppingshops o
1		spdbtable.php	1027	inform_ide			SQLSTATE[23000]: Integrity constraint violation: 1052 Column 'shopperid' in field list is ambiguous
2		spdbtable.php	275	Open	spdbtable	:: shoppinglist_dbtable_1	SQLSTATE[23000]: Integrity constraint violation: 1052 Column 'shopperid' in field list is ambiguous
3		spapplication.php	424	__activate	spdbtable	:: shoppinglist_dbtable_1	SQLSTATE[23000]: Integrity constraint violation: 1052 Column 'shopperid' in field list is ambiguous
4		shoppinglist.php	287	init	spapplication	:: shoppinglist	SQLSTATE[23000]: Integrity constraint violation: 1052 Column 'shopperid' in field list is ambiguous
1		spdbtable.php	253	inform_ide			SELECT name FROM sqlite_master WHERE type='table' AND name='T8C7C571FACC44AAE81D7C9D
2		spdbtable.php	1032	__SQLiteTableExists	spdbtable	:: shoppinglist_dbtable_1	SELECT name FROM sqlite_master WHERE type='table' AND name='T8C7C571FACC44AAE81D7C9D
3		spdbtable.php	275	Open	spdbtable	:: shoppinglist_dbtable_1	SELECT name FROM sqlite_master WHERE type='table' AND name='T8C7C571FACC44AAE81D7C9D
4		spapplication.php	424	__activate	spdbtable	:: shoppinglist_dbtable_1	SELECT name FROM sqlite_master WHERE type='table' AND name='T8C7C571FACC44AAE81D7C9D
5		shoppinglist.php	287	init	spapplication	:: shoppinglist	SELECT name FROM sqlite_master WHERE type='table' AND name='T8C7C571FACC44AAE81D7C9D
1		spdbtable.php	1046	inform_ide			create table T8C7C571FACC44AAE81D7C9D2CABAF9B9 (shoppinglist_shopperid int,shoppinglist_id
2		spdbtable.php	275	Open	spdbtable	:: shoppinglist_dbtable_1	create table T8C7C571FACC44AAE81D7C9D2CABAF9B9 (shoppinglist_shopperid int,shoppinglist_id
3		spapplication.php	424	__activate	spdbtable	:: shoppinglist_dbtable_1	create table T8C7C571FACC44AAE81D7C9D2CABAF9B9 (shoppinglist_shopperid int,shoppinglist_id
4		shoppinglist.php	287	init	spapplication	:: shoppinglist	create table T8C7C571FACC44AAE81D7C9D2CABAF9B9 (shoppinglist_shopperid int,shoppinglist_id
1		spdbtable.php	1064	inform_ide			insert into t8c7c571facc44aae81d7c9d2cabaf9b9 (shoppinglist_shopperid,shoppinglist_id,shopping
2		spdbtable.php	275	Open	spdbtable	:: shoppinglist_dbtable_1	insert into t8c7c571facc44aae81d7c9d2cabaf9b9 (shoppinglist_shopperid,shoppinglist_id,shopping
3		spapplication.php	424	__activate	spdbtable	:: shoppinglist_dbtable_1	insert into t8c7c571facc44aae81d7c9d2cabaf9b9 (shoppinglist_shopperid,shoppinglist_id,shopping

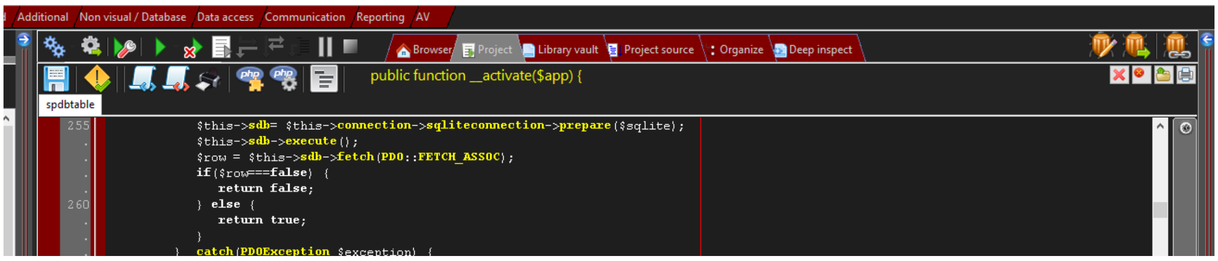
Now you see a list of procedures that are called (in reverse order, most recent on the bottom), and again, you have a button to jump to the file/line that cause the error:



```
            $this->sdb= $this->connection->sqliteconnection->prepare($sqlite);
            $this->sdb->execute();
            $row = $this->sdb->fetch(PDO::FETCH_ASSOC);
            if($row===false) {
               return false;
            } else {
               return true;
            }
         }  catch(PDOException $exception) {
            outputDebugStringOut(__FILE__, __LINE__, $exception->getMessage());
         }
      }

      // if there is no resultset then create a new resultset by
      // selecting data from the original table. Otherwise
      // use the resultset

      public function __activate($app) {
        if($this->active=='Y') {
           if($this->tableName!='') {
              $this-> Open();
           }
        }
```

The next step, is to set a breakpoint here, run the application in debug mode, and after stopping at the breakpoint you can step through the code to find the spot causing the issue. In the following paragraphs we will show you how you can debug your PHP code, by running the application in Debug mode.

Running you application in Debug Mode to debug PHP.



```
            $this->sdb= $this->connection->sqliteconnection->prepare($sqlite);
            $this->sdb->execute();
            $row = $this->sdb->fetch(PDO::FETCH_ASSOC);
            if($row===false) {
               return false;
            } else {
               return true;
            }
         }  catch(PDOException $exception) {
```

When you have opened the PHP editor you will find the debugging buttons in the right upper corner. For PHP debugging the first two toggle buttons are used:

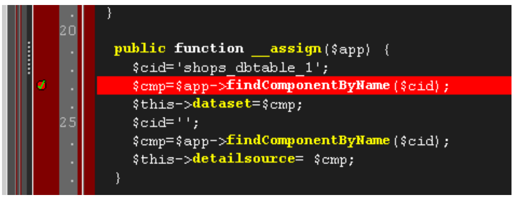
1. If set, when you run the application, then the application will stop at the first line, allowing you to step through your code, and follow the application flow.

2. If set (it will set 1 automatically), when you run the application, then the application will only stop if a set breakpoint is met (this button works in conjunction with the previous button).

If you want to set breakpoints in your generated code, then you have to open the module from the generated apps tab. In the following sample we opened shops_form_1 and have set a breakpoint at line 26.

Setting and resetting breakpoint

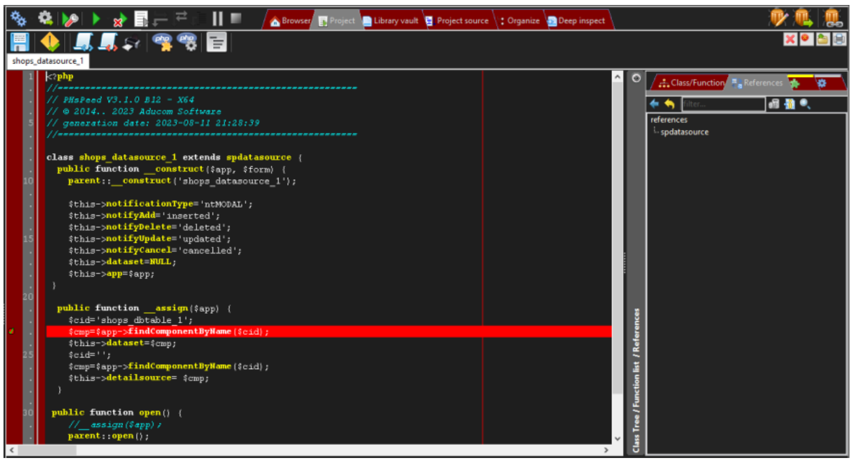
A breakpoint is a row where execution of the program will pause, allowing you to inspect values of variables, or as a starting point to step through your program line-by-line. To set a breakpoint, open the PHP source code file where you want to set the breakpoint and locate the line. Click with the mouse in the first column of the red marked gutter. This will set the breakpoint and shows with a red icon, and the line will turn red.



```
}

public function __assign($app) {
  $cid='shops_dbtable_1';
  $cmp=$app->findComponentByName($cid);
  $this->dataset=$cmp;
  $cid='';
  $cmp=$app->findComponentByName($cid);
  $this->detailsource= $cmp;
}
```

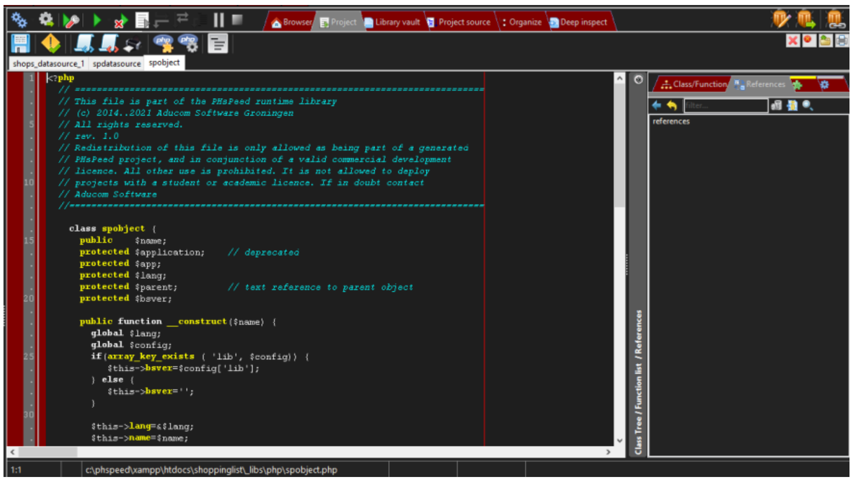
To unset, simply click again on the line in the red gutter area.

NOTE. You can only set breakpoints in the generated code. That might sound obvious, but is easily be overlooked. If you want to set a breakpoint in the runtime then do NOT use the files that are shown in the phspeed runtime section of the project tree, as these files are not executed, but copied into your project. To do this, open the class tree / function / reference on the right side of the editor. Use the reference tab.



```
<?php
//=====================================================
// PHsPeed V3.1.0 B12 - X64
// © 2014.. 2023 Aducom Software
// generation date: 2023-08-11 21:28:39
//=====================================================

class shops_datasource_1 extends spdatasource {
  public function __construct($app, $form) {
    parent::__construct('shops_datasource_1');

    $this->notificationType='ntMODAL';
    $this->notifyAdd='inserted';
    $this->notifyDelete='deleted';
    $this->notifyUpdate='updated';
    $this->notifyCancel='cancelled';
    $this->dataset=NULL;
    $this->app=$app;
  }

  public function __assign($app) {
    $cid='shops_dbtable_1';
    $cmp=$app->findComponentByName($cid);
    $this->dataset=$cmp;
    $cid='';
    $cmp=$app->findComponentByName($cid);
    $this->detailsource= $cmp;
  }

 public function open() {
   //__assign($app);
   parent::open();
```

This form shows the component(s) that the shown component is inherited of. Actually this is a visual way to follow the object tree to the basic object (spobject). To follow the tree, click on the object in the references and it will open. You can continue this until you reach the root object.



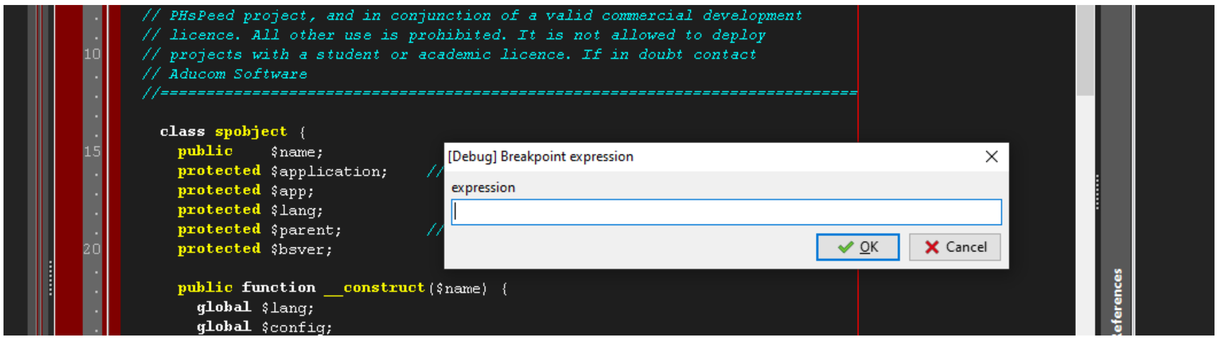
```
<?php
  // ======================================================================
  // This file is part of the PHsPeed runtime library
  // (c) 2014..2021 Aducom Software Groningen
  // All rights reserved.
  // rev. 1.0
  // Redistribution of this file is only allowed as being part of a generated
  // PHsPeed project, and in conjunction of a valid commercial development
  // licence. All other use is prohibited. It is not allowed to deploy
  // projects with a student or academic licence. If in doubt contact
  // Aducom Software
  //======================================================================

   class spobject {
     public    $name;
     protected $application;    // deprecated
     protected $app;
     protected $lang;
     protected $parent;         // text reference to parent object
     protected $bsver;

     public function __construct($name) {
       global $lang;
       global $config;
       if(array_key_exists ( 'lib', $config)) {
          $this->bsver=$config['lib'];
       } else {
          $this->bsver='';
       }

       $this->lang=&$lang;
       $this->name=$name;
```

```
1:1    c:\phspeed\xampp\htdocs\shoppinglist\_libs\php\spobject.php
```

The files that are shown in the editor are the actual files in your project. So you can set breakpoints here.

Setting conditional breakpoints

If you need to set a breakpoint that will pause the program on a certain condition, then hold the shift key while clicking with the mouse on the gutter. Next a dialog will open to enter your expression. This needs to be a valid PHP conditional expression like: $this->ver = '5'



```
// PHsPeed project, and in conjunction of a valid commercial development
// licence. All other use is prohibited. It is not allowed to deploy
// projects with a student or academic licence. If in doubt contact
// Aducom Software
//==========================================================================

  class spobject {
    public     $name;
    protected  $application;
    protected  $app;
    protected  $lang;
    protected  $parent;
    protected  $bsver;

    public function __construct($name) {
      global $lang;
      global $config;
```

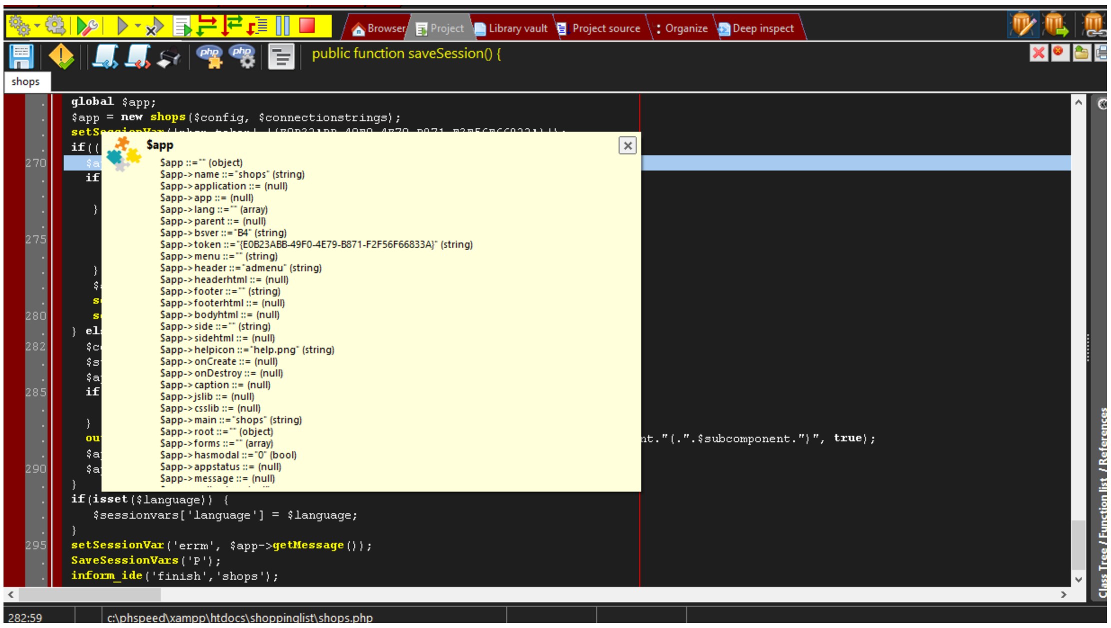
[Debug] Breakpoint expression

expression

Options when the program pauses on a breakpoint (or step)

If the application has stopped at a certain line, then you will have a number of options for debugging:

1. **Inspect values of variables in execution mode**
   When you are stepping through the code or have paused in a breakpoint then you are able to inspect the values of your program. The first option is to set the focus to the editor (click with your mouse in the editor) and then hover above the variable to inspect:



public function saveSession() {

```
global $app;
$app = new shops($config, $connectionstrings);
```

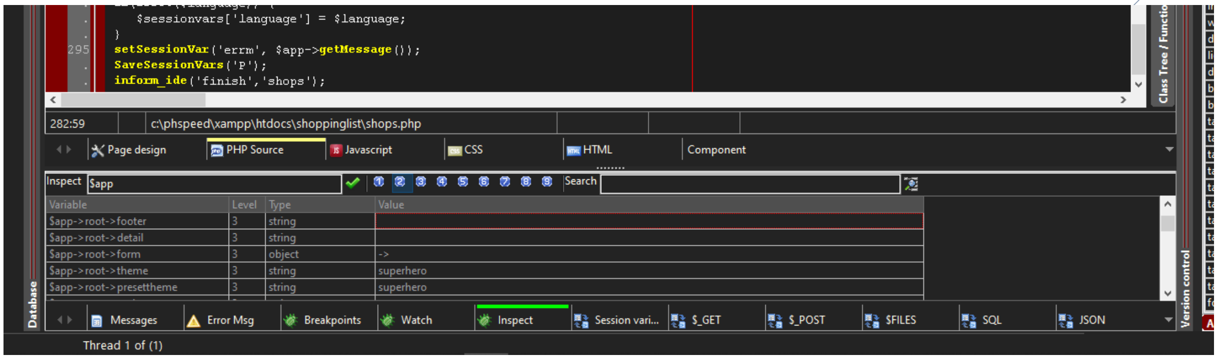
$app

```
$app ::="" (object)
$app->name ::="shops" (string)
$app->application ::= (null)
$app->app ::= (null)
$app->lang ::="" (array)
$app->parent ::= (null)
$app->bsver ::="B4" (string)
$app->token ::="{E0B23ABB-49F0-4E79-B871-F2F56F66833A}" (string)
$app->menu ::="" (string)
$app->header ::="admenu" (string)
$app->headerhtml ::= (null)
$app->footer ::="" (string)
$app->footerhtml ::= (null)
$app->bodyhtml ::= (null)
$app->side ::="" (string)
$app->sidehtml ::= (null)
$app->helpicon ::="help.png" (string)
$app->onCreate ::= (null)
$app->onDestroy ::= (null)
$app->caption ::= (null)
$app->jslib ::= (null)
$app->csslib ::= (null)
$app->main ::="shops" (string)
$app->root ::="" (object)
$app->forms ::="" (array)
$app->hasmodal ::="0" (bool)
$app->appstatus ::= (null)
$app->message ::= (null)
```

```
}
if(isset($language)) {
   $sessionvars['language'] = $language;
}
setSessionVar('errm', $app->getMessage());
SaveSessionVars('P');
inform_ide('finish','shops');
```

282:59   c:\phspeed\xampp\htdocs\shoppinglist\shops.php

   This will give a first look at the values of the fields. Notice that objects and properties are not shown. To inspect these values you need to use the inspect option that you find in the tabs on the bottom of the IDE:

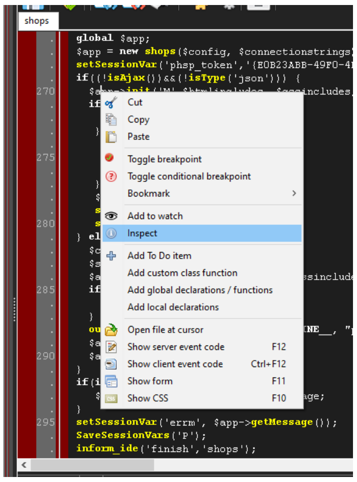


```
    $sessionvars['language'] = $language;
}
setSessionVar('errm', $app->getMessage());
SaveSessionVars('P');
inform_ide('finish','shops');
```

282:59 c:\phspeed\xampp\htdocs\shoppinglist\shops.php

Inspect $app

Variable	Level	Type	Value
$app->root->footer	3	string	
$app->root->detail	3	string	
$app->root->form	3	object	->
$app->root->theme	3	string	superhero
$app->root->presettheme	3	string	superhero



```
global $app;
$app = new shops($config, $connectionstrings
setSessionVar('phsp_token','{E0B23ABB-49F0-4
if((!isAjax())&&(!isType('json'))) {
setSessionVar('errm', $app->getMessage());
SaveSessionVars('P');
inform_ide('finish','shops');
```

shops
Cut
Copy
Paste
Toggle breakpoint
Toggle conditional breakpoint
Bookmark
Add to watch
Inspect
Add To Do item
Add custom class function
Add global declarations / functions
Add local declarations
Open file at cursor
Show server event code F12
Show client event code Ctrl+F12
Show form F11
Show CSS F10

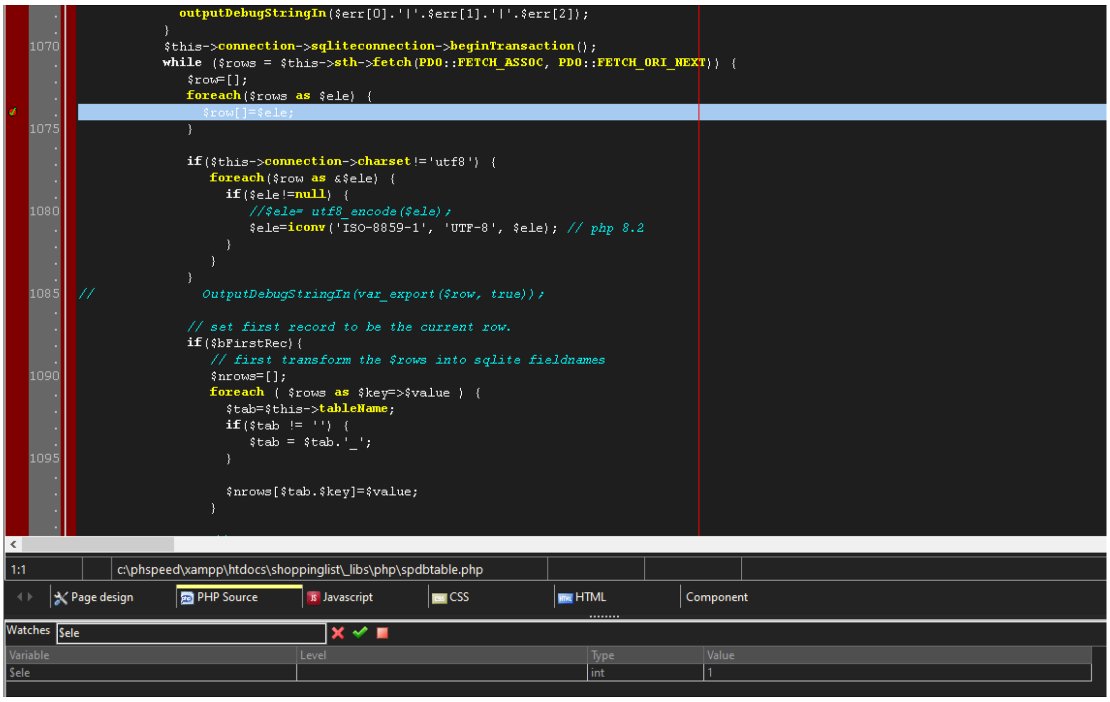
You can enter the name of the variable you want to inspect or point to the variable in the editor and use the right mouse click to move the variable to the inspector.

The depth of the inspector can be set by clicking on the level indicators (1-9). Keep in mind that the deeper you inspect, the more data will be displayed. To help you locate the variable you can search in the search box. This searches in the shown result.

If the window is too small to show the full content, then double click in the value column and the data will be shown in a popup box.

2. Watch variables

If you step through your code, i.e. in iterations, then it can be handy to follow the state of a certain variable during execution. You can do so by adding the variable to the watch list. This window is refreshed on each step of execution.



```
            outputDebugStringIn($err[0].'|'.$err[1].'|'.$err[2]);
        }
        $this->connection->sqliteconnection->beginTransaction();
        while ($rows = $this->sth->fetch(PDO::FETCH_ASSOC, PDO::FETCH_ORI_NEXT)) {
           $row=[];
           foreach($rows as $ele) {
             $row[]=$ele;
           }

           if($this->connection->charset!='utf8') {
              foreach($row as &$ele) {
                if($ele!=null) {
                   //$ele= utf8_encode($ele);
                   $ele=iconv('ISO-8859-1', 'UTF-8', $ele); // php 8.2
                }
              }
           }
//            OutputDebugStringIn(var_export($row, true));

           // set first record to be the current row.
           if($bFirstRec){
              // first transform the $rows into sqlite fieldnames
              $nrows=[];
              foreach ( $rows as $key=>$value ) {
                $tab=$this->tableName;
                if($tab != '') {
                   $tab = $tab.'_';
                }

                $nrows[$tab.$key]=$value;
              }
```

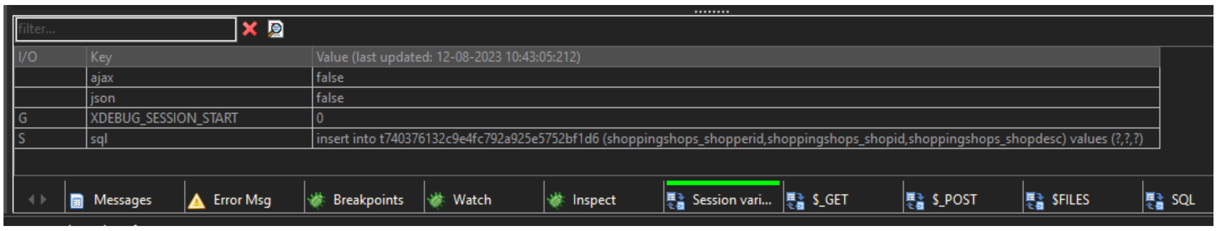
1:1 c:\phspeed\xampp\htdocs\shoppinglist_libs\php\spdbtable.php

Watches $ele

Variable	Level	Type	Value
$ele		int	1

You can only watch 'simple variables', if you need to look into objects or arrays, you need to use the inspector. You can add more variables to watch, but keep in mind that the content is refreshed on each step of execution, increasing execution time.

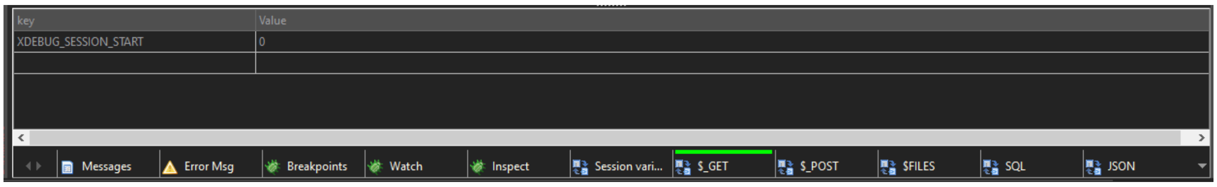
3. **Investigate session variable values**
   PHsPeed maintains variables in the session space. This means that your webform will not contain application data other than the necessary fields on the form, csrf token etc. When the execution of a PHP module ends, then the data is saved in the session space, and retrieved when you open a new session. The content of the area can be viewed in the session variable space:



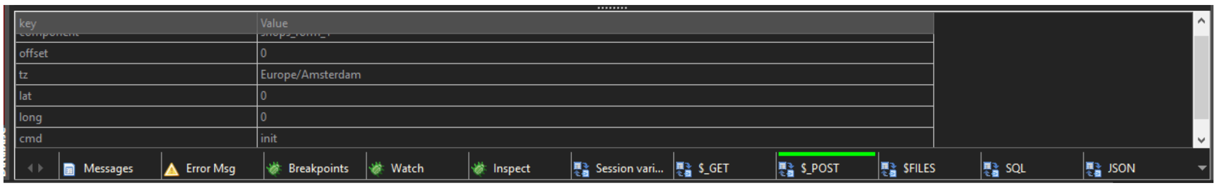
I/O	Key	Value (last updated: 12-08-2023 10:43:05:212)
	ajax	false
	json	false
G	XDEBUG_SESSION_START	0
S	sql	insert into t740376132c9e4fc792a925e5752bf1d6 (shoppingshops_shopperid,shoppingshops_shopid,shoppingshops_shopdesc) values (?,?,?)

   This area can be quite large.

4. **Inspect $_GET, $POST and $_FILE arrays**
   PHP works with $_GET (url variables), $_POST (form submits / ajax) and $_FILE (uploads) variables. You can inspect those variables in the appropriate tabs:



key	Value
XDEBUG_SESSION_START	0

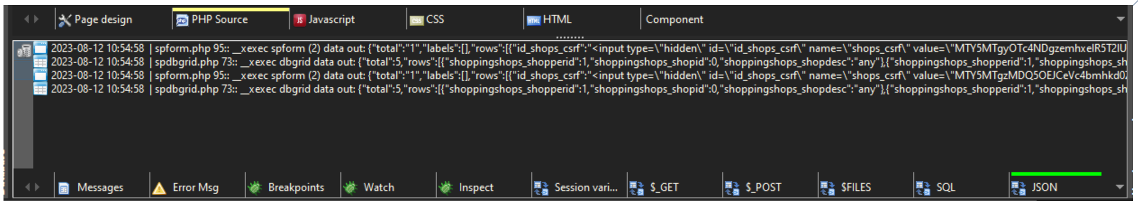


key	Value
offset	0
tz	Europe/Amsterdam
lat	0
long	0
cmd	init

   Hint: NEVER use the $_POST and $_GET array directly unless you have to. Use the rtrGetVar('variable', '') procedure instead. In the first place, this routine takes care of injection problems and second, it will return a default value if the variable is not found in the input space.

5. **Inspect SQL trace**
   This allows you to follow the executed SQL statements. PDO does not enable you to see the SQL statements including the ??? values. The use is explained in the top of this document.

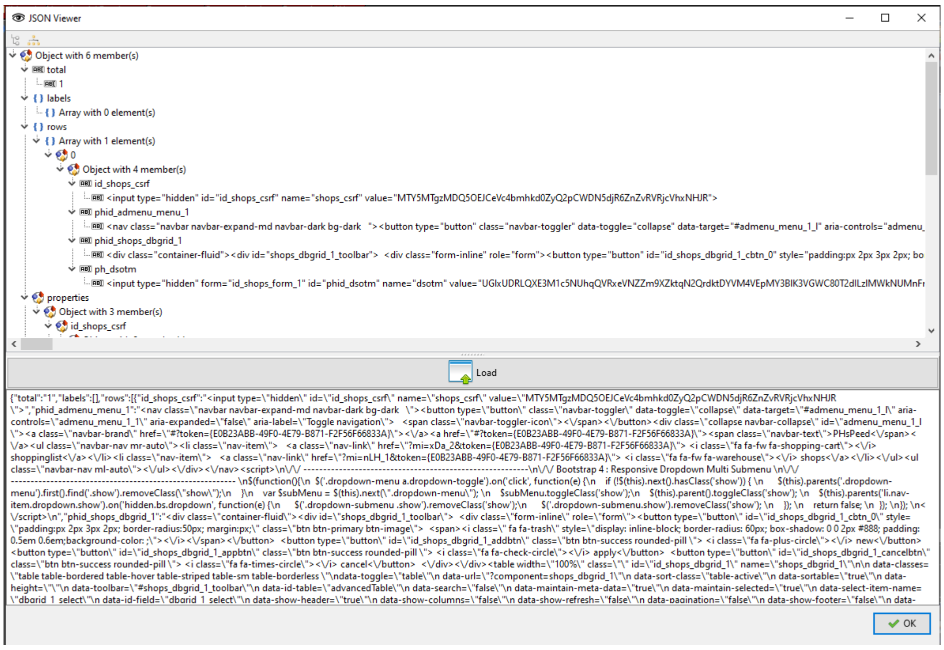
6. **Inspect JSON messaging between the form and application**
   Explained in an earlier document we showed the execution flow of a PHsPeed application. In the first stage, only the template is sent, next an ajax event will retrieve the rendered data and puts that in the form. The messages that are involved here can be inspected in the JSON tab:



```
2023-08-12 10:54:58 | spform.php 95:: _xexec spform (2) data out: {"total":"1","labels":[],"rows":[{"id_shops_csrf":"<input type=\"hidden\" id=\"id_shops_csrf\" name=\"shops_csrf\" value=\"MTY5MTgyOTc4NDgzemhxelR5T2IU
2023-08-12 10:54:58 | spdbgrid.php 73:: _xexec dbgrid data out: {"total":5,"rows":[{"shoppingshops_shopperid":1,"shoppingshops_shopid":0,"shoppingshops_shopdesc":"any"},{"shoppingshops_shopperid":1,"shoppingshops_sh
2023-08-12 10:54:58 | spform.php 95:: _xexec spform (2) data out: {"total":"1","labels":[],"rows":[{"id_shops_csrf":"<input type=\"hidden\" id=\"id_shops_csrf\" name=\"shops_csrf\" value=\"MTY5MTgzMDQ5OEJCeVc4bmhkd0
2023-08-12 10:54:58 | spdbgrid.php 73:: _xexec dbgrid data out: {"total":5,"rows":[{"shoppingshops_shopperid":1,"shoppingshops_shopid":0,"shoppingshops_shopdesc":"any"},{"shoppingshops_shopperid":1,"shoppingshops_sh
```

For a deeper insight, double click on a row. Be careful with grids, as it will show all the records. On huge arrays it could take a massive amount of time to display.

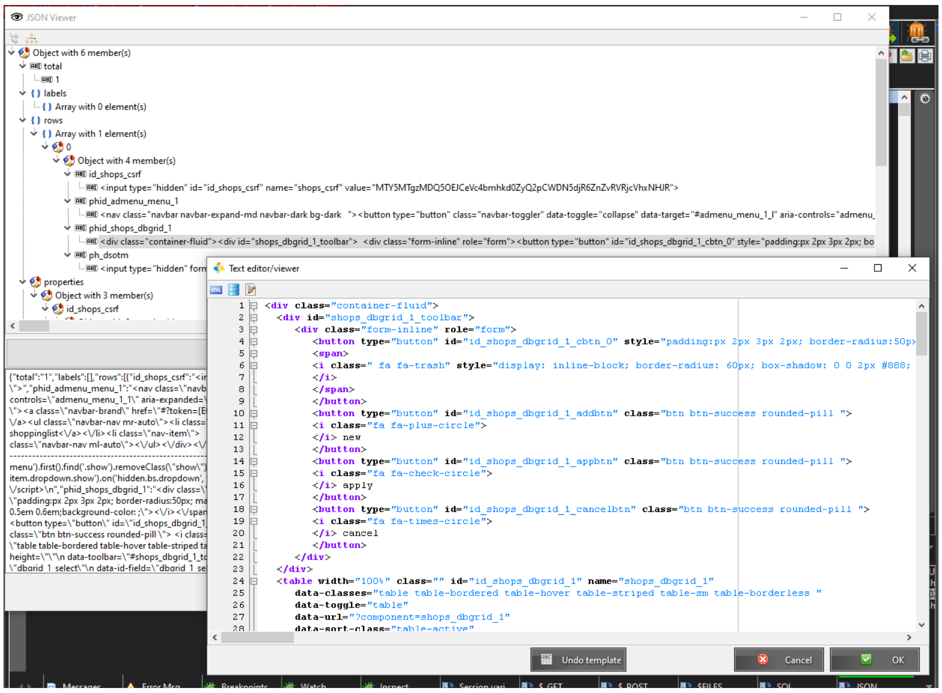
Double clicking on the form will open the JSON viewer:



```
Object with 6 member(s)
  total
    1
  labels
    Array with 0 element(s)
  rows
    Array with 1 element(s)
      0
        Object with 4 member(s)
          id_shops_csrf
            <input type="hidden" id="id_shops_csrf" name="shops_csrf" value="MTY5MTgzMDQ5OEJCeVc4bmhkd0ZyQ2pCWDN5djR6ZnZvRVRjcVhxNHJR">
          phid_admenu_menu_1
            <nav class="navbar navbar-expand-md navbar-dark bg-dark   "><button type="button" class="navbar-toggler" data-toggle="collapse" data-target="#admenu_menu_1_l" aria-controls="admenu_
          phid_shops_dbgrid_1
            <div class="container-fluid"><div id="shops_dbgrid_1_toolbar">  <div class="form-inline" role="form"><button type="button" id="id_shops_dbgrid_1_cbtn_0" style="padding:px 2px 3px 2px; bo
          ph_dsotm
            <input type="hidden" form="id_shops_form_1" id="phid_dsotm" name="dsotm" value="UGlxUDRLQXE3M1c5NUhqQVRxeVNZZm9XZktqN2QrdktDYVM4VEpMY3BIK3VGWC80T2dlLzlMWkNUMnFr
  properties
    Object with 3 member(s)
      id_shops_csrf
```

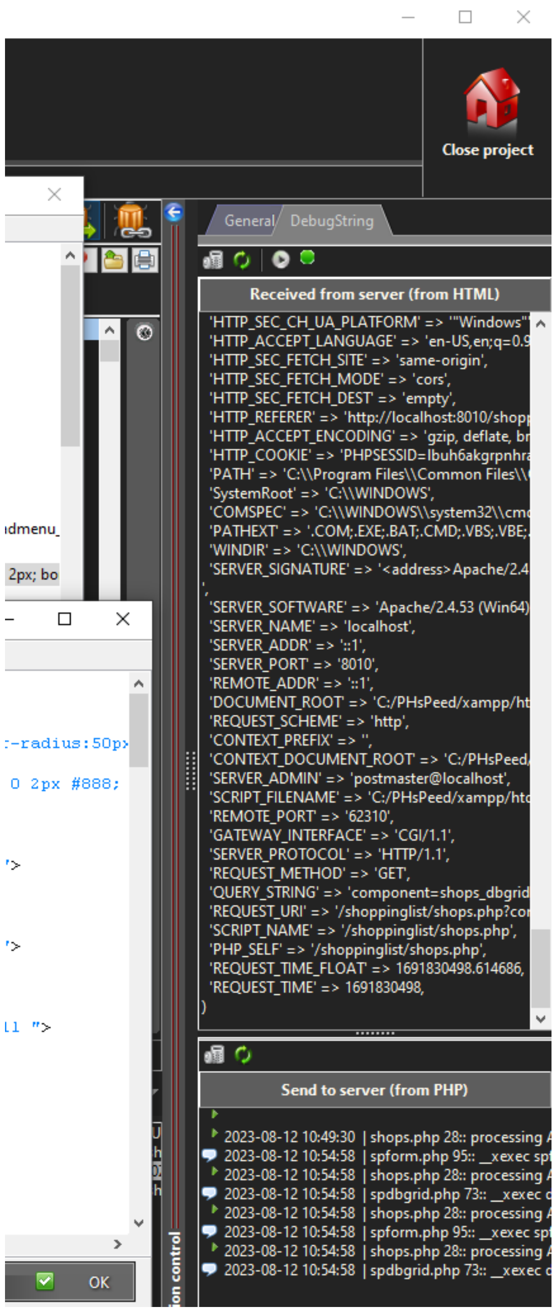
{"total":"1","labels":[],"rows":[{"id_shops_csrf":"<input type=\"hidden\" id=\"id_shops_csrf\" name=\"shops_csrf\" value=\"MTY5MTgzMDQ5OEJCeVc4bmhkd0ZyQ2pCWDN5djR6ZnZvRVRjcVhxNHJR
\">","phid_admenu_menu_1":"<nav class=\"navbar navbar-expand-md navbar-dark bg-dark \"><button type=\"button\" class=\"navbar-toggler\" data-toggle=\"collapse\" data-target=\"#admenu_menu_1_l\" aria-
controls=\"admenu_menu_1_1\" aria-expanded=\"false\" aria-label=\"Toggle navigation\">   <span class=\"navbar-toggler-icon\"><\/span><\/button><div class=\"collapse navbar-collapse\" id=\"admenu_menu_1_l
\"><a class=\"navbar-brand\" href=\"#?token={E0B23ABB-49F0-4E79-B871-F2F56F66833A}\"><\/a><a href=\"#?token={E0B23ABB-49F0-4E79-B871-F2F56F66833A}\"><span class=\"navbar-text\">PHsPeed<\/span><
\/a><ul class=\"navbar-nav mr-auto\"><li class=\"nav-item\">   <a class=\"nav-link\" href=\"?mi=xDa_2&token={E0B23ABB-49F0-4E79-B871-F2F56F66833A}\"> <i class=\"fa fa-fw fa-shopping-cart\"><\/i>
shoppinglist<\/a><\/li><li class=\"nav-item\">   <a class=\"nav-link\" href=\"?mi=nLH_1&token={E0B23ABB-49F0-4E79-B871-F2F56F66833A}\"> <i class=\"fa fa-fw fa-warehouse\"><\/i> shops<\/a><\/li><\/ul><ul
class=\"navbar-nav ml-auto\"><\/ul><\/div><\/nav><script>\n\/\/ ---\n\/\/ Bootstrap 4 : Responsive Dropdown Multi Submenu \n\/\/
-- \n$(function(){\n $('.dropdown-menu a.dropdown-toggle').on('click', function(e) {\n   if (!$(this).next().hasClass('show')) { \n $(this).parents('.dropdown-
menu').first().find('.show').removeClass(\"show\");\n   }\n   var $subMenu = $(this).next(\".dropdown-menu\"); \n $subMenu.toggleClass('show');\n $(this).parent().toggleClass('show'); \n $(this).parents('li.nav-
item.dropdown.show').on('hidden.bs.dropdown', function(e) {\n $('.dropdown-submenu .show').removeClass('show');\n $('.dropdown-submenu.show').removeClass('show'); \n   }); \n   return false; \n }); \n}); \n<
\/script>\n","phid_shops_dbgrid_1":"<div class=\"container-fluid\"><div id=\"shops_dbgrid_1_toolbar\"> <div class=\"form-inline\" role=\"form\"><button type=\"button\" id=\"id_shops_dbgrid_1_cbtn_0\" style=
\"padding:px 2px 3px 2px; border-radius:50px; margin:px;\" class=\"btn btn-primary btn-image\"> <span><i class=\" fa fa-trash\" style=\"display: inline-block; border-radius: 60px; box-shadow: 0 0 2px #888; padding:
0.5em 0.6em;background-color: ;\"><\/i><\/span><\/button> <button type=\"button\" id=\"id_shops_dbgrid_1_addbtn\" class=\"btn btn-success rounded-pill \"> <i class=\"fa fa-plus-circle\"><\/i> new<\/button>
<button type=\"button\" id=\"id_shops_dbgrid_1_appbtn\" class=\"btn btn-success rounded-pill \"> <i class=\"fa fa-check-circle\"><\/i> apply<\/button> <button type=\"button\" id=\"id_shops_dbgrid_1_cancelbtn\"
class=\"btn btn-success rounded-pill \"> <i class=\"fa fa-times-circle\"><\/i> cancel<\/button> <\/div><\/div><table width=\"100%\" class=\"\" id=\"id_shops_dbgrid_1\" name=\"shops_dbgrid_1\"\n\n data-classes=
\"table table-bordered table-hover table-striped table-sm table-borderless \"\ndata-toggle=\"table\"\n data-url=\"?component=shops_dbgrid_1\"\n data-sort-class=\"table-active\"\n data-sortable=\"true\"\n data-
height=\"\"\n data-toolbar=\"#shops_dbgrid_1_toolbar\"\n data-id-table=\"advancedTable\"\n data-search=\"false\"\n data-maintain-meta-data=\"true\"\n data-maintain-selected=\"true\"\n data-select-item-name=
\"dbgrid_1_select\"\n data-id-field=\"dbgrid_1_select\"\n data-show-header=\"true\"\n data-show-columns=\"false\"\n data-show-refresh=\"false\"\n data-pagination=\"false\"\n data-show-footer=\"false\"\n data-

To see the rendered text in a better format, double click on the row of choices like:



```
<div class="container-fluid">
  <div id="shops_dbgrid_1_toolbar">
     <div class="form-inline" role="form">
        <button type="button" id="id_shops_dbgrid_1_cbtn_0" style="padding:px 2px 3px 2px; border-radius:50p>
        <span>
        <i class=" fa fa-trash" style="display: inline-block; border-radius: 60px; box-shadow: 0 0 2px #888;
        </i>
        </span>
        </button>
        <button type="button" id="id_shops_dbgrid_1_addbtn" class="btn btn-success rounded-pill ">
        <i class="fa fa-plus-circle">
        </i> new
        </button>
        <button type="button" id="id_shops_dbgrid_1_appbtn" class="btn btn-success rounded-pill ">
        <i class="fa fa-check-circle">
        </i> apply
        </button>
        <button type="button" id="id_shops_dbgrid_1_cancelbtn" class="btn btn-success rounded-pill ">
        <i class="fa fa-times-circle">
        </i> cancel
        </button>
     </div>
  </div>
  <table width="100%" class="" id="id_shops_dbgrid_1" name="shops_dbgrid_1"
     data-classes="table table-bordered table-hover table-striped table-sm table-borderless "
     data-toggle="table"
     data-url="?component=shops_dbgrid_1"
     data-sort-class="table-active"
```

7. **Inspect raw messages between the application and the web server**
   PHP sends data to the web server. This data and the raw data that is sent back to the application can be viewed in the DebugString tab. The detail can be set in the run/debug options. The more detail you show, the more time it will take. On slower systems it might be handy to enable/disable options.

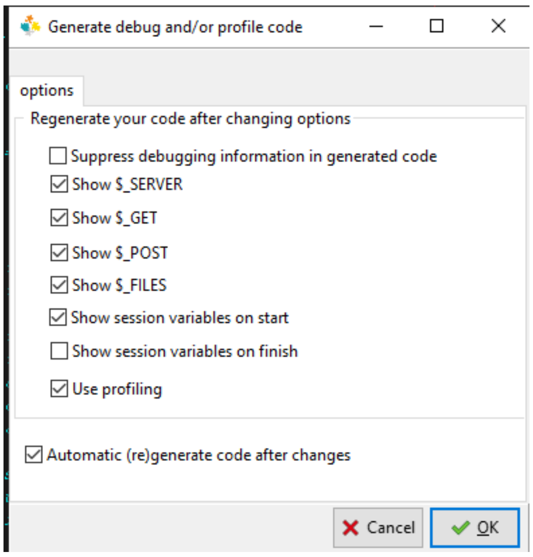


Received from server (from HTML)

```
 'HTTP_SEC_CH_UA_PLATFORM' => '"Windows"
 'HTTP_ACCEPT_LANGUAGE' => 'en-US,en;q=0.9
 'HTTP_SEC_FETCH_SITE' => 'same-origin',
 'HTTP_SEC_FETCH_MODE' => 'cors',
 'HTTP_SEC_FETCH_DEST' => 'empty',
 'HTTP_REFERER' => 'http://localhost:8010/shopp
 'HTTP_ACCEPT_ENCODING' => 'gzip, deflate, br
 'HTTP_COOKIE' => 'PHPSESSID=lbuh6akgrpnhra
 'PATH' => 'C:\\Program Files\\Common Files\\
 'SystemRoot' => 'C:\\WINDOWS',
 'COMSPEC' => 'C:\\WINDOWS\\system32\\cmd
 'PATHEXT' => '.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.
 'WINDIR' => 'C:\\WINDOWS',
 'SERVER_SIGNATURE' => '<address>Apache/2.4
',
 'SERVER_SOFTWARE' => 'Apache/2.4.53 (Win64)
 'SERVER_NAME' => 'localhost',
 'SERVER_ADDR' => '::1',
 'SERVER_PORT' => '8010',
 'REMOTE_ADDR' => '::1',
 'DOCUMENT_ROOT' => 'C:/PHsPeed/xampp/ht
 'REQUEST_SCHEME' => 'http',
 'CONTEXT_PREFIX' => '',
 'CONTEXT_DOCUMENT_ROOT' => 'C:/PHsPeed/
 'SERVER_ADMIN' => 'postmaster@localhost',
 'SCRIPT_FILENAME' => 'C:/PHsPeed/xampp/htc
 'REMOTE_PORT' => '62310',
 'GATEWAY_INTERFACE' => 'CGI/1.1',
 'SERVER_PROTOCOL' => 'HTTP/1.1',
 'REQUEST_METHOD' => 'GET',
 'QUERY_STRING' => 'component=shops_dbgrid
 'REQUEST_URI' => '/shoppinglist/shops.php?cor
 'SCRIPT_NAME' => '/shoppinglist/shops.php',
 'PHP_SELF' => '/shoppinglist/shops.php',
 'REQUEST_TIME_FLOAT' => 1691830498.614686,
 'REQUEST_TIME' => 1691830498,
)
```

Send to server (from PHP)

```
2023-08-12 10:49:30 | shops.php 28:: processing A
2023-08-12 10:54:58 | spform.php 95:: __xexec spf
2023-08-12 10:54:58 | shops.php 28:: processing A
2023-08-12 10:54:58 | spdbgrid.php 73:: __xexec d
2023-08-12 10:54:58 | shops.php 28:: processing A
2023-08-12 10:54:58 | spform.php 95:: __xexec spf
2023-08-12 10:54:58 | shops.php 28:: processing A
2023-08-12 10:54:58 | spdbgrid.php 73:: __xexec d
```



Generate debug and/or profile code

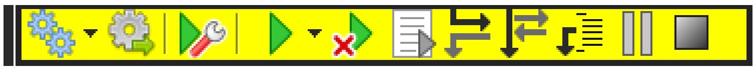
options

Regenerate your code after changing options

- ☐ Suppress debugging information in generated code
- ☑ Show $_SERVER
- ☑ Show $_GET
- ☑ Show $_POST
- ☑ Show $_FILES
- ☑ Show session variables on start
- ☐ Show session variables on finish
- ☑ Use profiling

☑ Automatic (re)generate code after changes

8. Step through your code
   Once your application is pause, you have several options to continue.



   You can use the mouse and the buttons above, but more convenient is to use the function keys:

   F7 : Step Into

   If the line is a function call then hitting F7 will enter this function. If that function is in another PHP file, then this file will be opened in a new tab. Otherwise it will execute the line and stops at the next.

   F8 : StepOver

   If the line is a function call then hitting F7 will enter this function. If that function is in another PHP file, then this file will be opened in a new tab.

   F9 : Continue

   If you want to continue to the next time a breakpoint is hit, or finish the application then use the continue key.

Notes on the debugger.

PHsPeed contains a client for XDebug. This debugger is installed in the PHP versions that comes with PHsPeed. If you use your own environment, then you need to install XDebug in order to have the debugger be able to work.