



# Zero-configuration Web Application Debugging with Xdebug and PhpStorm

This tutorial describes how to get started with PHP web application debugging in PhpStorm IDE using Xdebug. Configuration for Zend Debugger is very similar (just follow [Zend Debugger installation guide](#) and proceed from [step 2](#) of this tutorial).

- 1. [Install Xdebug](#)
- 2. [Prepare PhpStorm](#)
- 3. [Set a breakpoint in the source code](#)
- 4. [Activate debugger on server](#)
- 5. [Start a debug session in browser](#)
- 6. [Reload the current page](#)
- 7. [Set initial path mappings](#)
- 8. [Debug!](#)
- [Troubleshooting](#)

## 1. Install Xdebug

To use Xdebug with PhpStorm for debugging PHP applications, you need to have a PHP development environment configured with Xdebug extension installed. This task is beyond PhpStorm’s control. More information on configuring PHP development environment can be found in [our web help](#).

Make sure that you have Xdebug installed properly. Please note that your Xdebug version should fit the PHP version you have (mind the PHP version number and thread-safe/non-thread-safe option) and installed as *zend\_extension*. Don’t forget to set the *xdebug.remote\_enable* setting to *1* in *php.ini* file. Detailed information on installing and configuring Xdebug can be found [here](#). You can download the latest version of the Xdebug plugin at its [official website](#). Recommended stable versions are 2.2.1 or 2.1.3.

You can check the configuration by executing *php --version* in the terminal

```
localhost:bin users$ /Applications/Mamp/bin/php/php5.4.10/bin/php --version
PHP 5.4.10 (cli) (built: Jan 21 2013 15:12:32)
Copyright (c) 1997-2012 The PHP Group
Zend Engine v2.4.0, Copyright (c) 1998-2012 Zend Technologies
    with XCache v2.0.1, Copyright (c) 2005-2012, by m0o
    with Xdebug v2.2.1, Copyright (c) 2002-2012, by Derick Rethans
localhost:bin users$
```

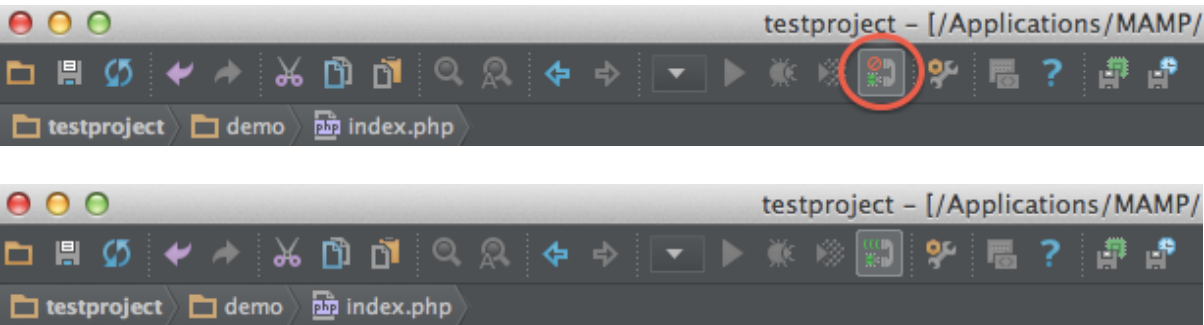
Alternatively, execute a page with *phpinfo()*; PHP function and make sure that you have Xdebug extension enabled. The xdebug section of the document with the installed version should be available.

xdebug

xdebug support	enabled
Version	2.2.1
IDE Key	PHPSTORM

## 2. Prepare PhpStorm

Toggle the “Listen debugger connections” button. Don’t use any run configurations.



## 3. Set a breakpoint in the source code

```
$smarty->assign("Name","Fred Irving Johnathan Bradley Peppergill",true);
$smarty->assign("FirstName",array("John","Mary","James","Henry"));
$smarty->assign("LastName",array("Doe","Smith","Johnson","Case"));
$smarty->assign("Class",array(array("A","B","C","D"), array("E", "F", "G", "H"),
    array("I", "J", "K", "L"), array("M", "N", "O", "P")));
$smarty->assign("contacts", array(array("phone" => "1", "fax" => "2", "cell" => "3"),
    array("phone" => "555-4444", "fax" => "555-3333", "cell" => "760-1234")));

$smarty->assign("option_values", array("NY","NE","KS","IA","OK","TX"));
$smarty->assign("option_output", array("New York","Nebraska","Kansas","Iowa","Oklahoma"));
$smarty->assign("option_selected", "NE");
```

Apart from setting the breakpoint manually, you can use the option *Run | Break at first line in PHP scripts* to have an automatic breakpoint at the first line of every PHP script you debug.

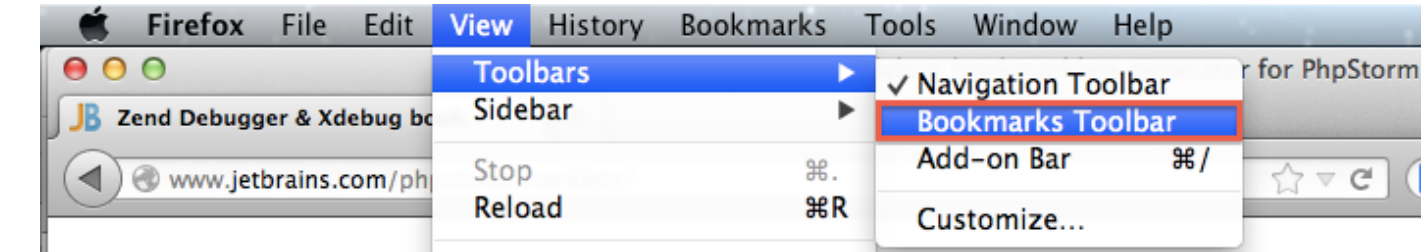
## 4. Activate debugger on server

In order to activate the debugger, you need to set a special GET/POST or COOKIE parameter ([click here for details](#)). You can do it manually, but it is much more convenient to use one a special tool such as browser toolbar or bookmarklet for that.

Try our [PhpStorm bookmarklets generator](#) and get bookmarklets which will work in any modern browser and allow you to start/stop a debugging session by controlling the Xdebug cookie.

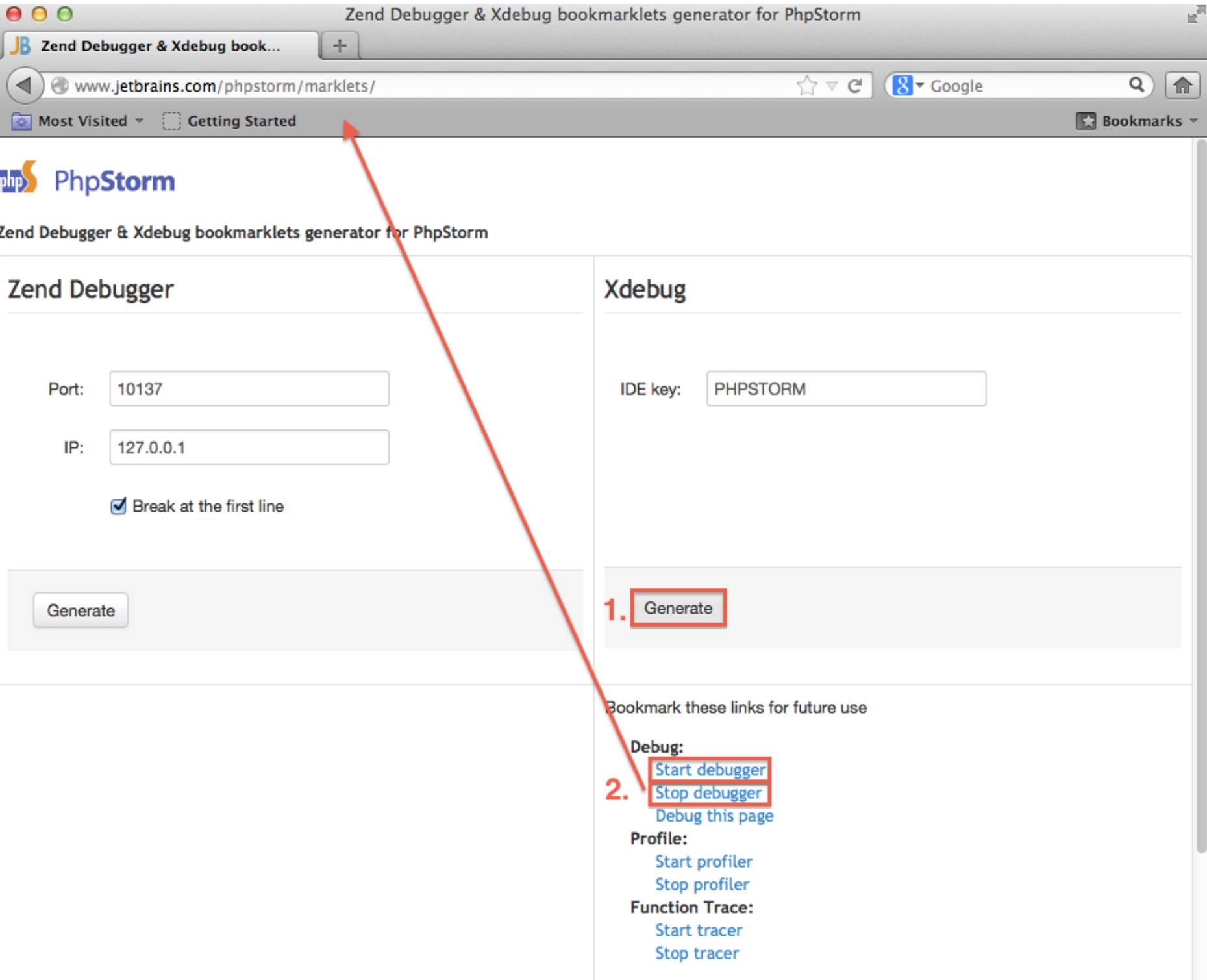
- ▶ How to add "Start/Stop Debugging" bookmarklets to your browser bookmarks toolbar  
*Names of menus and options may vary depending on your Internet browser. This section uses Firefox.*

### 4.1. Enable your Bookmarks Toolbar (View | Toolbars | Bookmarks Toolbar)

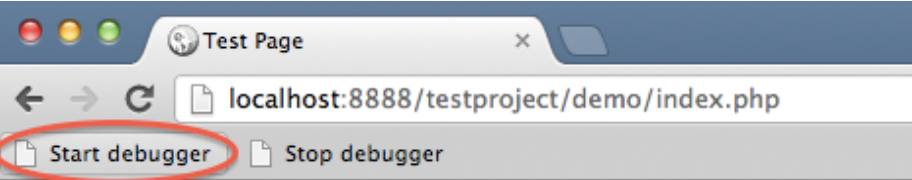


### 4.2. Go to <http://www.jetbrains.com/phpstorm/marklets/>

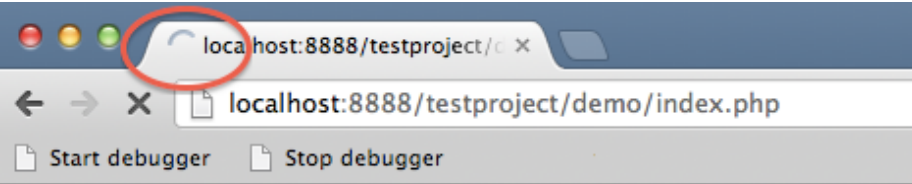
### 4.3. Click the **Generate** button under the **Xdebug** section, then drag the links **"Start debugger"** and **"Stop debugger"** from the **Debug** section that appears to the *Bookmarks Toolbar*.



### 5. Start a debug session in browser

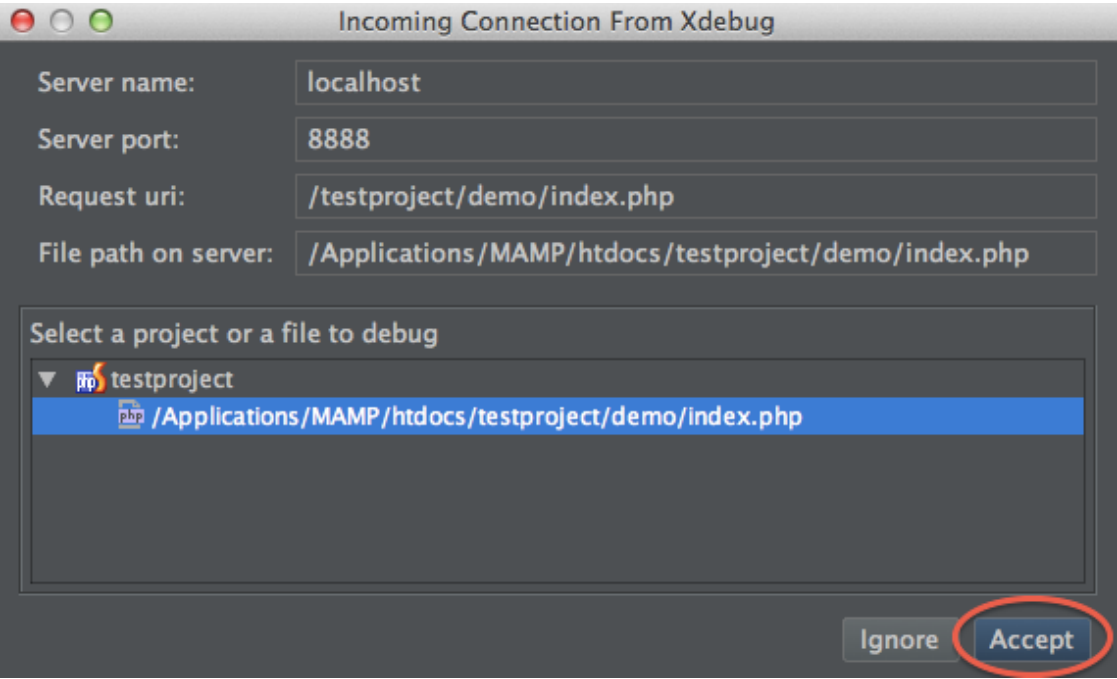


### 6. Reload the current page



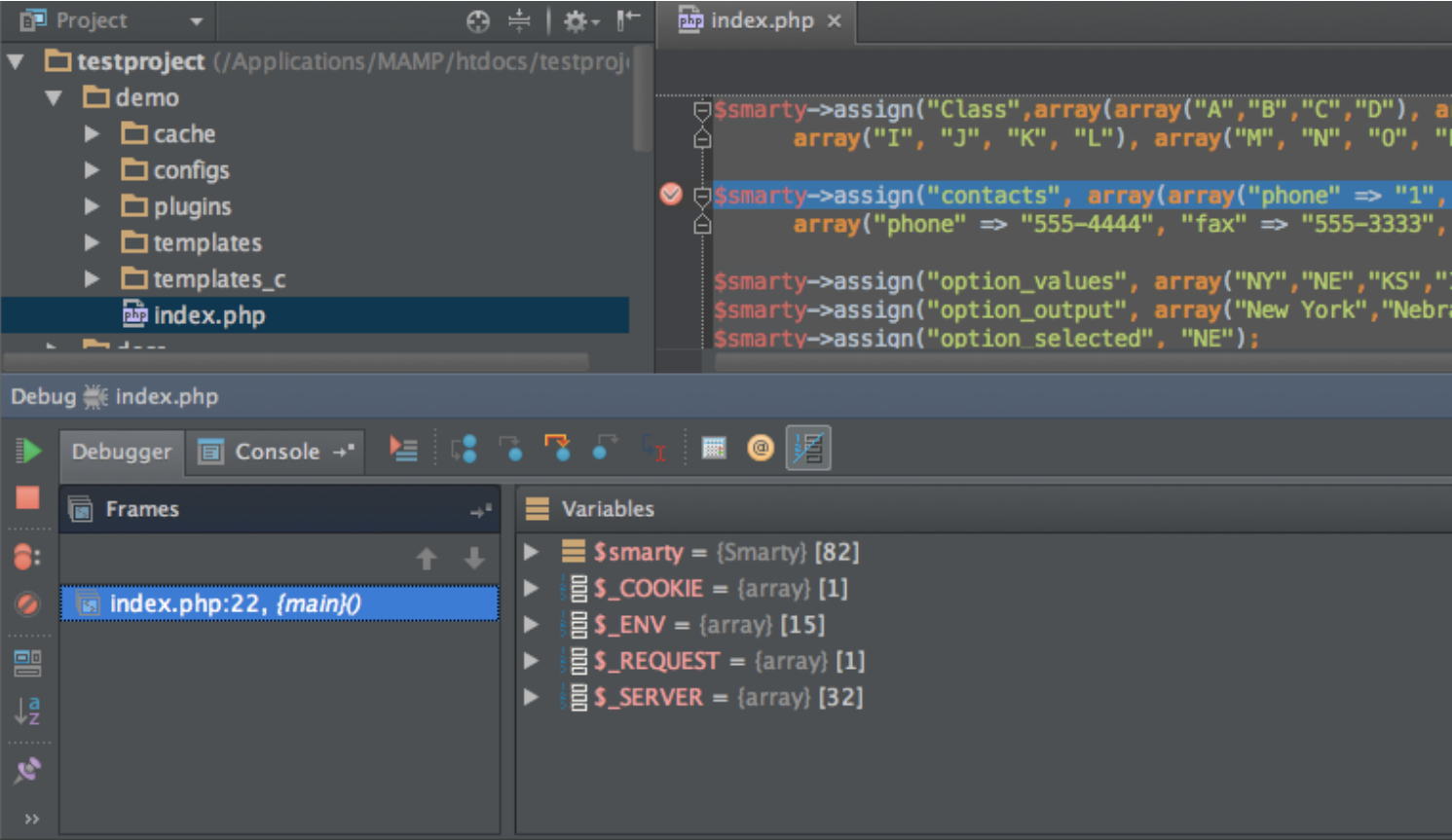
### 7. Set initial path mappings

Switch to PhpStorm and set the initial path mappings by selecting the path to the local copy of the script that you are debugging. These paths are similar if you are editing files directly on the server. The “Incoming Connection” dialog should appear automatically.



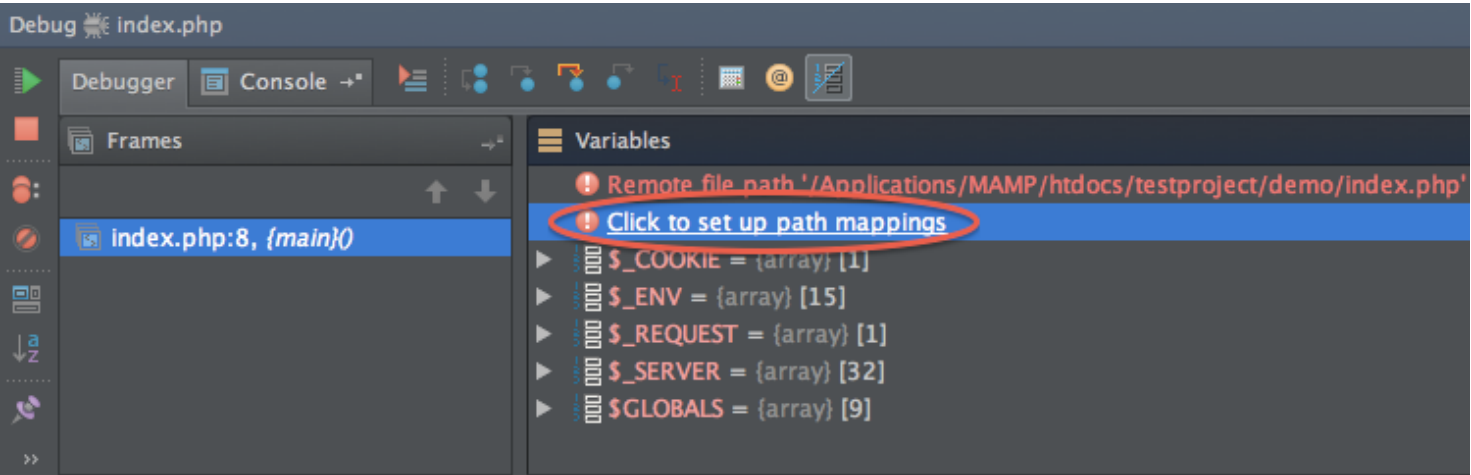
## 8. Debug!

After reaching the breakpoint the debugger is suspended. Now let’s investigate the application.



## Troubleshooting

In some cases you may get the error message “Remote file path ‘path/to/script/on/the/server.php’ is not mapped to any file path in project” or “The script ‘path/to/script/on/the/server.php’ is outside the project.” This means that PhpStorm is not sure which local file corresponds to the specified remote file path.



You can solve this problem quickly. Just set up the necessary path mappings by clicking the relevant link.

Resolve Path Mapping Problem

File path on server:

/Applications/MAMP/htdocs/testproject/demo/index.php

File path in project:

undefined

☒ Use path mappings (select if the server is remote or symlinks are used)

File/Directory	Absolute path on the server
▼ Project files	
▼ /Applications/MAMP/htdocs/testproject	
▶ .idea	/Applications/MAMP/htdocs/testproject/demo
▶ demo	/Applications/MAMP/htdocs/testproject
▶ docs	/Applications/MAMP/htdocs
▶ img	/Applications/MAMP
▶ js	/Applications
▶ less	
▶ libs	
.DS_Store	
.gitignore	
.travis.yml	
CONTRIBUTING.md	

⚡ Remote file path '/Applic...object/demo/index.php' is not mapped to any file path in project

Cancel

OK