Debugging PHP and JavaScript code at the same time in PhpStorm

This tutorial gives an overview of debugging PHP and JavaScript code simultaneously from within PhpStorm. It covers setting up the Run/Debug Configuration to start the two debuggers at once so we can step into our PHP and JavaScript code at the same time.

Web applications typically consist of both PHP and JavaScript code. PHP code will run on the server side, JavaScript will run in the browser. With PhpStorm, we can easily debug the PHP code to inspect what is happening on the server, modify variables and so on. We can also debug the JavaScript running in the browser by starting a JavaScript debugging session from our IDE. But what if we want to do both at the same time? The answer would be to start two debugging sessions at once. Let's see how we can do that!

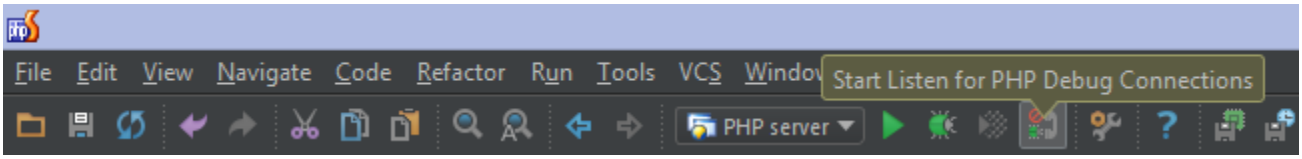
Prerequisites

The following prerequisites will have to be in place for debugging PHP and JavaScript applications:

- Xdebug or Zend Debugger should be installed and configured.
  - Xdebug Installation Guide
  - Zend Debugger Installation Guide
- Install PhpStorm debugger bookmarklets or one of the Browser Debugging Extensions
- Install the Chrome browser extension as outlined in Live Edit in PhpStorm

1. Listen for PHP debug connections

In PhpStorm, toggle the Listen for PHP Debug Connections button in the toolbar. Alternatively use the Run | Start listen for PHP Debug Connections menu.



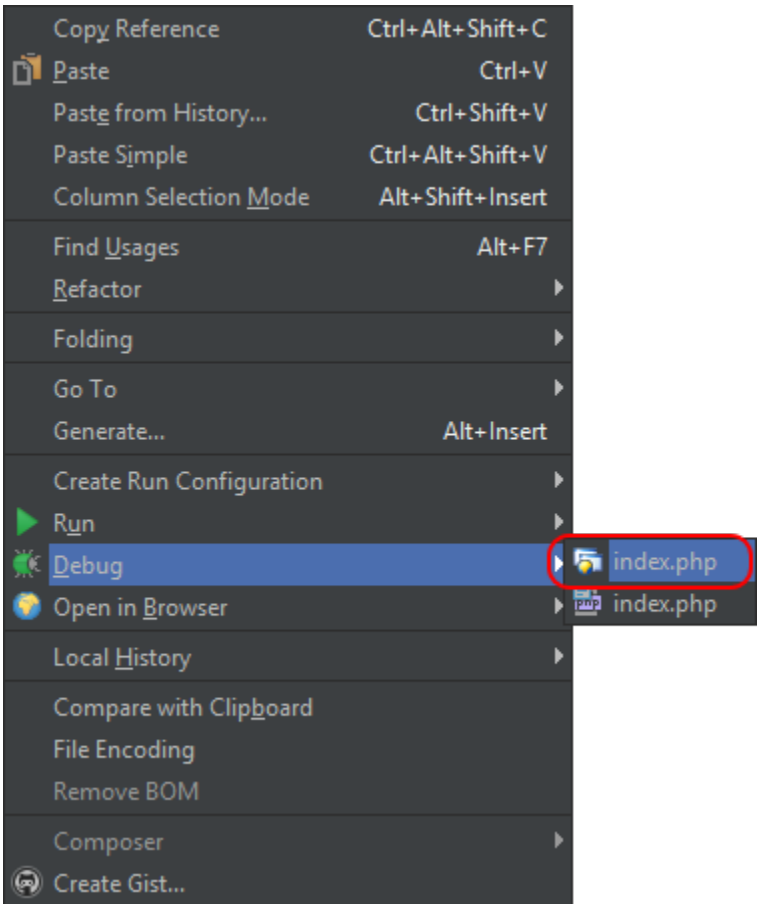
We do not have to set up any Run/Debug Configurations for this. As outlined in Zero-configuration Web Application Debugging with Xdebug and PhpStorm, Listen for PHP Debug Connections will make PhpStorm accept incoming debugger connections initiated by Xdebug or Zend Debugger.

2. Start the JavaScript debugger

Depending on preference and/or application requirements, we can use PhpStorm's built-in webserver to run our application locally, or make use of any other webserver running locally or on a remote machine.

2.1. (option 1) Using the built-in webserver

The JavaScript debugger in PhpStorm can started from the editor or from the Project tool window, using the Debug | <filename> context menu. If the selected file is a PHP file, two entries will be available. It is important to select the first one in this case, which will start the JavaScript debugger.

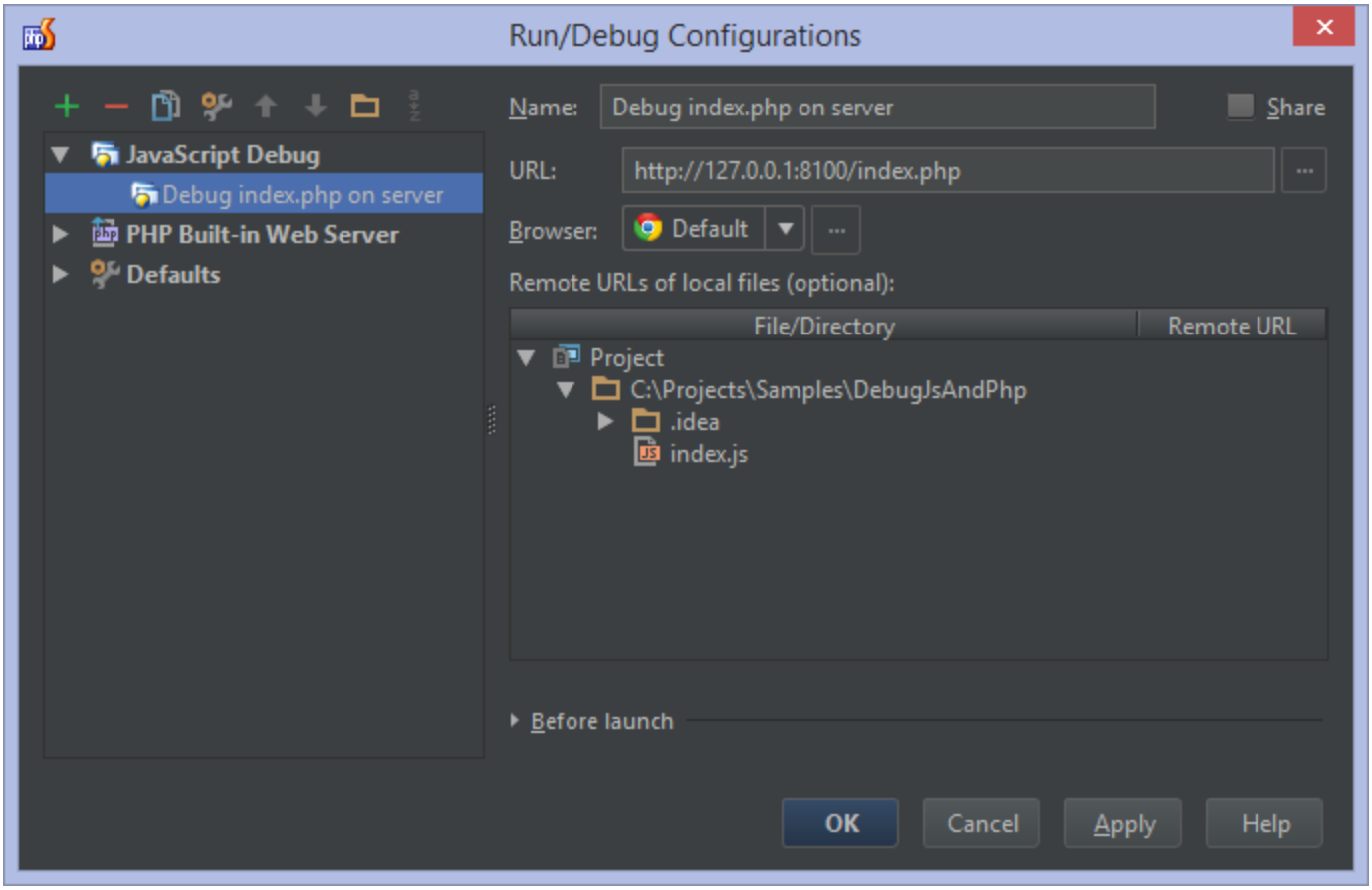


Once started, we can place breakpoints in JavaScript code and use the JavaScript debugger.

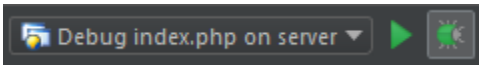
2.2. (option 2) Using a different webserver

When using a local webserver such as Apache or Nginx, or when developing on a remote web server or a Vagrant machine, we can start the JavaScript debugger using a Run/Debug configuration. We can create one from the toolbar or the Run | Edit Configurations... menu.

- Using the + button in the toolbar, add a new JavaScript Debug configuration.
- Enter the full URL to the page we want to debug on the webserver.
- Optionally, provide some mappings so PhpStorm can determine where to find local files relative to the remote URL. This will only be required when we have a different project structure locally and on the remote server. Note that if you are D eploying PHP applications with PhpStorm, mappings will be reused from the deployment configuration.



Once that is configured, we can start the JavaScript Debug session from the toolbar.



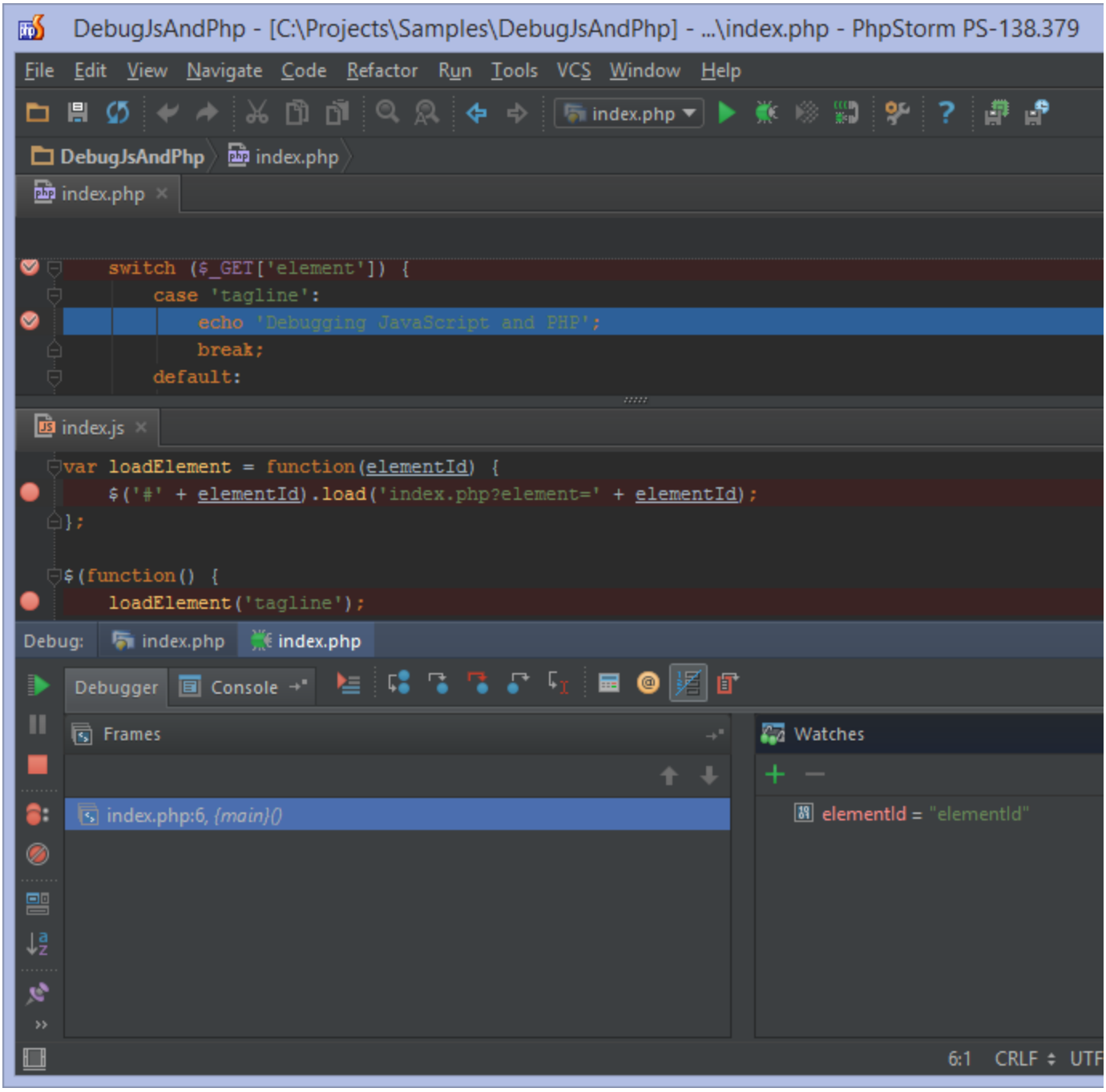
Before starting the JavaScript debug session, make sure the webserver is already running.

3. Start a PHP debugging session from the browser

To start a PHP debugging session, we will make use of the approach outlined in Zero-configuration Web Application Debugging with Xdebug and PhpStorm.

From the browser, we can use the PhpStorm debugger bookmarklets or one of the Browser Debugging Extensions to start a PHP debugging session.

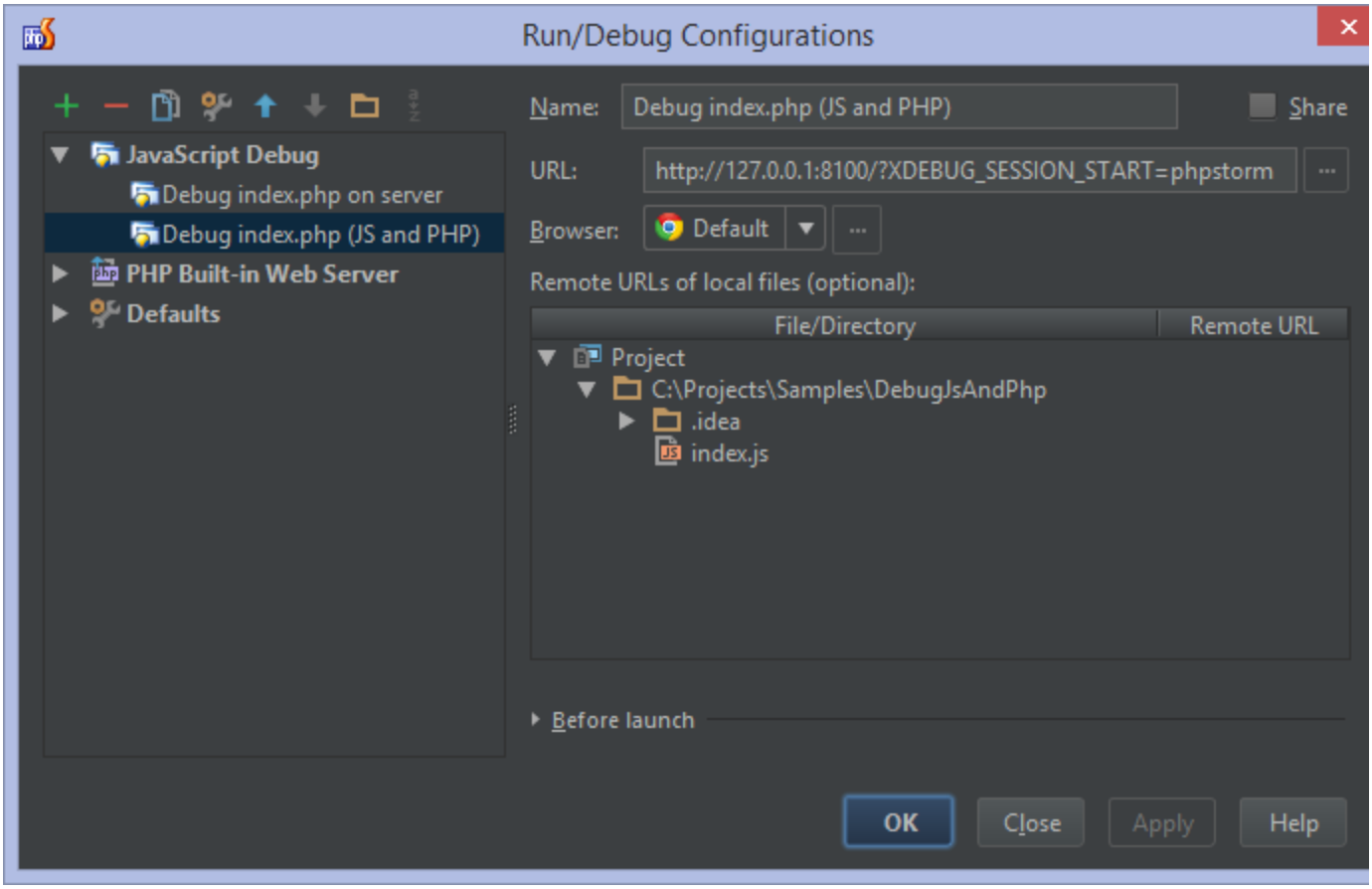
This will instruct the PHP server to make a connection to PhpStorm and open the debugger. Note that the IDE may initially ask you to provide the necessary path mappings. Once the debugger is attached, we will be able to debug both JavaScript and PHP at the same time. PhpStorm will switch between the debuggers as necessary.



4. (optional when using Xdebug) Launching the JavaScript and PHP debugger at the same time

In the previous steps, we started the JavaScript and PHP debugger separately. When using Xdebug, we can pass a XDEBUG_SESSION_START URL parameter to our server to start PHP debugging simultaneously with JavaScript debugging. We can do this using a custom Run/Debug configuration which can be created from the toolbar or the Run | Edit Configurations... menu.

- Using the + button in the toolbar, add a new JavaScript Debug configuration.
- Enter the full URL to the page we want to debug on the webserver. Make sure to append the XDEBUG_SESSION_START =some-session-name URL parameter, e.g. ?XDEBUG_SESSION_START=phpstorm



Once that is configured, we can start our combined PHP and JavaScript debugging session. To do that:

- listen for PHP debug connections;
- start the newly created Debug configuration.

5. Troubleshooting

I cannot place breakpoints in the JavaScript parts of a .php file

The current version of PhpStorm does not support setting both PHP and JavaScript breakpoints in one file. For example, no JavaScript breakpoints can be set in the following code:

```
<?php
// ...
?><!doctype html>
<html lang="en">
<head>
    <script>
        /* javascript code */
    </script>
</head>
<body>
</body>
</html>
```

To be able to debug the PHP and JavaScript code simultaneously, it's best to move the JavaScript code into a separate .js file and reference it from the HTML:

```
<?php
// ...
?><!doctype html>
<html lang="en">
<head>
    <script src="index.js"></script>
</head>
<body>
</body>
</html>
```

We can then place PHP breakpoints in the .php file, and set JavaScript breakpoints in the .js file.

Tweet