

Package Control

- Installation
- Browse
- Search
- Docs
- News
- Stats
- About
- Say Thanks

Search

BROWSE

Xdebug

by Kindari ST2

Xdebug Interface for Sublime Text 2

Details

VERSION	2013.03.29.04.29.00
Homepage	github.com
Issues	github.com
Modified	8 years ago
Last seen	1 hour ago
First seen	9 years ago

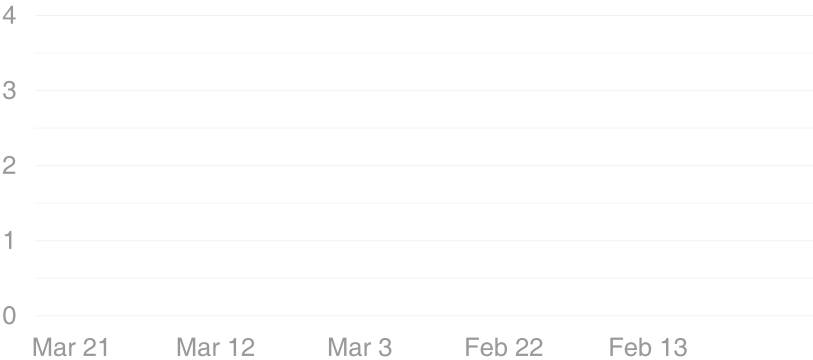
Installs

TOTAL 23K

WIN 11K

MAC 7K

LINUX 5K



Readme

SOURCE [raw.githubusercontent.com](#)

SublimeXDebug

Simple client to connect with XDebug.

Features

- Automatically display scope variables and stack trace
- Debugging layout for stack and variables
- Click variable to inspect value
- Auto-launch web browser for session based debugging (see below)

Quick start

Use `Shift+f8` to show a list of actions:

- **Start debugger:** Start listening for an XDebug connection
- **Add/Remove Breakpoint:** A marker in the gutter shows the breakpoint

Once the XDebug connection is captured, using the same shortcut shows these XDebug actions:

- **Continue:** Shows the debugger control menu (see below)
- **Stop debugger:** Stop listening
- **Add/remove breakpoint**
- **Status:** Shows the client status in the status bar

Debugger control menu

- **Run:** run to the next breakpoint or end of the script
- **Step Over:** steps to the next statement, if there is a function call on the line from which the `step_over` is issued then the debugger engine will stop at the statement after the function call in the same scope as from where the command was issued
- **Step Out:** steps out of the current scope and breaks on the statement after returning from the current function
- **Step Into:** steps to the next statement, if there is a function call involved it will break on the first statement in that function
- **Stop:** stops script execution immediately
- **Detach:** stops interaction with debugger but allows script to finish

Shortcut keys

- `Shift+f8` : Open XDebug quick panel
- `f8` : Open XDebug control quick panel when debugger is connected
- `Ctrl+f8` : Toggle breakpoint
- `Ctrl+Shift+f5` : Run to next breakpoint
- `Ctrl+Shift+f6` : Step over
- `Ctrl+Shift+f7` : Step into
- `Ctrl+Shift+f8` : Step out

Session based debugging

This plugin can initiate and terminate a debugging session by launching your default web browser with the `XDEBUG_SESSION_START` or `XDEBUG_SESSION_STOP` parameters. The debug URL is defined in your `.sublime-project` file like this:

```
{
  "folders":
  [
    {
      "path": "...",
    },
  ],

  "settings": {
    "xdebug": { "url": "http://your.web.server" }
  }
}
```

If you don't configure the URL, the plugin will still listen for debugging connections from XDebug, but you will need to trigger XDebug for a remote session. The IDE Key should be "sublime.xdebug".

Gutter icon color

You can change the color of the gutter icons by adding the following scopes to your theme file: `xdebug.breakpoint`, `xdebug.current`. Icons from Font Awesome.

Installing XDebug

Of course, SublimeXDebug won't do anything if you don't install and configure XDebug first.

Here's how I setup XDebug on Ubuntu 12.04:

- `sudo apt-get install php5-xdebug`
- Configure settings in `/etc/php5/conf.d/xdebug.ini`
- Restart Apache

Troubleshooting

XDebug won't stop at breakpoints on empty lines. The breakpoint must be on a line of PHP code.

If your window doesn't remove the debugging views when you stop debugging, then you can revert to a single document view by pressing `Shift+Alt+1`

The debugger assumes XDebug is configured to connect on port 9000.

Fixing pyexpat module errors. In Ubuntu you might need to do the following because Ubuntu stopped shipping Python 2.6 libraries a long time ago:

```
$ sudo apt-get install python2.6  
$ ln -s /usr/lib/python2.6 [Sublime Text dir]/lib/
```

On Ubuntu 12.04, Python 2.6 isn't available, so here's what worked for me:

- Download python2.6 files from [Ubuntu Archives](#)
- Extract the files: `dpkg-deb -x python2.6_2.6.5-1ubuntu6_i386.deb python2.6`
- Copy the extracted `usr/lib/python2.6` folder to {Sublime Text directory}/lib

In theory, it should work with any XDebug client, but I've only tested with PHP.