

Docs » System Administration Category » Docker with PHP and Caddy » Xdebug

Xdebug

Debugging with Xdebug

Anonymous · 09/09/2026

Debugging with Xdebug

This Docker container includes Xdebug to facilitate both interactive debugging and code coverage reporting. By default, Xdebug is installed but disabled to prevent performance impact in production environments.

Enabling Xdebug

To activate Xdebug for interactive debugging:

1. Access the container:

```
docker exec -it php84-caddy bash
```

2. Edit the Xdebug configuration file:

```
vim /usr/local/etc/php/conf.d/docker-php-ext-xdebug.ini
```

3. Change the mode setting:

```
xdebug.mode = develop,debug
```

4. Restart PHP-FPM:

```
supervisorctl restart php-fpm
```

Code Coverage for Unit Tests

To run tests with code coverage without permanently modifying the configuration:

```
docker exec -it php84-caddy bash -c "cd /var/www/sites/your-domain &&
XDEBUG_MODE=coverage vendor/bin/phpunit --coverage-html ./coverage"
```

This will generate an HTML coverage report in the `coverage` directory of your project.

Visual Studio Code Configuration

1. Install the PHP Debug extension.
2. Add this configuration to your `launch.json`:

```
{
  "name": "Listen for Xdebug",
  "type": "php",
  "request": "launch",
  "port": 9003,
  "pathMappings": {
    "/var/www/sites/your-domain": "${workspaceFolder}"
  }
}
```

Performance Considerations

Xdebug can significantly impact performance:

- **Production environments:** Keep Xdebug disabled (`xdebug.mode = off`).
- **Development environments:** Only enable when necessary.
- **Performance impact:** PHP execution can be 2-3x slower with Xdebug enabled.
- **Memory usage:** Increased memory consumption when active.

Always return the configuration to `xdebug.mode = off` when you've finished debugging.

Available Xdebug Modes

Xdebug 3 offers different modes that can be configured based on your needs:

Mode	Description
<code>off</code>	Disables all functionality (default)
<code>develop</code>	Development features (enhanced error messages, <code>var_dump</code> improvements)
<code>coverage</code>	Code coverage analysis for PHPUnit
<code>debug</code>	Interactive debugging with IDE

Mode	Description
profile	Performance profiling
trace	Function call tracing

Troubleshooting

No Connection to IDE

1. Check if Xdebug is properly enabled:

```
docker exec -it php84-caddy php -i | grep xdebug.mode
```

2. Ensure your IDE is listening for connections.

3. Verify the host and port settings:

```
xdebug.client_host = host.docker.internal  
xdebug.client_port = 9003
```

4. Check Docker networking:

```
docker exec -it php84-caddy ping host.docker.internal
```

Connection Timeouts

If the connection times out, add the following to your Xdebug configuration:

```
xdebug.start_with_request = yes  
xdebug.discover_client_host = false  
xdebug.log = /var/log/xdebug.log  
xdebug.log_level = 7
```

Then check the log file for connection issues:

```
docker exec -it php84-caddy cat /var/log/xdebug.log
```

Last updated on 09/09/2026