



[Knowledgebase](#) > [SSD Shared Hosting](#) > [Help! My website is showing a PHP-related error](#)

Help! My website is showing a PHP-related error

Helpdesk Staff - 2025-09-29 - [Comments \(0\)](#) - [SSD Shared Hosting](#)

PHP errors can disrupt your website's functionality, display unwanted messages, or even cause complete outages. Whether you're a developer or a site owner, understanding how to identify and fix these errors is essential for maintaining a smooth user experience.

□ Common Types of PHP Errors

Before diving into solutions, it's helpful to recognize the types of errors you might encounter:

- **Parse Errors (Syntax Errors):** Mistakes in the code structure, such as missing semicolons or unmatched brackets.
- **Fatal Errors:** Occur when PHP cannot execute a function, often due to undefined functions or missing files.
- **Warning Errors:** Non-critical issues like including a missing file or using deprecated functions.
- **Notice Errors:** Minor issues such as using an undefined variable.

□ Step-by-Step Troubleshooting Guide

1. □ Enable Error Reporting

To see what's going wrong, enable error reporting in your PHP configuration:

```
ini_set('display_errors', 1); ini_set('display_startup_errors', 1); error_reporting(E_ALL);
```

Alternatively, in cPanel:

- Go to **Select PHP Version > Options**
- Enable `display_errors`

2. □ Check the Error Logs

Most hosting environments store PHP errors in log files. In cPanel:

- Navigate to **Metrics > Errors**
- Or check `/error_log` file in your website's root directory

These logs provide detailed messages about what went wrong and where.

3. □ Revert Recent Changes

If errors appeared after a recent update or code change, try reverting those changes. Common culprits include:

- Plugin or theme updates (especially in WordPress)
- Manual edits to `.php` files
- Changes to `.htaccess` or `php.ini`

4. ☐ Verify PHP Version Compatibility

Some scripts require specific PHP versions. If you're a polur.net customer, [use either MultiPHP or PHP Selector as described in this article](#):

- Go to **Select PHP Version**
- Choose a compatible version (e.g., PHP 7.4 for older WordPress sites)
- Enable required extensions like `mysqli`, `mbstring`, or `curl`

5. ☐ Check File Permissions

Incorrect file or folder permissions can trigger errors. Use cPanel's File Manager or FTP to ensure:

- Files: 644
- Folders: 755

Avoid setting permissions to 777 as it poses a security risk.

6. ☐ Clear Cache and Sessions

Cached data or session files can cause unexpected behavior. Clear:

- CMS cache (e.g., WordPress plugins like WP Super Cache)
- Browser cache
- Server-side session files (if applicable)

7. ☐ Use a Debugging Tool

For complex issues, consider using tools like:

- **Xdebug** (for local development)
- **Query Monitor** (for WordPress)
- **PHPStorm** or **VS Code** with debugging extensions

☐ When to Contact Support

If you've tried the steps above and still see errors, reach out to our support with:

- A screenshot or copy of the error message
- The URL where the error occurs

- Any recent changes made to the site

Please note that polur.net Technical Support does **not** troubleshoot errors caused by third-party scripts (including WordPress) or websites for customers as this is beyond the scope of our complimentary support. However, if the error is determined to be caused by a server configuration issue or if there is a known server-side issue, we will fix it. Paid support is available if you need more assistance beyond our complimentary technical support.

□ Final Tips

- Keep your PHP scripts updated and well-documented.
- Avoid editing core files unless necessary.
- Test changes in a staging environment before deploying live.

By following this guide, you'll be well-equipped to diagnose and resolve PHP errors efficiently. If you'd like a printable checklist or a version tailored to WordPress, Joomla, or another CMS, I can help with that too!