Install PlaceholderAPI expansions and fix raw placeholders

PlaceholderAPI lets compatible plugins display information such as a player's name or economy balance. A placeholder appearing as `%player_name%` usually means part of the integration is missing or the receiving setting does not support it. Installing every available expansion is unnecessary.

Check the receiving plugin

Start with the plugin that will display the text: a scoreboard, chat formatter or menu, for example. Read its documentation to confirm that the specific configuration field supports PlaceholderAPI. Support elsewhere in the same plugin does not guarantee every text field parses placeholders.

Record the placeholder exactly, including punctuation and capitalisation where relevant. Confirm the Minecraft version, PlaceholderAPI version and receiving plugin version. Save a copy of the configuration before editing it. If you need to install or replace a JAR, take a completed backup and follow the normal plugin installation process.

Install only the required expansion

Some plugins register their own placeholders when they load. Others require an expansion downloaded separately. Check the official placeholder listing or expansion page to distinguish these cases and identify any required provider plugin.

For a simple Player expansion example, run these commands in game with appropriate administrator permissions:

```
/papi ecloud download Player
/papi reload
/papi list
/papi parse me %player_name%
```

The first command downloads the named expansion from the official eCloud. The reload command refreshes PlaceholderAPI; it is not the server-wide Bukkit reload command. Check that the expansion appears in the list and that the parse result matches your player's name.

In Console, omit leading slashes and replace `me` with an actual player's name for player-dependent tests. A null player is not a substitute for a real player when the expansion needs account-specific information.

Add the working token to your display

Put the exact tested placeholder into a supported text field, preserve YAML indentation and apply the receiving plugin's documented configuration refresh. Test with the intended player account. Start with one placeholder and one line so that a malformed surrounding configuration is easier to identify.

A Vault balance placeholder also needs the appropriate expansion and a working economy provider. Installing a display expansion does not create the underlying economy or player data. Check the provider independently before changing scoreboard formatting.

Separate download errors from parsing errors

If eCloud downloads time out, save the destination and exact error with its timestamp. Check the project's current connection guidance and ask support to investigate outbound connectivity. Do not disable certificate verification or download an expansion from an untrusted mirror to bypass a network problem.

If the expansion loads but still returns raw text, verify its documented placeholder syntax, dependency requirements and the receiving field's support. If a direct parse works but the display does not, investigate that plugin's integration and refresh method.

Keep diagnostic uploads private and review them for tokens or database credentials. Send support the failing placeholder, redacted configuration section, direct parse result and relevant versions. Avoid public broadcasts or command-execution parsing tests when a private text parse is enough.

Related CloudExa guides

- Install plugins on a Paper or Purpur Minecraft server
- Find Minecraft logs and send useful troubleshooting details

Further reading

- Using placeholders
- Commands
- Expansion Cloud
- Common issues

Revision #2
Created 2026-11-08 21:45:13 UTC by CloudExa Hosting Admin
Updated 2026-11-08 21:45:13 UTC by CloudExa Hosting Admin