Keep the TShock REST API disabled unless you need it

TShock’s REST API lets external programs interact with the server. A normal Terraria player does not need it to join. If you have no approved integration using that interface, leave it disabled and avoid exposing another management path.

Check whether anything depends on it

Open the game panel and record the installed TShock version. Download the generated TShock configuration as a private recovery copy. Configuration can contain credentials and tokens, so do not attach the full file to a public Discord message or publish it in a guide.

Ask whoever manages your website, bots or statistics tools whether they call TShock REST. Write down the integration name and the person responsible. A tool that reads a console log may not use REST at all. Confirm the dependency rather than enabling the API for a vague requirement.

The upstream REST documentation warns that parts of its endpoint list are old. Use the generated configuration and your installed version as the authority for supported fields. This guide disables an unneeded interface; it does not provide a public API deployment recipe.

Disable the listener

1. Choose a short maintenance window, notify players and stop the game server through the panel.
2. Find the generated TShock `config.json`, keeping it distinct from Terraria’s own configuration file.
3. Locate `RestApiEnabled`. If the existing field is `true` and no approved integration needs it, change it to `false`.
4. Preserve the other fields, commas and data types. Do not insert the same key twice.
5. Start the game server and inspect its startup log for configuration errors and REST-listener messages.

Verify both access paths

Join Terraria through the game address and port allocated to your service. Confirm ordinary gameplay and staff login still work. Disabling REST should not require changing the game’s assigned port or editing the host firewall.

If you have an authorised way to test the former REST address from outside, confirm that it no longer serves the API. A timeout by itself cannot identify whether TShock stopped listening or a network rule blocked it. Ask support to verify the process listener if the panel’s logs do not make that clear.

Handle old integrations and secrets

Update your staff notes so a later plugin setup does not re-enable the API without review. Check automation for saved REST credentials or application tokens. Remove unused secrets from those tools using their supported settings, and rotate any token you exposed in a ticket or screenshot.

If an approved integration breaks, keep the recorded configuration and error message. Re-enable the listener only after reviewing its authentication, permissions and restricted network path with support. Do not open a guessed port to the whole Internet as a quick test.

A working token can grant powerful actions depending on its account group. Restrict that group to the integration’s required actions. Keeping a spare SuperAdmin token for convenience increases the impact of a leak and makes later access reviews harder.

Related CloudExa help

- Keep your CloudExa account secure
- Set up TShock accounts and limited administrator permissions
- Open a support ticket

Official references

- TShock REST setup and token cautions

Revision #1
Created 2026-11-09 10:42:42 UTC by CloudExa Hosting Admin
Updated 2026-11-09 10:42:42 UTC by CloudExa Hosting Admin