Guide contents

Choose a build and a modding layer

Make a first item

Add Lua behavior deliberately

Maps, distribution and testing

Sources / References

Documentation: In progress **Quick start: In progress**

Offline copy from ModDocs. Source links remain clickable.

Project Zomboid modding

Project Zomboid has built-in mod management and several modding layers: **item scripts**, **Lua behavior**, **maps**, models, animation and translations. The PZwiki modding hub links each route. Its pages distinguish the stable game from Build 42's unstable branch, so choose a target build before copying an example.

> **Quick start:** create a tiny item-script mod for the game build you use. Make it appear in the mod list, enable it on a disposable save, and confirm the item behaves as intended. Add Lua only when the script format cannot express your behavior.

Choose a build and a modding layer

The modding overview explicitly warns that some pages describe the Build 42 unstable branch while the stable release has different APIs and data. Record the game's version and select documentation for that build. Test with an otherwise minimal mod list; a stable-build item example should not be assumed to work unchanged in an unstable beta.

Goal	Start with	Why
Add or tune an item	Scripts	Simple item parameters are data-driven

Goal	Start with	Why
Add gameplay behavior	Lua API	Events and exposed game objects
Add a location	Mapping	Map metadata, cells, tiles and zones
Change core Java behavior	Java modding	More invasive and harder to maintain
Translate content	Translations	Separate text resources

The PZwiki getting-started page and mod structure reference explain how the game discovers a mod. Build the folder structure and metadata first, then check that the mod appears in the game's manager before adding content. A mod that is not listed has a discovery problem; changing Lua code will not fix it.

Make a first item

Create a mod with a unique identifier and a small script defining one item or changing one value. Use the current Scripts reference for the syntax and fields supported by your target build. Keep the script focused on a single visible result, such as a display name or an item property you can inspect in game. Enable the mod on a new test save. Use debug tools only for testing and record how you obtained the item so a later test can reproduce it.

If the item never appears, inspect the mod manager first, then the script's folder and syntax. If it appears but a property is unchanged, verify that the game loaded your definition rather than another mod's definition. Disable other mods and retry before assuming the field is unsupported.

Add Lua behavior deliberately

Lua is the route for logic that cannot be expressed by item data alone. The Lua API reference lists events and exposed objects. Start with one event handler that writes a diagnostic message or applies an easily reversible effect. Confirm when it runs: at game start, world load, player creation or an interaction. A handler firing at the wrong lifecycle point can look like a broken item even when the script loaded.

Keep client, server and shared behavior in the appropriate locations described by the current mod-structure documentation. For multiplayer, test both the client display and server-authoritative behavior; a single-player success is not proof of multiplayer compatibility. The game's modding overview links current multiplayer guidance and build-specific notes.

Maps, distribution and testing

A map needs more than an image. The mapping reference points to `map.info`, tile definitions, room and item spawns, and vehicle zones. Start with one small location and test entry, collision, loot zones and save/reload before expanding. Record dependencies on custom tiles and avoid collisions with other mods' tile identifiers.

Package only your original mod files and required dependencies. Test the package from a clean mod profile, with your development copy disabled. State the supported game build and whether the mod supports multiplayer. The PZwiki hub also links the game's modding policy and Workshop route; check those before publication.

This guide is **unfinished**: it gives a version-aware first-mod route and points to the major systems, but it does not yet include a locally tested Lua item, full map build or server deployment walkthrough.

Sources / References

External project and documentation links cited in this guide. Check the upstream release and version before following version-sensitive steps.

- PZwiki modding hub
- Scripts
- Lua API
- Mapping
- Java modding
- Translations
- PZwiki getting-started page

- mod structure reference