Guide contents

Create the folder and metadata

Register a first node

Move beyond one node

Sources / References

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

Offline copy from ModDocs. Source links remain clickable.

Luanti / Minetest modding

Luanti is an engine that runs different games. A mod written for one game may depend on nodes or APIs absent from another, so choose the target game before copying an example. Mods use Lua, with `init.lua` as their entry point. The Luanti API and Modding Book document the first-mod route.

> **Quick start:** create a mod folder with `mod.conf` and `init.lua`, enable it for a disposable world, and register one uniquely named node. Use an original texture and check that the node is placeable before adding crafting or callbacks.

Create the folder and metadata

Open **About** → **Open user data directory**, then enter `mods` and create a folder with a lowercase technical name such as `moddocs_firstnode`. The Modding Book recommends a distinct name and explains the global mod path. A simple project has:

```text
mods/moddocs_firstnode/
  mod.conf
  init.lua
  textures/moddocs_firstnode_node.png
```

The texture can be a small PNG you made yourself. In `mod.conf`, start with `name = moddocs_firstnode`, a title and description. If your recipe uses `default:dirt` and

`default:stone`, add `depends = default` and target a game that actually contains the `default` mod, such as Minetest Game. **Do not make that dependency for a different Luanti game without checking it.** The mod directory reference explains `depends`, `optional_depends` and media folders.

Register a first node

In `init.lua`, use the documented node-registration pattern:

```lua
core.register_node("moddocs_firstnode:node", {
    description = "First Node",
    tiles = {"moddocs_firstnode_node.png"},
    groups = {cracky = 1}
})
```

The prefix before `:` should match the mod name. It prevents collisions with other mods. The texture filename must match the file in `textures`. This registers a block but does not yet make it craftable in survival. For a first test, enable the mod in the world's settings, enter a creative world and search the inventory for **First Node**. Place and dig it, then exit and reload the world. A missing texture or unknown-node appearance usually means a filename, folder, registration or dependency problem.

Next add a recipe, if the target game has the named ingredients:

```lua
core.register_craft({
    type = "shapeless",
    output = "moddocs_firstnode:node 3",
    recipe = {"default:dirt", "default:stone"}
})
```

The Modding Book example uses this route. Test the recipe in a new world with only the required game/mods enabled. If the node works in creative mode but the recipe does not, check that the `default` game content is present; that is a distinct problem from registration.

Move beyond one node

Luanti's `core` namespace includes registration and callbacks for items, entities, chat commands, formspecs and world interactions. Read the definition tables for required fields before changing the node's visual or physical behavior. Registration normally belongs at mod load time in `init.lua`; runtime callbacks execute later. For entities with persistent custom state, the object and entity chapter explains `get_staticdata` and `on_activate`. Do not assume a Lua table field survives save/reload by itself.

Package the mod folder with its own code, metadata and texture. Test it after moving your working copy away and installing the package into a fresh user-data directory or test profile. State the Luanti version, target game and dependencies. The release chapter covers ContentDB metadata and licensing.

The node, original texture, optional recipe and metadata form a complete small Luanti mod. The walkthrough is checked against the linked API and Modding Book; it has not been run in a local Luanti installation here. Entity, GUI, map-generation and multiplayer projects use additional APIs, so follow the linked references for those features rather than treating this node example as a template for every mod type.

Sources / References

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

- Luanti API
- Modding Book
- mod directory reference
- definition tables
- object and entity chapter
- release chapter