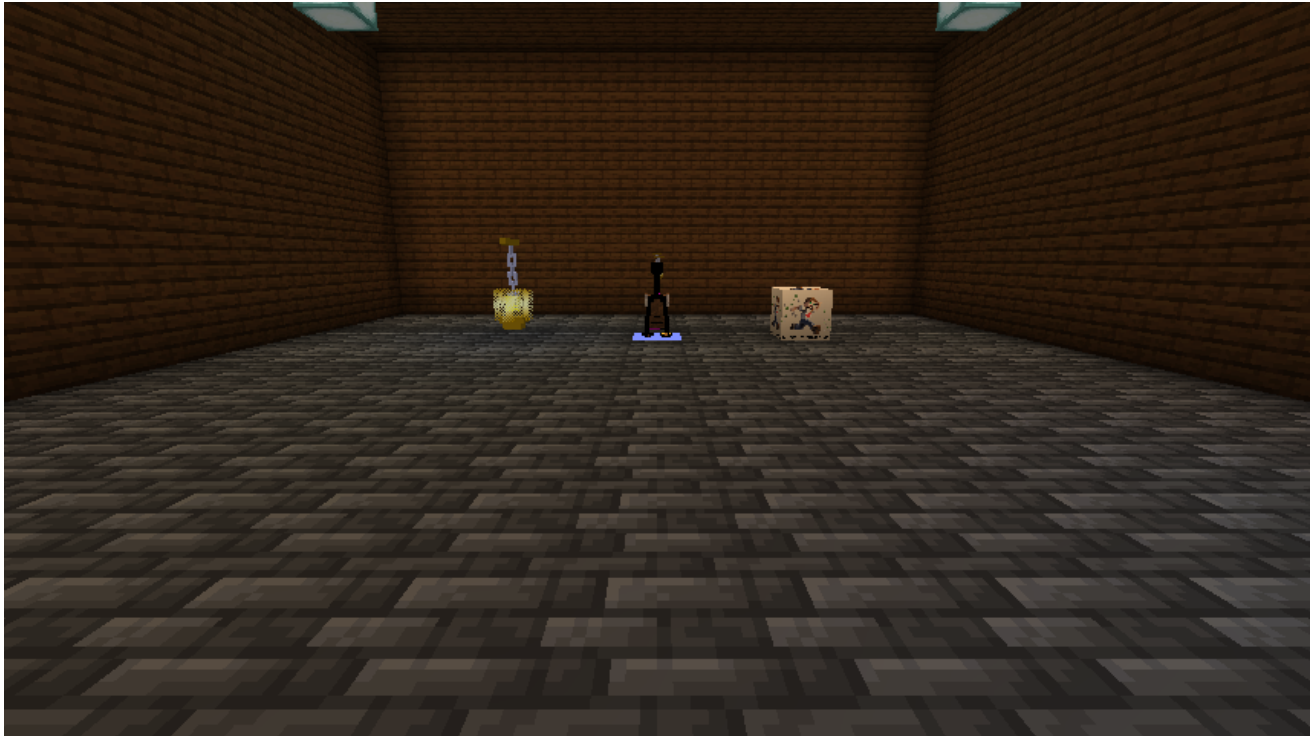


FABRIC 1.20 / JDK 21

LESSON 05

Connect models, textures and drops

Connect the world model to the item in your inventory



A real Fabric client screenshot from the original lesson.

WHAT YOU WILL MAKE

Check models, textures, names and drops independently, making a block you can obtain, place and mine back.

110–130 minutes | [simon-minecraft-template-1.20](#)

Complete the resource chain in a fixed order so each file has a clear role and a clear way to debug it.

Before you start: Complete lesson 4 with a registered lamp and working properties.

teacher-ma.de/minecraft-mod-lesson.php?lesson=5&lang=en

01 / 7

Complete resources and drops

Follow the model in the world all the way back to the player inventory.

Lesson 4 gave the block its properties. Now complete its resources and drops. Java registration tells the game that it exists; a model, inventory icon and mining drops still need the full resource chain.

1. Understand the reference direction: blockstates → block model → texture.
 2. Provide the BlockItem with an item model and localized names.
 3. Use a loot table to decide what drops when the block breaks.
1. Blockstates / 2. Block model / 3. Texture / 4. Item model / 5. Loot table

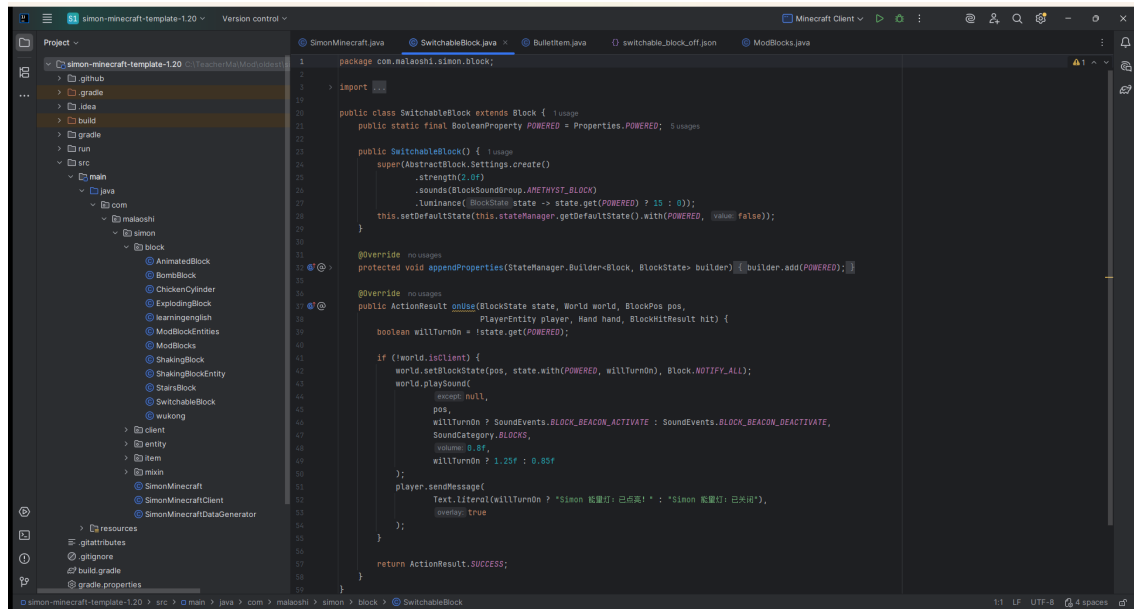
Use a fixed checking order

Check the world model first, the inventory icon second, and drops last. Verify one observable result at each step.

02 / 7

Locate resources in IntelliJ

Both assets and data are inside `src/main/resources`.



Real IntelliJ screenshot: locate resources in the same template project.

Paths relative to `src/main/resources`

```
assets/simon-minecraft/blockstates/light.json
assets/simon-minecraft/models/block/light.json
assets/simon-minecraft/textures/block/light_texture.png
assets/simon-minecraft/models/item/light.json
data/simon-minecraft/loot_tables/blocks/light.json
```

Models and textures belong in assets. Loot tables belong in data, not assets. Continue using `lang/zh_cn.json` and `lang/en_us.json` for names.

Version checkpoint

This lesson targets Minecraft 1.20, Yarn 1.20+build.1 and Fabric API 0.83.0+1.20. Run the supplied template with JDK 21; its Java compilation target is 17. Do not mix in method signatures from newer tutorials.

03 / 7

World and item models have different jobs

These are two separate JSON files.

```
assets/simon-minecraft/blockstates/light.json
```

```
{
  "variants": {
    "": { "model": "simon-minecraft:block/light" }
  }
}
```

```
assets/simon-minecraft/models/item/light.json
```

```
{
  "parent": "simon-minecraft:block/light"
}
```

Read each line, then save the files separately. blockstates selects the model for the placed block; the item model selects its inventory and hand appearance. Both can share simon-minecraft:block/light, which resolves to models/block/light.json.

04 / 7

Make mining return an item

Match the loot table namespace and name to the Java registration.

```
data/simon-minecraft/loot_tables/blocks/light.json

{
  "type": "minecraft:block",
  "pools": [
    {
      "rolls": 1,
      "entries": [
        {
          "type": "minecraft:item",
          "name": "simon-minecraft:light"
        }
      ],
      "conditions": [
        { "condition": "minecraft:survives_explosion" }
      ]
    }
  ]
}
```

Check the connection

The filename light matches the block registration name. simon-minecraft:light inside the JSON is the registered item ID. This pool rolls once and checks survives_explosion. Without a loot table, the block may still be available in Creative while producing no item when mined in Survival.

05 / 7

The six links in the resource chain

Use the symptom to find the relevant link, then trace its references.

Registration ID	simon-minecraft:light is the starting point for the references.
blockstates	Selects a block model for the block state.
block model	Defines elements, a parent model and texture variables.
texture	PNG paths, capitalization and alpha must be correct.
item model + lang	Control the inventory appearance and display name respectively.
loot table	Determines the dropped item and whether explosion conditions allow a drop.

Check paths carefully

Use lowercase letters, digits and underscores for resource filenames and slashes between directories. Keep the existing simon-minecraft namespace. Do not rely on Windows tolerating capitalization differences; the same error may fail in a case-sensitive environment.

06 / 7

Verify the whole chain in game

Test the world, inventory, drops and saved data.



Real game screenshot: the lamp, ordinary models, maze block and dropped items in one test room.

1. The placed block has the correct model and textures.
2. The inventory icon and hand model are correct.
3. Mine with the correct tool in Survival and obtain the expected item.
4. Save and reopen the world, then check the block and drops again.

The original image came from a copied test world in runClient and was saved by Minecraft ScreenshotRecorder.

Temporary capture code was removed and the project rebuilt at that time. Repeat all four checks in your own project.

07 / 7

Troubleshooting and the final challenge

Use the symptom to find the missing connection.

Missing world texture	Start with blockstates and trace the model and texture references.
Missing inventory icon	Check parent in models/item/light.json.
Mining produces no drop	Check the data namespace and loot_tables/blocks path, then verify Survival mode and tool conditions.
Missing display name	Check block.simon-minecraft.light in the file for the selected game language: en_us for English or zh_cn for Chinese.

Silk Touch challenge

Add a condition so ordinary tools drop materials while a Silk Touch tool drops the complete lamp. Test both branches in Survival and record the actual drops.

Next, use BooleanProperty to make the same block remember whether it is on or off.

Official references

The tutorial uses the supplied Fabric 1.20 template. Check documentation for the same version.

[Yarn 1.20 documentation](#)