

## 2. Create worlds (Builder)

In GWorld v2, new worlds are created using the **builder pattern**. This separates the **configuration** (registration) from the **generation** (load).

### Step 1: The Builder (Registration)

With the `WorldCreationBuilder` you define all the properties of the world. Calling `.build()` registers the world in the database, but does not yet physically create it.

```
import de.gilljan.gworld.api.IManageableWorld;
import de.gilljan.gworld.enums.WorldTypeMapping;

// Start the builder
IManageableWorld newWorld = worldManager.createBuilder("MeineEventWelt")
    .worldType(WorldTypeMapping.NORMAL) // Environment: Normal, Nether, End, Large Biomes,
    Amplified, Flat
    .generator("PlotSquared")           // Optional: Custom Generator Name
    .seed(987654321L)                   // Optional: Set seed
    .build();                           // -> Saves the world to Config/DB
```

**Note:** The `.build()` method internally calls `addWorldFromBuilder`. The world is then known (registered), but still unloaded.

### Step 2: Load (generation)

```
// Creates the bukkit world and loads chunks
boolean success = newWorld.createMap();

if (success) {
    getLogger().info("World is ready!");
}
```

The same applies to imports. Use the folder name of the world and use the `importMap()` method instead of `createMap()`.

# Why two steps?

1. **Performance:** You can register worlds when the server starts, but only load them when a player starts a minigame.
2. **Security:** You can set flags (e.g., PvP off) before the first player enters the world.

---

Revision #3

Created 2026-02-09 19:05:16 UTC by Gilljan (Jannik)

Updated 2026-02-09 19:27:44 UTC by Gilljan (Jannik)