Guide contents

Pick the right path

Create a first map

Understand vehicles and JBeam

Pack and test a mod

A complete first project: a short drivable loop

Beyond the first map

Sources / References

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

Offline copy from ModDocs. Source links remain clickable.

BeamNG.drive modding

BeamNG.drive has several modding paths. A level is made in the World Editor; a vehicle combines a visible model with a **JBeam** node-and-beam structure; UI apps, game modes, and Lua have their own workflows. The official modding index is the source for current editor and file-format details.

> **Quick start:** create a small level from the Starter Level template. Make it load, give it one safe vehicle spawn and a drivable surface, save, reload, and drive it. Then package it as a ZIP and test that ZIP as a normal installed mod. A whole city or a new car is a much bigger first project.

Pick the right path

You want to make	Start with	Main complication
A drivable map	Level Creation and World Editor	Spawn, terrain, materials, performance
A vehicle part or configuration	Vehicle documentation	Part slots and JBeam relationships

You want to make	Start with	Main complication
A full vehicle	Vehicle getting started	Model, physics structure, materials, tuning
A UI app or scripted feature	UI and programming index	Different runtime and packaging rules

These are related, but a map tutorial does not teach vehicle deformation. Choose one outcome for the first release and follow that branch of the official docs.

Create a first map

Open the World Editor with **F11**, choose **File** → **New Level**, and start with the current Starter Level template. The terrain and new-level instructions describe how the editor creates a new folder under `levels/`. Give the map a unique name and save it before editing many objects.

The first playable milestone is modest: a default spawn, a ground or terrain surface, a short road or test area, a few props, readable daylight, and no missing materials. The official first-playable checklist recommends this exact sequence because it isolates path and material mistakes early. Move the spawn onto a safe surface, save, reload the level, and verify the player vehicle still appears in the intended place. Test one imported asset before filling the world.

A level is more than scenery. Spawn points, lighting, drivable surfaces, and resource paths all affect whether another player can load it. Keep source meshes and other working files separate from the distributable level, and inspect the editor console/log for missing textures or load errors. Missions, traffic, and Flowgraph are distinct follow-on systems; complete the world foundation first.

Understand vehicles and JBeam

JBeam defines the physics structure: **nodes** have mass and position, while **beams** connect them and act like springs. A visible mesh alone is not a usable BeamNG vehicle. The JBeam introduction explains why a structure needs bracing and how it deforms; the syntax reference documents its JSON-like tables and case-sensitive keys.

For a first vehicle project, the official getting-started advice recommends a small project and a text editor that can show syntax errors. The basic-car tutorial uses a simple chassis and existing drivetrain/suspension components as a learning path. Treat its provided files as a tutorial template, not as a universal vehicle skeleton. A missing bracket, wrong part slot, or mismatched mesh can prevent the car from appearing or behaving correctly. Change one subsystem at a time, reload it, and observe the result.

Pack and test a mod

BeamNG's packing guide says the ZIP's first visible directory should be the content directory, such as `levels`, `vehicles`, `lua`, or `ui`. The wrapper project folder should not sit above that directory inside the ZIP. For a map, the structure begins like this:

```text
my-map.zip
└── levels/
    └── my_map/
        └── ...level files...
```

Install the ZIP through the game's user-folder `mods` location and inspect it in Mod Manager, following the official installation instructions. Test the **packed** copy, ideally with the working unpacked copy disabled. Confirm it appears in the level selector, loads from a fresh game session, spawns correctly, and has no missing resources. If it does not appear, open the ZIP and inspect its top-level folders before changing gameplay data. The mod packing page lists this wrapper-folder mistake as a common cause.

A complete first project: a short drivable loop

Make a new level from the Starter Level template, rename its folder, and confirm it reloads before adding content. Move the default spawn onto a flat area. Make a short road or test surface with enough space to turn around, place a few props, and assign a material that does not show a warning texture. Save and reload, then drive the loop. The official first-level checklist treats spawn, ground, road, props, and daytime light as the minimum playable map.

Fill in `info.json` and provide a representative `preview.png` using the current level packaging guide. It says the package should include the level and any required art paths, but exclude source meshes, caches, and editor backups. Open the ZIP to ensure its top directory is `levels/`, not a wrapper named after your project. Remove the unpacked working folder, install the ZIP alone, clear cache if needed, and launch the map again. Drive the complete loop and check the log for invalid JSON, missing material files, and broken spawn names.

If it only works while the loose project is installed, the ZIP is incomplete. If it appears in the selector but vehicles start in the wrong place, check the spawn object and its `info.json` reference. A map that loads and drives from the ZIP is a complete first release; the vehicle tutorial is a separate, more complex project.

Beyond the first map

The packaged, driveable map is a complete first mod. This walkthrough is source-reviewed against BeamNG's documentation; it has not been run in a local BeamNG.drive installation here. Vehicle, mission and Lua projects have distinct requirements: follow the level guide and vehicle guide for their specific files and tests. The official documentation itself notes that some sections remain incomplete.

Sources / References

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

- official modding index
- Level Creation
- Vehicle documentation
- Vehicle getting started
- terrain and new-level instructions
- official first-playable checklist
- JBeam introduction
- syntax reference
- basic-car tutorial

- packing guide
- official installation instructions
- level packaging guide