DRUPAL
Opens Minds

Learn Drupal. Elevate Your Skills.

DRUPAL VERSION 11

Beyond WordPress

by Daniele Venditti

DRUPAL Opens Minds: Beyond WordPress

Daniele Venditti

DRUPAL Opens Minds: Beyond WordPress

INTRODUCTION

Why I wrote this book (and why you should look beyond WordPress)

If you use WordPress, you know its greatest strength well: the ability to set up intuitive sites in record time. It is a tool I have used and appreciate, especially when speed and simplicity are everything.

But over time, I asked myself: what happens when projects become truly complex? I often found myself facing clients who were no longer asking for simple “pages” or “articles,” but for articulated data structures, scalable systems, and security that wouldn’t leave me anxious every time I updated a plugin, hoping nothing would conflict. It was at that moment I realized I needed to make a leap in quality.

If WordPress is an excellent production car, I see Drupal as a specialized workshop: the place where I can design a custom vehicle, tailoring it exactly to the needs of professional and ambitious contexts.

Why I chose to add Drupal to my skillset

Learning to use Drupal didn’t mean abandoning WordPress, but rather stopping being a prisoner of the wrong tool for the wrong project. Here is what changed in my way of working:

- **I model content, not just layouts:** With Drupal, I learned to structure data with surgical precision. Whether dealing with massive catalogs or complex relationships, I now manage information with a level of accuracy that would require too many compromises on other CMSs.
- **I offer Enterprise-level security:** It is no coincidence that major universities and international institutions choose Drupal. Knowing I can offer such a solid architecture allows me to present much more robust and reliable solutions to my clients.
- **I eliminated the “dictatorship” of plugins:** In Drupal, modularity is native. I stopped accumulating dozens of third-party extensions; now I integrate functionalities in a coherent, clean way, without fear of sudden bottlenecks.
- **I am ready for the modern web:** I use Drupal as a true content engine. Whether it’s traditional sites, headless systems, or advanced AI integrations, I feel I have a tool in my hands that does not fear technological evolution.

Conclusion: Leveling Up

WordPress remains the ideal tool for speed and simplicity. But Drupal has become my tool for ambition, structure, and control.

In this ebook, you won’t find useless theory, but my practical method: the right way to look at Drupal with the mindset of someone who, like you and me, comes from the WordPress world and finally wants to raise the bar.

How to read this ebook

You are not holding a theoretical manual.
You have a Drupal that works on "paper."
Each chapter is built like a real work session.
As you read, **you will already see where to click, what to write, and which option to choose**. You don't need to install anything to understand. Screenshots are described, steps are precise sequences, and examples are exactly what you will encounter when you open the administration panel. You can study on the train, during your lunch break, or in the evening on the couch. And when you finally open Drupal, it will feel like you've used it a thousand times before.
Because that's exactly how a good manual works: it doesn't just explain; it makes you do it.

(This ebook covers the latest version, Drupal 11)

If you would like to receive further insights and updates regarding this ebook:

Subscribe to the Newsletter here

CHAPTER 1 - The Drupal Mindset

Structure, control, and content management

Many think Drupal and WordPress are the same thing just because they both serve to create websites. In reality, the difference lies in how they are built “inside.” While **WordPress** is a package of ready-made services, **Drupal** is an intelligent system where every part communicates perfectly with the others.

Let’s look in detail at why this difference is fundamental to your work.

A system that truly communicates In WordPress, you often find yourself installing many different plugins: one for the image gallery, one for contact forms, one for extra fields. The problem? These plugins are like strangers who don’t speak the same language; they often don’t “know” each other and can create conflicts.
In Drupal, modules are designed to work together from the start. It’s like a well-coordinated team: the module that manages data naturally talks to the one managing searches and the one deciding how to display images. Everything is part of a single, coherent system.

Long-term savings At first glance, WordPress seems cheaper because many plugins are free. However, for professional projects, you soon realize you almost always need the paid “Pro Version.” Between annual subscriptions and licenses for every single feature, costs accumulate and weigh on the client’s quote.

Drupal already includes most advanced features “as standard.” Tools for creating complex searches or custom fields are free and native. This means no financial surprises and no need to ask your client to pay for dozens of different subscriptions every year.

Less maintenance, more peace of mind Anyone who uses WordPress knows the “update moment” well: you click a button and hope no plugin stops working or conflicts with the theme.
With Drupal, maintenance is much more solid. Since modules are made to integrate perfectly, updates are safer and more predictable. This translates into fewer hours spent fixing sudden bugs and more time dedicated to growing the project.

Data is free (Scalability and Flexibility) In WordPress, content is often “trapped” within the design. If you want to show the same data in different ways, you often have to duplicate the work or use complicated code.
In Drupal, data is free. Imagine a real estate agency with 5,000 listings:
With WordPress, managing such a vast search by price, square footage, and area requires heavy plugins, expensive hosting, and a lot of optimization to avoid slowing down the site.
With Drupal, this structure is native. The system is built to handle thousands of pieces of content without losing speed. You can show the same listing on the homepage, in a search list, or even send it to a mobile app, without ever duplicating a single word.

In conclusion: Drupal allows you to build a solid structure once. The result is a faster, more secure site ready to grow with your client’s needs, without hidden costs or compatibility issues.

Getting an edge

To use Drupal successfully, you need to make a small “mental shift.” If WordPress asks you: “*What should my agency’s site look like?* (We will often use the practical example of a hypothetical Real Estate Agency)”, Drupal asks you: “*How is my real estate data structured?*”.

Switching to the Drupal mindset means moving away from thinking about “descriptive pages” and “graphic themes” and starting to think about structures made of data sets.

1. **Structure first, design later** In WordPress, you often look for a real estate theme and try to adapt your houses to that design. With Drupal, we do the opposite:

    - **Planning:** First, decide what information defines a house (Price, Square Footage, Number of rooms, Energy class).

    - **Construction:** Create the technical structure to host exactly this information.

    - **Presentation:** Only at the end do you decide how to display it graphically. This approach gives you total control. You aren’t limited by what the theme provided; you decide every single detail.

2. **The concept of “Pure Data” (The Property example)** Imagine entering a listing for a penthouse downtown. In WordPress, you might write everything (description, price, and sqm) in one large block of text. The problem is that the computer reads that text as a simple story, not as data.
   In Drupal, you separate each element into specific fields:

    - A field just for the Price (number).
    - A field for Square Footage (number).
    - A dropdown for the Property Type (Apartment, Villa, Office).

   Why do this? Because if the data is separated, Drupal can “reason” with it. It can instantly create a list of “all Villas costing less than €300,000” or “all apartments with at least 3 rooms.” In WordPress, achieving this level of precision often requires complex and heavy plugins; in Drupal, it’s the natural way of working.

3. **Total management control** The Drupal mindset allows you to manage the agency professionally:

    - **Precise roles:** You can create an account for the “Photographer” who can only upload images, an account for the “Agent” who enters technical data, and an account for the “Director” who must approve the listing before it goes live.
    - **Workflows:** The listing doesn’t have to go online immediately. You can set up a path: Draft -> Review -> Published.

4. **Building for the future** Adopting the Drupal mindset means building an archive that doesn’t age. If tomorrow you want to create a smartphone app for your agency, your data is already ready. You won’t have to rewrite anything, because Drupal already knows how to distinguish what is a price and what is an address.

In summary, the Drupal mindset for real estate is:

- **Analyze:** What characteristics do my houses have?
- **Organize:** Create precise fields for each characteristic.

- **Visualize:** Show the data in lists, maps, or galleries, wherever they are needed.

1.1 Case study: the real estate listing site we will build

To make this journey concrete, we won't limit ourselves to theory. Throughout this ebook, we will build a real project together: a portal for a modern real estate agency (or at least a part of it).

We chose this example because a real estate site is the ultimate test for any developer. It contains all the challenges a professional site must handle: advanced search, photo galleries, maps, structured data, and multi-user management.

Here is what we will build step by step:

The structure (The project's backbone) Instead of creating simple pages, we will define the **Content type** "Property". We will create precise fields for price, surface area, number of rooms, and area. You will learn how Drupal organizes this information so it is always easy to update and impossible to lose.

The search engine (The power of Drupal) We will create a search system that will allow your clients to find their dream home in just a few clicks. *Want to filter for "Apartments with a garden for under €200,000"?* With Drupal, we will do it without writing a single line of code, using a powerful tool called **Views**.

Why is this example important for you? Learning to manage a real estate site with Drupal will give you the skills to tackle any other project: whether it’s an e-commerce site, a news portal, or a complex corporate platform. Once you understand how to manage the data for a house, you will hold the keys to building any type of digital architecture.

1.2 The Concept of the “Node”

Imagine having a large box of colored bricks. With these bricks, you can build a house, a castle, a bridge. Drupal is like a big box. You have the bricks; then you decide what to build.

WordPress is different. WordPress is like a house that is already built. You go in, furnish it, and paint the walls. But if you want an extra room or a different staircase, you have to force the structure with plugins. Sometimes it works, sometimes it doesn’t.

Drupal gives you the bricks. At first, it seems harder. But then you can build exactly what you want.

The bricks are called “Node”
In Drupal, every piece of content is a **Node**. An article is a **Node**. A page is a **Node**. A real estate listing is a **Node**.

In WordPress, when you create something, you immediately think of a “Post” or a “Page.” In Drupal, however, we use a single term for everything: the **Node**.

Don’t be intimidated by the technical name. A **Node** is simply “a thing” on your site.

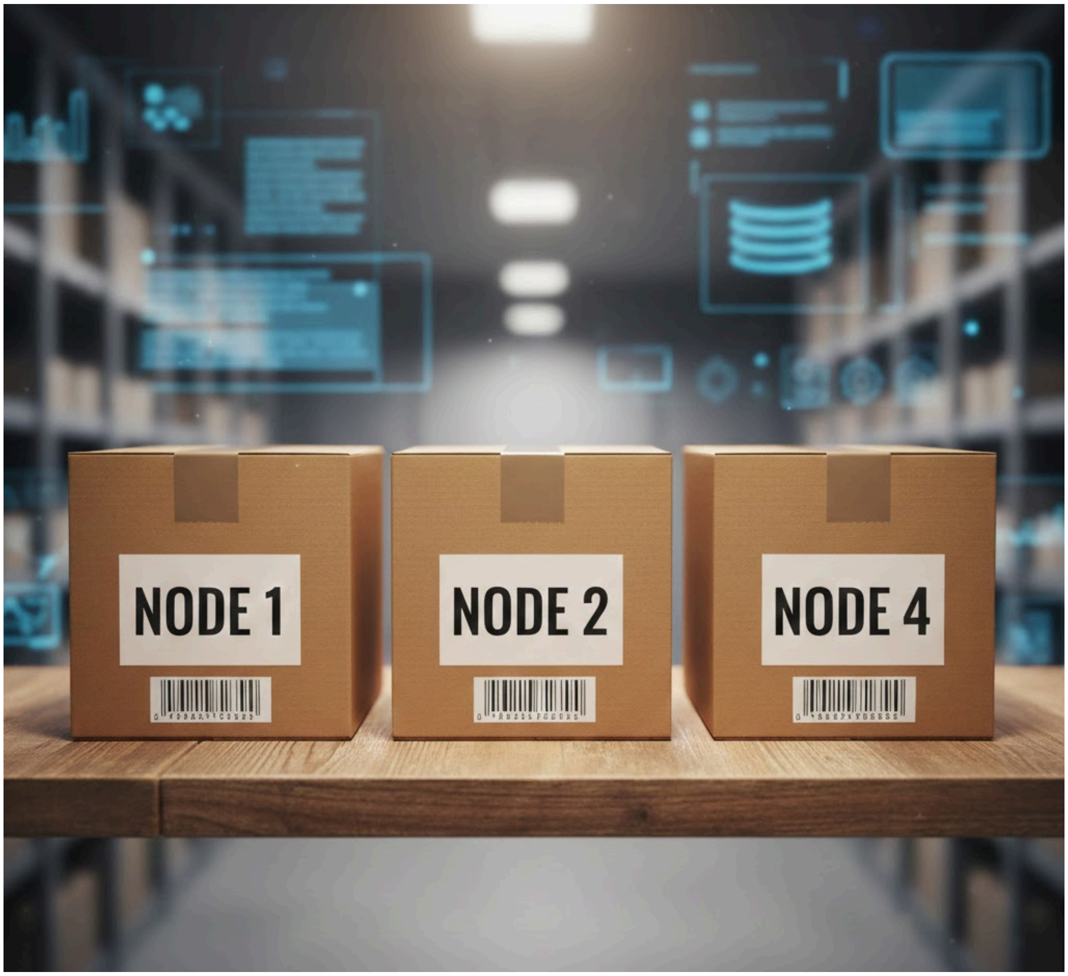
- Writing an article? It’s a **Node**.
- Creating a services page? It’s a **Node**.
- Entering a listing for a house? It’s a **Node**.

The shoe store example
To understand better, imagine a large shoe warehouse. Every single box on the shelves has a label with a unique barcode. That code tells the warehouse worker: “This is a specific box in my store.”

In Drupal, that code is called the **Node ID** (or **nid** for short). Every piece of content you create receives its own identification number.

- Box number 1 is your blog post.
- Box number 2 is the “About Us” page.
- Box number 3 is the “Apartment on Via Roma” listing.

Each is a separate box; each is a **Node**.



drupal node

What’s inside the box? (The Fields)

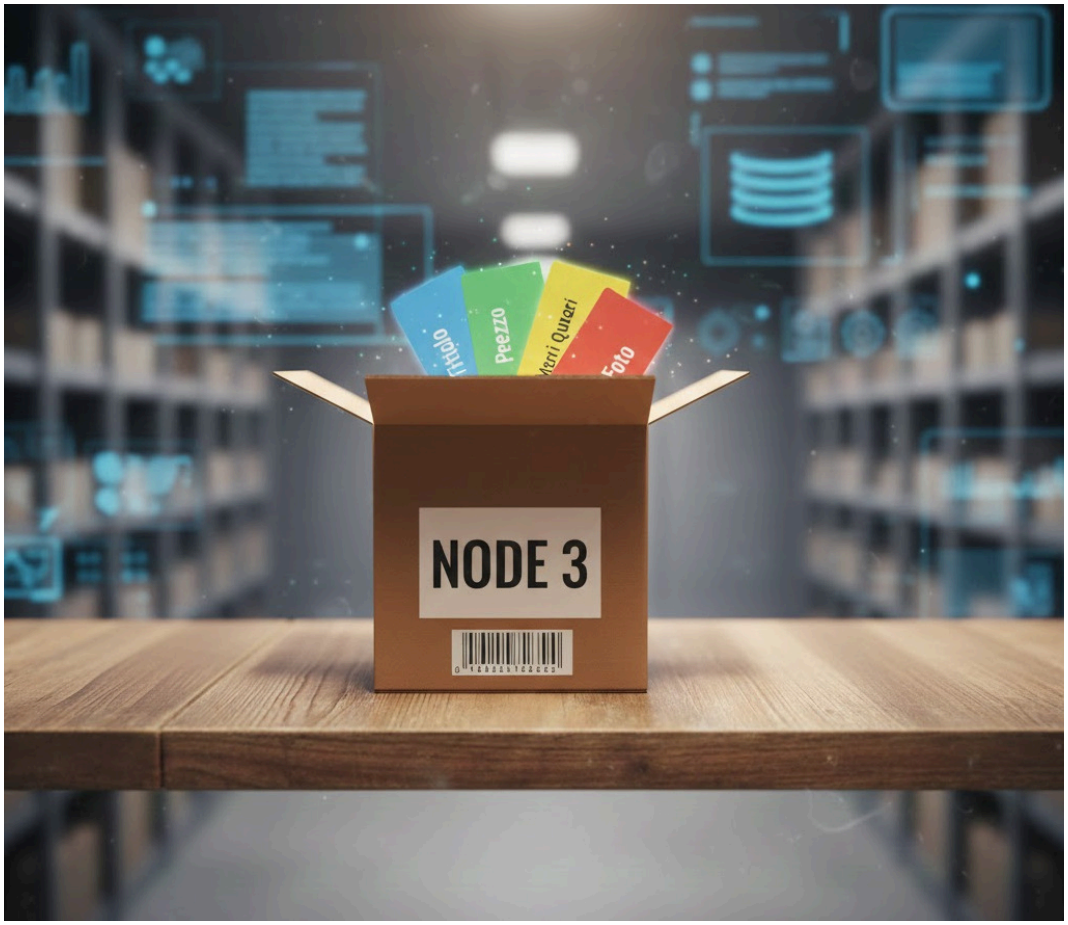
If we open our real estate listing box (Node 3), what do we find? We find a series of cards with specific information. In Drupal, these pieces of information are called **Fields**.

Inside the “Apartment on Via Roma” box we will find:

- **Title**: Apartment on Via Roma
- **Price**: 250,000 euro
- **Surface**: 85 sqm

- **Image**: the photo of the house.

Please note: The title is not a Node. The price is not a Node. They are just pieces of paper inside the box. The **Node** is the container that holds everything together—that is, the data we enter through the **Fields**.



drupal node

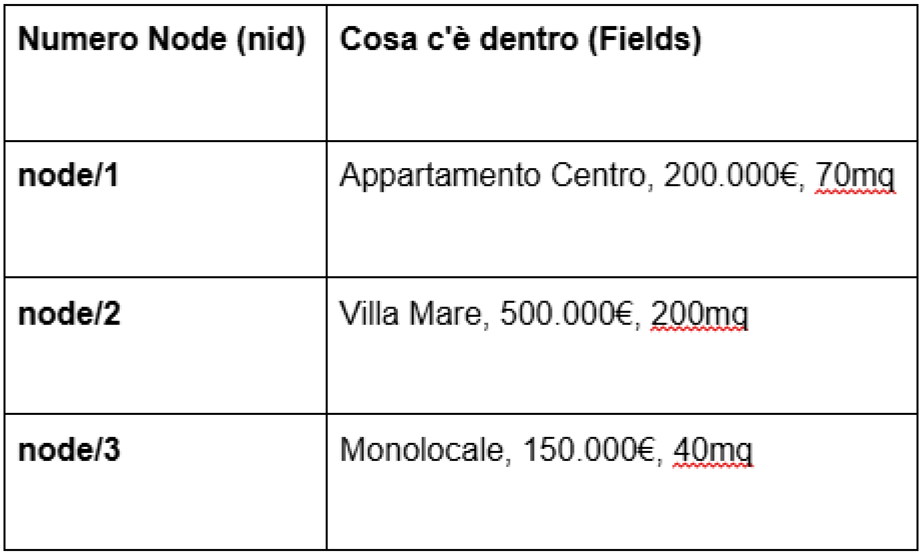
The content address (URL)

When browsing a Drupal site, you might come across web addresses (URLs) like this:

```
your-site.com/node/3
```

This is how Drupal talks to its database. That link means: “Go to the warehouse, get box number 3, open it, and show me what’s inside.”

Drupal opens the box, reads the **Fields** (title, price, etc.), and displays them in a beautiful web page for the visitor.



Numero Node (nid)	Cosa c'è dentro (Fields)
node/1	Appartamento Centro, 200.000€, 70mq
node/2	Villa Mare, 500.000€, 200mq
node/3	Monolocale, 150.000€, 40mq

drupal node

Why is this system so powerful?

In WordPress, the post *is* the page. In Drupal, the box (**Node**) exists independently of how you decide to display it.

This gives you incredible freedom. Because information is separated into numbered boxes, you can decide to:

- Display the box alone as a full page.
- Take only the “Price” and “Photo” from ten different boxes and put them all together in a list (using **Views**, which we will discuss shortly).
- Take the box’s address and show it as a dot on a map.

In short: the **Node** is how Drupal keeps your content organized. You fill the boxes, and Drupal makes sure never to lose sight of them.

Quick Summary:

- **Node** = A box with a unique ID.
- **Fields** = The content of the box (title, price, photo).
- **node/3** = The instruction to “open box number 3.”

1.3 The “Blueprint” of the boxes: the Content Type

We’ve seen that every listing in our agency is a box called a **Node**. *But how does Drupal know that inside the apartment box it should ask for the “Price,” while in the blog post box it should ask for the “Author”?*

The answer is the **Content Type**.

If the **Node** is the physical box, the **Content Type** is the mold (or the default template) we use to create it. Think of the **Content Type** as the machinery used to produce those boxes, each with specific characteristics or data (**Fields**).

Why do we need the “mold”? Because by creating a **Content Type** called “Real Estate Listing,” we tell Drupal: “Every time I create a box of this type, always prepare these compartments (**Fields**):”

1. A place for the Title.
2. A place for the Price.
3. A place for the Photos.
4. A place for the Description.

The pre-printed form example Think of the **Content Type** as a pre-printed paper form found inside the box. When a new agent enters the office to list a house, they don't take a blank sheet. They take the "Real Estate Listing" form.

- The form already has empty spaces with labels (Price, SQM, Area).
- The agent only has to fill in the blanks.
- Once filled out, that sheet becomes a **Node**.

In Drupal, you can have many different "molds"
The great thing is that you can create as many **Content Types** as you want. On your agency's site, you could have:

- **Content Type "Property":** with fields for price, rooms, and energy class.
- **Content Type "Agent":** with fields for name, headshot, and phone number.
- **Content Type "Article":** with fields for text, category, and date.

This way, every time you add content, Drupal already knows which questions to ask you and how to organize the data.

Why is this convenient for WordPress users?
In WordPress, you only have two main molds: "Post" and "Page." If you want to add a "Price" field to a post, you have to install external plugins. In Drupal, creating a custom mold (a **Content Type**) with all the fields you want is a core feature. No plugin is needed: it's in the system's DNA.

In summary:

- **Content Type**: The blueprint, the empty form (the mold) inside the box.
- **Node**: The final result, the box full of data (the completed form).

Content Type (The Mold) + Data Entered = Node (The Result/The Box)

So, can we say that Node and Content Type are two connected concepts?

Absolutely! They are two sides of the same coin. One cannot exist without the other. To close this topic and lock in the concept, we can say their link is a relationship of “Rule” and “Application of the rule.”

The inseparable link between Content Type and Node Think of them like the relationship between a Recipe and a Cake:

- **The Content Type is the Recipe:** It tells you which ingredients are needed (Flour, Eggs, Sugar). You don’t eat the recipe; it’s just an instruction sheet.
- **The Node is the Cake:** It’s the actual object you baked following the recipe. If the recipe says “Flour,” your cake will have 500g of flour. If the recipe changes and says “add Chocolate,” the next cake you make will have chocolate.

Why is it important to understand they are connected? Because in Drupal, every time you want to create new content, the system will always ask the same question: “*Which Content Type should this Node belong to?*”. You cannot create a **Node** “in a vacuum.” It must always follow the rules of a **Content Type**.

1. Choose the **Content Type** (e.g., Real Estate Listing).
2. Enter the data (e.g., “Beautiful Villa,” “€300,000”).
3. Save the **Node** (and the listing is online!).

A small trick to never get confused again
If you find yourself struggling, use this phrase:
“The **Content Type** is the TYPE of thing I am creating; the **Node** is the specific THING I have created.”

- **Type**: Movie Review → **Thing**: The review of “The Matrix.”
- **Type**: Product Sheet → **Thing**: The “Model X” drill.
- **Type**: Real Estate Listing → **Thing**: The apartment on Via Roma.

Let’s stay in touch

If you would like to receive further insights and updates regarding this ebook:

Subscribe to the Newsletter here

newsletter

CHAPTER 2 - The heart of the site: creating the “Property” Content Type with its fields

If you have used WordPress, you are familiar with Custom Post Type UI and Advanced Custom Fields.
Two plugins. Two interfaces. Two updates. Two potential conflicts.
In Drupal, everything is right here. In the same place. Instantly.
In this chapter, we build the heart of our real estate site:
the **Content Type** “Property” with all the fields a real estate agency needs.
By the end, you will have your first listing online.
And you will know exactly how to create a hundred, a thousand, or tens of thousands more.

2.1 What is a Content Type?

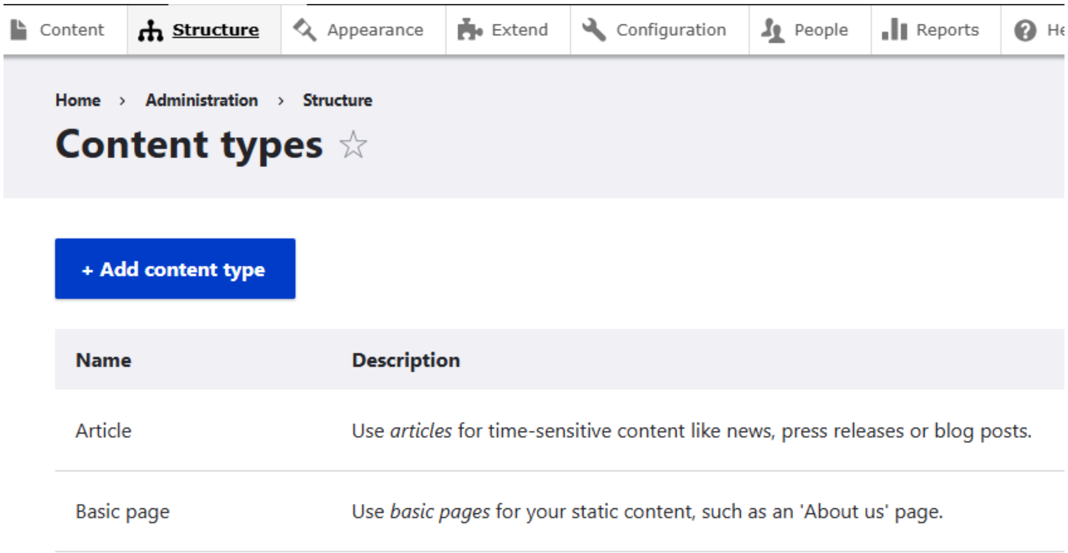
In WordPress, you have Posts and Pages.
If you want something different (listings, products, events), you must install a plugin.
In Drupal, the **Content Type** is a native feature.
A **Content Type** is a mold. It defines:

- Which fields each piece of content must have
- How it behaves (publishing, URL, comments)
- How it is displayed (we will see this later)

For our Beta Real Estate Agency, we need a mold called “Property”

Step 1 – Access the Content Type section

From the administration menu: **Structure** → **Content types**



Content types ☆

Name	Description
Article	Use *articles* for time-sensitive content like news, press releases or blog posts.
Basic page	Use *basic pages* for your static content, such as an 'About us' page.

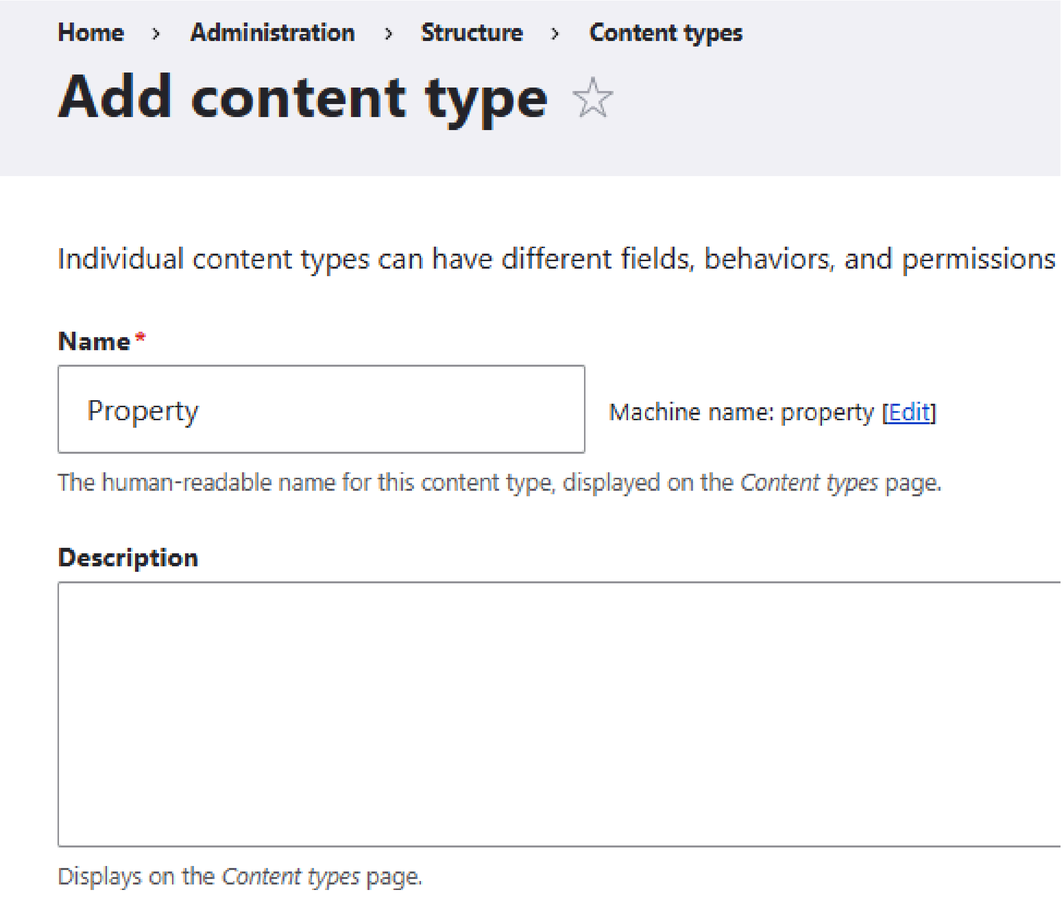
content type

What it shows: The main Content types page.
You will see a list with at least “Article” and “Basic page.”
At the top left, there is the **“Add content type”** button.
Click on that button.

Step 2 – Fill in the basic information

- **Name:** Property
- **Description:** *“Enter a real estate listing with price, surface area, photos, and features.”*

This description will help whoever enters the listings understand what they need to do.



Home › Administration › Structure › Content types

Add content type ☆

Individual content types can have different fields, behaviors, and permissions

Name*

Property

Machine name: property [Edit]

The human-readable name for this content type, displayed on the *Content types* page.

Description

Displays on the *Content types* page.

content type

What it shows: The form to create a new Content type.
Fields: Name, Description, various Settings.
Action: Fill in the Name and Description. Leave the rest as default.

Step 3 – Save

Click **Save**.

Well done.
Now you have the mold. But it’s empty.
We need to fill it with the fields that, in WordPress, you would create using a plugin like ACF.

2.2 Fields: the property data

Put yourself in the shoes of a real estate agent. *When entering a new listing, what do they need?*

Field	Description	Example
Title	Name of the listing	“Apartment with Cathedral view”
Price	Value in euros	250,000 €
Surface	Square meters	85 sqm
Rooms	Number of rooms	3
Floor	Property position	2nd, Ground, Top floor
Photos	Property images	Gallery

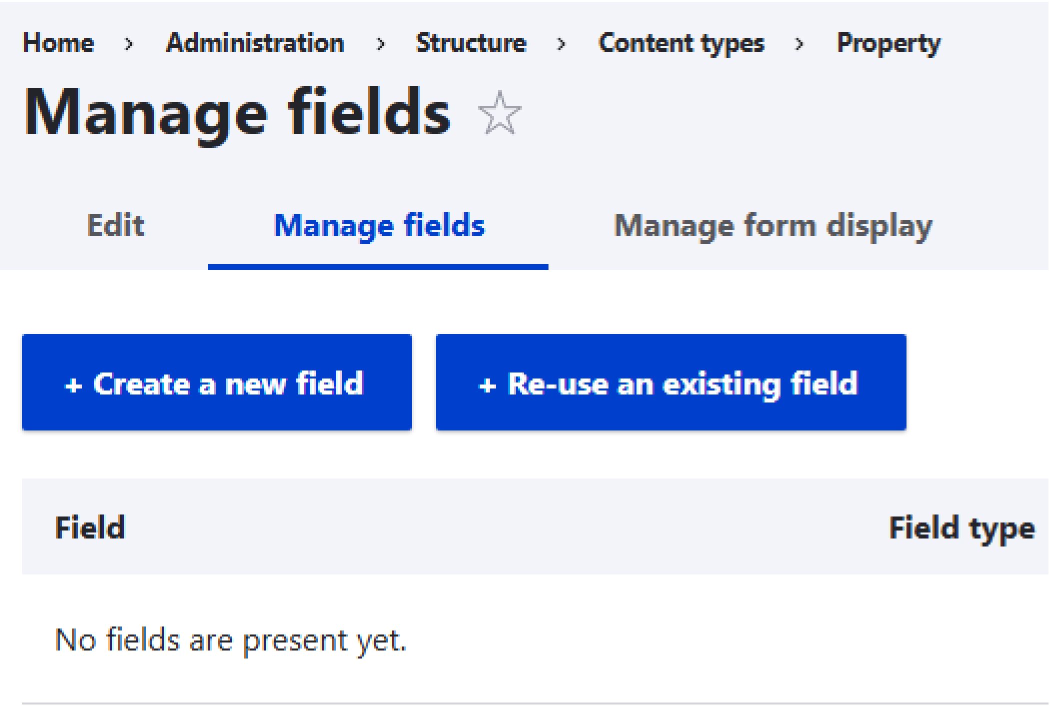
In this chapter, we will focus on these.
To be thorough, there would be much more information, such as the property’s energy class or an interactive reference map. However, at this stage, the goal is to become familiar with the Drupal CMS, so we will explore all these other elements in future Progettimultimediali blog articles distributed for free via our Newsletter.

Taxonomies (Type, Area, Features) will be added in **Chapter 3**.

2.3 The universal mechanism: adding a field

Every field in Drupal is added in the same way. **Where to do it:**

- Go to your **Property** Content Type
- Click on **Manage fields**


Home › Administration › Structure › Content types › Property
Manage fields ☆
Edit
Manage fields
Manage form display
+ Create a new field
+ Re-use an existing field

Field	Field type
No fields are present yet.	

content type

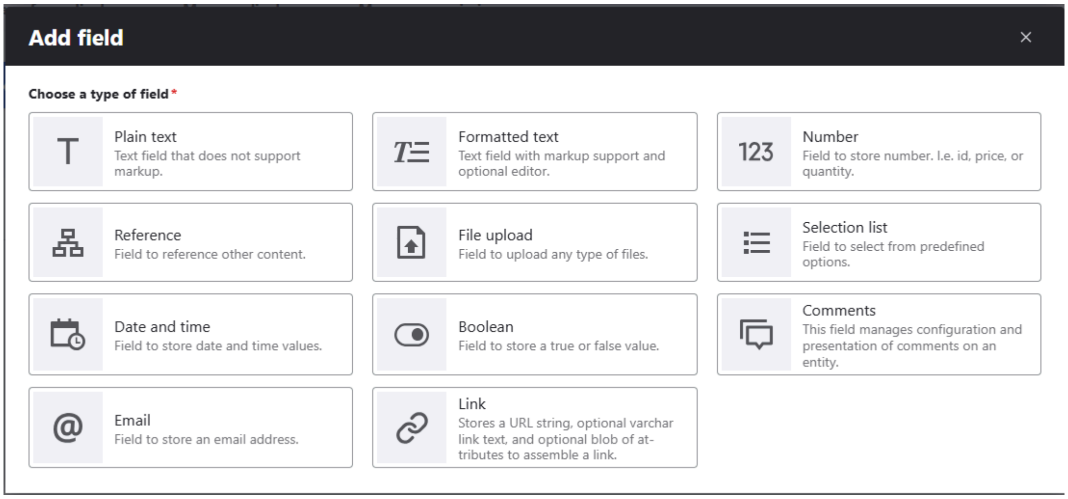
What it shows: The “Manage fields” tab of a Content Type.
By default (even if they are not immediately visible), there are already existing fields (“Title”).

The button: “Add a new field.”
Click on “Add a new field”.

The process is always the same:

1. Choose the field type (Text, Number, Image, etc.)
2. Give it a label (e.g., “Price”)
3. Configure the settings (minimum, maximum, dimensions)
4. Save

Learn this, and you learn it all.



Add field ×

Choose a type of field *

Plain text
Text field that does not support markup.

Formatted text
Text field with markup support and optional editor.

Number
Field to store number. I.e. id, price, or quantity.

Reference
Field to reference other content.

File upload
Field to upload any type of files.

Selection list
Field to select from predefined options.

Date and time
Field to store date and time values.

Boolean
Field to store a true or false value.

Comments
This field manages configuration and presentation of comments on an entity.

Email
Field to store an email address.

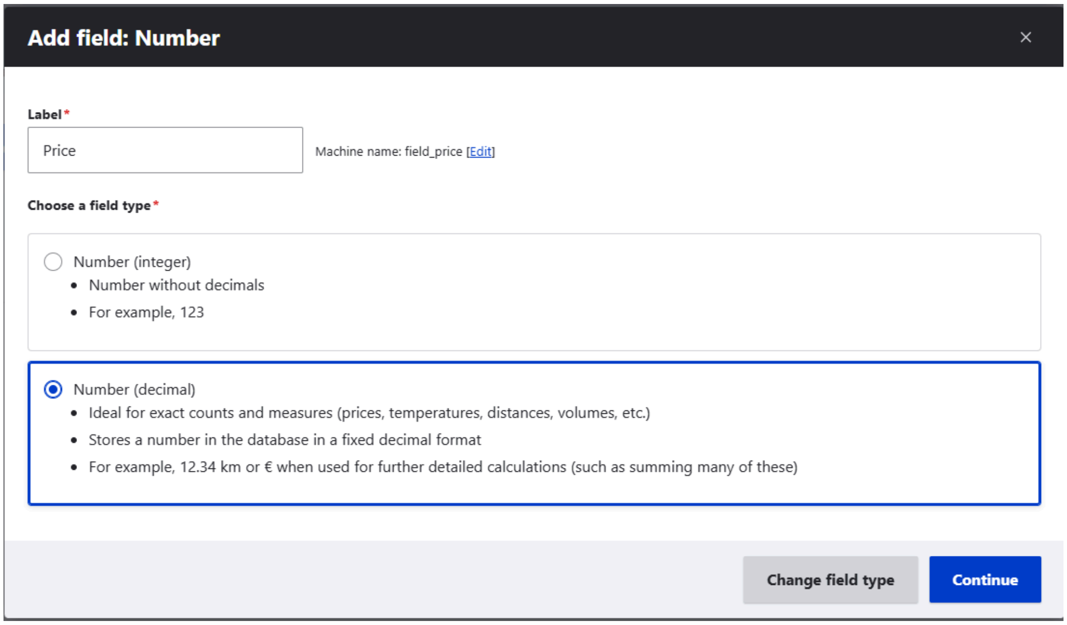
Link
Stores a URL string, optional varchar link text, and optional blob of attributes to assemble a link.

content type

What it shows: The field types tab.
Action: Click on the various fields you wish to create.

Adding the “Price” field

1. **Add field**
   - **Field type:** Number (decimal)
   - **Label:** Price



Add field: Number

Label*

Price

Machine name: field_price [Edit]

Choose a field type*

○ Number (integer)

- Number without decimals
- For example, 123

● Number (decimal)

- Ideal for exact counts and measures (prices, temperatures, distances, volumes, etc.)
- Stores a number in the database in a fixed decimal format
- For example, 12.34 km or € when used for further detailed calculations (such as summing many of these)

content type

What it shows: The Number field type.
Select "Number (decimal)".
Click on Continue

2. Field settings

- Minimum: **0**
- Maximum: **leave empty**
- Scale (Decimal places): **2**

3. Save.
Now, when you create a listing, a field will appear where you can enter the price with two decimal places.

Adding the "Surface" field

1. Add field

- **Field type**: **Number (integer)**
- **Label**: **Surface (sqm)**

2. Field settings

- Minimum: **0**
- Maximum: **leave empty**

3. Save.

Adding the “Rooms” field

1. Add field

- **Field type**: **Number (integer)**
- **Label**: **Rooms**

2. Settings

- Minimum: **1**
- Maximum: **20**

3. Save

Adding the “Floor” field (Selection)

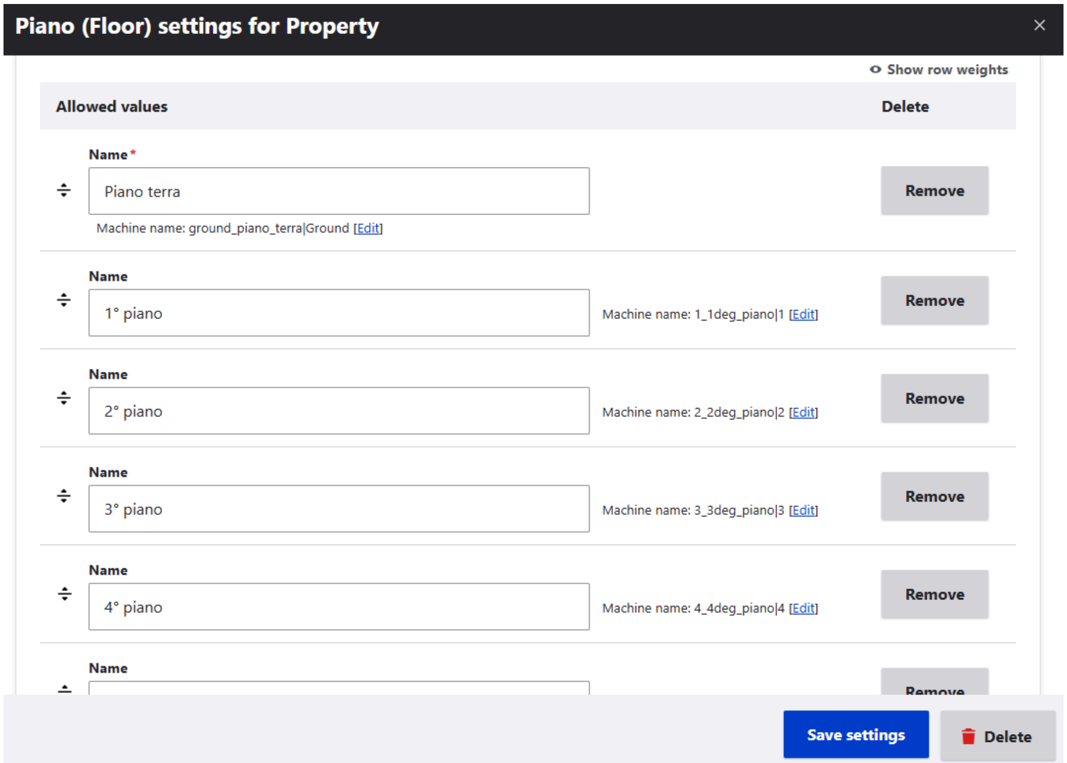
1. Add field

- **Field type**: **List (text)**
- **Label**: **Floor**

2. Allowed values

Enter this list (one row per option):

Ground|Ground floor
1|1st floor
2|2nd floor
3|3rd floor
4|4th floor
Top|Top floor



Piano (Floor) settings for Property

Allowed values

Name*
Piano terra
Machine name: ground_piano_terra|Ground [Edit]

Name
1° piano
Machine name: 1_1deg_piano|1 [Edit]

Name
2° piano
Machine name: 2_2deg_piano|2 [Edit]

Name
3° piano
Machine name: 3_3deg_piano|3 [Edit]

Name
4° piano
Machine name: 4_4deg_piano|4 [Edit]

Name

content type

What it shows: An example of a “List (text)” field with “pipe” or “vertical bar” values.

You will see the textarea where you can enter the options.
Copy the values above. Save.

What does the pipe (|) mean?

- Before the pipe: the **value saved in the database** (e.g., “Ground”, “1”, “2”)
- After the pipe: the **label the user sees** (e.g., “Ground floor”, “1st floor”)

3. Save

What does the pipe or vertical bar (|) mean?

- Before the pipe: the **value saved in the database** (e.g., “Ground”, “1”, “2”)
- After the pipe: the **label the user sees** (e.g., “Ground floor”, “1st floor”)

3. Save

Adding the “Photos” field (multiple)

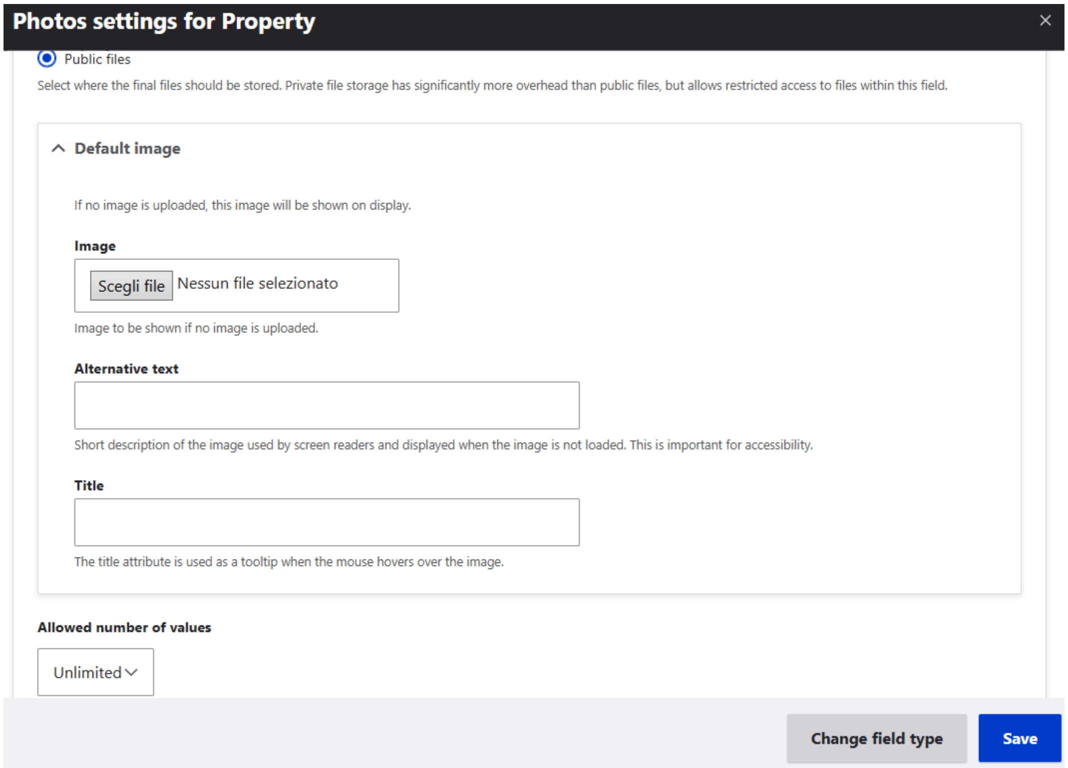
1. **Add field**

- **Field type**: **Image** – *File Upload → Image*
- **Label**: **Photos**

2. Settings

- **Number of values**: **Unlimited**

- **Maximum file size**: **5 MB**
- **Recommended resolution**: **1200x800** (optional)



Photos settings for Property ×

● Public files

Select where the final files should be stored. Private file storage has significantly more overhead than public files, but allows restricted access to files within this field.

Default image

If no image is uploaded, this image will be shown on display.

Image

Scegli file Nessun file selezionato

Image to be shown if no image is uploaded.

Alternative text

Short description of the image used by screen readers and displayed when the image is not loaded. This is important for accessibility.

Title

The title attribute is used as a tooltip when the mouse hovers over the image.

Allowed number of values

Unlimited

content type

What it shows: The settings for an Image field.
You will see the "Number of values" option and the file size limit.
Set "Number of values" to "Unlimited".

3. Save Now you can upload as many photos as you want for each listing.

Summary: here is what you have built

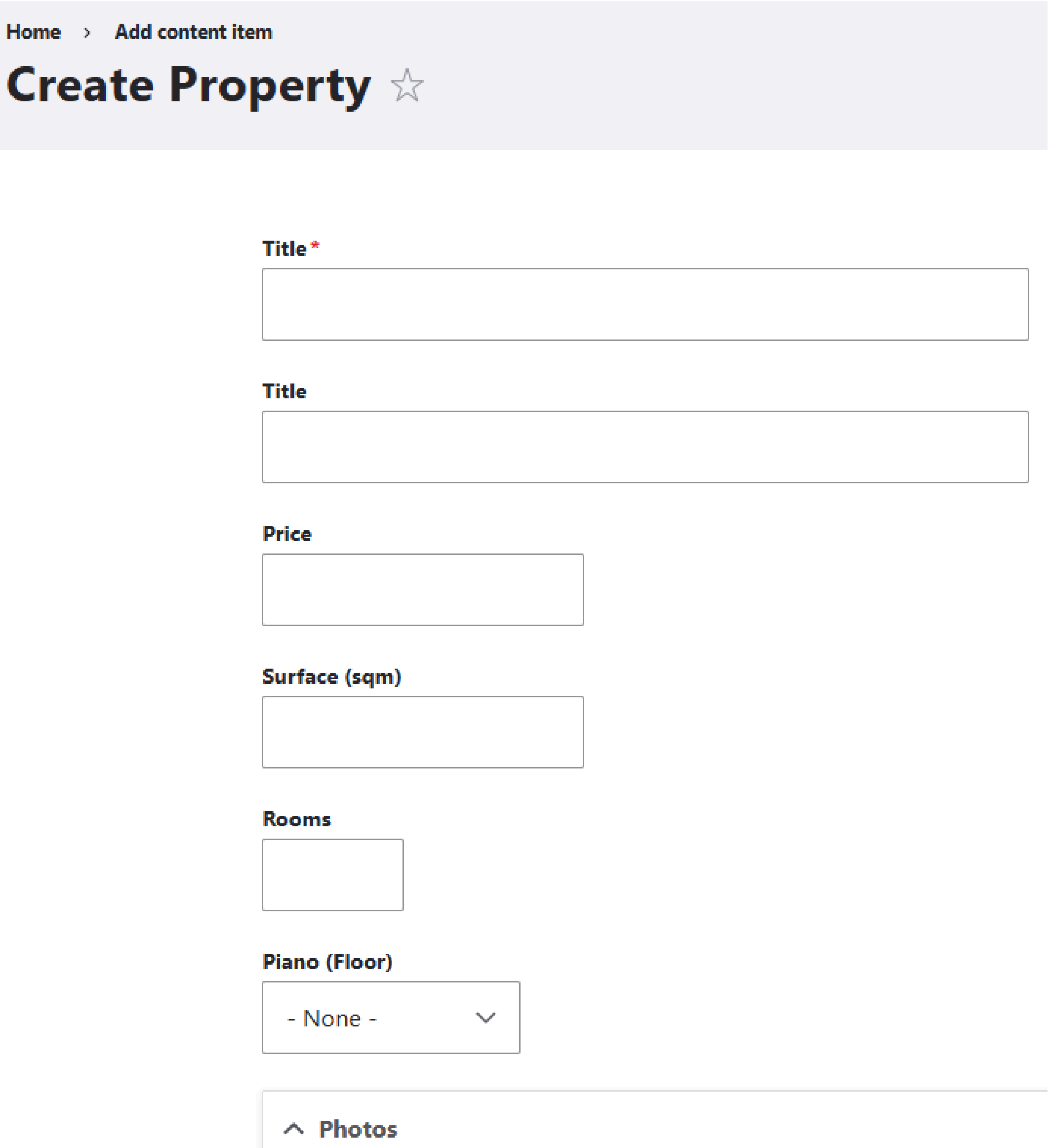
Field	**Type**	**Purpose**
Title	Text	Listing name

Field	Type	Purpose
Price	Number (decimal)	Economic value
Surface	Number (integer)	Square meters
Rooms	Number (integer)	Number of rooms
Floor	List (text)	Selection from a list
Photos	Image	Image gallery

Without a plugin. Without code. Without costs.

The test: creating the first listing Now let’s verify that everything works.

Go to → Content → Add content → Property



Home › Add content item

Create Property ☆

Title*

Title

Price

Surface (sqm)

Rooms

Piano (Floor)

- None -

^ Photos

content type

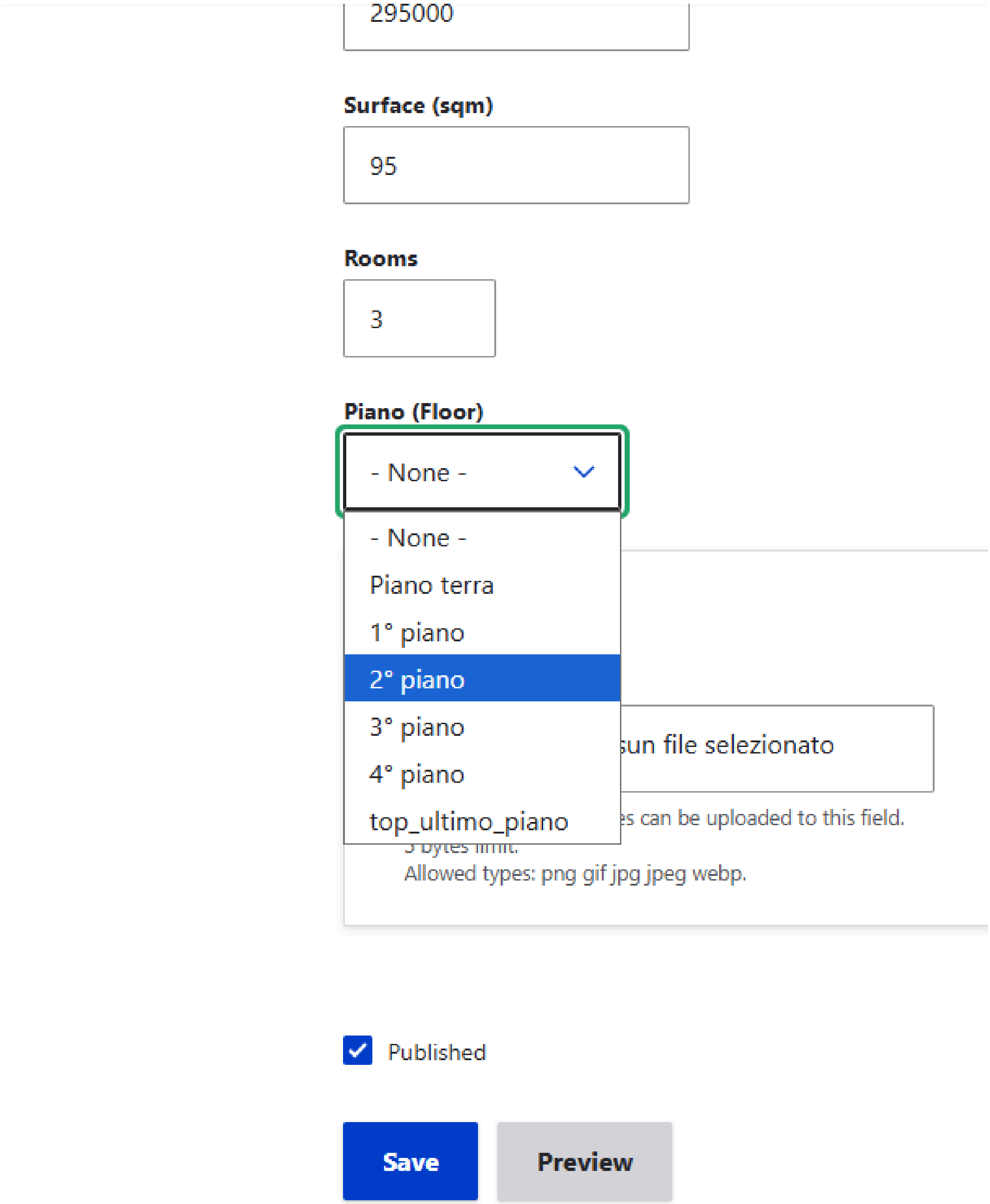
What it shows: The input form for a custom Content Type.
You will see the fields you just created: Price, Surface, Rooms, Floor, Photos.
Fill in the fields. Upload some photos.

Save.

2. Fill in the fields:****

- **Title**: Bright apartment in the city center
- **Price**: 295000
- **Surface**: 95
- **Rooms**: 3
- **Floor**: 2nd floor
- **Photos**: Upload 2-3 images

3.Click Save



295000

Surface (sqm)

95

Rooms

3

Piano (Floor)

- None -

un file selezionato

s can be uploaded to this field.

5 bytes limit.

Allowed types: png gif jpg jpeg webp.

☑ Published

content type

What it shows: The full view of a Node.
You will see the title, price, surface, rooms, floor, and the uploaded photos.
Here is your first listing online.

You have just created your first Node (content) of the Property type.
This is not an exercise.
It is the first brick of your real estate website.

The difference you will see (and your client won't) In WordPress, that data ends up in the `postmeta` table of the MySQL database:

post_id	meta_key	meta_value
123	price	295000
123	surface	95
123	rooms	3

Ogni campo è **una riga**.
Per trovare "tutti gli appartamenti sotto i 300.000€ con almeno 80 mq", WordPress deve **scandire milioni di righe** e confrontare stringhe.
In Drupal, ogni campo è **una colonna**:

nid	title	field_price	field_surface	field_rooms
123	Downtown Apartment	295000.00	95	3
124	Villa with Garden	550000.00	120	5

The query is direct. Indexed. Fast. This isn't theory.
It's the reason why Drupal can handle portals with hundreds of thousands of listings, while WordPress cannot.

WordPress and Drupal: It's not a competition

Use WordPress for:

- Blogs, showcase sites, small e-commerce.
- Projects that won’t grow beyond 500 content items.
- Clients with limited budgets and simple needs.

Use Drupal for:

- Portals, marketplaces, intranets, complex multilingual sites.
- Projects that are small today but will be large tomorrow.
- Clients who want to invest in a solid structure.

You don’t have to choose. You can have both in your arsenal.

CHECKPOINT – OUR SITE STATUS

Now we have:

- A Content Type named **“Property”**
- The fields: **Price, Surface, Rooms, Floor, Photos**
- The first listing entered and published
- A high-performance data structure ready to scale

We don’t have yet:

- A system to classify properties (**Type, Area, Features**)
- A search page
- A customized homepage
- Agent management

Everything else is coming in the next chapters.

FOR THOSE IN A HURRY

- **Content Type** = the mold. In Drupal, it’s native, no plugins required.
- **Fields** = the specific data. You add them from the UI in 30 seconds.
- **Each field is a database column** → superior performance on large volumes.

Today you built the container. Tomorrow you learn how to classify the content.

Farewell… and See You Next Time!

We have reached the end of this journey into the world of Drupal. If you've made it this far, you've learned that Drupal isn't that difficult "sacred monster" reserved only for high-level developers. You have seen that:

- Content is created with ease.
- Taxonomies organize the chaos.
- Tailor-made Content Types are not a luxury.
- Appearance is separated from logic.
- Multilingual support is native, not an afterthought.

You have transformed a "naked" Drupal installation into something with a personality: **the Beta Agency website.** But this is not the finish line. It is the starting point.

Continuing Your Training

The Drupal world is constantly evolving, and with it, the possibilities for creating even better sites. Here is how we can stay in touch and continue this path together:

****The Newsletter** Every week in your inbox:

- Previews of upcoming ebooks
- Practical Drupal tips
- Articles from the Blog

Subscribe to the Newsletter here

The YouTube Channel Do you prefer watching over reading? On our channel, you will find:

- Step-by-step video tutorials
- In-depth webinars
- Real-world case studies
- Video versions of ebook excerpts and blog articles

Subscribe to the YouTube channel

The Website All materials, updates, resources, and upcoming ebooks can be found on our site:

Visit the official website

Looking Ahead: My Vision Beyond the CMS

If you have reached the end of these pages, I hope one thing is clear: my goal was not to convince you that WordPress is “wrong,” but to show you that a vaster territory exists—one where you can finally regain control over your projects.

For years, I too lived the frustration of having to force a tool to do what it wasn’t born to do. I lived the dependency on third-party plugins and that sense of limitation that appears when a client asks for “something more.” When I chose to specialize in Drupal, I did so because I was looking for stability and, above all, professional freedom.

My Advice for Your Path

Today, I no longer see Drupal merely as a CMS, but as the perfect engine for the challenges of the future. Whether it’s integrating Artificial Intelligence into workflows or pushing toward **Headless architectures with Next.js**, Drupal gives me the certainty of standing on steel foundations.

My invitation to you is not to stop at the theory. Start getting your hands dirty: model your first Content Type, understand the logic of Views, and you will see that, step by step, what seemed like a steep learning curve will become your ladder toward more ambitious projects and better margins.

Let’s Stay in Touch

This ebook is only the beginning of a journey. With **Progetti Multimediali**, my mission is exactly this: to accompany professionals through this leap in quality, transforming technological complexity into a real competitive advantage.

The web is changing fast, and those who master data structure and enterprise-level security will always be one step ahead of the rest. I have made my choice. I’ll be waiting for you on the other side, ready to build something truly solid.

Happy coding and happy evolution.

Daniele Venditti