Add a dropzone | WeWeb Developer Docs

|  |  |
| --- | --- |
| URL source | <https://developer.weweb.io/add-dropzone.html> |
| Date de scraping | 2025-04-08T15:52:24.514720+00:00 |

## Add a dropzone ​

In WeWeb you can easily create a dropzone, an area where the user will be able to drop in any element. It's useful to build very flexible components.  
See the Development process to load a base section in dev mod.  
We will add a header dropzone instead of our title. The user will be able to drop containers, icons, etc...

## ww-config.js ​

The first step is to declare a new property which will hold the references to all the elements dropped.  
We declare this property on the properties field of the configuration:

### Code

Langage: unknown

properties: {  
 // [...]  
 header: {  
 hidden: true,  
 defaultValue: []  
 },  
}

## Use the value inside the template ​

We will use a special component: wwLayout .  
Full documentation for wwLayout possibilities here  
We also need to add some style  
This is the simpliest way to use wwLayout .  
We just need to provide the path property which is the path inside the content to access our previously created array.  
We also give a direction so that the drop zone appears in the right way.

### Code

Langage: unknown

<div class="my-section">  
 <wwLayout path="header" direction="column" class="header"/>  
 <p>  
 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et  
 dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex  
 ea commodo consequat.  
 </p>  
</div>

Langage: unknown

.my-section {  
 // [...]  
 .header {  
 min-height: 20px;  
 display: flex;  
 flex-direction: column;  
 }  
}

## Customize the item ​

wwLayout by default just repeats the elements. But it also has a scoped slot you can customize to add extra style or property. This is useful to create a card for example or for the editing experience.  
Here is an example:  
You can see more options of wwLayout here

### Code

Langage: unknown

// ww-config.js  
properties: {  
 // [...]  
 cards: {  
 hidden: true,  
 defaultValue: []  
 },  
}

Langage: unknown

<div class="my-section">  
 <wwLayout path="cards" direction="row" class="cards">  
 <template v-slot="{ item }">  
 <wwLayoutItem class="card">  
 <wwObject v-bind="item">  
 </wwLayoutItem>  
 </template>  
 </wwLayout>  
</div>

Langage: unknown

.my-section {  
 .cards {  
 display: flex;  
 justify-content: space-around;  
 }  
 .card {  
 border: 1px solid dashed;  
 padding: 4px;  
 border-radius: 8px;  
 min-width: 100px;  
 }  
}

## Bind the content ​

If you want, you can mark your content property linked to your dropzone as bindable, with a special binding: repeatable  
When you do that, wwLayout will repeat its first child element for each item in the collection list bound to it, and it will set a binding context for each item.  
You can learn more about binding here

### Code

Langage: unknown

// ww-config.js  
properties: {  
 // [...]  
 cards: {  
 hidden: true,  
 bindable: 'repeatable',  
 defaultValue: []  
 },  
}