Add another element as property | WeWeb Developer Docs

|  |  |
| --- | --- |
| URL source | <https://developer.weweb.io/add-element-property.html> |
| Date de scraping | 2025-04-08T15:52:25.887084+00:00 |

## Add another element as property ​

## Why? ​

Sometimes we will need to have other element as children, and have more control on it than a dropzone.  
For example, to add an icon inside a button or to display a text where you forced the text value, but want to allow users to change its style.  
This is also a common pattern to initialize a dropzone content.

## Add an element on creation ​

See the Development process to load a base section in dev mod.  
We will replace the title by a ww-text component.

## ww-config.js ​

The first step is to declare this property on the properties field of the configuration:

### Code

Langage: unknown

properties: {  
 // [...]  
 title: {  
 hidden: true,  
 defaultValue: { isWwObject: true, type: 'ww-text' },  
 },  
}

## Use the value inside the template ​

And it's done!

### Code

Langage: unknown

<div class="my-section">  
 <wwElement v-bind="content.title" />  
 <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>

## Add a forced props ​

Some elements can take special props when they are used inside another one.  
This is the case for ww-text : you can force the text content. In that case, users will no longer be able to edit the text content but can still customize the style of the element.  
This is very useful when your text is computed by internal logic. This is what we use for the Paginator element for example.

### Code

Langage: unknown

// inside ww-config.js  
properties: {  
 // [...]  
 priceElement: {  
 hidden: true,  
 defaultValue: { isWwObject: true, type: 'ww-text' },  
 },  
}

Langage: unknown

<div class="my-section">  
 <wwElement v-bind="content.priceElement" :ww-props="{ text: computedPrice }"></wwElement>  
</div>

## Add an element dynamically ​

Sometimes, you will need to create an element after your component has been mounted, in response to a property change for example.  
Be aware that element creation is only available in the Editor context, so your code needs to be stripped from the production build. (See stripping here )  
Example: you have a toggle to indicate if an icon is present or not. You create the icon element only when this option is active.  
You can use wwLib.createElement for that or wwLib.createElementFromTemplate , and the update:content event:

### Code

Langage: unknown

export default {  
 // [...]  
 methods: {  
 createIcon() {  
 const icon = wwLib.createElement({ type: "ww-icon" })  
 this.$emit('update', { icon })  
 }  
 }  
 // [...]  
}