

29

216: Official WP Block Dev

Block.json is new

@wordpress/scripts has more Utility

Interactivity API is new

Summary: introduction to the new
Wordpress tools

29

217: How to Setup Multi Blocks

New File structure: SRC

- SRC is needed to have @wordpress/scripts detect each Block.Json File inside



Inside block.json

```

{
  "$schema": "https://schemas.wp.org/trunk/block.json",
  "apiVersion": 3,
  "name": "ourblocktheme/footer",
  "title": "Fictional University Footer",
  "editorScript": "file:./index.js",
  "render": "file:./render.php"
}
  
```

- This is the Blocks config file

Inside functions.php

- Setup Function to create new blocks

```

// Register our new blocks
function our_new_blocks() {
    register_block_type_from_metadata(__DIR__, './build/footer');
}
  
```

Inside package.json

- Add "blocks" to the scripts

```

"blocks": "wp-scripts start --experimental-modules",
  
```

Inside render.php

- Insert HTML that needs to be displayed in Front end

Inside index.php

- this file is used to display block in admin

```
import { registerBlockType } from "@wordpress/blocks"
import metadata from "../block.json"

function Edit() {
    return <div>Hello, I am the footer block in the editor.</div>
}

registerBlockType(metadata.name, {
    edit: Edit
})
```

Files Changed

SRC/Footer/

Block.JSON
functions.php
Render.php
index.php

themename/

Package.JSON

29

218: Practice makes Perfect

Start by running 2 commands

- npm run start
- npm run build

Inside index.JS

Removed edit Function
edit.JS File

into its own

Inside edit.JS

- Add useBlockProps so element can be selected
in Admin editor

```
import { useBlockProps } from "@wordpress/block-editor"

export default function Edit() {
  const blockProps = useBlockProps()

  return (
    <div {...blockProps}>
      <div className="our-placeholder-block">University Footer Place
    </div>
  )
}
```

Note: Remember for each new block to create its own Folder in SRC and register it within functions.php

Files Changed

SRC/Footer/

index.JS
edit.JS

29.

219: Our Banner Block

- Making Banner Block Interactive

NOTE: An old Banner Block already exists so we are just moving components

Inside edit.js

- We add the imports from the old Banner block

```
import apiFetch from "@wordpress/api-fetch"
import { Button, PanelBody, PanelRow } from "@wordpress/components"
import {
  useBlockProps,
  InnerBlocks,
  InspectorControls,
  MediaUpload,
  MediaUploadCheck
} from "@wordpress/block-editor"
import { registerBlockType } from "@wordpress/blocks"
import { useEffect } from "@wordpress/element"
```

Inside block.json

- we add any "supports" and "attributes" to the existing JSON

NOTE: We will address background image later

```
{
  "$schema": "https://schemas.wp.org/trunk/block.json",
  "apiVersion": 3,
  "name": "ourblocktheme/banner",
  "title": "Fictional University Banner",
  "supports": {
    "align": ["full"]
  },
  "attributes": {
    "align": { "type": "string", "default": "full" },
    "imgID": { "type": "number" },
    "imgURL": { "type": "string" }
  },
  "editorScript": "file:./index.js",
  "render": "file:./render.php"
}
```

Inside edit.js

Add JSX & Functions From Old Banner

Note: don't forget to add props in edit function

Copy and paste these two blocks of code

```
useEffect(  
  function () {  
    if (props.attributes.imgID) {  
      async function go() {  
        // ...  
      }  
    }  
  },  
  [props.attributes.imgID]  
)  
  
function onFileSelect(x) {  
  props.setAttributes({ imgID: x.id })  
}
```

Extra Code

```
<InspectorControls>  
  <PanelBody title="Background" initialOpen={true}>  
    <PanelRow>  
      <MediaUploadCheck>  
        // ...  
      </MediaUploadCheck>  
    </PanelRow>  
  </PanelBody>  
</InspectorControls>
```

Extra Code

```
<InnerBlocks allowedBlocks={['  
  // ...  
</div>  
</div>
```

~ Stopped @ ~

12:40