

Hello, everyone.

In this lesson, we're going to start a brand new block type just for a quick review.

Here's a sneak preview of the finished product of what we're going to build together.

So from the edit post screen, if I try to add a new block right about here, there's one called Professor

Callout.

The idea here is within a blog post, this will let me sort of feature or highlight a professor, right.

One of our posts of the professor post type.

So as soon as I click to add that this new area appears, there's just this select dropdown.

And if I click on it, there's only three because I only have three professor posts on my website.

But if you had 20 or 50, it would show all of them here.

And then you just select the one that you want to highlight or feature here.

So I'll click Dr. Rogerson and now we see a live preview appear right here.

And if I click update and then we go view this on the front end.

Well, if we scroll down to that area, cool.

There it is.

And I do want to point out that it's not as if we pulled the data for Rogerson and then stored a static

snapshot of it within this blog post.

Instead, this is actually pulling the real current values from the Frogger sign post.

So if we went in and edited the professor post for Rogerson, changed the text here, if we reloaded

this blog page, we would see that change reflected here.

So really, this blog post is just storing the ID of the professor post as a piece of metadata for this

blog post.

And then we just run a query to load in the actual post data for that.

Professor.

That way it's all truly dynamic and we're learning how to create a real relationship between two pieces

of content.

And we do want to make sure that that relationship is a two way street.

What I mean is if we click this link to view the professor detail screen, well, if we scroll down a bit, we can see this new area.

Dr. Rogerson is mentioned in the following posts.

So this will be a good bit of practice to make sure we know how to display the relationship from both

sides, from both angles.

At this point, let me switch back to the tab where I have not built the featured Professor Post type

and let's start moving forward together.

Now, at this point in the course, you and I have created at least three plug ins together.

So to save a bit of the boilerplate typing of code that has zero new concepts to explain and understand,

I actually want you to pause the video right now and download the zip file associated with this lesson

called Starter Zip.

So please follow along with me.

Do that right now, pause the video, go download this file and then when you extract the zip file,

it should create a folder named Featured Dash Professor.

If it doesn't, if it creates a folder named starter, you can just rename the folder to be featured dash professor.

Also make sure that it doesn't have a folder nested inside a folder.

So it should just be featured.

Professor And then inside it there should be one PHP file and one folder called CRC.

Anyways, we just want to move this folder into the WordPress plugins folder and don't worry we will

review the code that is in the starter files.

I just think it's a good idea to avoid typing out some of this boilerplate again and again.

So to move this into your WordPress plugins folder, just find where your WordPress is installed.

Go into the WP content folder and then go into the plugins folder.

And then I'm just going to drag featured professor into this folder.

Then let's open up this folder in vs code.

Let's start by looking at the PHP file.

So if we open that up, it's just going into PHP.

It has the basic comment.

You can add your name here if you want to.

This line of code prevents executing this file directly by visiting its URL.

And then just like all of our other plugins, I'm creating a class.

I have my constructor function, I'm using the init action hook and then I'm giving it a method right

here where I'm registering a script, registering a style, and then most importantly, registering a block type.

So this should look very familiar from our quiz block type.

Right now I just have the render callback as a function that returns a paragraph.

We will obviously work on this together.

And then down at the very bottom, I'm just creating a new instance from our class.

So that's it for that file in our SRC folder.

We don't need to go into the CSS file.

It's just a bit of basic CSS because this course really is about WordPress and JavaScript, not CSS.

But this will give us a bit of basic styling for the block and then inside the source folder, we also

have index dot JS.

So I'm just importing the sassy CSS file so that the WordPress JavaScript package will extract it into

its own file.

But we do want to include it in sort of the file tree within our JavaScript here.

And then just like before, I'm registering the block type on the JavaScript side of things and then

for the edit component, that's what you see in the admin area.

Well, for now the function just returns a bit of JS X, that's just an overall div with two divs inside

it.

Now this class name here does need to be exactly this just for a bit of CSS reasoning, but hopefully

now you feel a bit at ease using this boilerplate code.

There's nothing advanced or magical going on, so at this point we can start to work together.

The first thing we need to do is install the official WordPress JavaScript package, right?

The workflow tool.

So open up your command line that's control J on Windows or command j on Mac and run this command.

It's.

PM.

Init and then a space dash y.

Let's press enter.

So that will create our packaged JSON file.

It will keep track of our ingredients, but for now we only need one ingredient.

So the only thing we need to install is this.

We say npm install at simple WordPress slash scripts and that's it.

Go ahead and press enter and that will take perhaps a minute or two or maybe even a little bit longer.

So please feel free to pause the video until your installation completes.

Once it completes, we'll want to run our NPM run start command.

But we first need to set that up so really quick.

Just jump into your package JSON file with me and you could actually just copy and paste and borrow

this code from the previous plugin we worked on together.

But inside of scripts we just want to add two extra lines here.

So right above the test line we'll say quotes, build colon, give it a string of text, WP Dash scripts,

build.

Be sure to end that line with a comma on a new line.

Let's say quotes, start colon quotes, WP scripts start and again.

Be sure to end that with a comma.

Let's give that a save and now we can indeed start running our task.

So it's just npm run start.

Okay.

You'll know that it's working.

If all of a sudden you see a new folder on the left hand side here called Build and it should have our

CSS and our JavaScript.

Cool.

So our boilerplate code in the PHP file is already loading the JavaScript and CSS from the build folder.

So now let's just go into the admin side of WordPress and activate our plugin.

So under plugins.

Just look for the new plug in.

So it's called featured Professor Bloch type.

Let's go ahead and activate that.

And let's go test it out.

So I will go into one of my blog posts.

And you can try to add it anywhere.

But for me, I'll add it maybe down here I immediately see this option called Professor Callout and

if I click on that perfect, it should look like this for you.

So there's this gray area that says we will have a select dropdown form element here.

And then also the HTML preview of the selected professor will appear here.

So where do we start from here?

Well, let's just start building the actual select dropdown element right here.

I think once that's actually in place here, even if it's just empty, it'll give us something to sink our teeth into and to build towards.

So back in our SR folder, jump into indexed JS down in the JS X, we're looking for this div that has

a class of professor select container.

So inside that, let's get rid of this placeholder text and let's instead have you can just type, select

and then hit the tab key.

We actually don't need name or ID, we just want opening and closing select tags.

And for now let's just hard code a few fake options.

So inside the select tag, let's say option hit tab for now, let's just say value of one and the readable

text is one and then you can just duplicate that line two more times.

So let's just have a fake one, two, three.

So that's two.

This one is three.

So if we save that and then refresh, well, I actually forgot to save this post, so I'll need to insert

the block type again.

Professor Callout.

So now we have this select dropdown.

Now imagine instead of just 1 to 3, if this was a list of all professor posts and the actual readable

text values were.

Dr. MEOWS A lot.

Dr. Barks A lot, frog.

Everson But then here's the crucial point.

The actual value for each one was just their post ID, right?

Which is just a number.

So maybe Doctor MEOWS a lot is number 47 and Barks a lot is 48 and Frog Person is 49.

Everyone will have different post ID values, but the point here is that whatever that post ID that

they select is we just want to store that as an attribute for this block type.

So let's go set up an attribute for that.

So if we scroll up just a little bit when we're registering our block type, we have title description,

icon, category, edit, save, well, maybe right above the edit line.

So below the category line, let's say attributes to TS at the start.

So attributes colon give it an object.

Be sure to have a comma right after that.

Inside the object I'm going to drop down now.

We can make up any name we want for this attribute.

I'm going to name it Prof ID colon, curly brackets.

And let's say the type is a string.

We know the post ID is actually a number, but in terms of how WordPress actually loads data out of

the database and just for general string comparison, I've had better luck using a string as the type

here with this in place.

Let's go back down to our select element and make it work with that attribute.

First of all, before we forget when we have our function definition for edit component here within

the parentheses for edit component, be sure to include a parameter and remember the industry standard

is called props.

I should have included this in the boilerplate code for us, but this is a good example, a good lesson

to remember to include that here.

Anyways, on the opening select tag, let's just give it an on change prop.

So equals curly brackets and we can just include an s six arrow function.

So I'll say e short for event arrow symbol don't need curly brackets.

We can just say props dot set attributes, parentheses to actually call it give it an object.

We just want to change the property or attribute that's named Prof ID Colon and then it would just be

e dot target dot value OC inside our options.

Why don't we actually have a very first option?

So I'll just say option one more time and let's have its readable text.

Say, select a professor.

Finally, let's set things up so that once you select one, even if you reload the page, it will take

the existing value from the database and it will know which one to sort of mark as selected so that

when you revisit the page it already appears highlighted.

So first, let's set that up for this number one entry.

You would just say selected equals.

And then if you give it a value of true, it will appear to be the selected option.

But we don't want to just hard code it to true.

Instead we can compare props, dot attributes, dot profile ID double sign to check for equality if

that equals well for this case, just number one.

So then if you wanted to, you could just copy that.

And use that here as well on the third line as well and just change these values that we're checking

for to two and three.

Let's give this a save and test it out.

So back in the admin panel, I'm going to remember to save this post so I don't have to insert that

block type again.

But now if I refresh.

And if I find that block, select a professor.

If I pick professor number two and then actually save and update the post and then reload the page.

I'm just trying to make sure that WordPress is actually remembering that value.

And if we scroll back down, perfect.

It remembered number two.

Now it might not seem like a big deal to have just a hard coded list of one, two and three that remembers

your preference.

But we're actually further along than you might realize because now we are laser focused on what our

next step needs to be.

We need a way, using client side JavaScript to load a list of all of our professor posts, right,

so that we can visibly display their names but also have their post ID numbers as the value here.

Well, that's exactly what we're going to learn how to do in our very next lesson.

Let's keep things rolling and I'll see you then.

このコースについて

Learn PHP, JavaScript, WordPress theming & the WP REST API to Create Custom & Interactive WordPress Websites

基本情報

スキルレベル: すべてのレベル

受講生: 107388

言語: 英語

字幕: はい

レクチャー: 212