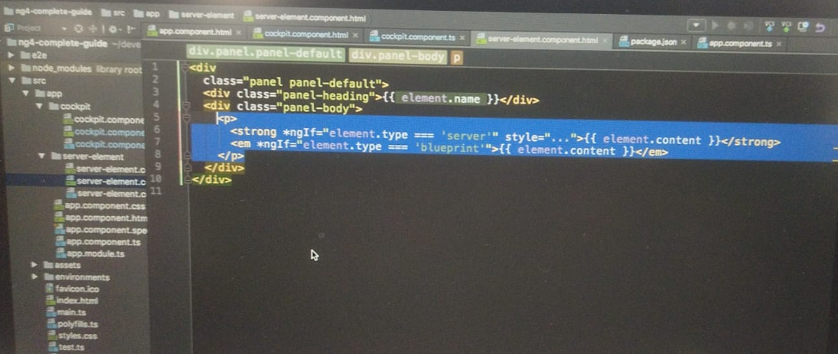
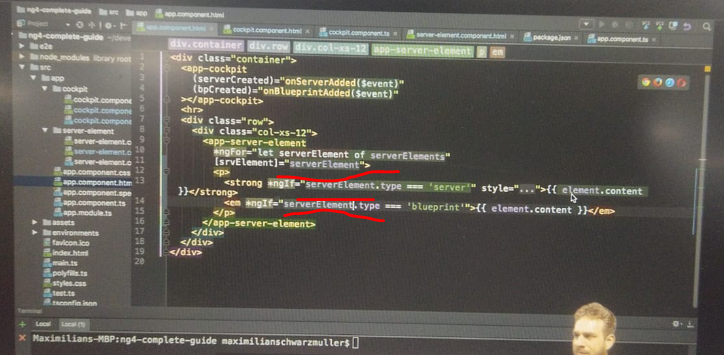
**76. Projecting Content into Components with ng-content**

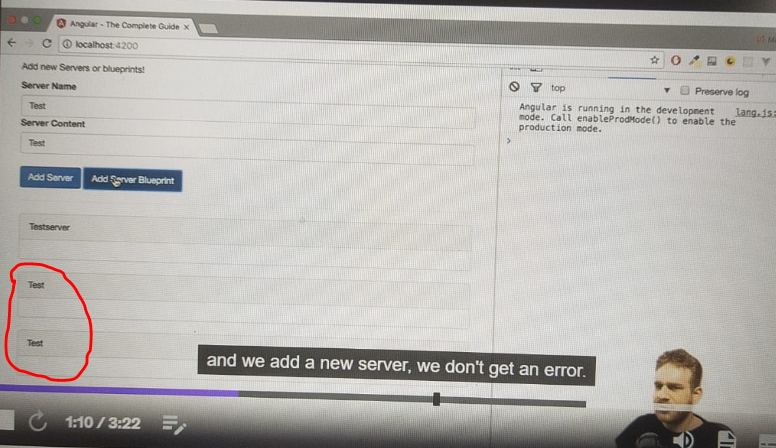
* -: So we learn a lot throughout this section.
* We learned how to pass data around.
* How to get access to elements in our dom.
* How to use local references.
* So our application now is much more dynamic.
* We got much more tools to interact between the different pieces we have in our app.
* Now there is another way we can pass data around.
* The last way for now in our server element component here.
* Right now we check if we have a type server or a type blueprint server, right? Nothing wrong with that but sometimes you have complex HTML code like this which you want to pass into a component from outside.
* So imagine you don't want to set this up inside the component here.



* I cut it instead in the app component you some want to add this here between the opening and closing element of your own component.
* And there we simply would have to replace element with server element.



* And we should expect us to work, right? So if we saved us and let it reload and we add a new server, we don't get an error.



* But the content is also not there and that is the default behavior.
* ***Everything you place between the opening and closing tag of your own component is lost by default.***
* It's simply removed from the dom.
* Angler will not take care about it, but you can change this.

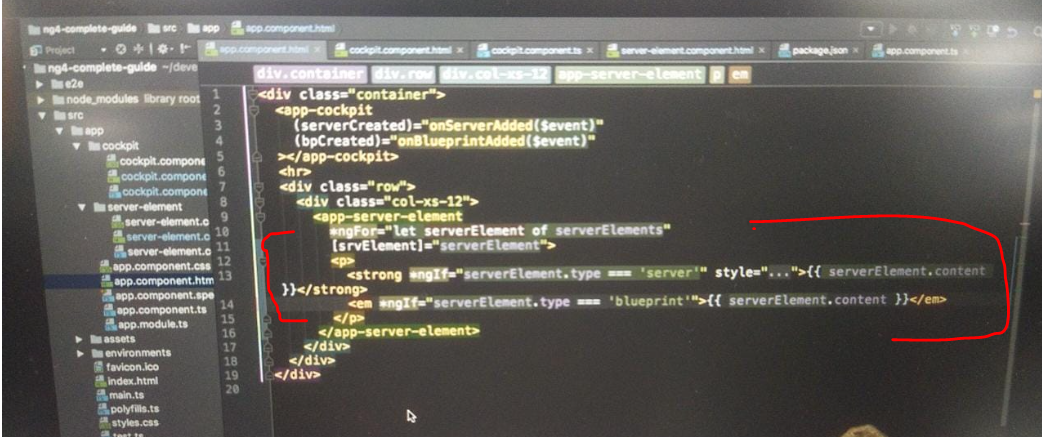
**<ng-content> directive:**

* There is a special directive and it is a directive even though it looks like a component but it doesn't have its own template.
* A special directive you can add in these server element components template.
* Here in the place where I want to render the content, I can add NG content opening and closing.

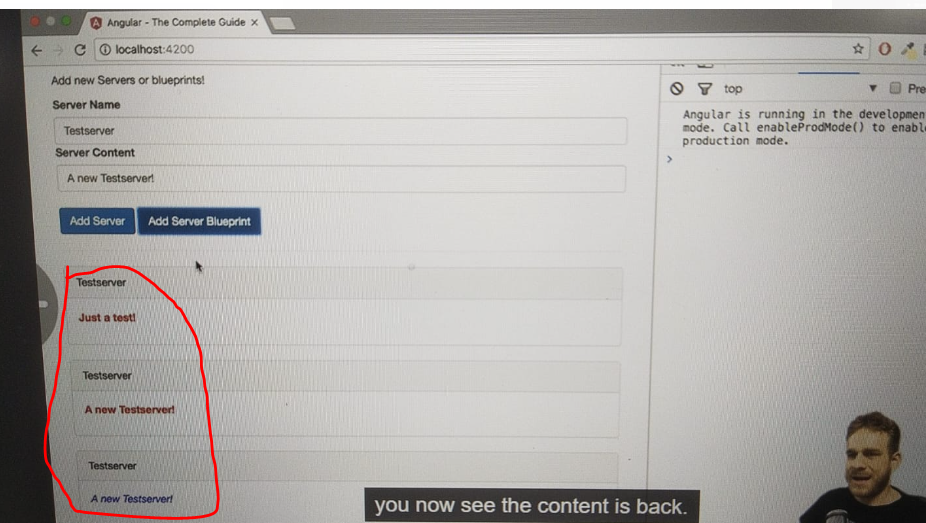
Graphical user interface, text

Description automatically generated

* *Again, it still is a directive just using this element like selector, and this serves as a hook you can place in your component to mark the place for Angular where it should add any content it finds between the opening and closing tag here.*



* So this content in this case with this tiny addition you should see that once you saved this and let it reload.
* If we add this and we click add server you now see the content is back.



* It looks exactly the way it looked before but technically something totally different works or happens here.
* Now we add this via the NG content hook.
* We add it between the opening and closing tags and therefore it will be projected into your component, projected into your server element component.
* A nice feature, especially if you think about building reusable widgets like a tab widget where each tab will have a content which probably comes from some other source and which you don't want to pass through property binding which always would be an alternative.
* But if it's more complex HTML code property binding really is not the best solution because Angular will escape HTML texts there to prevent cross sites scripting attacks from happening.
* And you could kind of work around that.
* But really energy content, that is how you want to display this and that is a great tool to have at your disposal.