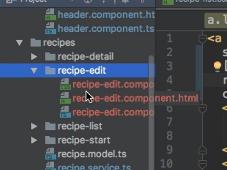
**164. Adding Editing Routes**

* Instructor: So with child routing almost finished, let's make sure that we can create new recipes and that we can edit the existing ones.
* So for this, we need a new component.
* So let's generate this first.
* I'll use the CLI again with ng g c, and I want to store it in the recipes folder.
* And here I will name it recipe-edit.

Text

Description automatically generated

* And I don't want to create this spec file which is why I set it to false.
* So with this, if I shrink down the other files, we get this new recipe-edit folder here with the RecipeEditComponent.



* We'll work on the content of this file after the form section, but I already want to load it, and I want to retrieve the route perimeter.
* So to load this, what I will do is I will go to my app-routing.
* module and I will register a new route here.
* The first route is for a new recipe.
* So maybe "new" would be a good path so that we have /recipes/new in the end.
* And with that, we could load our RecipeEditComponent here.
* And of course, you need to add the import here from the newly created recipes/recipe-edit folder.
* Also, make sure that it was added to the app-module.
* And this is done automatically as always by the CLI but you need to ensure that it is added to declarations and as an import here at the top to have it work correctly and to be able to load it.
* So with that added here, we have our new recipe route.
* Now I'll duplicate it to also have a :id/edit route maybe, so that we have the ID available in this route because we need to load it.
* But I want to have a /edit at the end so that it's clear that we're in edit mode.
* And this is the same component, that's also the RecipeEditComponent.
* I will, in the component, determine whether we are in edit mode or not.
* Now with this setup here, if we go back to the application and we enter /edit after this, we see "recipe-edit works!".
* So that's great.
* But we will see something maybe unexpected if I try to go to /recipes/new.
* Now we try to load the detail component and therefore we get an error because it basically failed to get a recipe with the ID "new".
* And that's exactly what went wrong here.
* If we visit recipes/new, due to the way how we ordered our routes, it'll try to parse "new" as an ID because the route with the dynamic ID parameter comes before the route definition where we have "new" hard coded into the path.
* So we should switch these two routes so that the new route comes before the route with the dynamic parameter.
* Otherwise, how would Angular know if whatever we pass right now should be interpreted as the new hard coded path or as dynamic parameter? So with this order, now we should be able to load "new" without any issues.
* If we hit Enter, that looks all right and we can also still load the existing recipes.
* So with that, these paths are set up too.
* The missing piece is now to retrieve the ID in the RecipeEditComponent and to determine whether we are in edit mode or not.