How to add a new Clinical Note Category (Type)

Step 1:

Add a new entry to the “ClinicalNoetCategories” table. Note that the “FriendlyName” field will be used throughout the UI to select the category, so it must be unique. Also, it is recommended that the “ComponentName” field matches the Angular UI component’s name, i.e. clinical-notes/clinical-notes-types/note-‘*ComponentName*’. Please see snippets 1 and 2.

Step 2:

Each Note type will be composed by two Angular components, one to show the note’s details and another one to edit/clone/create a note. Each one of these components includes one .ts, one .html and one .css file. The components will be located in the path “clinical-notes/clinical-notes-types/”. The folder distribution for the component will be the illustrated in snippet 3. There is a component named “template” intended to facilitate the creation of new notes categories. A new category could be created by copying this “template” one, renaming it and simply adding the new required fields.

Note that besides renaming the folders and files, also the component actual classes and selectors must be renamed. Please see snippet 4.

Step 3:

After copying the template component and renaming it accordingly with the “ComponentName” field assigned for the new category in the DB, please add the new components references (for both edit and details new components) to the “clinical-notes” module. Snippets 5 and 6.

Step 4:

Add the new components references (for both edit and details new components) to the ClinicalNotesComponent within the “clinical-notes-index.component.ts” file. See snippet 7.

Step 5:

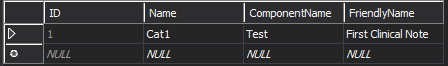
Modify the resolveComponentType function in the ClinicalNotesComponent class by adding an entry to the “Switch” clause related to the new category. Please note the switch clause uses the lower case value of the “ComponentName” field from the DB. Please see snippet 8.

Note: This step might eventually not be needed, but for the moment, I have found no better way to dynamically resolve the components type.

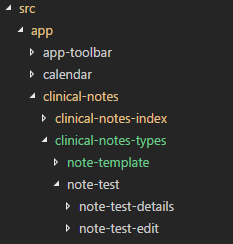
Step 6:

Edit the typescript and html code for both the new edit and details modules according to the new Category’s requirements. The html files for both modules within the “template” folder, have marks indicating were in the markup this new code is recommended to be added. Also, the typescript code already contains most of the functions and variables needed to make the components functional. However, both the typescript and html code can be 100% edited id needed, at the developers discretion.

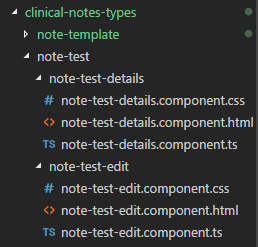
Snippets



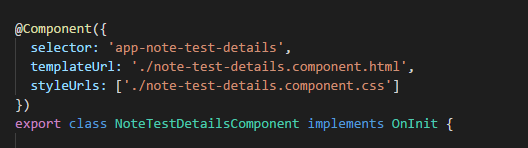
Snippet 1



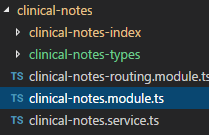
Snippet 2



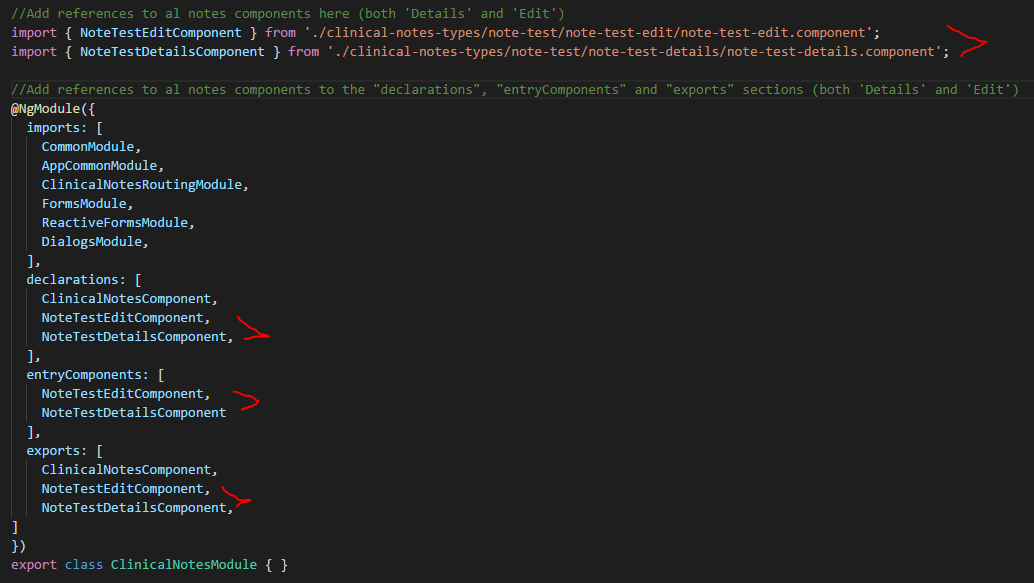
Snippet 3



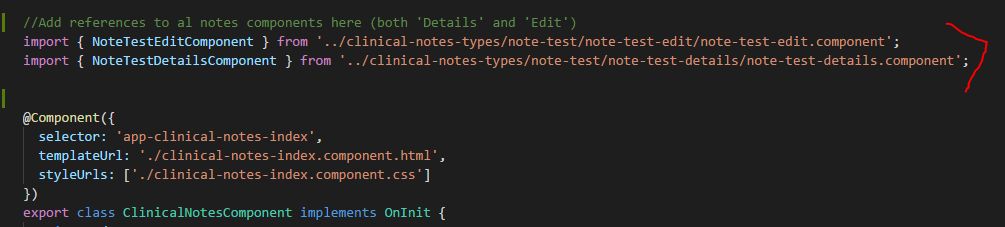
Snippet 4



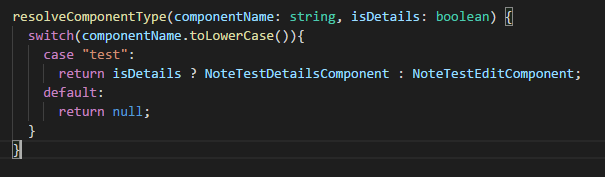
Snippet 5



Snippet 6



Snippet 7



Snippet 8