CreateRecipeFragment

- -recipeTitle:EditText,-preppingTime:EditText
- -cookingTime:EditText,
- -tagsList:TextView,
- -tagsAdd:Button.
- -servings:EditText,
- -ingredients:EditText,
- -description:EditText,
- -public:Switch,
- -save:Button,
- -share:Button

Note..

Beobachter Entwurfsmuster zwischen den LiveData-Objekt in VM und Fragment-Atrributer

«observe»

Note..

addTags() öffnet das "SearchWithTagsCreteRecipeFrage

setContent(...) schreibt die bisher eingetragenen LiveData-Attribute (!=null) in das Rezept

addToRepository(rec:PrivateRecipe) ruft erst setContent auf und gibt dann das privateRecipe an das Repository zum Speichern in der SQLite DB

parse(rec:PrivateRecipe):wandelt das PrivateRecipe rec PublicRecipe um. Dazu prüft die Methode alle Attribute PrivateRecipe.

save() unterscheidet, ob isPublic={true,false}.
true: rufe parse() mit privateRec als Parameter auf
und schicke dann das Resultat ins Repository
zum Speichern auf dem Server
addToRepository(parse(privateRec))

false: rufe addToRepository(privateRecipe) auf und lasse somit das privateRec vom Repository in der SQLite DB speichern

CreateRecipeViewModel

«construct»privateRec:PrivateRecipe

- -title:LiveData<String>?
- -preppingTime:LiveData<int>?
- -cookingTime:LiveData<int>?
- -servings:LiveData<int>?
- -ingredients:LiveData<IngredientAmoun
- -description:LiveData<String>?
- -isPublic:LiveData<boolean>
- +addTags():void
- + «private» setContent(title, preppingtim servings, ingredients, description, idPub
- +addToRepository(rec:PrivateRecipe)
- +addToRepository(rec:PublicRecipe)
- +save()
- +share()
- +parse(rec:PrivateRecipe):PublicRecipe

0 1 mrepo

Repository

+addPrivateRecipe(rec:PrivateRec

- +getTagList():List<String>
- +getRecipe():PrivateRecipe