

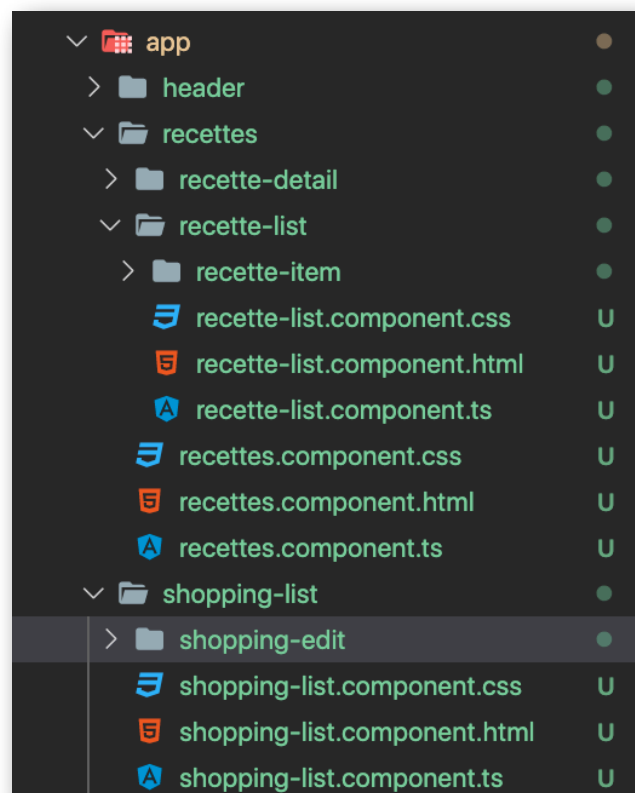
Angular

Exo Récap III

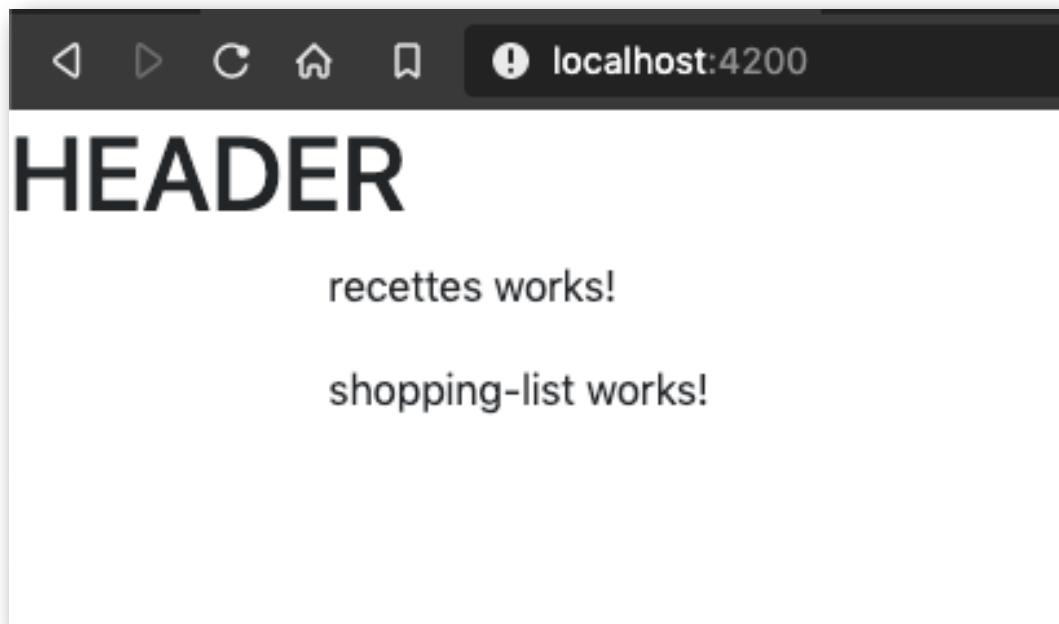
- Créer un bouton Afficher Détails
- Créer un paragraphe (contenu au choix)
- Le bouton affiche le paragraphe
- Compter chaque click du bouton dans un array et afficher cet array en dessous du paragraphe
- À partir du 3e élément du array on change l'affichage des prochains éléments du array (ngStyle) (ngClass)

Projet Recette

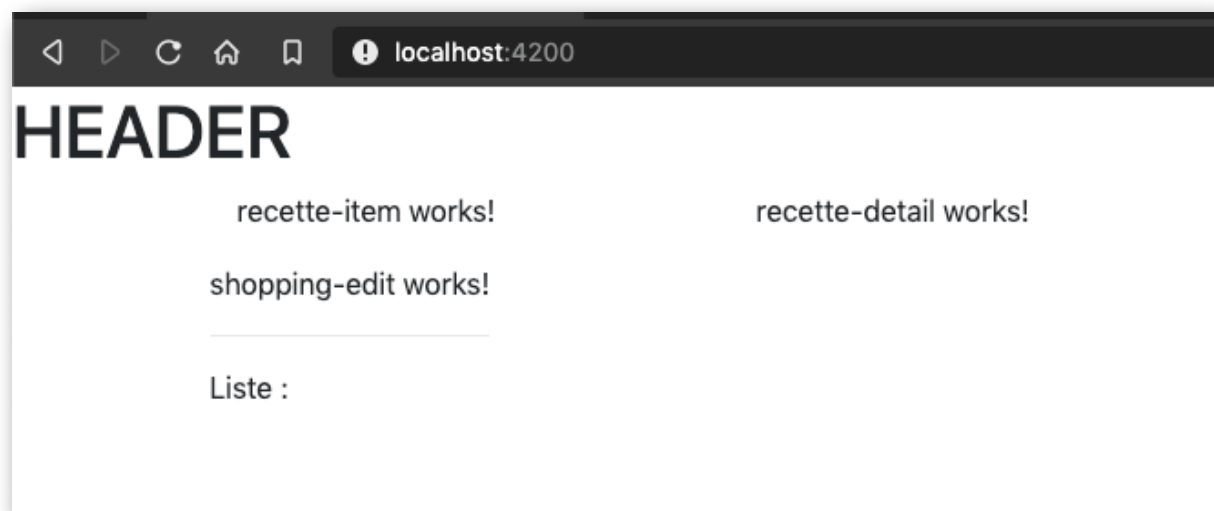
- Créer un nouveau projet angular qui gère des recettes de cuisine, (nom au choix)
 - Installer bootstrap dans le projet
 - Dans l'export de class (app.component.ts) on supprime le code de base (on part de 0)
 - Dans app.component.html idem on supprime tout (on part de 0)
 - Dans app.component.html tester bootstrap avec des div container, Row, col-md-12, avec un titre.
- Créer des composants
 - Créer manuellement un composant header et l'intégrer dans app.component.html 🤪
 - Avec le CLI créer un composant recettes et un composant shopping-list
 - Dans le dossier recettes, créer un composant recette-list
 - Dans le dossier recettes, créer un composant recette-detail
 - Dans le dossier recette-list, créer un composant recette-item
 - Dans le dossier shopping-list, créer un composant shopping-edit



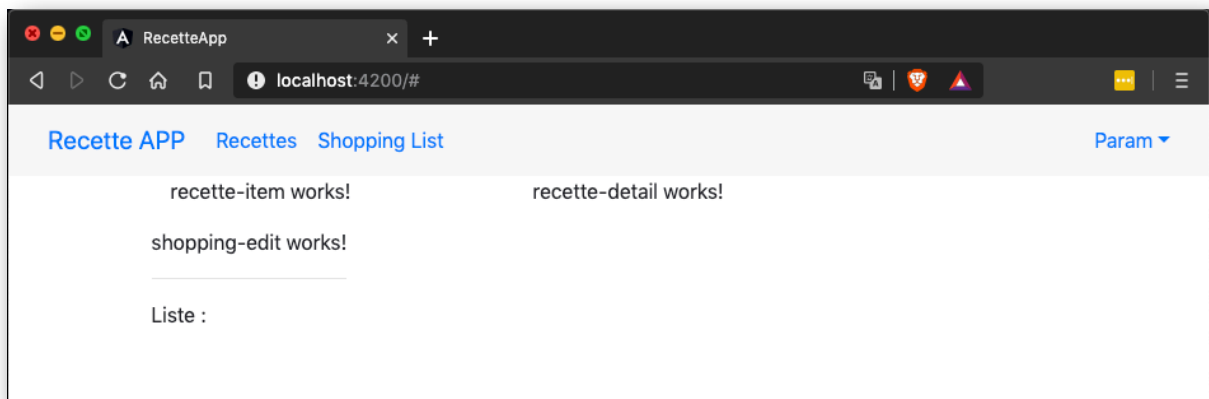
- Afficher les composants recettes et shopping-list dans app.component.html



- Afficher les composants recette-list et recette-detail dans recettes.component.html
- Afficher le composant recette-item dans recette-list.component.html
- Afficher le composant shopping-édit dans shopping-list.component.html

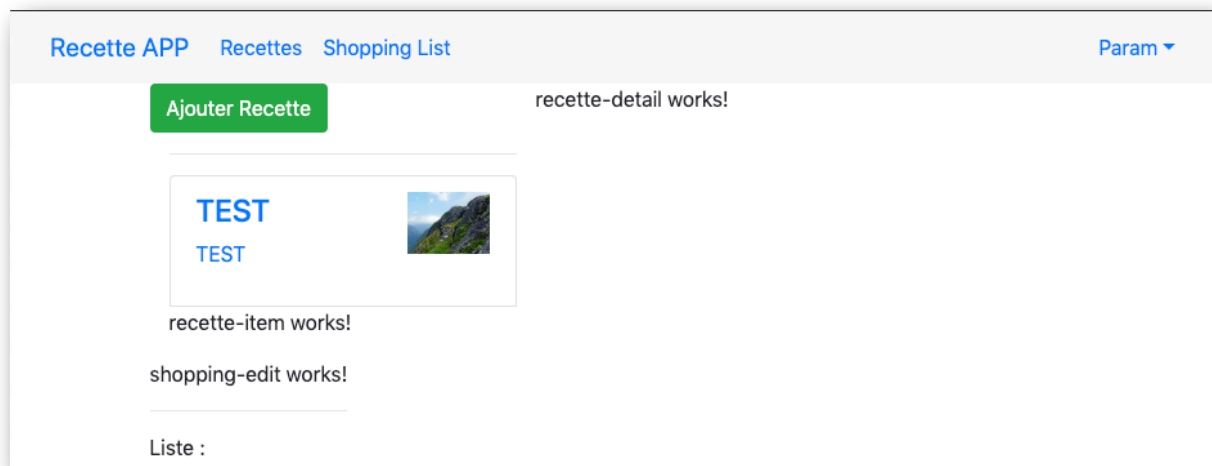


- Customizer le header (au choix),
 - un lien avec le nom de l'appli
 - Un lien vers Recettes
 - Un lien vers La shopping list
 - Un menu dropdown Paramètres dans lequel on a 2 liens « Sauvegarder » et « Charger »

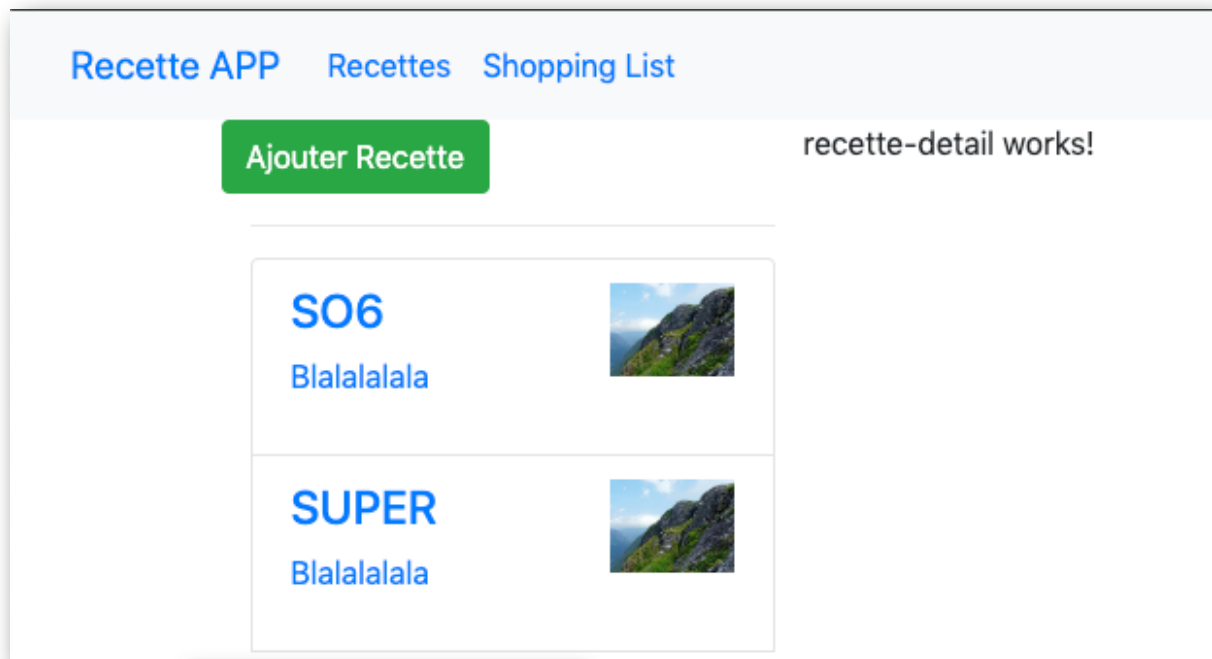


- Aller se renseigner sur ce qu'est un model en angular
 - C'est quoi ?
 - À quoi cela sert ?
- Créer un model recette
 - Dans `recette-list.component.ts`
 - Ajouter une propriété `recettes` qui sera un tableau vide
 - Créer un nouveau fichier `recette.model.ts` dans dossier `recettes`
 - Dans `recette.model.ts`
 - Exporter une classe `Recette`
 - Dans cette class on ajoute 3 propriété public de type string un `name`, une `description`, une `Url d'image`
 - La class `Recette` doit aussi avoir un constructeur pour instancier les données.

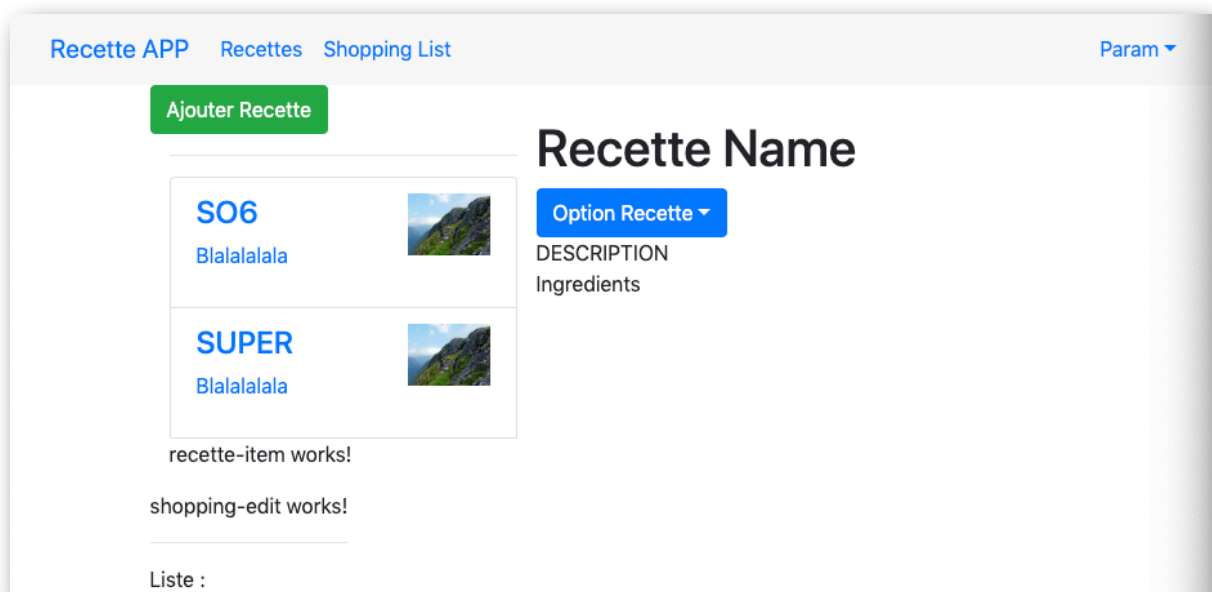
- Utilisation / test du model recette
 - Dans `recette-list.component.ts` :
 - On va typer notre tableau recettes pour qu'il soit de type `Recette[]`
 - Importer notre model Recette
 - Dans le tableau recettes on va créer une new Recette en lui passant un titre, une description, une url de votre choix
 - Dans `recette-list.component.html` au dessus de notre selector `<app-recette-item>` :
 - Ajouter un bouton « Nouvelle Recette »
 - Préparer l'affichage d'une Recette une div dans laquelle on affiche le name, la description, l'image



- Dans `recette-list.component.html`
 - Sur notre div qui affiche une recette on va placer une directive `*ngFor` avec `let — of —` de manière à répéter ce pattern Html
 - Dans cette div utiliser string interpolation `{{ }}` pour dynamiser les données de chaque recette.
 - Pour l'image de la recette :
 - utiliser Property Binding sur `src`
 - Rajouter le nom de l'image dans `alt`



- Dans `recette-detail.component.html` on rajoute :
 - 1 image dans 1 div class `col-md-12` dans une div class `row`
 - 1 titre `Recette Name` dans 1 div class `col-md-12` dans une div class `row`
 - 1 bouton dropdown avec un dropdown-menu qui contient 3 liens
 - + ingrédients
 - `Modif Recette`
 - `Suppr Recette`
 - 1 texte `Description` dans 1 div class `col-md-12` dans une div class `row`
 - 1 texte `Ingrédients` dans 1 div class `col-md-12` dans une div class `row`



- Composant Shopping List

- Dans shopping-list.component.html à la place d'afficher Liste
- Rajouter une liste de class list-group qui contient :
- 1 lien <a> de class list-group-item
- Dans shopping-list.component.ts on va (dans export class)
- Ajouter un tableau vide d'ingrédients

Réflexion :

Une fonction clé de notre app c'est aussi la gestion des ingrédients.

De la même manière on a déjà créé un recette.model.ts

C'est pertinent de créer un model pour ingrédient

Où va-t-on stocker ce model ?

Dans un nouveau dossier shared dans le dossier app

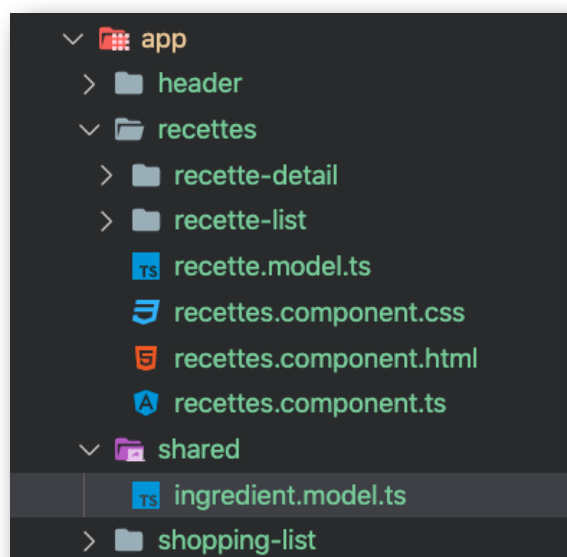
Pourquoi shared ?

Dans ce dossier il y aura des fonctionnalités, ou des éléments de notre app qui seront partagés par plusieurs composants

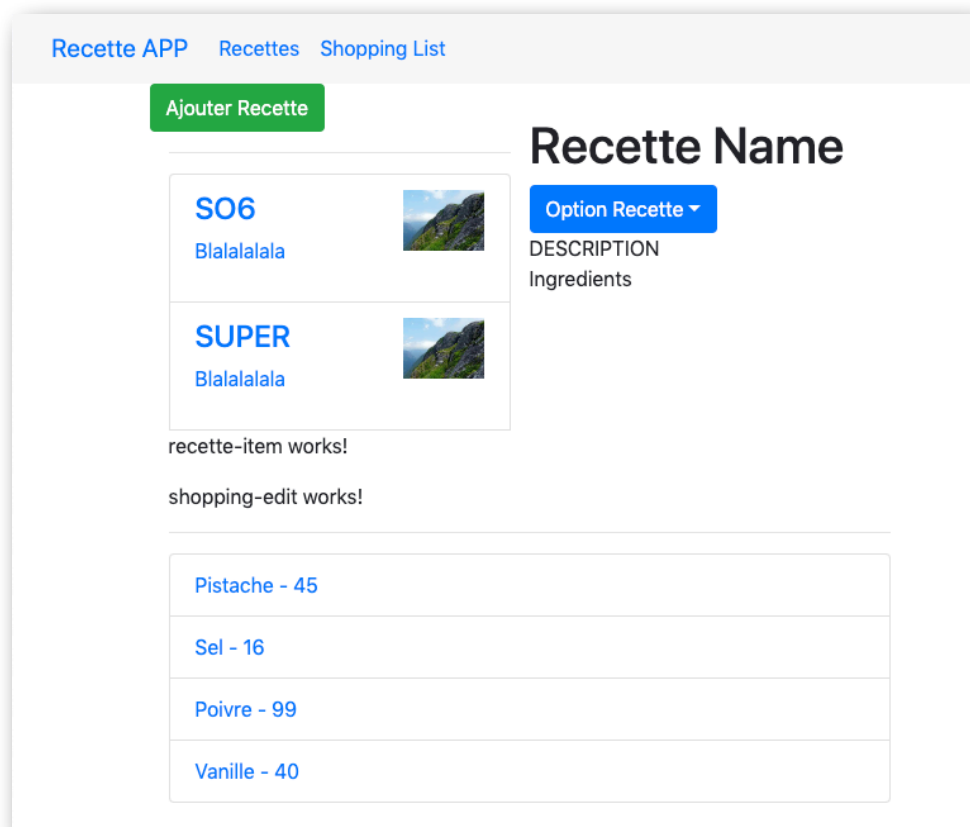
Un ingrédient est utilisé à la fois par Recette et par ShoppingList

- Créer le model pour ingredient

- Dans le Dossier app
- Créer un nouveau dossier shared
- Dans ce dossier créer le fichier ingredient.model.ts



- Créer le model ingredient
 - Dans shared / ingredient.model.ts
 - Exporter la class Ingredient qui possède :
 - 2 propriétés public :
 - name (chaîne de caractères)
 - Une quantité de type nombre
 - Un constructeur qui prend 2 paramètres et les assigne à nos propriétés public
 - Dans shopping-list.component.ts
 - On va typer notre tableau ingredients avec le type Ingredient[]
 - Dans ce tableau on va créer plusieurs new Ingredient (grâce au constructeur) en passant en paramètre un nom et une quantité
 - Dans shopping-list.component.html
 - Au niveau de notre balise <a> on rajoute une directive *ngFor qui parcourt ingredient de ingredients
 - Dans notre balise <a> on affiche :
 - le nom de l'ingrédient
 - La quantité de l'ingrédient




- Template de shopping-edit
 - Dans shopping-edit.component.html
 - 1 formulaire contient
 - Une div avec un label et un input pour le nom de l'ingrédient
 - Une div avec un label et un input pour la quantité de l'ingrédient
 - 1 bouton pour ajouter un ingrédient
 - 1 bouton pour Supprimer un ingrédient
 - 1 bouton pour Reset le formulaire


Recette APP
Recettes
Shopping List
Param ▼

Ajouter Recette

SO6
Blalalalala



SUPER
Blalalalala



recette-item works!

NAME
QTE

+

Suppr

Reset

Option Recette ▼

DESCRIPTION
Ingredients

Pistache - 45

Sel - 16

Poivre - 99

Vanille - 40