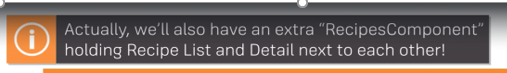
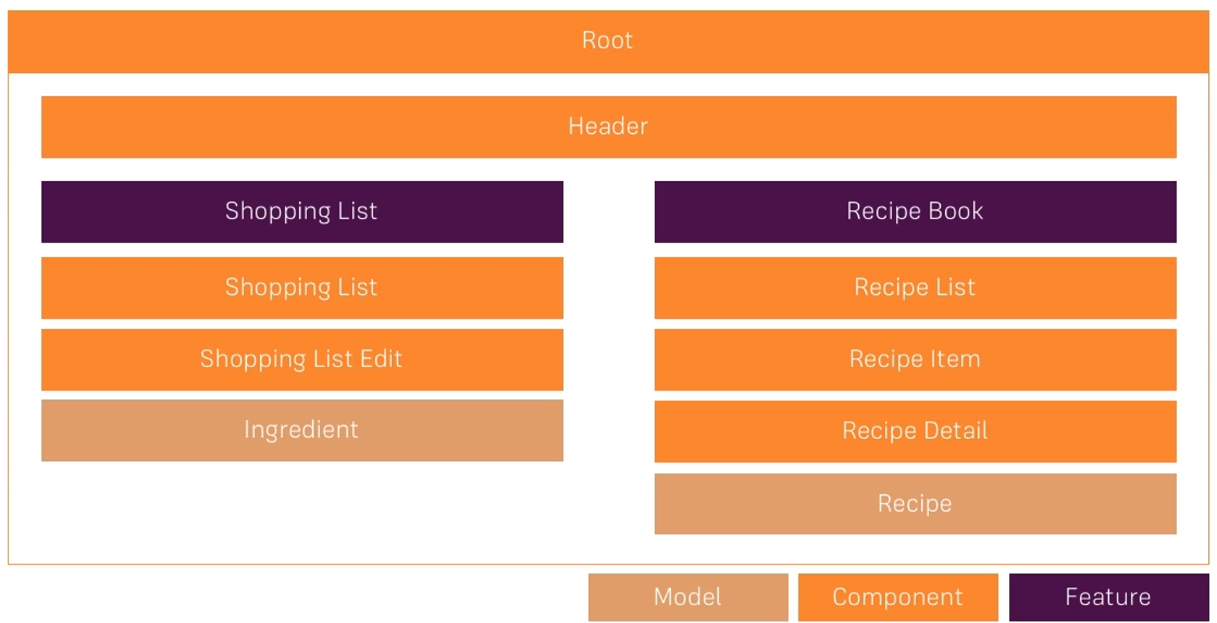
1. planning the app





**Creating a New App Correctly**

**MUST READ**

In the next lecture, we set up the course project.

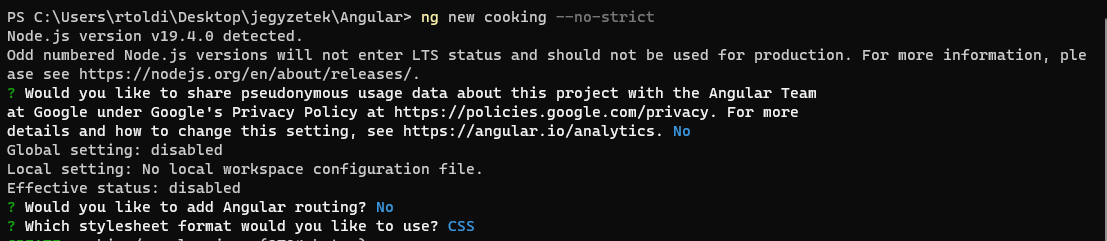
Make sure, you do create that app by also adding the --no-strict flag to the ng new command - otherwise you will run into issues later on (we'll still dive into that "Strict Mode" later in the course of course, no worries)!

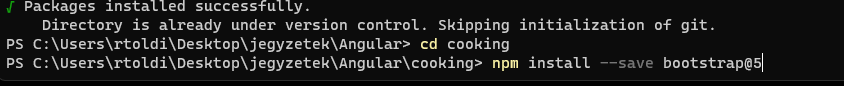
We'll also install the Bootstrap CSS Framework and in this course, we use version 3 of the framework. Install it via npm install --save bootstrap@3  => The @3  is important!

Additionally, when using a project created with Angular CLI 6+ (check via ng v ), you'll have an angular.json  file instead of an .angular-cli.json  file. In that file, you still need to add Bootstrap to the styles[]  array as shown in the next video, but the path should be node\_modules/bootstrap/dist/css/bootstrap.min.css , **NOT** ../node\_modules/bootstrap/dist/css/bootstrap.min.css . The leading ../  must not be included.

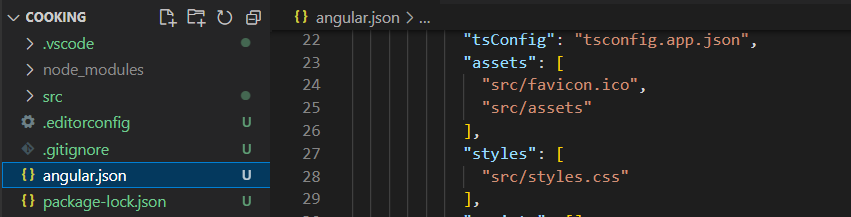
Also see this lecture - I do show the complete setup process there: <https://www.udemy.com/the-complete-guide-to-angular-2/learn/v4/t/lecture/6655614/>

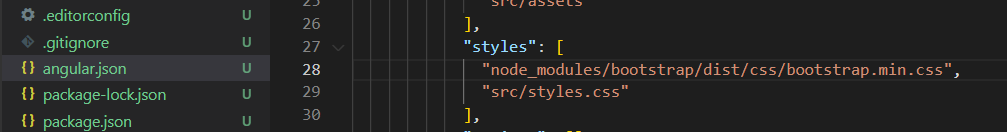
If you're facing any problems, please have a look at this very thorough thread by Jost: <https://www.udemy.com/course/the-complete-guide-to-angular-2/learn/lecture/17862130#questions/10444944>





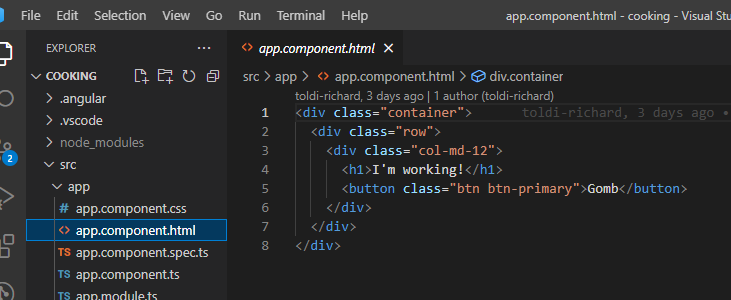
npm install –save bootstrap is elég, a legújabbat rakja majd fel





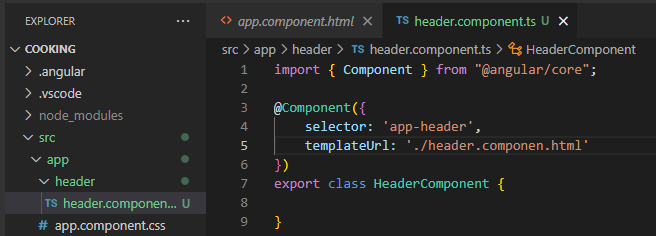
Kitakarítottuk az app.component.html-t és component.ts-ből is a title kukázható.

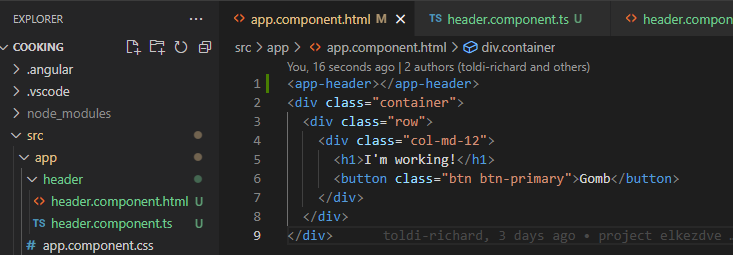
**npm install** a node pagkage-k letöltésére



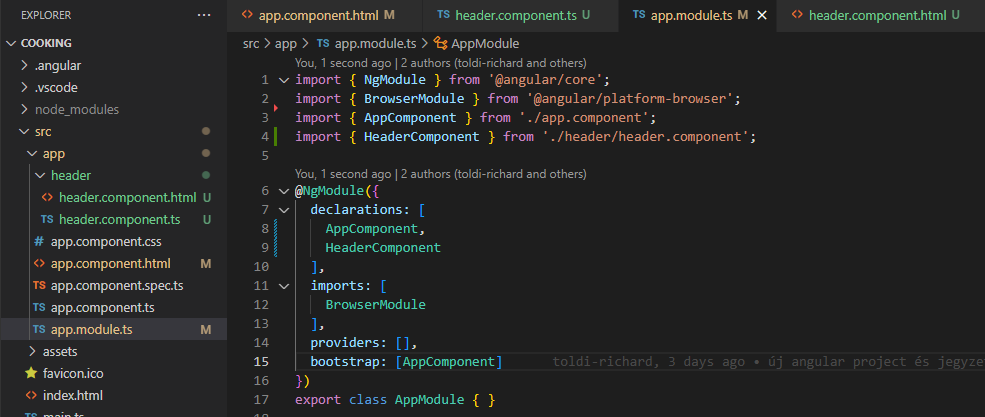
és **ng serve**, hogy lássuk, hogy minden jól megy

**Header componens elkészítése:**



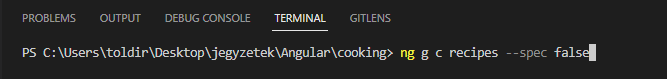


megcsináltuk a componens-nek a html file-ját is és a fő html file-ba beleraktuk a componenst

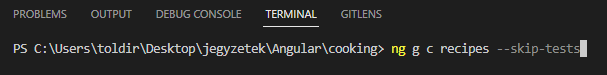


Végül hogy a componens-t lássa a fő html az app.module.ts-be is hozzá kell adni a componenst.

Vagy komponens így is létrehozható:



--spec false - hogy ne hozzon létre test file-t, régen ez volt a command



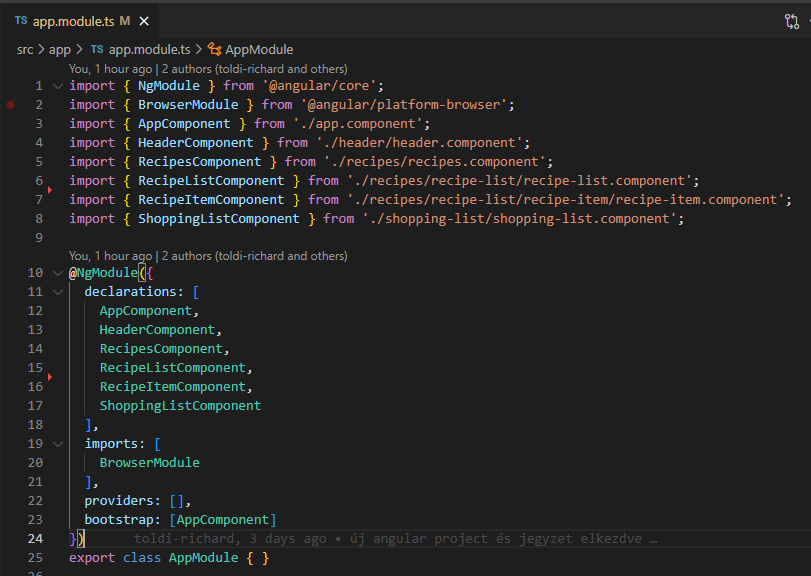
ez az új command



így kerül bele a recept lista a recept mappába



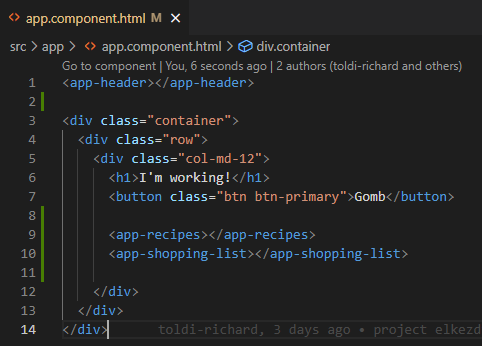


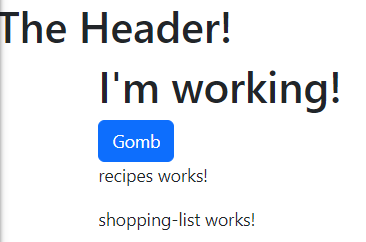


app.module-ba is berakta az új komponenseket



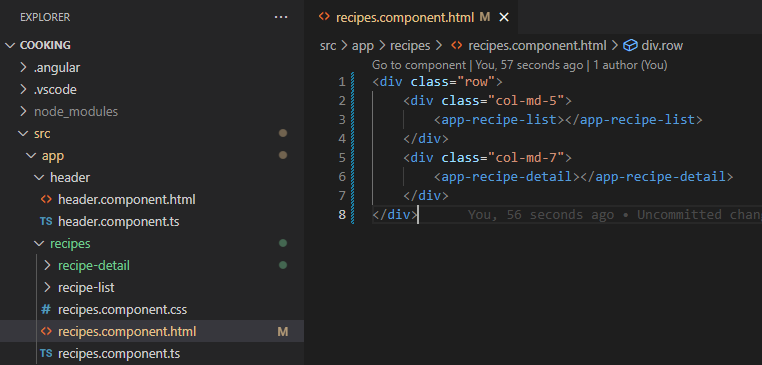
Componensek tesztelése:

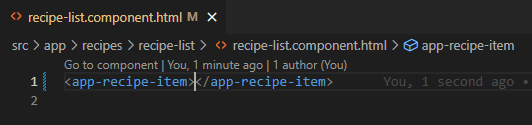


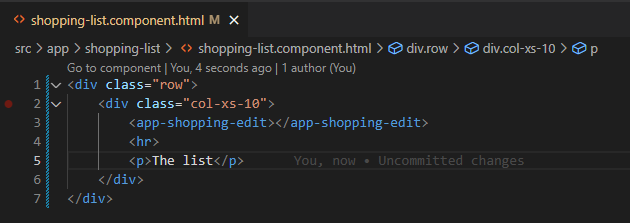


html-jükbe alapvetően beletolja hogy xy works…

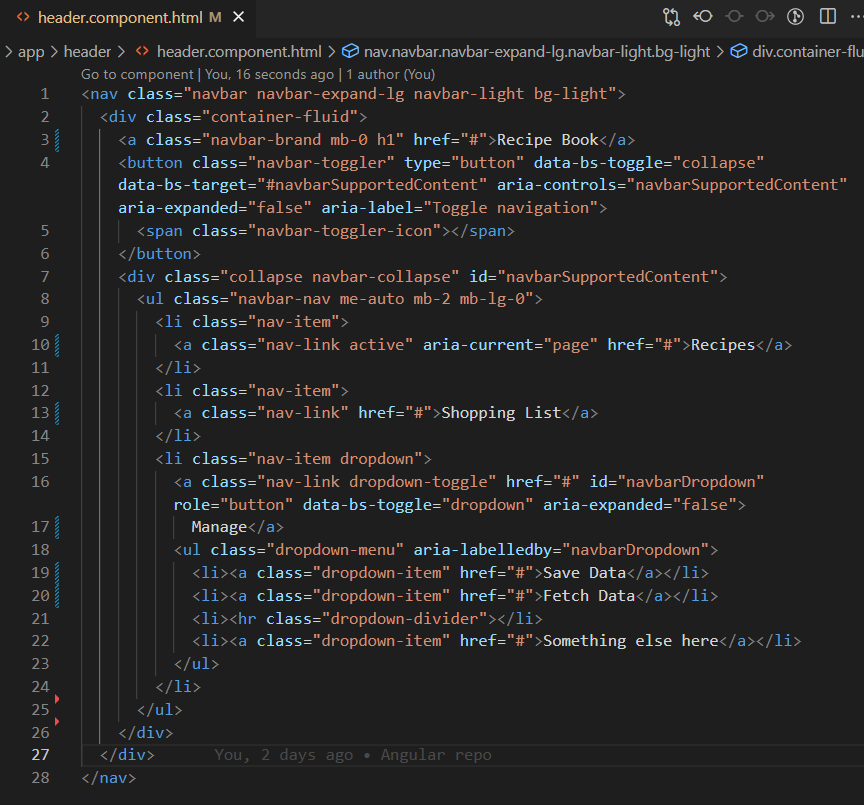




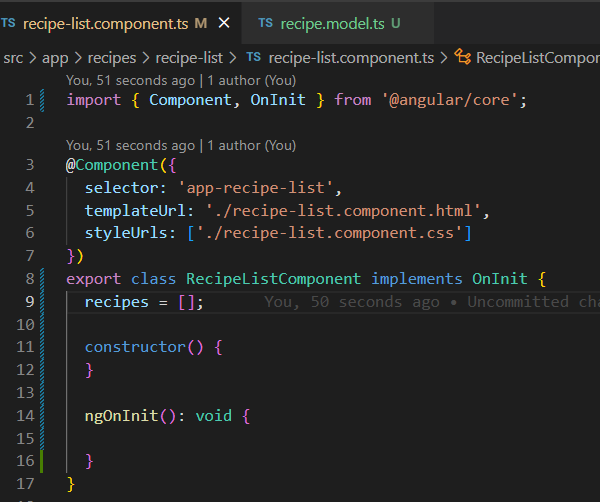




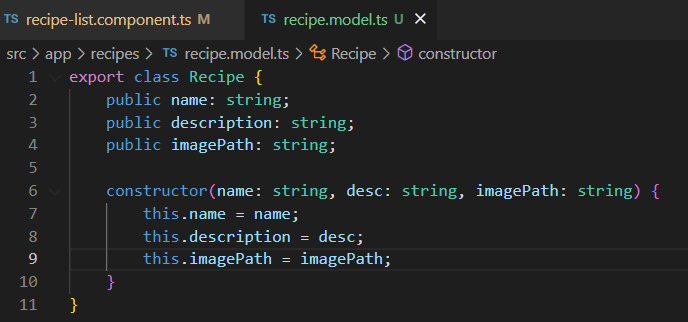
Legegyszerűbb, ha belinkeljük a bootstrap script-eket. Ha telepítjük, nem biztos hogy megy sőt….valamiért gond van. Linkelve meg telepítve pláne gond lesz. Fontos, hogy a script a componens felett legyen, hogy az érvényesüljön először. Szóval a head-ben és a stylesheet.css se mindegy hogy a bootstrap után vagy előtt van, az érvényesül ami lentebb van!



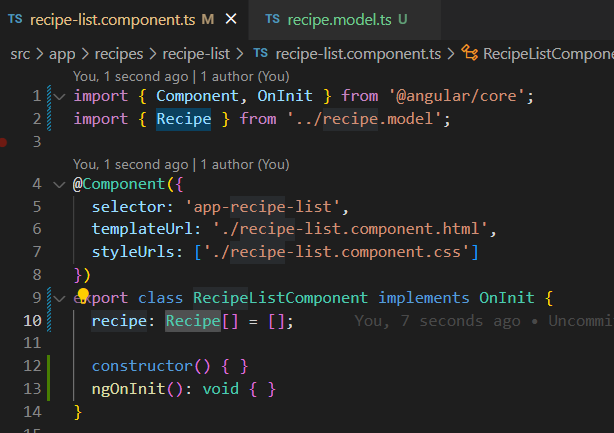
Navbar az újabb bootstrappel. https://getbootstrap.com/docs/5.2/components/navbar/



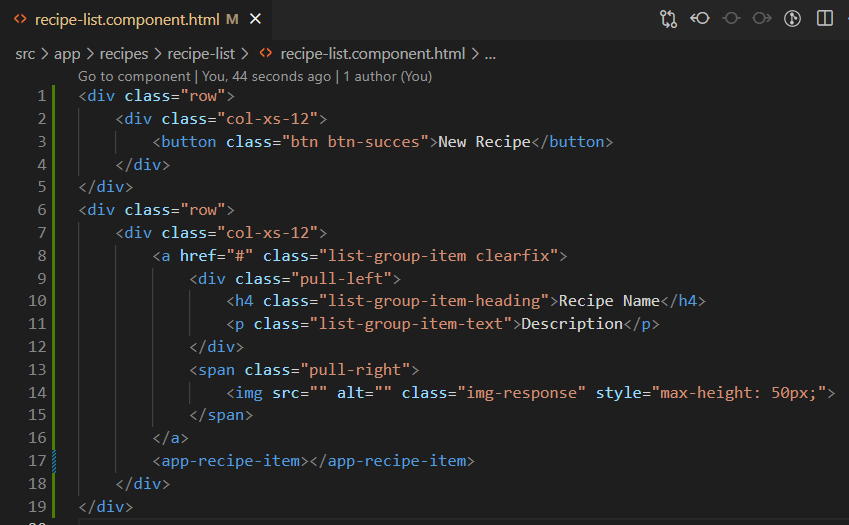
recipe-list komponenst elkezdtük csinálni és csináltunk a recipes mappába egy recipe.model.ts file-t



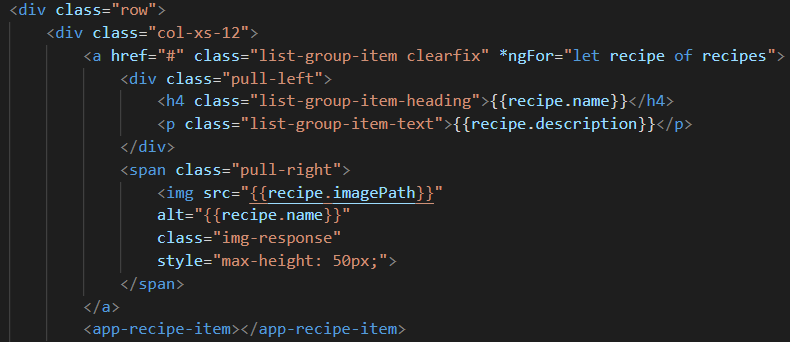
Megcsináltuk a recipe modelt.



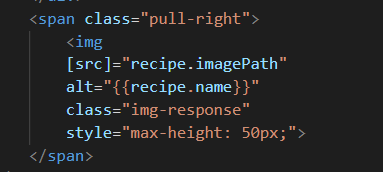
Megcsináltuk a Recipe modelt, szóval a tömbünk most már Recipe osztály tömb lesz, fontos hogy importálni is kell a modelt.

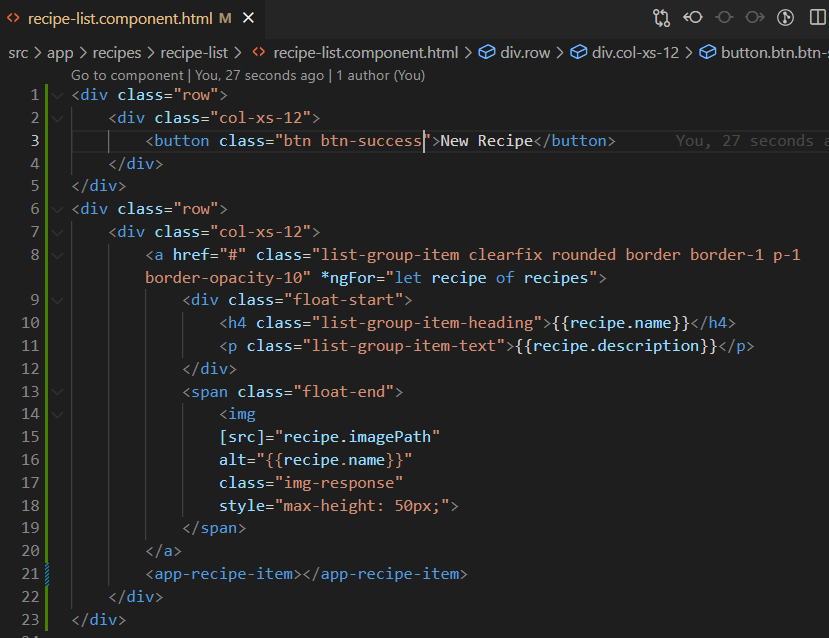


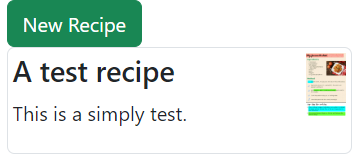
Megcsináltuk a receptek kilistázásához a skeletont.

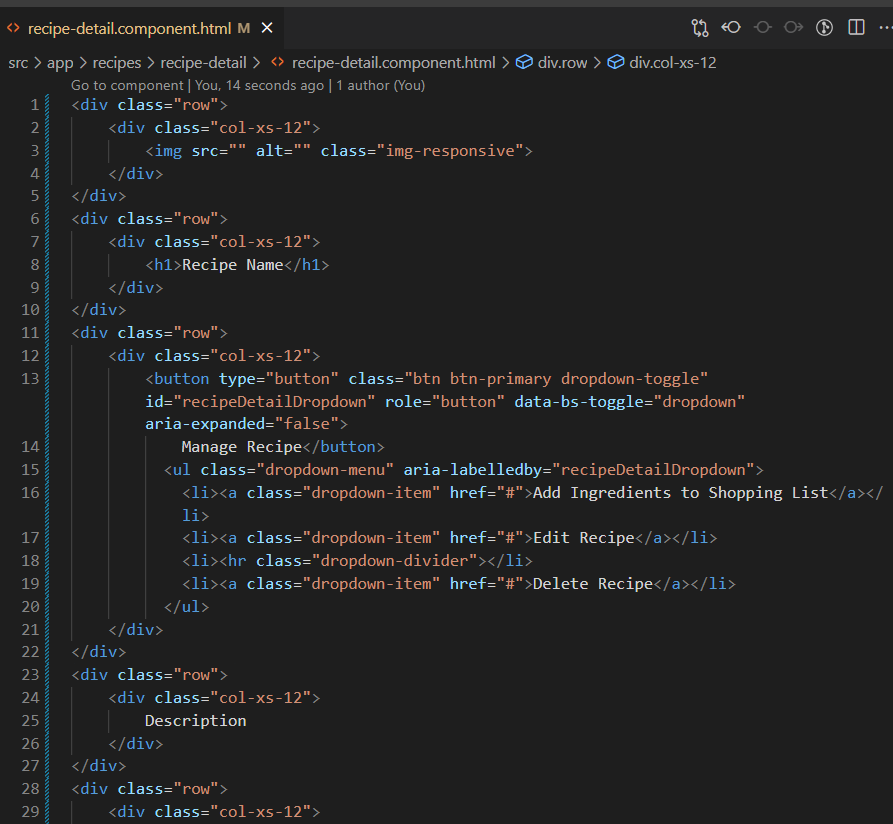


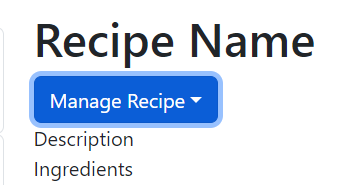
egyik lehetőség

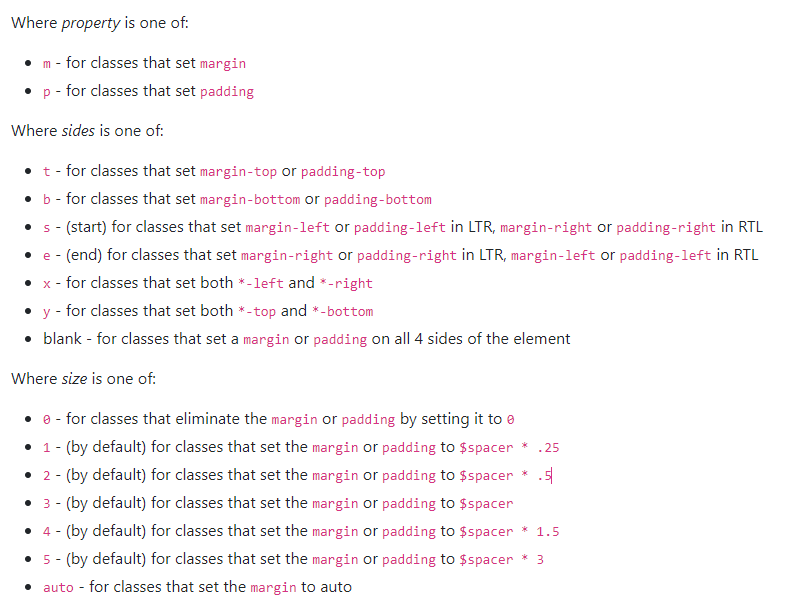








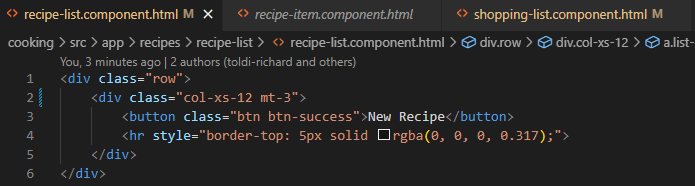




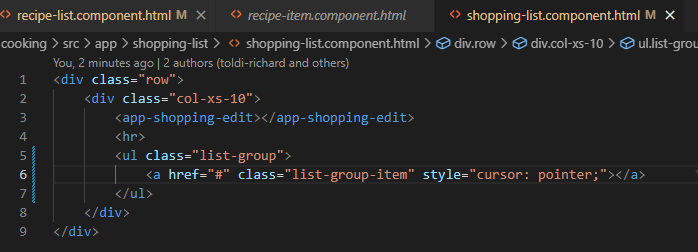
<https://getbootstrap.com/docs/5.2/utilities/spacing/>



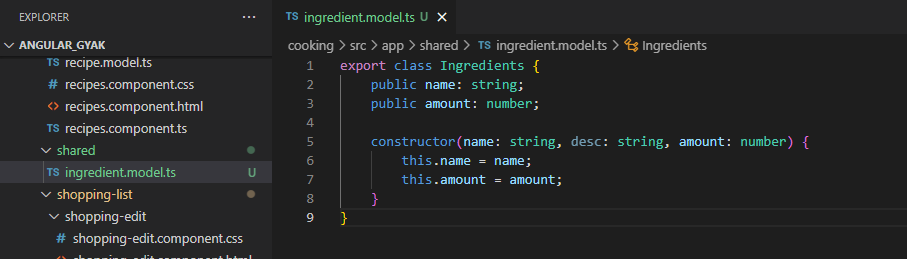
hr stilizálása így lehetséges



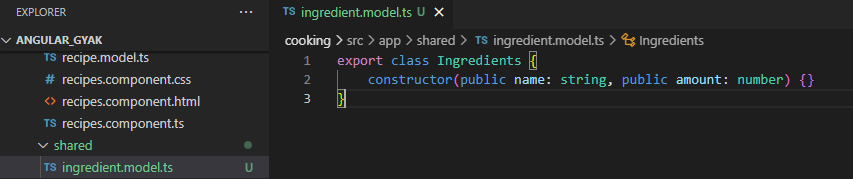
adtam egy kis margin-t a gombra



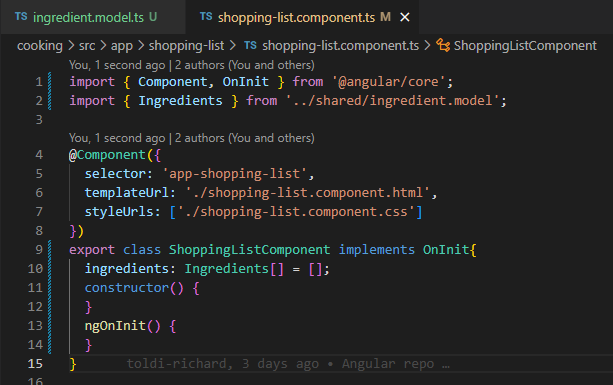
shopping-list componenst elkezdtük

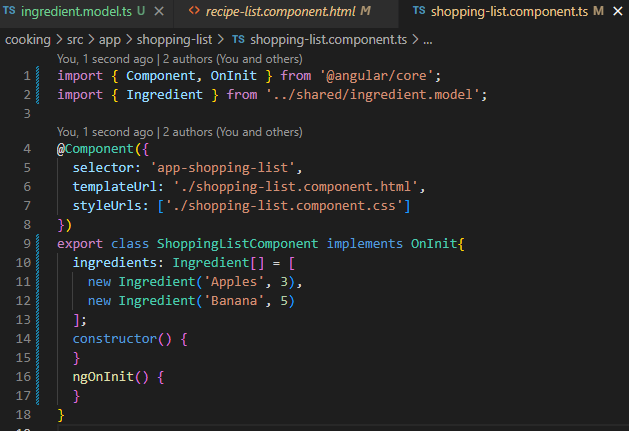


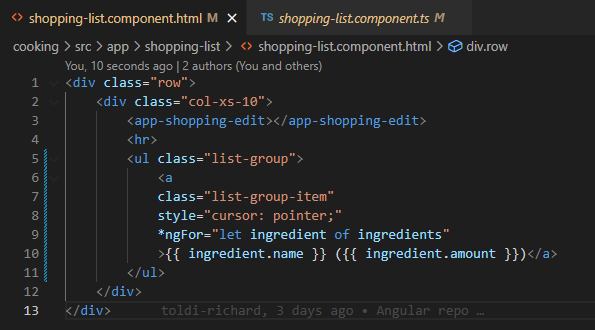
ingredients-hez elkezdtük a modelt csinálni

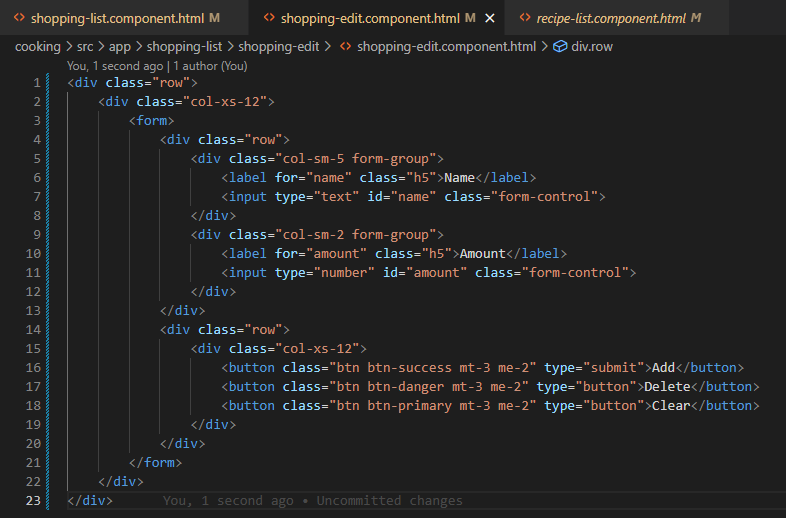


röviden így is lehet csinálni



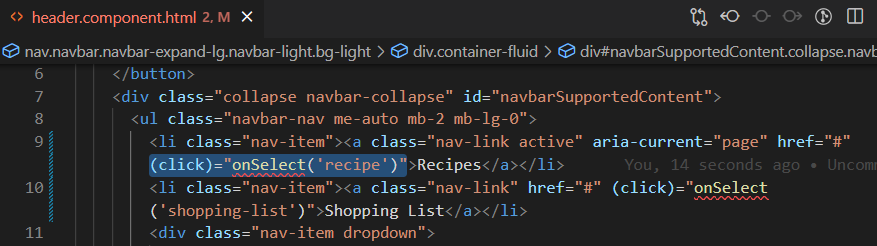




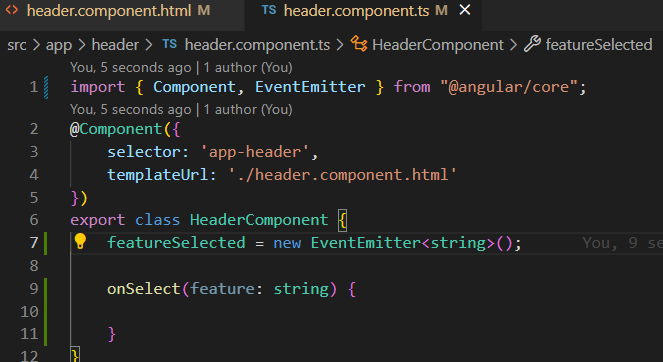


**Navigation:**

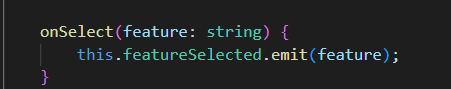
ng-if hogy 1 komponenst töltsünk be egy időben és ne mindet egyszerre



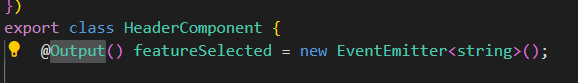
beleraktuk a click eventet az onSelect funkcióval



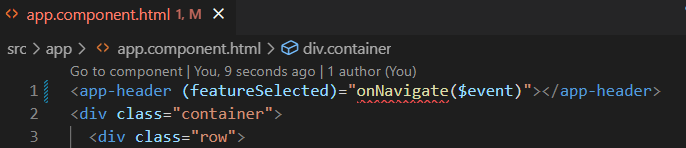
Elkezdtük az onselect metódust írni és létrehoztunk egy új event figyelőt az eventemitter-t propertyként.



Létrejött a metódus, a kapott string értéket át

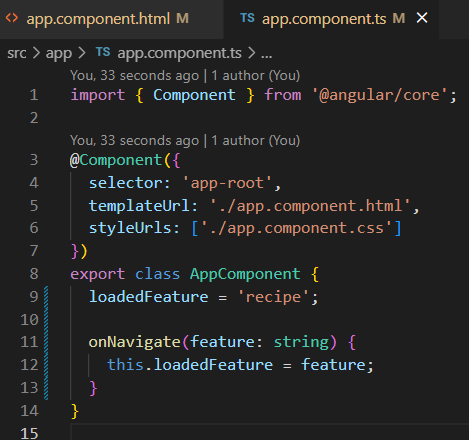


Output rákerült, hogy a szülő (app komponens az esetünkben) komponens elkaphassa az adatot.



Az app komponensnél létrehoztuk a metódust és a $event segjtségével kapja el a komponens event által generált adatot, amit átadunk az onNavigate metódusnak.

$event mindig az event adatára hivatkozik



Elkészült a metódus ami a loadedFeature értékét szabályozza majd.

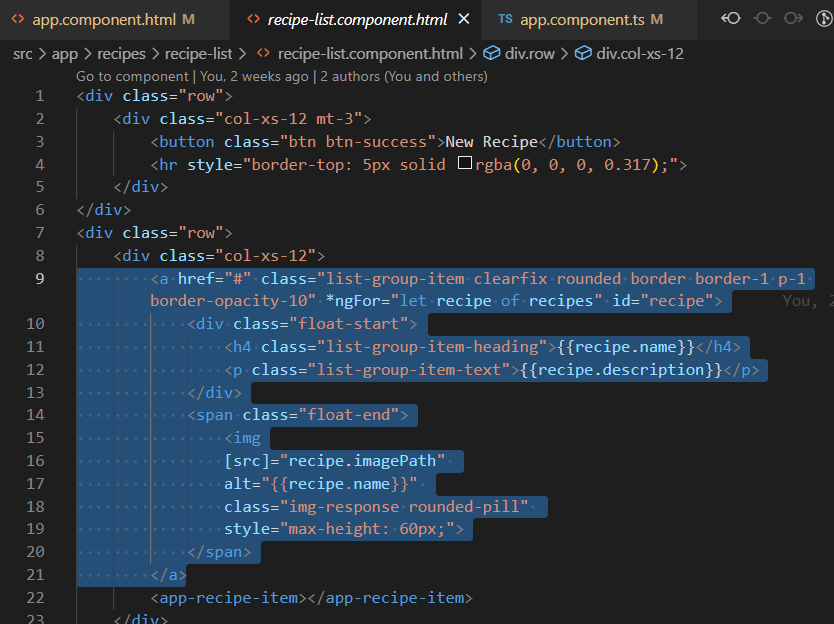


ngIf használata navigációra

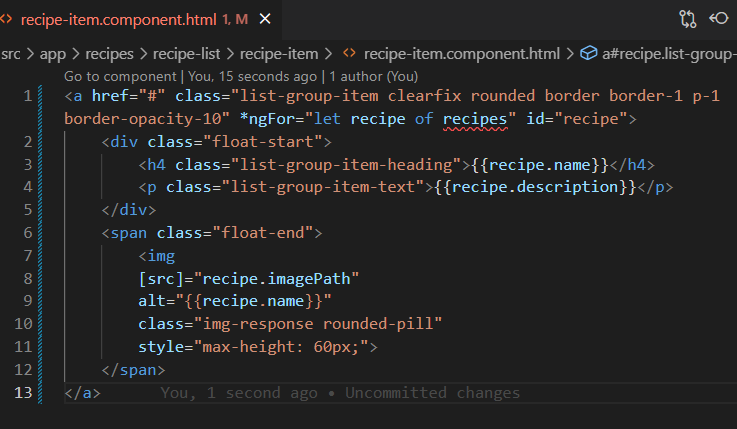
navigáció ezzel megy a 2 tab között már

**Property binding:**

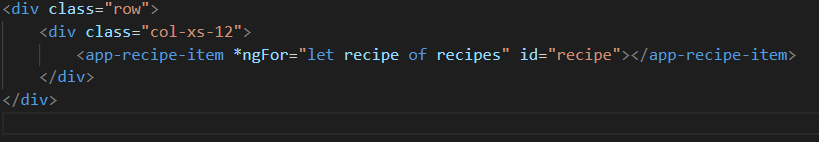
property bindinggal történő adatátadás



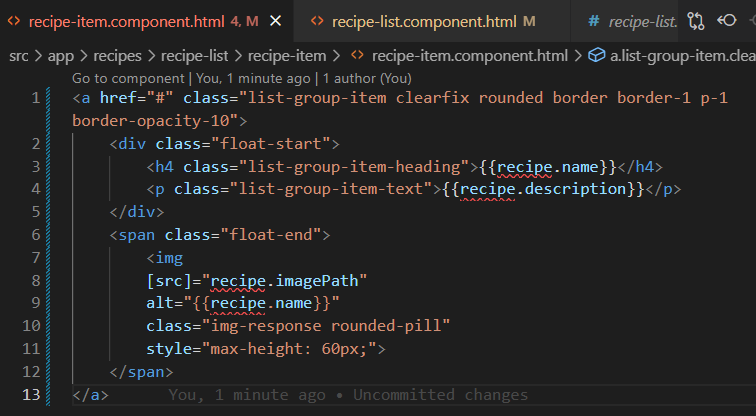
a lista egy eleme ez a rész a listában, de ez úgy szép ha a lista egy komponens és az elem is 1-1 komponens



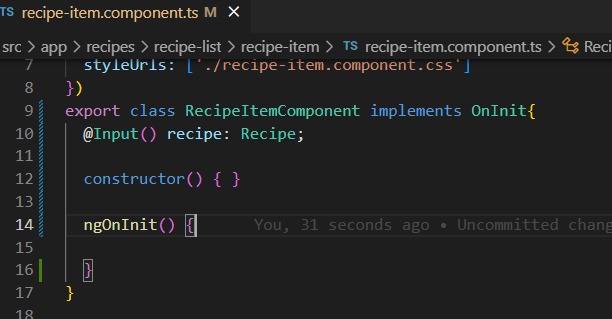
áthoztuk a szakaszt a recipe item komponensbe



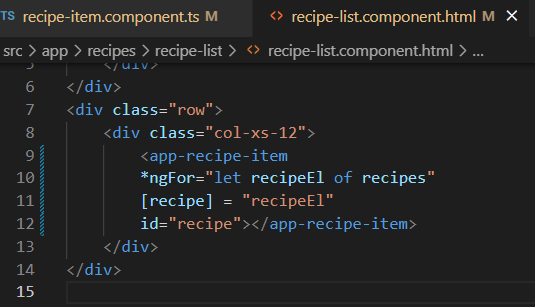
ngFor-t visszaraktuk



innen meg kiszedtük



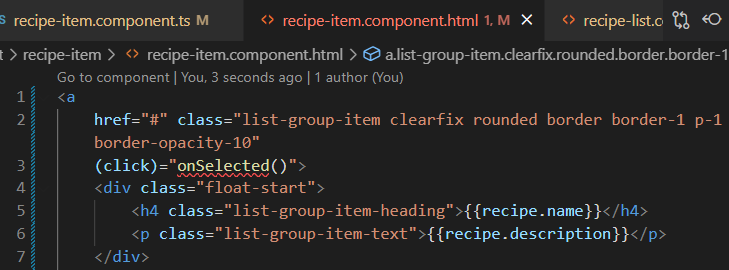
Létrehoztuk a recept objektumot az item komponensnél is amit majd átvesz egy külső komponenstől, ezért kell az input elé



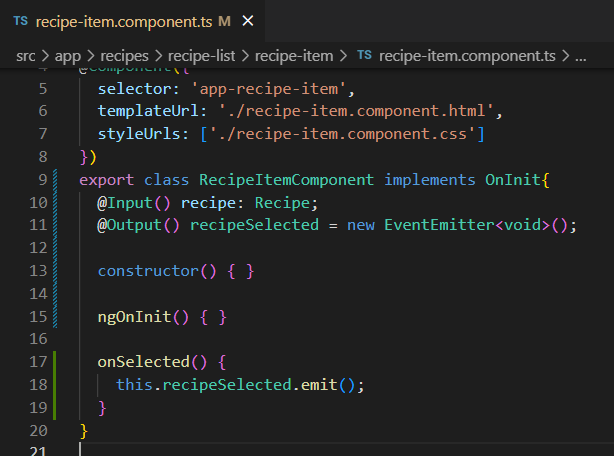
megcsináltuk, hogy a listánál található recept objektumokat átadjuk attributumként a recipe item komponensnek

Most a recipe listában szereplő recepteket átadjuk az itemnek és így kül komponensben van már.

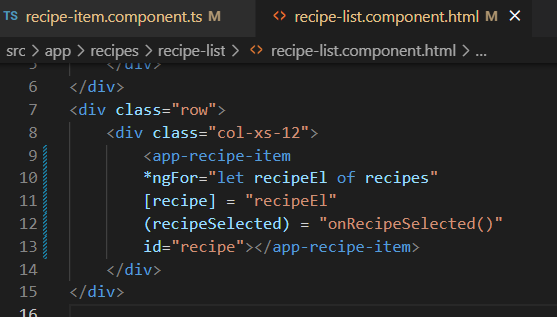
**Lista elemre kattintva megjelenjenek az elemek dolgai:**



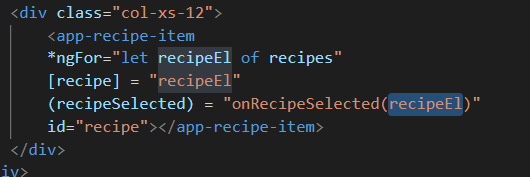
click evenetet létrehoztuk



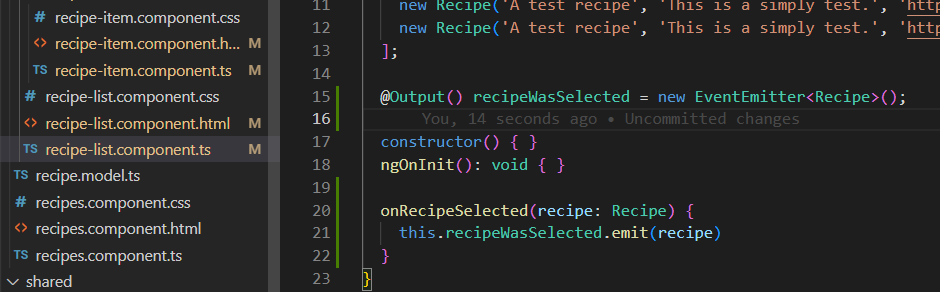
Megcsináltuk a metódust és az eventet ami létrejön, nem kell neki átadnunk semmit, mert az eventet figyeli a parent komponens.



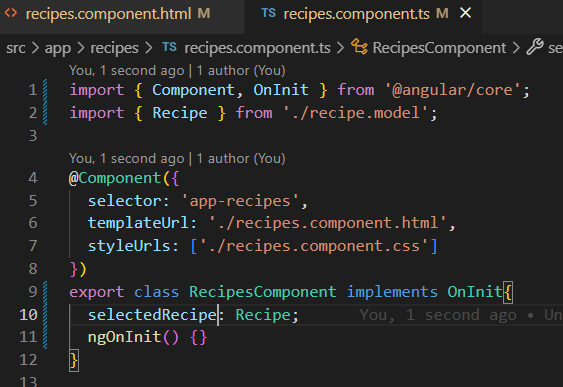
megcsináltuk a lista komponensben hogy figyelje az eventet a html oldalon



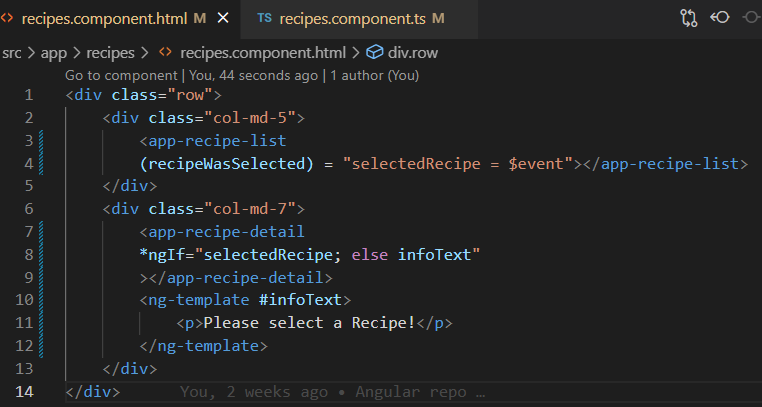
átadjuk neki, hogy melyik recept volt kiválasztva, ezért nem adtuk át neki az item-nél mert innen kapja az item a recept objektumot



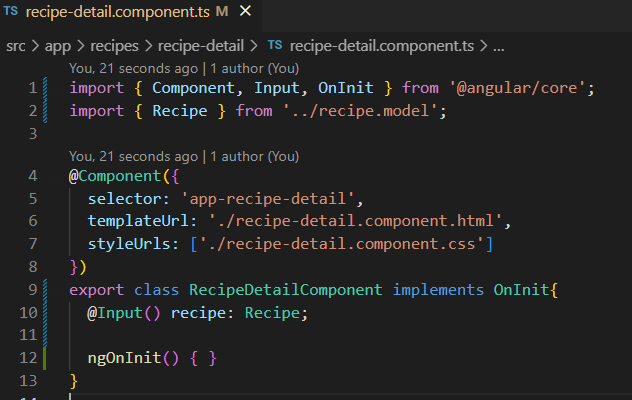
megcsináltuk a metódust és a property-t az eventhez, azért kell az output mert ezt az objektumot adjuk át majd annak a komponensnek amelyik megjeleníti az objektumunk adatait amire kattintottunk



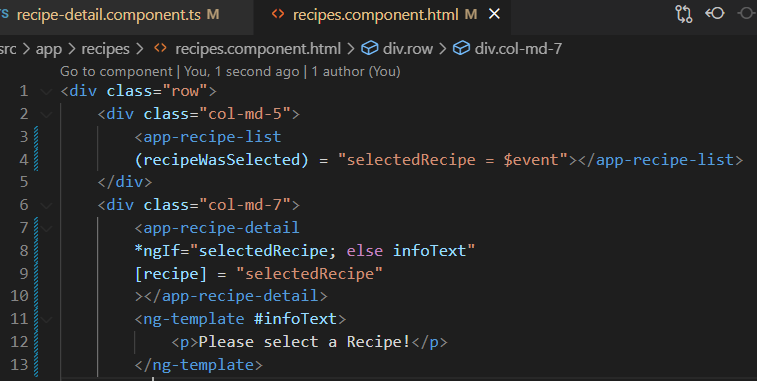
A komponensbe létrehoztuk a property-t ahol meg fogjuk jeleníteni az objektumot.



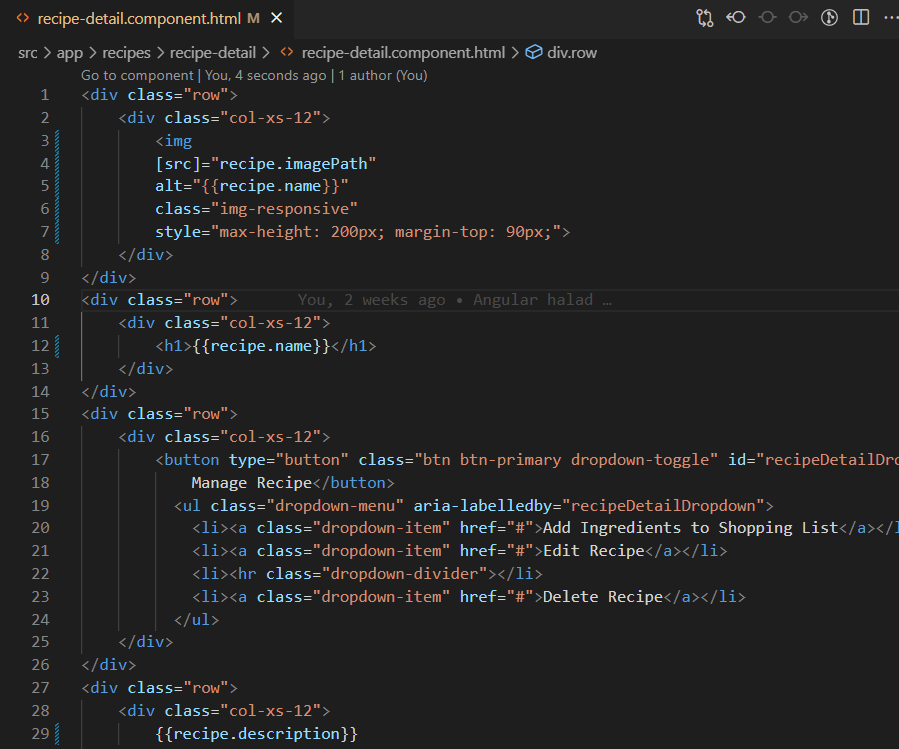
ngif-fel megnézzük hogy választott e receptet, ha nem akkor az infotext fog megjelenni, ha meg igen akkor a recept adatai, de még az adatok nincsenek átadva



Létrehoztuk a details-ben is a propertyt inputtal hogy át tudja venni az adatokat a komponens



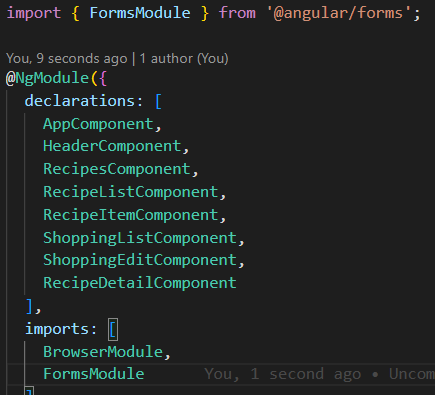
így átadjuk a detail komponensnek a selected receptet



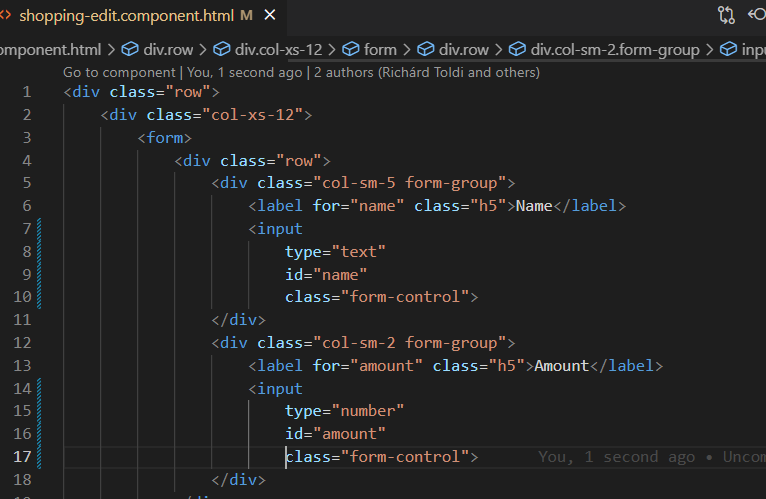
kicseréltük a text placereket az objektum egyes tulajdonságaira és a kép formázásán is változtattunk kicsit

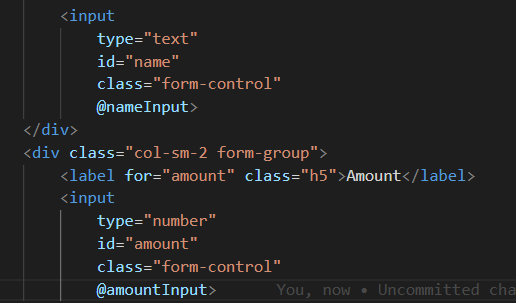
**Make sure you have FormsModule added!**

One quick note: In case you're hitting an error in the next lecture, make sure you have FormsModule added to your imports[] in the AppModule. Also have a look at the following Q&A thread for more info: <https://www.udemy.com/the-complete-guide-to-angular-2/learn/v4/questions/4924644>

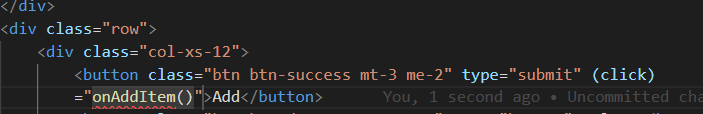


**Ingredients adding to shopping list:**

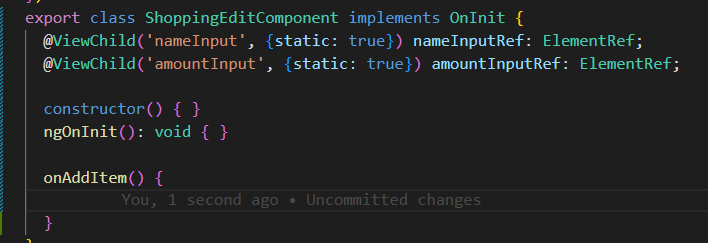




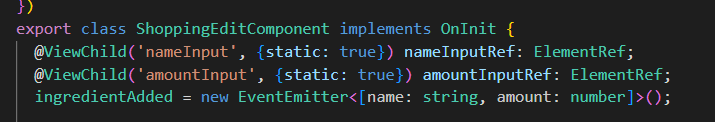
Lokális referenciákat adtam az inputokhoz.



ráraktam a gombra egy klikk eventlistenert az onAddItem metódussal



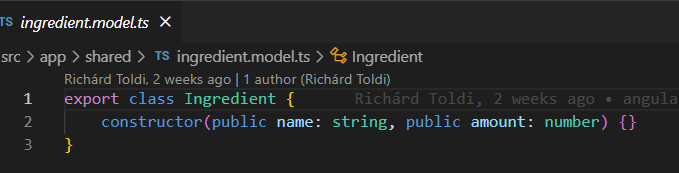
létrehoztuk az adatokat az eventhez

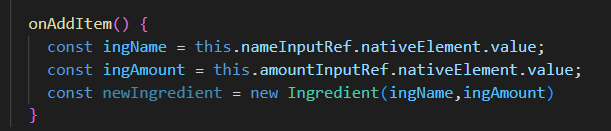


Megcsináltuk az eventemitter-t de azok még nem értékek, hanem típus meghatározások



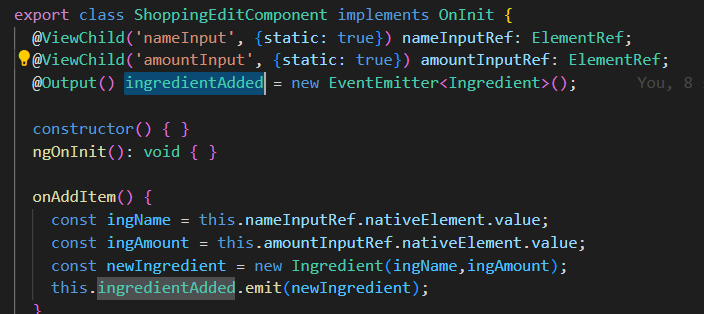
de így is elég mivel már van ingredient modellünk



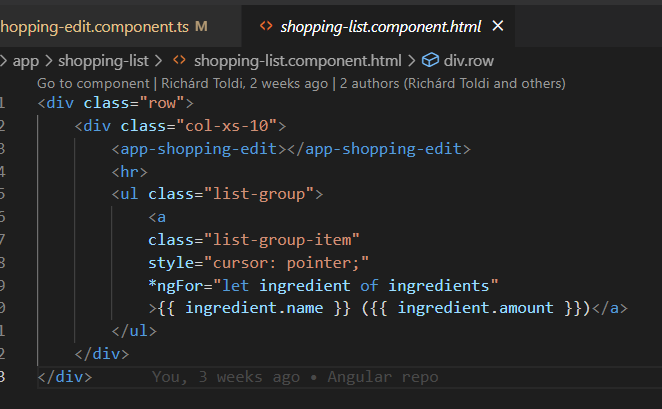


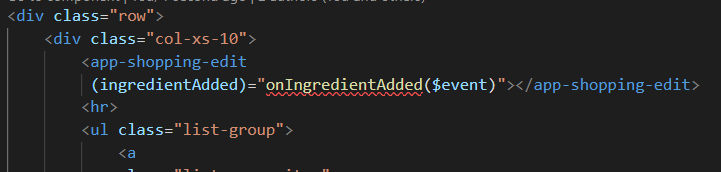
alakul a metódusunk

Mivel ezeket nem fogjuk változtatni, ezért érdemes constans-ként használni őket.



Szóval van egy click eventünk, ami a nameInput és az amountInput lokális változókat átadja az ingName-nek és az ingAmount-nak majd a newIngredient objektumnak őket. Az egész objektum átkerül az eventemitter-hez -> eventsugrázó és arra egy output, mert egy másik komponens felel az adatok megjelenítéséért. a shopping-list komponens

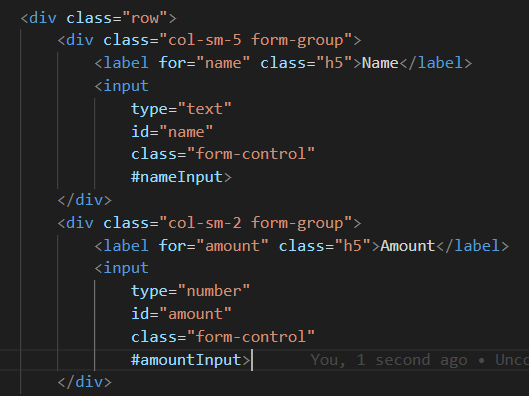




így megkapja az adatokat és meghívódik a metódus is rá



és a ts-be a metódust is megírtuk



Local variable #-el hozható létre, de a metódusnak is közvetlen átadhatjuk paraméterként és akkor nem kell a viewchild extra bele.

**Attributum vs Structural directives:**

Mind a 2 a html elem-en ül rajta. De a structural directives a DOM struktúráját is megváltozatják.

Az attributum direktíva sose töröl el egy elemet a DOM-ból, de a structural egy ngIf-el már igen, ha nem true akkor az elem kikerült a DOM-ból.

**Attributum:**

sima html attribútum-nak néznek ki

csak az adott elemre vannak hatással

**Structural:**

mint a sima html attributum, de van egy \* jel előttük

az egész DOM-ra hatással van, hozzáad vagy eltárolít elemet

89. lecke jön