



Codergirl – Frontend

Unit 2 - Class 2

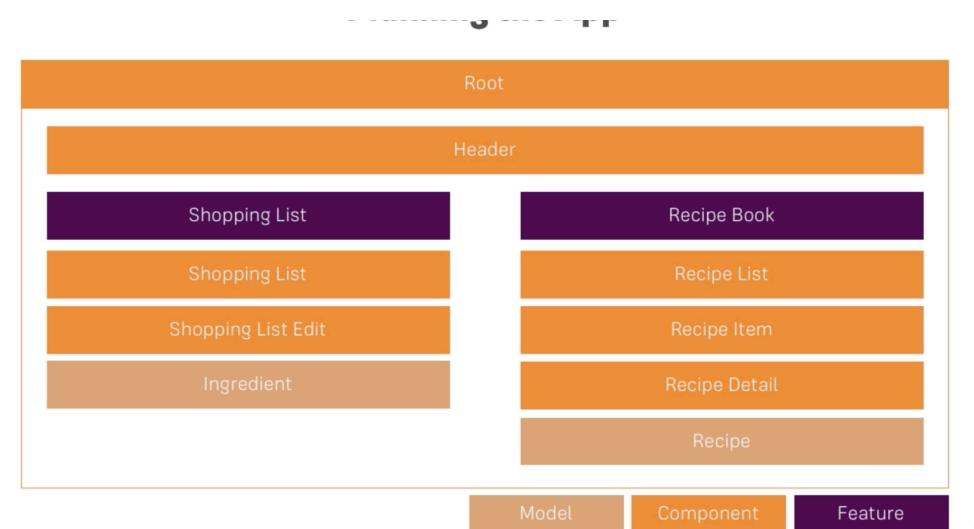
November 30, 2020

# Agenda

- Course Project – The Basics
- Debugging
- Talk on Learning Environment
- Studio

# Planning the app

- App or root component
- Header or Navigation component
- One component for each feature
- Shopping List
  - Shopping List
  - Shopping List edit (data)
- Recipe Book
  - Recipe List
  - Recipe Item
  - Recipe Detail



# Creating the app

```
ng new shopping
```

```
npm install --save bootstrap@3
```

```
"styles": [
```

```
  "node_modules/bootstrap/dist/css/bootstrap.min.css",
```

```
  "src/styles.css"
```

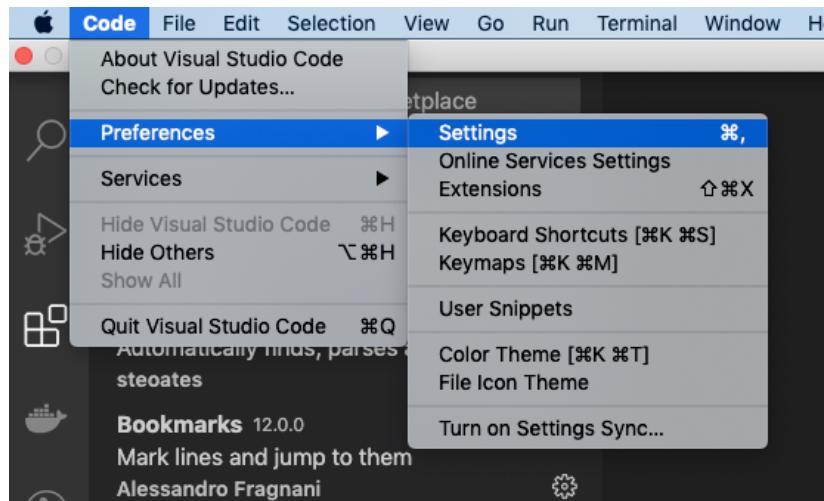
```
],
```

```
ng serve
```



# Adding Emmet

"emmet.triggerExpansionOnTab": true,  
.container in html file and tab gives the div tag with  
class container!



```
18 ],
19   "eslint.codeAction.showDocumentation": {
20     "editor.codeActionsOnSave": [
21       "source.fixAll.eslint": true
22     },
23     "vetur.completion.scaffoldSnippetSources": [
24       "editor.tabCompletion": "onlySnippets",
25       "emmet.triggerExpansionOnTab": true,
26       "emmet.excludeLanguages": [
27         "markdown"
28       ]
29     }
30   }
31 }
```

The screenshot shows the Visual Studio Code interface with the settings.json file open in the editor. The file contains JSON configuration for various extensions like eslint, editor, and emmet. The 'emmet.triggerExpansionOnTab' setting is explicitly set to true. The code editor shows syntax highlighting for the JSON file.

# Bootstrap

**Everything is now divided into Rows and Columns.**

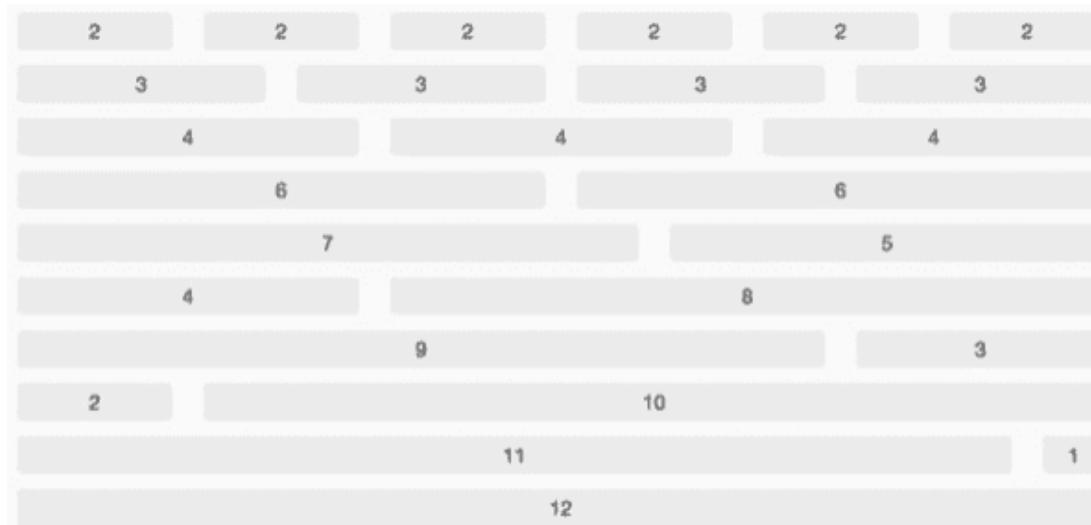
Bootstrap creators replicated the same concept that <tables> had, but instead of using tables, they used <div> (boxed containers). Tags must stay the same – that's why they decided to override the <div> default behaviors with classes.

This is a row: <div class="row">

This is a column: <div class="col-sm-x">

# Bootstrap

---



## Grid system

Bootstrap includes a responsive, mobile first fluid grid system that appropriately scales up to 12 columns as the device or viewport size increases. It includes [predefined classes](#) for easy layout options.

# Responsiveness

## Bootstrap is 100% Responsive

It's very easy to decide how your website will render in different screen sizes; when you add each column into the rows you need to assign a class with the following format:

```
1 | <div class="col-md-x">
```

**Col**

Means that this element should behave like a Bootstrap column.

**md**

Means that I am specifying only for the devices with a "medium" sized screen.

**x**

Specifies how many slots I want this column to take (remember you can take a max of 12 slots per row).

<b>Bootstrap device sizes:</b>	<b>Phones</b>	<b>Big-phone/small-tablet</b>	<b>Tablets</b>	<b>Desktops</b>	<b>Extra-large desktops</b>
--------------------------------	---------------	-------------------------------	----------------	-----------------	-----------------------------

Nothing	sm	md	lg	xl
---------	----	----	----	----

**Note:** if you don't specify the screen size (ex. by using 'sm', 'md', or 'xl'), the website will be rendered for mobile phones by default.

## Containers #

Containers are the most basic layout element in Bootstrap and are **required when using our default grid system**. Choose from a responsive, fixed-width container (meaning its `max-width` changes at each breakpoint) or fluid-width (meaning it's `100%` wide all the time).

While containers *can* be nested, most layouts do not require a nested container.



```
<div class="container">  
  <!-- Content here -->  
</div>
```

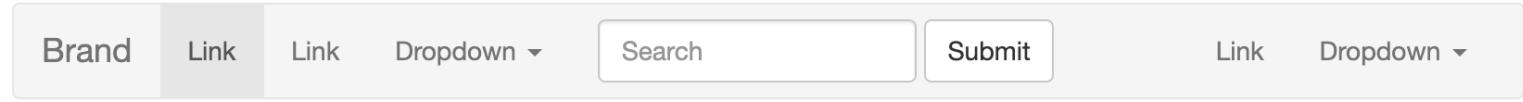
Copy

# Bootstrap Components

## The NavBar

This is so popular that it's in the menu of 99% of all websites. It normally has the logo of the company and a series of links – depending on each website's business logic.

### EXAMPLE



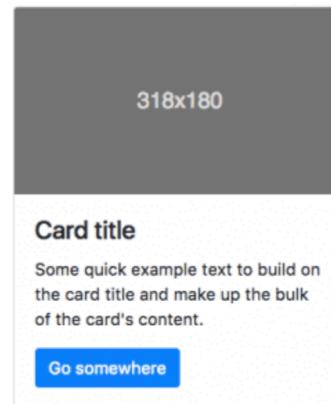
# Bootstrap Components

## The Card

This is probably the most used Bootstrap component, every website has a few cards because it is ideal to list items in a beautiful way. Some examples of the Card used can be:

- The "team" section of a website where you list the different employees.
- The typical Pinterest wall.
- Any social media feed like Instagram, Facebook, twitter, etc.

Here is an example of how a "The Card" may look on a website:

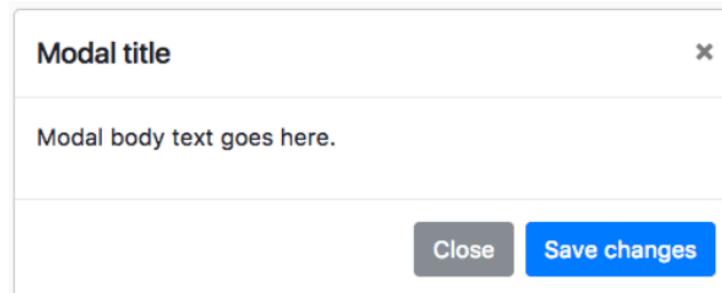


# Bootstrap Components

## The Modal

Everyone hates a modal, it is super annoying, always asking you to subscribe to a newsletter! 😊

Here is how a modal looks by default on Bootstrap.



{ } angular.json      ◊ app.component.html X

src > app > ◊ app.component.html > div.container

You, seconds ago | 1 author (You)

```
1 <div class="container">
2   <div class="row">
3     <div class="col-md-12">
4       <h2>I'm working</h2>
5     </div>
6   </div>
7 </div>
```

You, 5 minutes ago • Uncommitted changes

Bootstrap Tutorial: Learn Bootstrap | Angular - The Complete Guide | Shopping

http://localhost:4200

Apps real-world-vue LaunchCode Dev tools Buying Fitness INSPIRE Vue learning Bayer Sites LEPSI Bayer Personal

I'm working

div.container 1140x46

Elements Sources Network Performance Memory Security Application Lighthouse

<!DOCTYPE html>
<html lang="en">
 <head></head>
 <body>
 <app-root \_ngcontent-apj-c11 ng-version="11.0.2">
 <div \_ngcontent-apj-c11 class="container">
 <div \_ngcontent-apj-c11 class="row">
 <div \_ngcontent-apj-c11 class="col-md-12">
 <h2>I'm working</h2>
 </div>
 </div>
 </div>
 <script src="runtime.js" defer=></script>
 <script src="polyfills.js" defer=></script>
 <script src="vendor.js" defer=></script>
 <script src="main.js" defer=></script>
 </body>
</html>

Styles Computed Layout Event Listeners DOM Breakpoints

:hover .cls + [ ]

element.style { }

.h2, h2 {
 font-size: 2rem;
}
.h1, .h2, .h3, .h4, .h5, .h6, h1, h2, h3, h4,
.h5, .h6 {
 margin-bottom: .5rem;
 font-weight: 500;
 line-height: 1.2;
}

h1, h2, h3, h4, h5, h6 {
 margin-top: 0;
 margin-bottom: .5rem;
}

\*::after, \*::before {
 box-sizing: border-box;
}

h2 {
 display: block;
 font-size: 2.5em;
 margin-block-start: 0.83em;
 margin-block-end: 0.83em;
 margin-inline-start: 0px;
 margin-inline-end: 0px;
 font-weight: bold;
}

Inherited from body

body {
 margin: 0;
 font-family: -apple-system,BlinkMacSystemFont,"Segoe UI",Roboto,"Helvetica Neue",Arial,"Noto Sans",sans-serif,"Apple Color Emoji","Segoe UI Emoji","Segoe UI Symbol","Noto Color Emoji";
 font-size: 1rem;
 font-weight: 400;
 line-height: 1.5;
 color: #212529;
 text-align: left;
 background-color: #fff;
}

# Creating the Components

```
ng g c header
```

--spec false will skip creating spec files.

Add <app-header> to the app.component.html

```
ng g c recipes
```

```
ng g c recipes/recipe-list
```

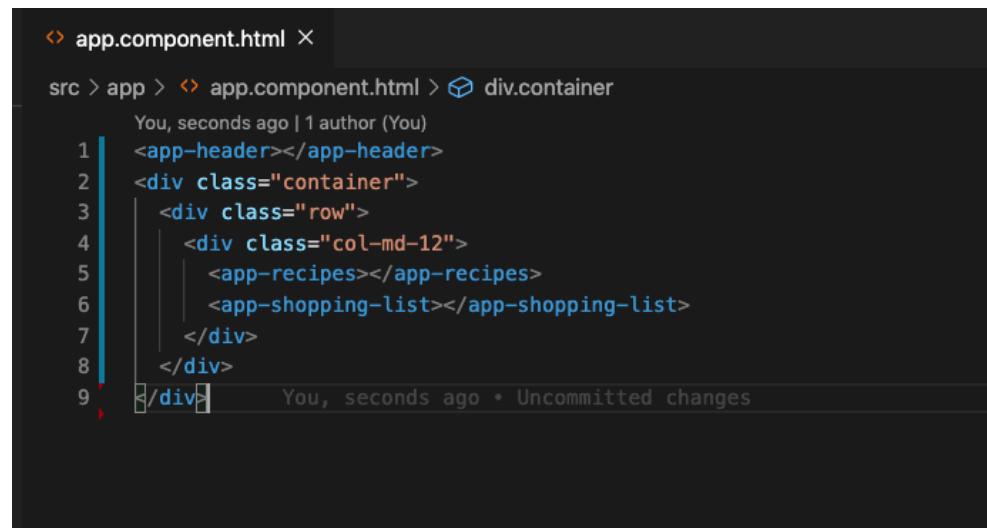
```
ng g c recipes/recipe-detail
```

```
ng g c recipes/recipe-list/recipe-item
```

```
ng g c shopping-list
```

```
ng g c shopping-list/shopping-edit
```

# Using the Components

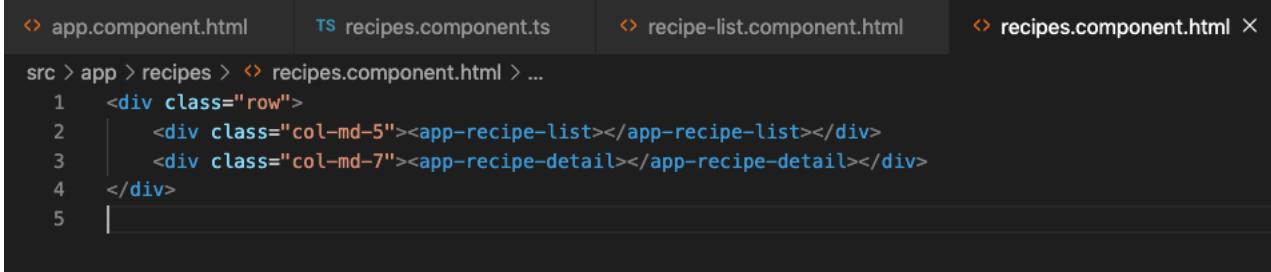


The screenshot shows a code editor window with the file `app.component.html` open. The code is as follows:

```
<app-header></app-header>
<div class="container">
  <div class="row">
    <div class="col-md-12">
      <app-recipes></app-recipes>
      <app-shopping-list></app-shopping-list>
    </div>
  </div>
</div>
```

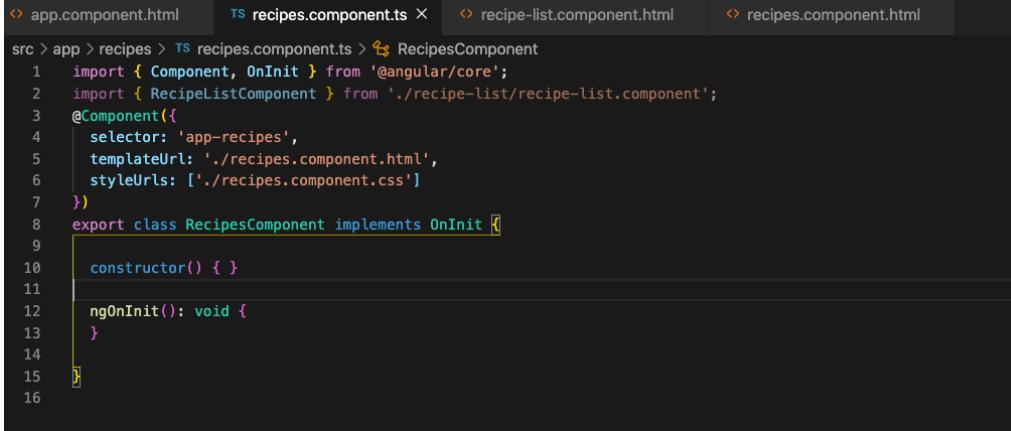
The code editor interface includes a navigation bar at the top with icons for file operations, and a status bar at the bottom indicating "You, seconds ago • Uncommitted changes".

# Using the Components



The screenshot shows a code editor with four tabs at the top: 'app.component.html', 'recipes.component.ts', 'recipe-list.component.html', and 'recipes.component.html'. The 'recipes.component.html' tab is active, showing the following template code:

```
src > app > recipes > recipes.component.html > ...
1  <div class="row">
2    <div class="col-md-5"><app-recipe-list></app-recipe-list></div>
3    <div class="col-md-7"><app-recipe-detail></app-recipe-detail></div>
4  </div>
5
```



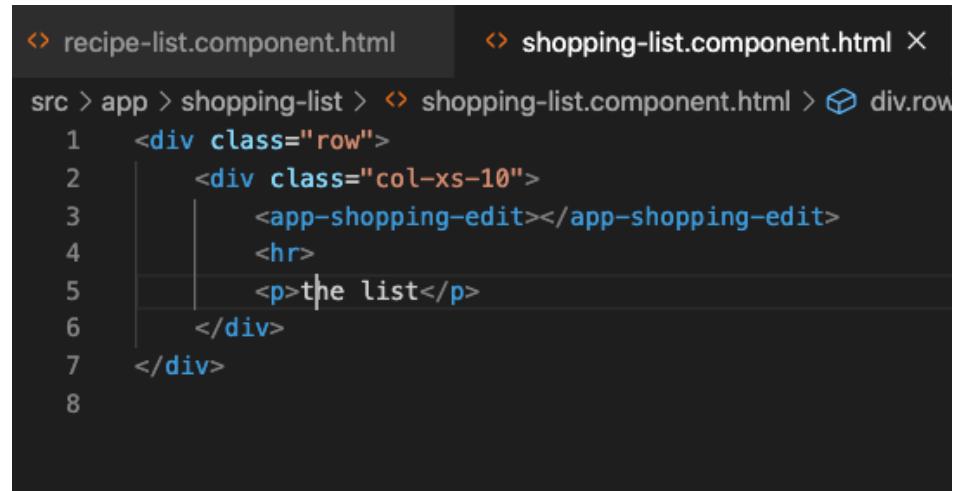
The screenshot shows a code editor with four tabs at the top: 'app.component.html', 'recipes.component.ts', 'recipe-list.component.html', and 'recipes.component.html'. The 'recipes.component.ts' tab is active, showing the following component class code:

```
src > app > recipes > recipes.component.ts > RecipesComponent
1  import { Component, OnInit } from '@angular/core';
2  import { RecipeListComponent } from './recipe-list/recipe-list.component';
3  @Component({
4    selector: 'app-recipes',
5    templateUrl: './recipes.component.html',
6    styleUrls: ['./recipes.component.css']
7 })
8 export class RecipesComponent implements OnInit {
9
10  constructor() { }
11
12  ngOnInit(): void {
13  }
14
15 }
16
```

# Using the Components

```
<> recipe-list.component.html X  
src > app > recipes > recipe-list > <> recipe-list.component.html > ...  
1   <app-recipe-item></app-recipe-item>  
2
```

# Adding the shopping-list Components



The image shows a code editor interface with two tabs: "recipe-list.component.html" and "shopping-list.component.html". The "shopping-list.component.html" tab is active, showing the following code:

```
src > app > shopping-list > shopping-list.component.html > div.row
1  <div class="row">
2    <div class="col-xs-10">
3      <app-shopping-edit></app-shopping-edit>
4      <hr>
5      <p>the list</p>
6    </div>
7  </div>
8
```

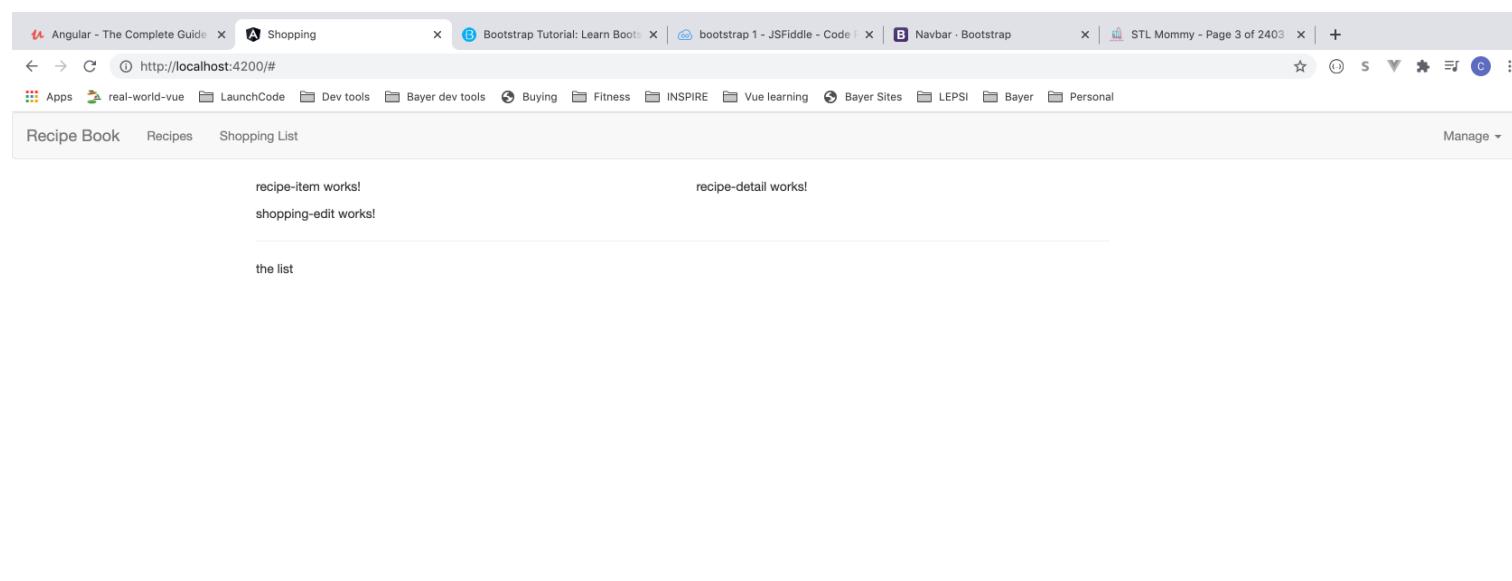
The code is written in HTML and uses Bootstrap's grid system with classes like "row" and "col-xs-10". It includes an Angular component tag "" and a simple paragraph element "

the list

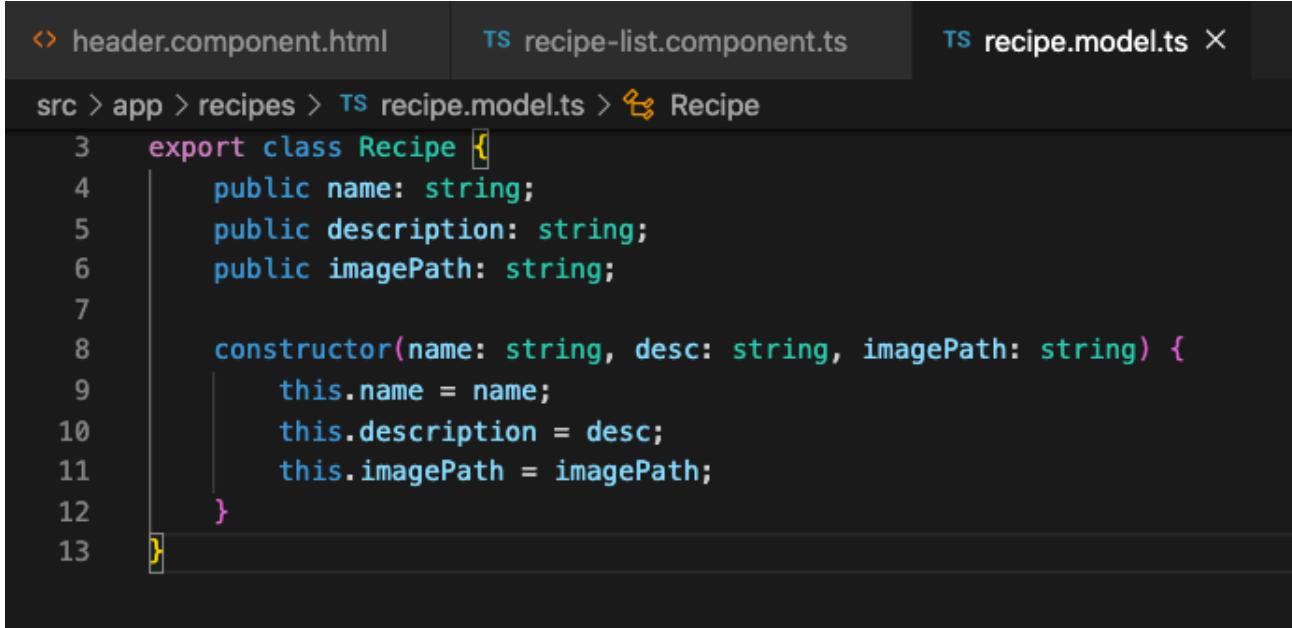
". Line numbers 1 through 8 are displayed on the left side of the code.

# Header Components

```
'c > app > header > <> header.component.html > nav.navbar.navbar-default > div.container-fluid > div.collapse.navbar-collapse
1   <nav class="navbar navbar-default">
2     <div class="container-fluid">
3       <div class="navbar-header">
4         <a href="#" class="navbar-brand">Recipe Book</a>
5       </div>
6       <div class="collapse navbar-collapse">
7         <ul class="nav navbar-nav">
8           <li><a href="#">Recipes</a></li>
9           <li><a href="#">Shopping List</a></li>
10      </ul>
11      <ul class="nav navbar-nav navbar-right">
12        <li class="dropdown">
13          <a href="#" class="dropdown-toggle" role="button">Manage <span class="caret"></span></a>
14          <ul class="dropdown-menu">
15            <li><a href="#">Save Data</a></li>
16            <li><a href="#">Fetch Data</a></li>
17          </ul>
18        </li>
19      </ul>
20    </div>
21  </div>
22 </nav>
```



# Recipe Component



The screenshot shows a code editor with three tabs at the top: "header.component.html", "recipe-list.component.ts", and "recipe.model.ts". The "recipe.model.ts" tab is active, showing the following TypeScript code:

```
src > app > recipes > recipe.model.ts > Recipe
3   export class Recipe {
4     public name: string;
5     public description: string;
6     public imagePath: string;
7
8     constructor(name: string, desc: string, imagePath: string) {
9       this.name = name;
10      this.description = desc;
11      this.imagePath = imagePath;
12    }
13 }
```

# Recipe Component

```
app/recipes/recipe-list/recipe-list.component.ts ...
import { Component, OnInit } from '@angular/core';
import { Recipe } from '../recipe.model';

@Component({
  selector: 'app-recipe-list',
  templateUrl: './recipe-list.component.html',
  styleUrls: ['./recipe-list.component.css']
})
export class RecipeListComponent implements OnInit {

  recipes: Recipe[] = [
    new Recipe('A Test Recipe', 'This is simple a test', 'https://pixabay.com/photos/vegetables-carrot-food-healthy-1085063/')
  ];

  constructor() { }

  ngOnInit(): void {
  }
}
```

```
app/recipes/recipe-list/recipe-list.component.html ...
<div class="row">
  <div class="col-xs-12">
    <button class="btn btn-success">New Recipe</button>
  </div>
  <div class="row"></div>
  <div class="col-xs-12">
    <a href="#" class="list-group-item clearfix">
      <div class="pull-left">
        <h4 class="list-group-heading">Recipe Name</h4>
        <p class="list-group-item-text">Description</p>
        <span class="pull-right">
          <img src="" alt="" class="img-responsive" style="max-height: 50px;">
        </span>
      </div>
    </a>
    <app-recipe-item></app-recipe-item>
  </div>
</div>
```

# Recipe Component

```
html      TS recipe-list.component.ts    TS recipes.component.ts    # recipe-list.component.css    TS recipe-item.component.ts    < re
rc > app > recipes > recipe-list > < recipe-list.component.html > ...
1   <div class="row">
2     <div class="col-xs-12">
3       <button class="btn btn-success">New Recipe</button>
4     </div>
5     <div class="row"></div>
6     <div class="col-xs-12">
7       <a href="#" class="list-group-item clearfix" *ngFor="let recipe of recipes">
8         <div class="pull-left">
9           <h4 class="list-group-heading">{{recipe.name}}</h4>
10          <p class="list-group-item-text">{{recipe.description}}</p>
11          <span class="pull-right">
12            
13          </span>
14        </div>
15      </a>
16      <app-recipe-item></app-recipe-item>
17    </div>
18  </div>
19
```

← → ⌂ ⓘ http://localhost:4200/#

\_apps \_real-world-vue \_LaunchCode \_Dev tools \_Bayer dev tools \_Buying \_Fitness \_INSPIRE \_Vue learning \_Bayer Sites \_LEPSI

Recipe Book   Recipes   Shopping List

New Recipe

A Test Recipe

This is simple a test



recipe-item works!

recipe-detail works!

shopping-edit works!

the list

# Recipe Component

```
nl | TS recipe-list.component.ts | TS recipes.component.ts | # recipe-list.component.css | TS recipe-item.component.ts | ⌂  
|> app > recipes > recipe-list > ⌂ recipe-list.component.html > ⌂ div.row > ⌂ div.col-xs-12  
<div class="row">  
  <div class="col-xs-12">  
    <button class="btn btn-success">New Recipe</button>  
  </div>  
  <div class="row"></div>  
  <div class="col-xs-12">  
    <a href="#" class="list-group-item clearfix" *ngFor="let recipe of recipes">  
      <div class="pull-left">  
        <h4 class="list-group-heading">{{recipe.name}}</h4>  
        <p class="list-group-item-text">{{recipe.description}}</p>  
        <span class="pull-right">  
          <img [src]="recipe.imagePath" alt="{{recipes.name}}" class="img-responsive" style="max-height: 50px;">  
        </span>  
      </div>  
    </a>  
    <app-recipe-item></app-recipe-item>  
  </div>  
</div>
```

← → ⌂ ⓘ http://localhost:4200/#

\_apps \_real-world-vue \_LaunchCode \_Dev tools \_Bayer dev tools \_Buying \_Fitness \_INSPIRE \_Vue learning \_Bayer Sites \_LEPSI

Recipe Book Recipes Shopping List

New Recipe

A Test Recipe

This is simple a test



recipe-item works!

shopping-edit works!

the list

recipe-detail works!

# Recipe Detail Component

recipe-detail.component.html ×

```
src > app > recipes > recipe-detail > recipe-detail.component.html > div.row > div.col-xs-12
1  <div class="row">
2    <div class="col-xs-12">
3      |   <img src="" alt="" class="img-responsive">
4    </div>
5  </div>
6  <div class="row">
7    <div class="col-xs-12">
8      |   <h1>Recipe Name</h1>
9    </div>
10 </div>
11 <div class="row">
12   <div class="col-xs-12">
13     <div class="btn-group">
14       <button type="button" class="btn btn-primary dropdown-toggle">
15         |   Manage Recipe <span class="caret"></span>
16       </button>
17       <ul class="dropdown-menu">
18         |   <li><a href="#">To Shopping List</a></li>
19         |   <li><a href="#">Edit Recipe</a></li>
20         |   <li><a href="#">Delete Recipe</a></li>
21       </ul>
22     </div>
23   </div>
24 </div>
25 <div class="row">
26   <div class="col-xs-12">
27     |   Description
28   </div>
29 </div>
30 <div class="row">
31   <div class="col-xs-12">
32     |   Ingredients
33   </div>
34 </div>
```

Recipe Book   Recipes   Shopping List

New Recipe

A Test Recipe

This is simple a test



A Test Recipe

This is simple a test



recipe-item works!

shopping-edit works!

Recipe Name

Manage Recipe ▾

Description  
Ingredients

the list

# Shopping List Component

```
src > app > shared > ts ingredient.model.ts > Ingredient
  3  export class Ingredient {
  4    constructor(public name: string, public amount: number) {
  5    }
  6  }

<div class="row">
  <div class="col-xs-10">
    <app-shopping-edit></app-shopping-edit>
    <hr>
    <ul class="list-group">
      <a
        class="list-group-item"
        style="cursor: pointer"
        *ngFor="let ingredient of ingredients"
        >
        {{ ingredient.name }} {{ ingredient.amount }}
      </a>
    </ul>
  </div>
</div>
```

```
ts shopping-list.component.ts ×
src > app > shopping-list > ts shopping-list.component.ts > ShoppingList
  1  import { Ingredient } from './shared/ingredient.model'
  2  import { Component, OnInit } from '@angular/core';
  3  @Component({
  4    selector: 'app-shopping-list',
  5    templateUrl: './shopping-list.component.html',
  6    styleUrls: ['./shopping-list.component.css']
  7  })
  8  export class ShoppingListComponent implements OnInit {
  9
 10    ingredients: Ingredient[] = [
 11      new Ingredient('Apple', 5),
 12      new Ingredient('Tomato', 10),
 13    ];
 14    constructor() { }
 15
 16    ngOnInit(): void {
 17    }
 18  }
```

New Recipe

A Test Recipe

This is simple a test



A Test Recipe

This is simple a test



recipe-item works!

shopping-edit works!

Apple 5
Tomato 10

Recipe Name

Manage Recipe ▾

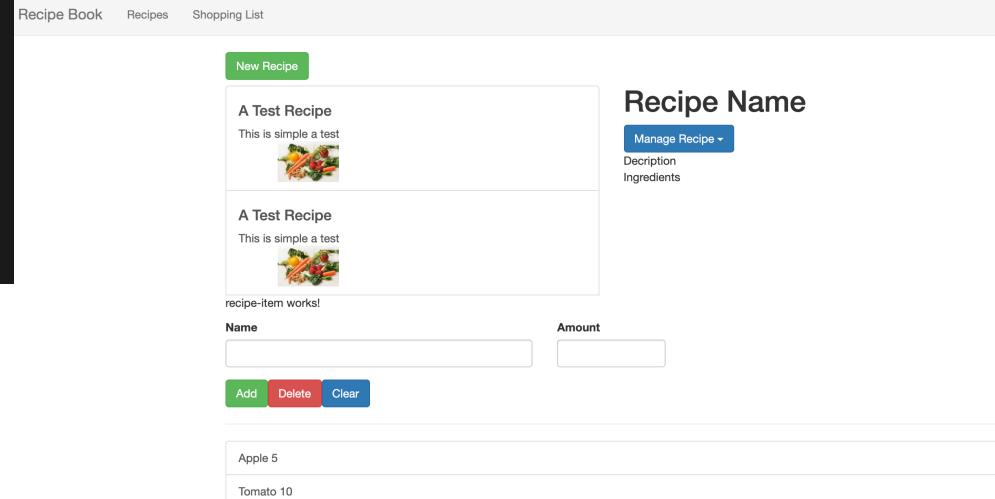
Description

Ingredients

# Shopping List Edit Component

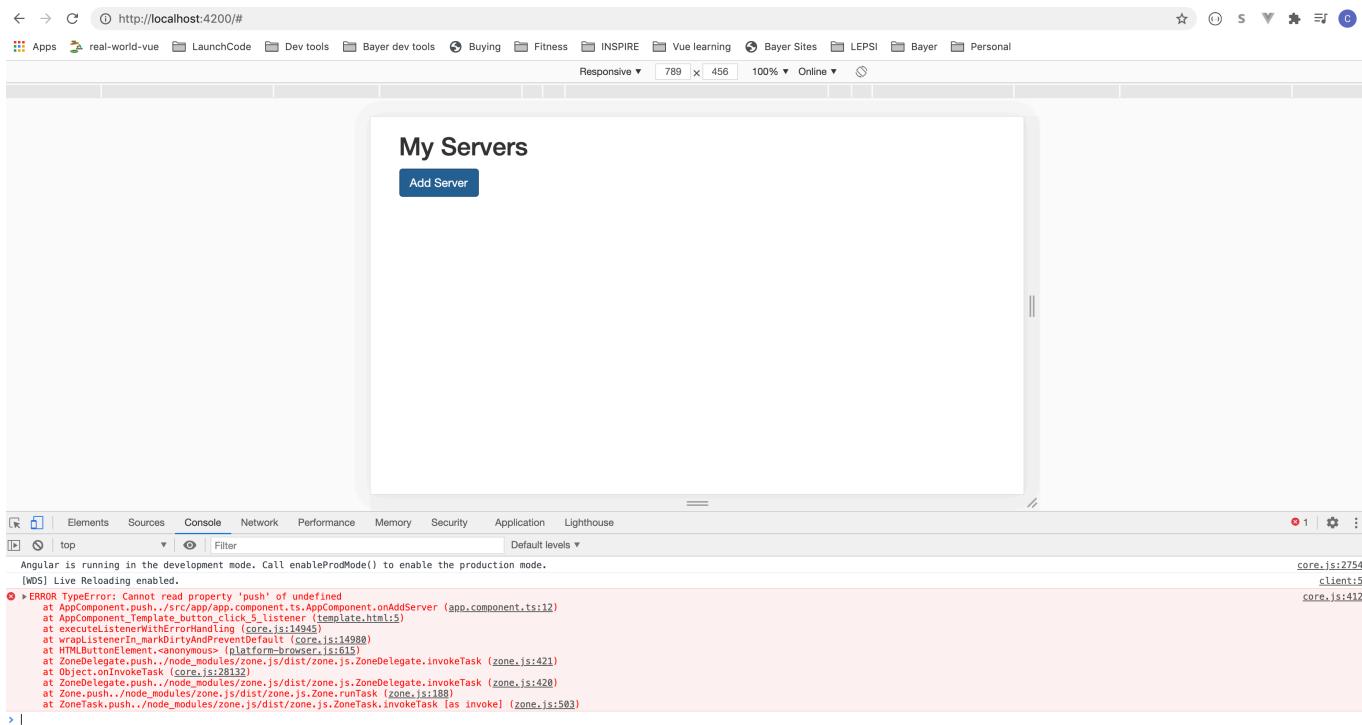
shopping-edit.component.html

```
src > app > shopping-list > shopping-edit > shopping-edit.component.html > ...
1  <div class="row">
2    <div class="col-xs-12">
3      <form>
4        <div class="row">
5          <div class="col-sm-5 form-group">
6            <label for="name">Name</label>
7            <input type="text" id="name" class="form-control">
8          </div>
9          <div class="col-sm-2 form-group">
10            <label for="amount">Amount</label>
11            <input type="number" id="amount" class="form-control">
12          </div>
13        </div>
14        <div class="row">
15          <div class="col-xs-12">
16            <button class="btn btn-success" type="submit">Add</button>
17            <button class="btn btn-danger" type="button">Delete</button>
18            <button class="btn btn-primary" type="button">Clear</button>
19          </div>
20        </div>
21      </form>
22    </div>
23  </div>
```



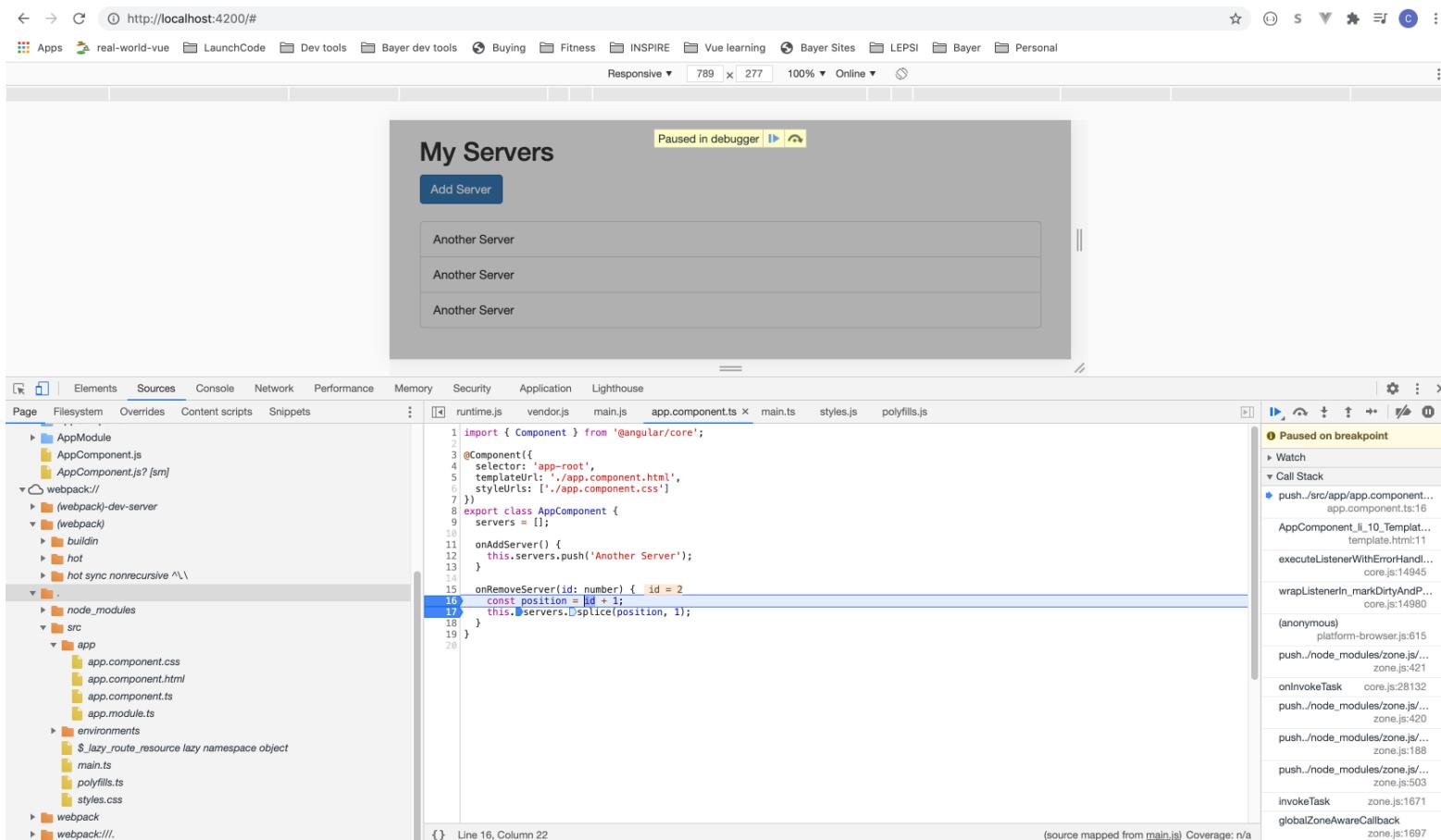
# Debugging

## Error handling – check console in Developer tools



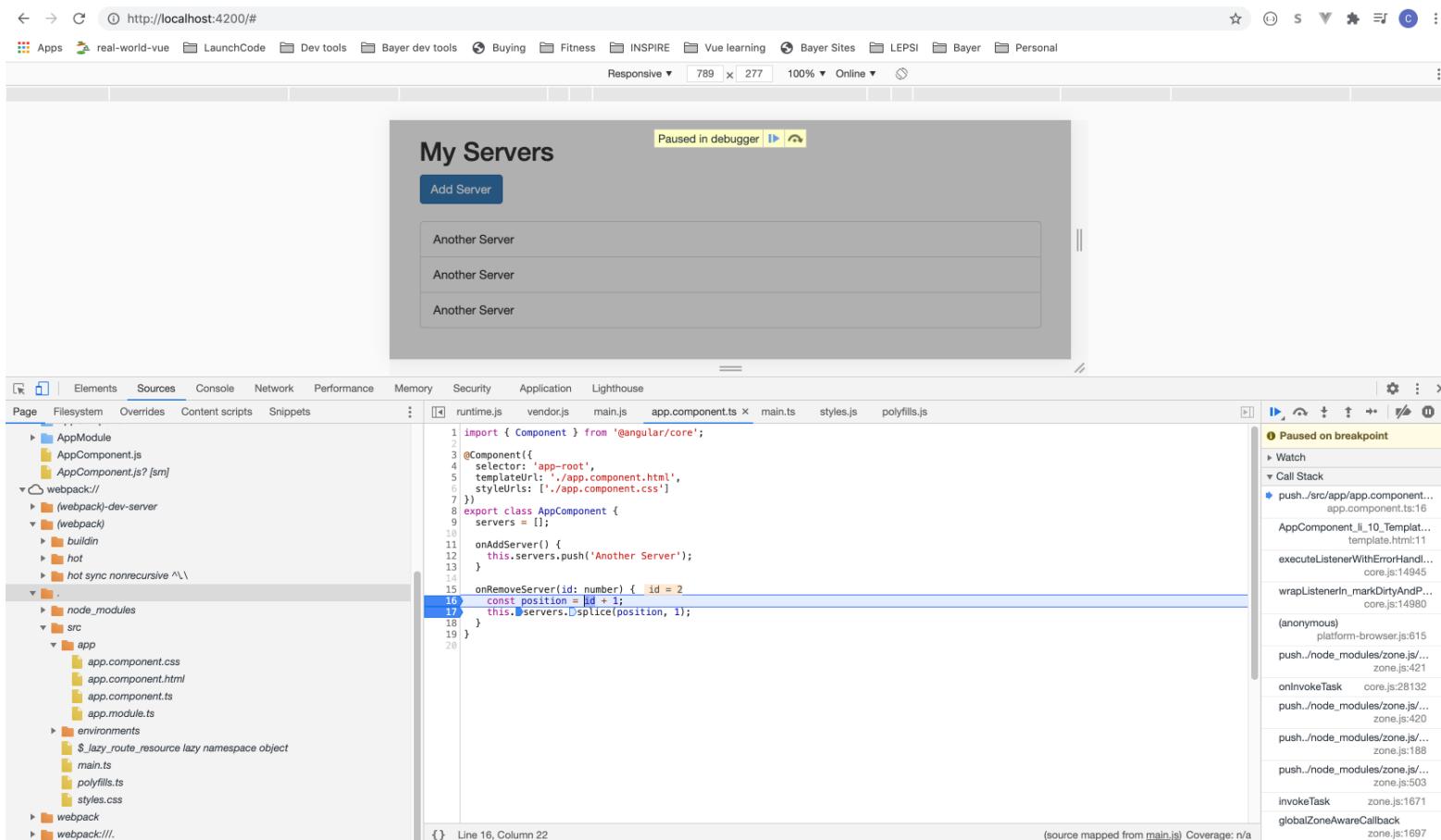
# Debugging

Can add breakpoints in code via

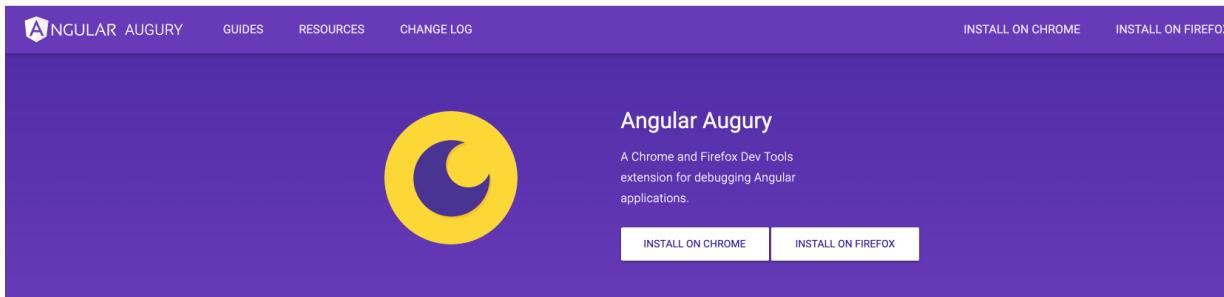


# Debugging with source

Can add breakpoints in code via



# Angular Augury



## What is Augury

Augury is the most used Developer Tool extension for debugging and profiling Angular applications inside the Google Chrome and Mozilla Firefox browsers.

## Why Use Augury?

Augury helps Angular developers visualize the application through component trees, and visual debugging tools. Developers get immediate insight into their application structure, change detection and performance characteristics.

## Open Source

Augury is an open source effort with [Google](#) and [Rangle.io](#). Developers can contribute and fork the

Tool to understand app at runtime and dependencies

Questions?

Learning Environment

# Studio!