

Custom Pipes

Introducing Custom Pipes

In the Angular Fundamentals course, we learned that pipes are handy ways of providing transformations to values on template binding. They're useful for all sorts of reasons, from formatting dates to capitalizing letters of lowercase-formatted data.

Angular already includes some “built-in” pipes that you can put to work immediately but what if the functionality doesn't exist in these built-in pipes, for a specific transformation that you want? This is where custom pipes come into play.

In a custom pipe, you write the logic in a function and that logic performs the necessary transformation that you desire.

Creating Custom Pipes

When we are ready to create our own custom pipes, we start by, as you might guess by now, with the CLI. Running the following command will generate a new custom pipe called favorite-text.

```
ng generate pipe favorite-text
```

Angular should inform you that the necessary files have been generated and that the app.module.ts file has been modified. Let's look at the changes to the app.module.ts file first.

The first item you might notice is a new import statement at the top of the file followed by the name of the new custom pipe in the @NgModule declarations section. The name of the pipe is derived from the name you supplied in the generate command.

The generate command also creates the favorite-text.pipe.ts file that looks like this:

```
import { Pipe, PipeTransform } from '@angular/core';

@Pipe({
  name: 'favoriteText'
})
export class FavoriteTextPipe implements PipeTransform {

  transform(value: any, args?: any): any {
    return null;
  }
}
```

Notice the import statement that brings in Pipe and PipeTransform from angular core.

Next you will notice the @Pipe declaration with a name specified. As in most Angular classes, the export statement is included so this can be used in other classes in your Angular application.

Your logic will exist within the transform method in this class. This logic is simply whatever is required to perform the necessary transformation that you desire. In task 2 of the Custom Pipe lab, you will implement the logic to perform a custom transform using this favorite text custom pipe.

To Create a Custom Pipe

In this lab, we'll create a custom pipe to modify our title to indicate a favorite when it is selected as one.

1. Open up your command prompt or terminal and point it at your advanced-angular project.
2. Type the following `ng generate pipe favorite-text`. You should see a confirmation that your pipe files were created, and that your `app.module.ts` was modified.
3. Open up your `src/app/favorite-text.pipe.ts` file in Visual Studio Code. It should look like the following:

```
import { Pipe, PipeTransform } from '@angular/core';

@Pipe({
  name: 'favoriteText'
})
export class FavoriteTextPipe implements PipeTransform {

  transform(value: any, args?: any): any {
    return null;
  }

}
```

As you can see, the Angular CLI has configured this to be ready to perform a `PipeTransform` on the value we are passing to it. The transform function is where you perform the actual transformation on the value being passed through the pipe's first parameter.

4. How we are going to build this is we are going to pass the argument of whether an item is favorited or not into the arguments for the pipe. These get passed to the second parameter of the transform function, if only one value is passed as a single value, otherwise as an array. We'll check that, and then add `* Favorite` to the end of the title if that is the case. Otherwise we'll return the value as-is.

```
export class FavoriteTextPipe implements PipeTransform {

  transform(value: any, args?: any): any {
    if (args) {
      return value + '* Favorite';
    }
    else {
      return value
    }
  }

}
```

```
}  
  
}
```

That's it! Our pipe is ready for use. Let's apply it to our template now.

5. Open up your `src/app/repository-display/repository-display.component.html` file in Visual Studio Code. It should look like this at this point:

```
<li [ngStyle]='{"background-color" : (i % 2 === 0) ? "silver" : "white"}' class="list_item" *ngFor="let result  
of searchResults.items; index as i;" appFade>  
  <a [href]="result.html_url">  
    <img class="avatar" [src]="result.owner.avatar_url" />  
    <h4 class="title">{{result.name}}  
      <small> by {{result.owner.login | uppercase}}</small>  
    </h4>  
  </a>  
  <span class="favorite_icon" *ngIf="!checkFavorite(result)" (click)="addFavorite(result)"  
>&#9734;</span>  
  <span class="favorite_icon favorited" *ngIf="checkFavorite(result)">&#9733;</span>  
  <p class="description"> {{result.description}}</p>  
  <p> Created On: {{result.created_at | date:'fullDate'}} </p>  
</li>
```

6. Let's apply our new pipe, `favoriteText` to our `result.name` binding, with the argument passed being our `checkFavorite(result)` function to determine if the item is a favorite.

```
<h4 class="title">{{result.name | favoriteText:checkFavorite(result)}}  
  <small> by {{result.owner.login | uppercase}}</small>  
</h4>
```

7. Apply the same pipe to your `src/app/code-display/code-display.component.html` file. When you're done it should look like this:

```
<li [ngStyle]='{"background-color" : (i % 2 === 0) ? "springgreen" : "palegreen"}' class="list_item"  
*ngFor="let result of searchResults.items; index as i;" appFade>  
  <a [href]="result.html_url">  
    <img class="avatar" [src]="result.repository.owner.avatar_url" />  
    <h4 class="title">Code Result: {{result.name | favoriteText:checkFavorite(result)}}  
      <small> by {{result.repository.owner.login | uppercase}}</small>  
    </h4>  
  </a>  
  <span class="favorite_icon" *ngIf="!checkFavorite(result)" (click)="addFavorite(result)"  
>&#9734;</span>  
  <span class="favorite_icon favorited" *ngIf="checkFavorite(result)">&#9733;</span>  
  <p class="description"> {{result.repository.description}}</p>  
  <p> Path: {{result.path}} </p>  
</li>
```

- And we're done! Open up a command prompt/terminal and point it to your advanced-angular project and type `ng serve` to start your server if it isn't already running.
- If you don't have a browser window open, open one and point it to `localhost:4200` and click "Search for Repositories". You should see the following:

Git Search - angular

q
 q is invalid.

language
 user
 size
 stars
 topic

Total Results:





[angular by ANGULAR](#)

One framework. Mobile & desktop.

Created On: Thursday, September 18, 2014

- Click on one of the stars, and you should see the text transform in the title to indicate that you marked it as a favorite - like so:

Git Search - angular

q
 q is invalid.

language
 user
 size
 stars
 topic

Total Results:





[angular* Favorite by ANGULAR](#)

One framework. Mobile & desktop.

Created On: Thursday, September 18, 2014





And that's it! Now you have learned some more advanced concepts in working with Components - including using custom Directives and Pipes. Next up is the self-assessment, where you'll put these skills to use on your own.