Tuesday: Create Pipes

Creating Custom Pipes

As we had stated earlier, if the inbuilt angular pipes don't help us to achieve the data transformation we want, we have the liberty to create our own custom pipes to do it. Let's create a pipe that will count the number of days left to the completion of a goal.

On our terminals, let's create a pipe using the following command:

```
$ ng generate pipe date-count
```

This command generates two files, *date-count.pipe.ts* which we'll use to write the code for counting the number of days left and *date-count.pipe.spec.ts* which is the test file for our pipe. The command also adds this pipe to our root level modules file, *app.module.ts*, for us.

Let's now define the logic to count the number of days left to the completion of a goal.

src/app/date-count.pipe.ts

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

@Pipe({
  name: 'dateCount'
})
export class DateCountPipe implements PipeTransform {

  transform(value: any): number {
    let today:Date = new Date(); //get current date and time
    let todayWithNoTime:any = new Date(today.getFullYear(), today.getMonth(), today.getDate())
    var dateDifference = Math.abs(value - todayWithNoTime) //returns value in miliseconds
    const secondsInDay = 86400; //60 seconds * 60 minutes in an hour * 24 hours in a day
    var dateDifferenceSeconds = dateDifference*0.001; //converts miliseconds to seconds
    var dateCounter = dateDifferenceSeconds/secondsInDay;

    if (dateCounter >= 1 && value > todayWithNoTime){
      return dateCounter;
    }else{
      return 0;
    }
  }
}
```

We first have the `@Pipe` decorator that defines the *name* of the pipe. We then have the `DateCountPipe` class that has a transform function that takes in a value of type any. This value will be our date object from the goals.

We have used the `new Date()`; function to get the current date and time. But since we do not need the time aspect we create a new date object for today's date which we store in the `todayWithNoTime` variable.

We have then calculated the difference of the date we get in the pipe input and subtracted the current date to get the difference in milliseconds.

We have converted the milliseconds to seconds and then divided these seconds by the seconds in a day to get how many days are left.

We have then returned the days left if the goal date has not been passed. But if it past due we post 0.

Let us add this pipe to our goal details and serve our application to see our custom pipe in action.

src/app/goal-details/goal-detail.component.html

```
<p>{{goal.description}}</p>
<p>This goal will be complete in {{goal.completeDate|dateCount}} days</p>
```

We have added the pipe to the `completeDate` property to calcute the number of days left to the completion of a goal.