Pipes

Introducing Pipes

A pipe allows you to use data, as an input to the pipe, and transform it to a more desirable output. As we think about our Angular applications, perhaps the most common reason for creating the application is to access and display data in our web applications. However, the data may be somewhat cryptic or not necessarily “user friendly” as in a data format.

Often times a date may come to in the form:

```
Sat Jun 20 2017 04:00:00 GMT -0800 (Pacific Standard Time)
```

While it is readable to a certain degree, not every user cares about the time, or the offset. Perhaps you want to display this in a different format that expands the month rather than using an abbreviation, or perhaps, you want to switch the day and month around to support a different locale. Pipes help you accomplish this.

Angular actually has a collection if built-in pipes for common tasks such as DatePipe, UpperCasePipe, LowerCasePipe, and CurrencyPipe. You can find out more about these in the API documentation for Angular. You will make use of the DatePipe and UpperCasePipe in the tutorial labs for this module.

Using Pipes

As mentioned in Introducing Pipes, pipes are a concept in Angular that takes data and applies a transformation to it before sending it to the template to be rendered. This can be particularly useful for formatting purposes where consistent data transformations need to be applied to your data.

If you are familiar with command-line or shell environments in operating systems, you may also be familiar with the pipe symbol |. In the command-line environments, this pipe symbol is used to send the output from one command to another command. Likewise, in Angular, we are going to use the pipe symbol to transform the data in our template, by passing it through a pipe to a transformation.

In this code sample, taken from the labs, we will demonstrate applying the built-in date pipe to an Angular binding. The way we accomplish this is to add the pipe or | character after the Angular binding, followed by the built-in pipe command. The command we use here is date:'fullDate' to indicate a format of full date. See the code sample below:

```
<div *ngIf="searchResults; else elseBlock">
  ...
     <p class="description"> {{result.description}}</p>
     <p> Created On: {{result.created_at | date:'fullDate'}} </p>
   </li>
  </ul>
</div>
<ng-template #elseBlock>Loading...</ng-template>
```

To Use Built In Pipes

In this lab, we are going to do two things: apply built-in Pipes to our Angular template to further transform the data output, and use the ngModel built-in directive to create a search input for our application to be able to send custom queries.

We are going to work with a couple of built-in Pipes in Angular: the date pipe and the uppercase pipe.

1. Open up your src/app/git-search/git-search.component.html file in Visual Studio Code. It should look like this at this point:

```
<div *ngIf="searchResults; else elseBlock">
 <h3 class="total">Total Results: {{searchResults.total_count}}</h3>
 <ul class="list">
  <li [ngStyle]="{'background-color' : (i % 2 === 0) ? 'silver' : 'white'}" class="list_item" *ngFor="let result of searchResults.items; index as i;">
   <a [href]="result.html_url">
    <img class="avatar" [src]="result.owner.avatar_url" />
    <h4 class="title">{{result.name}}
     <small> by {{result.owner.login}}</small>
    </h4>
   </a>
   <p class="description"> {{result.description}}</p>
  </li>
 </ul>
</div>
<ng-template #elseBlock>Loading...</ng-template>
```

2. Next we are going to add an additional field from the GitSearch output, created_at, below the description.

```
<div *ngIf="searchResults; else elseBlock">
 <h3 class="total">Total Results: {{searchResults.total_count}}</h3>
 <ul class="list">
  <li [ngStyle]="{'background-color' : (i % 2 === 0) ? 'silver' : 'white'}" class="list_item" *ngFor="let result of searchResults.items; index as i;">
   <a [href]="result.html_url">
    <img class="avatar" [src]="result.owner.avatar_url" />
    <h4 class="title">{{result.name}}
     <small> by {{result.owner.login}}</small>
    </h4>
   </a>
   <p class="description"> {{result.description}}</p>
   <p> Created On: {{result.created_at}} </p>
  </li>
 </ul>
</div>
<ng-template #elseBlock>Loading...</ng-template>
```

3. Now we are going to apply a date pipe to our result.created_at binding. To do so, you add the pipe or | character after your binding, followed by the pipe command, which in this case will be date:'fullDate' to indicate that we want to format this as a full date.

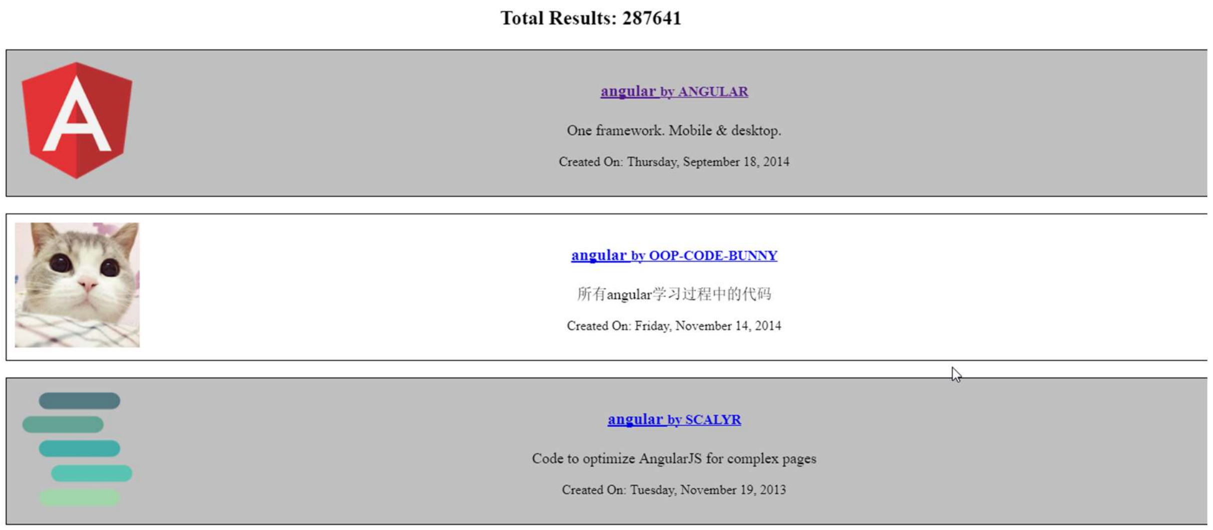
```
<div *ngIf="searchResults; else elseBlock">
  <h3 class="total">Total Results: {{searchResults.total_count}}</h3>
  <ul class="list">
    <li [ngStyle]="{'background-color' : (i % 2 === 0) ? 'silver' : 'white'}" class="list_item" *ngFor="let result of searchResults.items; index as i;">
      <a [href]="result.html_url">
        <img class="avatar" [src]="result.owner.avatar_url" />
        <h4 class="title">{{result.name}}
          <small> by {{result.owner.login}}</small>
        </h4>
      </a>
      <p class="description"> {{result.description}}</p>
      <p> Created On: {{result.created_at | date:'fullDate'}} </p>
    </li>
  </ul>
</div>
<ng-template #elseBlock>Loading...</ng-template>
```

4. Now let's add an uppercase pipe to our repository owner's names in the same fashion, by adding a | followed by uppercase after the template binding.

```
<div *ngIf="searchResults; else elseBlock">
  <h3 class="total">Total Results: {{searchResults.total_count}}</h3>
  <ul class="list">
    <li [ngStyle]="{'background-color' : (i % 2 === 0) ? 'silver' : 'white'}" class="list_item" *ngFor="let result of searchResults.items; index as i;">
      <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>
      <p class="description"> {{result.description}}</p>
      <p> Created On: {{result.created_at | date:'fullDate'}} </p>
    </li>
  </ul>
</div>
<ng-template #elseBlock>Loading...</ng-template>
```

5. Save the file, and then, if your app is not already running: open up your terminal or command prompt, navigate to your angular-fundamentals folder, and use ng serve to start your app.

6. Open up a browser and navigate to localhost:4200 and you should see the following.



Total Results: 287641

angular by ANGULAR

One framework. Mobile & desktop.

Created On: Thursday, September 18, 2014

angular by OOP-CODE-BUNNY

所有angular学习过程中的代码

Created On: Friday, November 14, 2014

angular by SCALYR

Code to optimize AngularJS for complex pages

Created On: Tuesday, November 19, 2013

Congratulations! You have implemented Pipes in your templates. You can see that the date is now formatted properly, and that the user's name is capitalized. For our next task, we will use ngModel to implement a search input in our application.