

ANGULAR 2

PIPES

ANGULAR 2 BOOTCAMP

EPAM SARATOV · SPRING 2017



ABOUT MYSELF

Front-end developer

- 2 years
- Angular 1.x
- C# - ASP.NET MVC 3

Last project - trading system
(Angular 1, WebSockets, React,
SASS, Gulp ...).



LESSON PLAN

- What are Pipes on Angular 2
- Use of Pipes
- Build-in Pipes
- Custom Pipes
- Pure and Impure Pipes
- Async Pipe



WHAT ARE PIPES ON ANGULAR 2?

- Pipes allow us to change data inside of a template

JavaScript

```
var now = new Date();
function formatDate(date) {
  var month = date.getMonth() + 1;
  return date.getDate() + '.'
  + (month < 10 ? '0' : '') + month + '.'
  + date.getFullYear();
}
$("#datepicker").val(formatDate(now));
```

Angular2

```
{{now | date: 'dd.MM.yyyy'}}
```

USING PIPES

- General syntax(template)

```
<p>{{ · expression · | · pipeName:parameter1:parameter2 · }}</p>
```

- Chaining Pipes



```
<p>{{ 'Angular2' | slice:3 | uppercase }}</p>
```

USING PIPES

- Using in code

```
import { AgePipe } from './pipes/custom.pipe';  
import { DatePipe } from '@angular/common';  
  
@Component({  
  selector: 'my-app',  
  templateUrl: '../app/app.html',  
  providers: [AgePipe, DatePipe],  
})
```

USING PIPES

- Using in code

```
export class AppComponent implements OnInit {  
  public currentDay: string = new DatePipe().transform(new Date(), 'yyyy/MM/dd');  
  constructor(private _agePipe: AgePipe) {}  
  ...  
  this.employees = this._agePipe.transform(this.allEmployees, 30);  
  ...  
}
```

BUILT-IN PIPES

- DatePipe

```
curDate: Date = new Date();  
{{expression | date[:format]}}
```

<code><p>output.is: {{curDate date}}</p></code>		Jul 26, 2016
<code><p>output.is: {{curDate date:'medium'}}</p></code>		Jul 26, 2016, 4:35:42 PM
<code><p>output.is: {{curDate date:'short'}}</p></code>		7/26/2016, 4:35 PM
<code><p>output.is: {{curDate date:'mediumTime'}}</p></code>		4:35:42 PM
<code><p>output.is: {{curDate date:'shortTime'}}</p></code>		4:35 PM
<code><p>output.is: {{curDate date:'dd/MM/yyyy'}}</p></code>		26/07/2016

BUILT-IN PIPES

- SlicePipe

```
{{expression | slice:start[:end]}}
```

BUILT-IN PIPES

```
public numCollection = [1, 2, 3, 4, 5, 6];
```

```
<ul>  
  <li *ngFor="let num of numCollection | slice: 2:5">{{num}}</li>  
</ul>
```



```
▼ <ul>  
  <li>3</li>  
  <li>4</li>  
  <li>5</li>  
</ul>
```

BUILT-IN PIPES

- UpperCasePipe

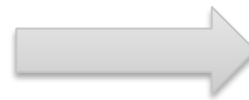
```
{{'SlicePipe' | uppercase}}
```



```
<span>SLICEPIPE </span>
```

- LowerCasePipe

```
{{'SlicePipe' | lowercase}}
```



```
<span>slicepipe </span>
```

BUILT-IN PIPES

- CurrencyPipe

```
{{expression | currency[:currencyCode[:symbolDisplay[:digitInfo]]]}}
```

```
{{0.259 | currency:'EUR':true}} → <p>€0.259</p>
```

```
{{0.259 | currency:'USD':false:'2.1-2'}} → <p>USD00.26</p>
```

https://en.wikipedia.org/wiki/ISO_4217


BUILT-IN PIPES


- DecimalPipe

```
{{expression | number[:digitInfo]}}
```

```
{minIntegerDigits}.{minFractionDigits}-{maxFractionDigits}
```

no formatting: `{{24.718281828459045}}`

`{{24.718281828459045 | number:'1.1-5'}}`  `<p> 24.71828 </p>`

`{{24.718281828459045 | number:'3.2-5'}}`  `<p> 024.71828 </p>`

BUILT-IN PIPES

- JsonPipe

```
{{object | json}}
```

Without JSON pipe:

```
[object Object]
```

With JSON pipe:

```
{  
  "name": "Ivan",  
  "surname": "Ivanov",  
  "details": {  
    "number": 3,  
    "city": "Saratov"  
  }  
}
```

EXAMPLE

<http://plnkr.co/edit/5AKKFMJXXJKYziWX7ODN?p=preview>

CUSTOM PIPES

- How to make the custom pipe available everywhere

```
34  
35 ▾ @NgModule({  
36     imports: [ BrowserModule ],  
37     declarations: [ App, StatusPipe ],  
38     bootstrap: [ App ]  
39 })  
40 export class AppModule {}
```


PURE AND IMPURE PIPES

- **Pure Pipes**

Angular executes a pure pipe only when it detects a pure change to the input value. A pure change is either a change to a primitive input value (String, Number, Boolean, Symbol) or a changed object reference (Date, Array, Function, Object).

Pipes are pure by default.

PURE AND IMPURE PIPES

- Impure Pipes

Angular executes an impure pipe during every component change detection cycle. An impure pipe will be called a lot, as often as every keystroke or mouse-move.

```
import {Pipe, PipeTransform} from '@angular/core';
@Pipe({
  name: 'status',
  pure: false
})
export class StatusImpurePipe implements PipeTransform {
```

EXAMPLE

<http://plnkr.co/edit/08HH8qArwZjOHqhGqXGK?p=preview>

<http://plnkr.co/edit/fJyipEblmeYypCFAPoiz?p=preview>

EXAMPLE

<http://plnkr.co/edit/9ICJ20kzzsHvwbrNS2Lk?p=preview>

HTML SANITIZE PIPE

```
<span [innerHTML]="termInfo.dictionary.disclaimer | safeHtml"></span>
```

```
import {Pipe, PipeTransform} from '@angular/core';
import { DomSanitizer} from '@angular/platform-browser';

@Pipe({
  name: 'safeHtml'
})
export class Safe implements PipeTransform {

  constructor(private sanitizer: DomSanitizer){}

  transform(html: string) {
    return this.sanitizer.bypassSecurityTrustHtml(html);
  }
}
```

USEFUL LINKS

1. <https://www.ng-book.com/2/>
2. <https://angular.io/docs/ts/latest/api/#!?apiFilter=pipe>
3. <https://angular.io/docs/ts/latest/guide/pipes.html>
4. <http://stackoverflow.com/questions/34456430/ngfor-doesnt-update-data-with-pipe-in-angular2>



VADIM KHASHEV, SARATOV, RUSSIA