Unicom TIC

Lab Session [05/10/2024]

Lab Objectives:

- Understand built-in Angular pipes.

- Use parameterized pipes.

- Create custom pipes.

- Implement pure and impure pipes.

Lab Setup:

Pre-requisites:

- Angular CLI installed. If not, install it by running:

```
npm install -g @angular/cli
```

- Create a new Angular project: [If you have already created any angular application, Please use it instead of creating new one.]

```
ng new angular-pipes-lab
```
[No standalone app is preferred]

```
cd angular-pipes-lab
ng serve
```

- Run the application in your browser by visiting: http://localhost:4200/

Lab Sections:

1. Built-in Pipes (20 mins)

1. Task: **Use built-in pipes like `uppercase`, `lowercase`, `currency`, `date`, `json`, etc.**
2. Steps:

- Open and Update ` src/app/app.component.ts` with the following:

```
export class AppComponent {
  today: number = Date.now();
  price: number = 2500.5;
  obj = { name: 'John', age: 25 };
}
```

- Open the src/app/app.component.html file.

- Modify it to display various data using built-in pipes:

```
<h2>Built-in Pipes Demo</h2>

<p><b>Original Text:</b> {{ 'Hello World' }}</p>
<p><b>Uppercase:</b> {{ 'Hello World' | uppercase }}</p>
<p><b>Lowercase:</b> {{ 'Hello World' | lowercase }}</p>
<p><b>Date:</b> {{ today | date }}</p>
<p><b>Currency:</b> {{ price | currency:'USD':true }}</p>
<p><b>JSON:</b> {{ obj | json }}</p>
```

3. **Expected Output**: You should see the following transformed data in your browser:
   - Text converted to uppercase and lowercase.
   - Current date formatted.
   - Price displayed in USD.
   - An object formatted as JSON.

2. Parameterized Pipes (20 mins)

1. Task: **Work with parameterized pipes like `slice`, `date` with custom formats, etc.**

2. Steps:

- Modify `app.component.html`:

```
<h2>Parameterized Pipes Demo</h2>

<p><b>Slice:</b> {{ 'Angular Pipes' | slice:0:7 }}</p>
<p><b>Custom Date Format:</b> {{ today | date:'fullDate' }}</p>
<p><b>Currency with Code:</b> {{ price | currency:'EUR' }}</p>
```

3. Expected Output: Data will be sliced, formatted in a custom date format, and the price will be shown in EUR.

3. Creating Custom Pipes (30 mins)

1. Task: **Create a custom pipe that converts a string into title case (first letter of each word capitalized).**

2. Steps:

- Generate a new pipe using the Angular CLI:

```
ng generate pipe titlecase
```

- Implement the `transform` method in the `titlecase.pipe.ts` file:

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

@Pipe({
  name: 'titlecase'
})
export class TitlecasePipe implements PipeTransform {

  transform(value: string): string {
    return value.split(' ')
                .map(word => word.charAt(0).toUpperCase() +
word.slice(1).toLowerCase())
                .join(' ');
  }

}
```

- Use the custom pipe in `app.component.html`:

```
<h2>Custom Pipe Demo</h2>

<p><b>Titlecase:</b> {{ 'angular custom pipe' | titlecase }}</p>
```

3. Expected Output: The string should be displayed as `Angular Custom Pipe`.

4. Pure vs Impure Pipes (30 mins)

1. Task: **Understand the difference between pure and impure pipes by creating an impure pipe.**
2. Steps:

- Modify the `titlecase.pipe.ts` to implement an impure pipe:

```
@Pipe({
  name: 'titlecase',
  pure: false  // Make the pipe impure
```

```
})
export class TitlecasePipe implements PipeTransform {
  // Same transform logic as before
}
```

- Add a button and a function to update the object in `app.component.ts`:

```
export class AppComponent {
  obj = { name: 'john doe' };

  updateName() {
    this.obj.name = 'jane doe';
  }
}
```

- In `app.component.html`, add a button and use the pipe on the object:

```
<h2>Impure Pipe Demo</h2>

 <p><b>Titlecase Object Name:</b> {{ obj.name | titlecase }}</p>
 <button (click)="updateName()">Update Name</button>
```

3. Expected Output: The name should change in real-time without manual refresh.

4. Challenge Question (30 mins)

You have a list of items in an array. Write a custom pipe that filters the array based on a search term input by the user.

- The pipe should take two inputs: the array and the search term