

# Testing Pipes



## Contents

Testing the `TitleCasePipe`

Writing DOM tests to support a pipe test

You can test [pipes](#) without the Angular testing utilities.

For the sample app that the testing guides describe, see the [sample app](#).

For the tests features in the testing guides, see [tests](#).

## Testing the `TitleCasePipe`

A pipe class has one method, `transform`, that manipulates the input value into a transformed output value. The `transform` implementation rarely interacts with the DOM. Most pipes have no dependence on Angular other than the `@Pipe` metadata and an interface.

Consider a `TitleCasePipe` that capitalizes the first letter of each word. Here's an implementation with a regular expression.

app/shared/title-case.pipe.ts

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

@Pipe({name: 'titlecase', pure: true})
/** Transform to Title Case: uppercase the first letter of the words in a string. */
export class TitleCasePipe implements PipeTransform {
  transform(input: string): string {
    return input.length === 0 ? '' :
      input.replace(/\w\S*/g, (txt => txt[0].toUpperCase() +
txt.substr(1).toLowerCase() ));
  }
}
```



Anything that uses a regular expression is worth testing thoroughly. Use simple Jasmine to explore the expected cases and the edge cases.

app/shared/title-case.pipe.spec.ts



```
describe('TitleCasePipe', () => {
  // This pipe is a pure, stateless function so no need for BeforeEach

  const pipe = new TitleCasePipe();

  it('transforms "abc" to "Abc"', () => {
    expect(pipe.transform('abc')).toBe('Abc');
  });

  it('transforms "abc def" to "Abc Def"', () => {
    expect(pipe.transform('abc def')).toBe('Abc Def');
  });

  // ... more tests ...
});
```

## Writing DOM tests to support a pipe test

These are tests of the pipe *in isolation*. They can't tell if the `TitleCasePipe` is working properly as applied in the application components.

Consider adding component tests such as this one:

app/hero/hero-detail.component.spec.ts (pipe test)

```
it('should convert hero name to Title Case', () => {
  // get the name's input and display elements from the DOM
  const hostElement = fixture.nativeElement;
  const nameInput: HTMLInputElement = hostElement.querySelector('input');
  const nameDisplay: HTMLSpanElement = hostElement.querySelector('span');

  // simulate user entering a new name into the input box
  nameInput.value = 'quick BROWN  f0x';

  // dispatch a DOM event so that Angular learns of input value change.
  // use newEvent utility function (not provided by Angular) for better browser
  // compatibility
  nameInput.dispatchEvent(newEvent('input'));

  // Tell Angular to update the display binding through the title pipe
  fixture.detectChanges();

  expect(nameDisplay.textContent).toBe('Quick Brown  Fox');
});
```

## RESOURCES

[About](#)

[Resource Listing](#)

[Press Kit](#)

[Blog](#)

[Usage Analytics](#)

## HELP

[Stack Overflow](#)

[Gitter](#)

[Report Issues](#)

[Code of Conduct](#)

## COMMUNITY

[Events](#)

[Meetups](#)

[Twitter](#)

[GitHub](#)

[Contribute](#)

## LANGUAGES

[简体中文版](#)

[正體中文版](#)

[日本語版](#)

[한국어](#)

Super-powered by Google ©2010-2020. Code licensed under an MIT-style License. Documentation licensed under CC BY 4.0.

Version 10.0.10-local+sha.84d1ba792b.