

Typescript Decorators

Angular 2.0 syntax inside
your 1.5 application

Background

- Angular 1.x directives were designed for Javascript
- Typescript implementations were good, but didn't reduce complexity
- Can Angular 2 syntax be applied to an Angular 1.5 application?
- Ng-forward (tried to do too much, never gained support)

What are Typescript Decorators?

- Similar to C# Attributes and Java Annotations
- Allows you to observe, modify, or replace Typescript entities

@Sealed

```
class Dog extends Animal {
```

```
    @ReadOnly
```

```
    sound = 'Woof';
```

```
    @Override
```

```
    speak() { ... }
```

```
}
```

Example

Before:

```
$inject: string[] = ['$log', '$window', '$state'];  
constructor(  
    $log: ng.ILogService,  
    $window: ng.IWindowService,  
    $state: ng.ui.IStateService) {  
  
}
```

After:

```
constructor(  
    @Inject('$log') $log: ng.ILogService,  
    @Inject('$window') $window: ng.IWindowService,  
    @Inject('$state') $state: ng.ui.IStateService) {  
  
}
```

Implementation

- Create a factory function with the desired inputs
- Return a decorator that manipulates the target

```
function Inject(provider: string): ParameterDecorator {  
    return (target: any, name: string, index: number) => {  
        target.$inject = target.$inject || [];  
        target.$inject.unshift(provider);  
    }  
}
```

Conversion Demo

Benefits

- Code is cleaner and more declarative
- Syntax is more inline with Angular 2
- Pattern is consistent across components, directives, and services
- Class-based architecture

Component Inheritance

- Leverage Typescript classes to easily extend components
- Dependencies, bindings, properties, and methods are inherited
- Ability to override/extend methods
- Access to base class implementations through **super**

Inheritance Demo

Thank You

<https://www.typescriptlang.org/docs/handbook/decorators.html>

<https://github.com/MikeRyan52/angular-decorators>

<https://github.com/ngUpgraders/ng-forward>

<https://github.com/brechtbilliet/angular-typescript-webpack>