Skip to main content

UpgradeComponent DIRECTIVE

A helper class that allows an AngularJS component to be used from Angular.

See more...

Description

Part of the upgrade/static library for hybrid upgrade apps that support AOT compilation.

This helper class should be used as a base class for creating Angular directives that wrap AngularJS components that need to be "upgraded".

Examples

Let's assume that you have an AngularJS component called `ng1Hero` that needs to be made available in Angular templates.

```
JS component will be "upgraded" to be

ng1AppModule.component('ng1Hero', {
  bindings: {hero: '<', onRemove: '&'},
  transclude: true,
  template: `<div class="title" ng-transclude></div>
             <h2>{{ $ctrl.hero.name }}</h2>
             <p>{{ $ctrl.hero.description }}</p>
             <button ng-
click="$ctrl.onRemove()">Remove</button>`,
});
```

We must create a `Directive` that will make this AngularJS component available inside Angular templates.

```
// This Angular directive will act as an interface to
the "upgraded" AngularJS component
@Directive({selector: 'ng1-hero'})
export class Ng1HeroComponentWrapper extends
UpgradeComponent {
  // The names of the input and output properties here
must match the names of the
  // `<` and `&` bindings in the AngularJS component
that is being wrapped
  @Input() hero!: Hero;
  @Output() onRemove!: EventEmitter<void>;

  constructor(elementRef: ElementRef, injector:
Injector) {
    // We must pass the name of the directive as used
by AngularJS to the super
    super('ng1Hero', elementRef, injector);
  }
}
```

In this example you can see that we must derive from the `UpgradeComponent` base class but also provide an `@Directive`

decorator. This is because the AOT compiler requires that this information is statically available at compile time.

Note that we must do the following:

- specify the directive's selector (`ng1-hero`)
- specify all inputs and outputs that the AngularJS component expects
- derive from `UpgradeComponent`
- call the base class from the constructor, passing
  - the AngularJS name of the component (`ng1Hero`)
  - the `ElementRef` and `Injector` for the component wrapper