

# Using ngxtension to migrate towards "Modern" Angular

The diagram illustrates the migration process from a legacy component to a modern one using the `ng g ngxtension:*` command.

**before:awesome.component.ts**

```
before:awesome.component.ts
  - X
  TS
  @Component({
    template: `
      @if (showTitle) {
        <h1>{{title}}</h1>
      }
      <button (click)="onClick()">Click me!</button>
    `
  })
  export class AwesomeComponent {
    @Input({required: true}) title: string = '';
    @Input({required: true, transform: booleanAttribute })
    showTitle: boolean = false;
    @Output() buttonClicked = new EventEmitter<string>();

    constructor(private service: AwesomeService){}

    onClick() {
      this.buttonClicked.emit(this.title);
      this.service.magic();
    }
  }
```

**ng g ngxtension:\***

- `ng g ngxtension:convert-di-to-inject`
- `ng g ngxtension:convert-signal-inputs`
- `ng g ngxtension:convert-outputs`

**after:awesome.component.ts**

```
after:awesome.component.ts
  - X
  TS
  @Component({
    template: `
      @if (showTitle()) {
        <h1>{{title()}}</h1>
      }
      <button (click)="onClick()">Click me!</button>
    `
  })
  export class AwesomeComponent {
    private service = inject(AwesomeService);
    title = input.required<string>();
    showTitle =
      input.required<boolean, boolean>({transform: booleanAttribute})
    buttonClicked = output<string>();

    onClick() {
      this.buttonClicked.emit(this.title());
      this.service.magic();
    }
  }
```

A large white arrow points from the legacy code on the left to the modern code on the right, indicating the transformation process.



timdeschryver.dev/bits

# Using ngxtension to migrate towards "Modern" Angular

## Constructor Dependencies => Inject method

before:awesome.component.ts

```
@Component()  
export class AwesomeComponent {  
  constructor(private service: AwesomeService) {}  
}
```

ngxtension:convert-di-to-inject.sh

```
ng g ngxtension:convert-di-to-inject
```



after:awesome.component.ts

```
@Component()  
export class AwesomeComponent {  
  private service = inject(AwesomeService)  
}
```



timdeschryver.dev/bits

# Using ngxtension to migrate towards "Modern" Angular

## @Input Decorator => Signal Input

before:awesome.component.ts

```
@Component()  
export class AwesomeComponent {  
  @Input({required: true}) title: string = '';  
  protected log() {  
    console.log(this.title);  
  }  
}
```

ngxtension:convert-signal-inputs.sh

```
ng g ngxtension:convert-signal-inputs
```

after:awesome.component.ts

```
@Component()  
export class AwesomeComponent {  
  title = input.required<string>();  
  protected log() {  
    console.log(this.title());  
  }  
}
```



timdeschryver.dev/bits

# Using ngxtension to migrate towards "Modern" Angular

## @Output Decorator => Output Function

before:awesome.component.ts

```
@Component()  
export class AwesomeComponent {  
  @Output() logOutput = new EventEmitter<string>();  
  protected log() {  
    this.logOutput.emit('👋 Hello');  
  }  
}
```

ngxtension:convert-outputs.sh

```
ng g ngxtension:convert-outputs
```

after:awesome.component.ts

```
@Component()  
export class AwesomeComponent {  
  logOutput = output<string>();  
  protected log() {  
    this.logOutput.emit('👋 Hello');  
  }  
}
```



timdeschryver.dev/bits