

Angular - Create New Component



Our Goal

- Create a new Angular component to display a table of data

The diagram illustrates a custom Angular component. A red callout bubble labeled "Custom component" points to a white card-like interface. On the card, the title "Sales Team" is displayed in large, bold, black font. Below the title is a table with a red dotted border. The table has four columns: "First Name", "Last Name", "Email", and "Sales Volume". The data in the table is as follows:

First Name	Last Name	Email	Sales Volume
Anup	Kumar	anup.kumar@luv2code.com	50000
John	Doe	john.doe@luv2code.com	40000
Claire	Murphy	claire.murphy@luv2code.com	90000
Mai	Truong	mai.truong@luv2code.com	60000

Development Process

Step-By-Step

1. Create a new project
2. Update main template page
3. Generate a new component
4. Add new component selector to app template page
5. Generate a SalesPerson class
6. In SalesPersonListComponent, create sample data
7. In sales-person-list template file, build HTML table by looping over data

Step 1: Create a new project

```
> ng new sales-project
```

```
> cd sales-project
```

Remember, this generates all of the
Angular starter files
for our project

Step 2: Update main template page

File: src/app/app.component.html

```
<h1>Sales Team</h1>
```

Remove all of the
Angular "placeholder" content

Just add basic HTML header

Step 3: Generate a new component

```
> ng generate component sales-person-list
```

```
CREATE src/app/sales-person-list/sales-person-list.component.css (0 bytes)
CREATE src/app/sales-person-list/sales-person-list.component.html (32 bytes)
CREATE src/app/sales-person-list/sales-person-list.component.spec.ts (693 bytes)
CREATE src/app/sales-person-list/sales-person-list.component.ts (311 bytes)
UPDATE src/app/app.module.ts (436 bytes)
```

About the Generated Files

`sales-person-list.component.ts`: the component class

`sales-person-list.component.html`: the component template HTML

`sales-person-list.component.css`: the component private CSS

`sales-person-list.component.spec.ts`: the unit test specifications

`UPDATE src/app/app.module.ts`: Adds the component to the main application module

Main Application Module

...

UPDATE src/app/app.module.ts: Adds the component to the main application module

...

```
import { AppComponent } from './app.component';
import { SalesPersonListComponent } from './sales-person-list/sales-person-list.component';

@NgModule({
  declarations: [
    AppComponent,
    SalesPersonListComponent
  ],
  ...
})
export class AppModule { }
```

Our new component
was automatically added by the
`ng generate component ...`
command

Step 4: Add new component selector to app template page

File: src/app/app.component.html

```
<h1>Sales Team</h1>  
  
<app-sales-person-list></app-sales-person-list>
```

1

File: src/app/sales-person-list/sales-person-list.component.ts

```
import { Component, OnInit } from '@angular/core';  
  
@Component({  
  selector: 'app-sales-person-list',  
  templateUrl: './sales-person-list.component.html',  
  styleUrls: ['./sales-person-list.component.css']  
})  
export class SalesPersonListComponent implements OnInit {  
  
  constructor() {}  
  
  ngOnInit() {}  
}
```

2

File: src/app/sales-person-list/sales-person-list.component.html

```
<p>sales-person-list works!</p>
```

Later on,
we'll add HTML table here

Later on,
we'll add sample data here

Step 5: Generate a SalesPerson class

```
> ng generate class sales-person-list/SalesPerson
```

File: src/app/sales-person-list/sales-person.ts

```
export class SalesPerson {  
  
    constructor(public firstName: string,  
               public lastName: string,  
               public email: string,  
               public salesVolume: number) {  
  
    }  
}
```

Creates a basic TypeScript class

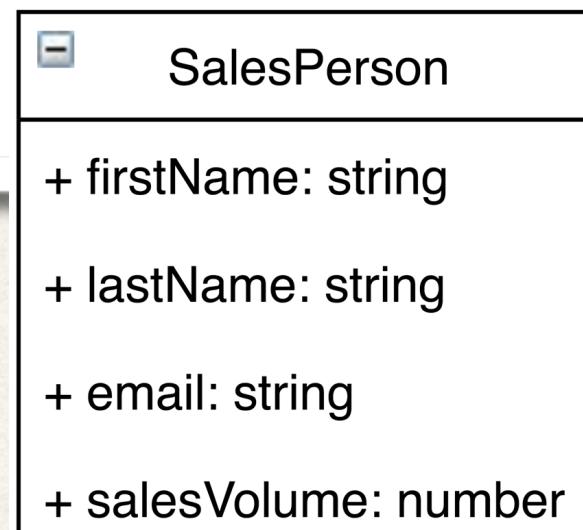
Step 5: Generate a SalesPerson class

```
> ng generate class SalesPerson
```

Remember, these are
Parameter Properties

File: src/app/sales-person-list/sales-person.ts

```
export class SalesPerson {  
  
  constructor(public firstName: string,  
              public lastName: string,  
              public email: string,  
              public salesVolume: number) {  
  
  }  
}
```



Declared by prefixing
constructor argument with access modifier:
public, **protected**, **private**, or **readonly**

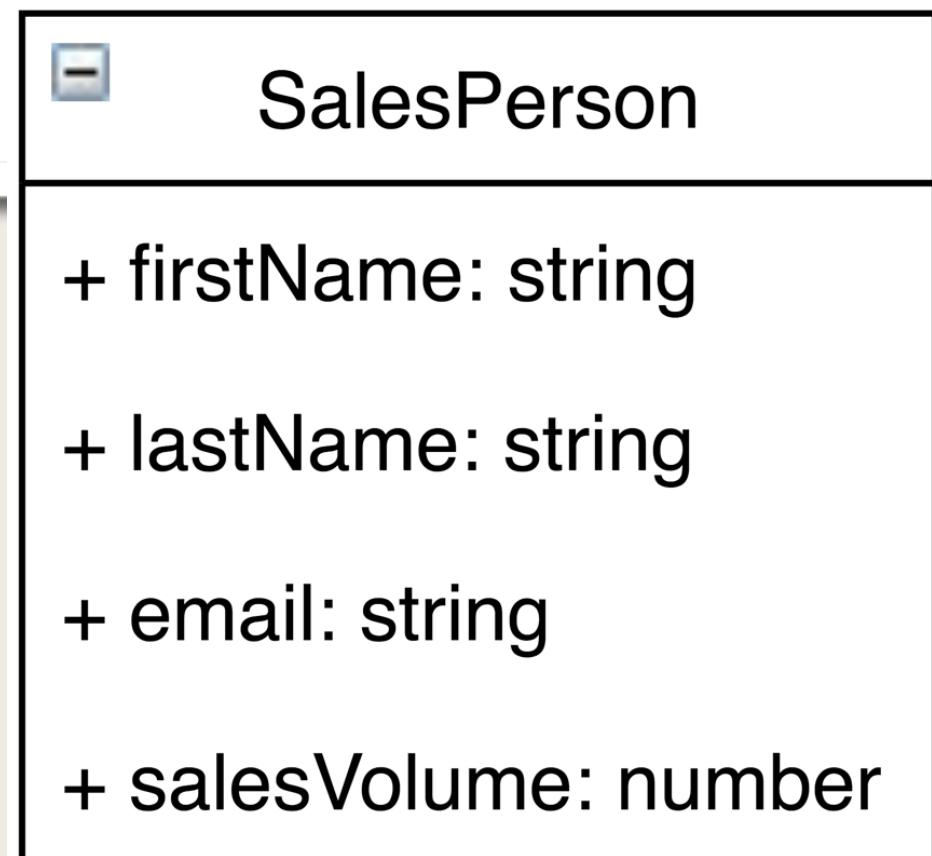
Declares properties and
assigns properties automagically.

Minimizes boilerplate coding!

About "public" properties

File: src/app/sales-person-list/sales-person.ts

```
export class SalesPerson {  
  
  constructor(public firstName: string,  
             public lastName: string,  
             public email: string,  
             public salesVolume: number) {  
  
  }  
}
```



In Angular world,
developers commonly
use "public" properties

Step 6: In SalesPersonListComponent, create sample data

File: src/app/sales-person-list/sales-person-list.component.ts

```
import { Component, OnInit } from '@angular/core';
import { SalesPerson } from './sales-person';

@Component({
  selector: 'app-sales-person-list',
  templateUrl: './sales-person-list.component.html',
  styleUrls: ['./sales-person-list.component.css']
})
export class SalesPersonListComponent implements OnInit {

  // create an array of objects
  salesPersonList: SalesPerson[] = [
    new SalesPerson("Anup", "Kumar", "anup.kumar@luv2code.com", 50000),
    new SalesPerson("John", "Doe", "john.doe@luv2code.com", 40000),
    new SalesPerson("Claire", "Murphy", "claire.murphy@luv2code.com", 90000),
    new SalesPerson("Mai", "Truong", "mai.truong@luv2code.com", 60000)
  ]
  ...
}
```

Import our new class

Create sample data

SalesPerson
+ firstName: string
+ lastName: string
+ email: string
+ salesVolume: number

Step 7: In sales-person-list template file, build HTML table by looping over data

File: src/app/sales-person-list/sales-person-list.component.html

```
<table border="1">
  <thead>
    <tr>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
      <th>Sales Volume</th>
    </tr>
  </thead>

  <tbody>
    <tr *ngFor="let tempSalesPerson of salesPersonList">
      <td>{{ tempSalesPerson.firstName }}</td>
      <td>{{ tempSalesPerson.lastName }}</td>
      <td>{{ tempSalesPerson.email }}</td>
      <td>{{ tempSalesPerson.salesVolume }}</td>
    </tr>
  </tbody>
</table>
```

```
import { Component, OnInit } from '@angular/core';
import { SalesPerson } from './sales-person';

@Component({
  selector: 'app-sales-person-list',
  templateUrl: './sales-person-list.component.html',
  styleUrls: ['./sales-person-list.component.css']
})
export class SalesPersonListComponent implements OnInit {

  // create an array of objects
  salesPersonList: SalesPerson[] = [
    new SalesPerson("Anup", "Kumar", "anup.kumar@luv2code.com", 50000),
    new SalesPerson("John", "Doe", "john.doe@luv2code.com", 40000),
    new SalesPerson("Claire", "Murphy", "claire.murphy@luv2code.com", 90000),
    new SalesPerson("Mai", "Truong", "mai.truong@luv2code.com", 60000)
  ]
  ...
}
```

First Name	Last Name	Email	Sales Volume
Anup	Kumar	anup.kumar@luv2code.com	50000
John	Doe	john.doe@luv2code.com	40000
Claire	Murphy	claire.murphy@luv2code.com	90000
Mai	Truong	mai.truong@luv2code.com	60000

ngFor

- **ngFor** is a structural directive
- It renders a template for each item in a collection
- For complete documentation and examples, see:

<http://angular.io/api/common/NgForOf>