luv2code

Angular - Create New Component

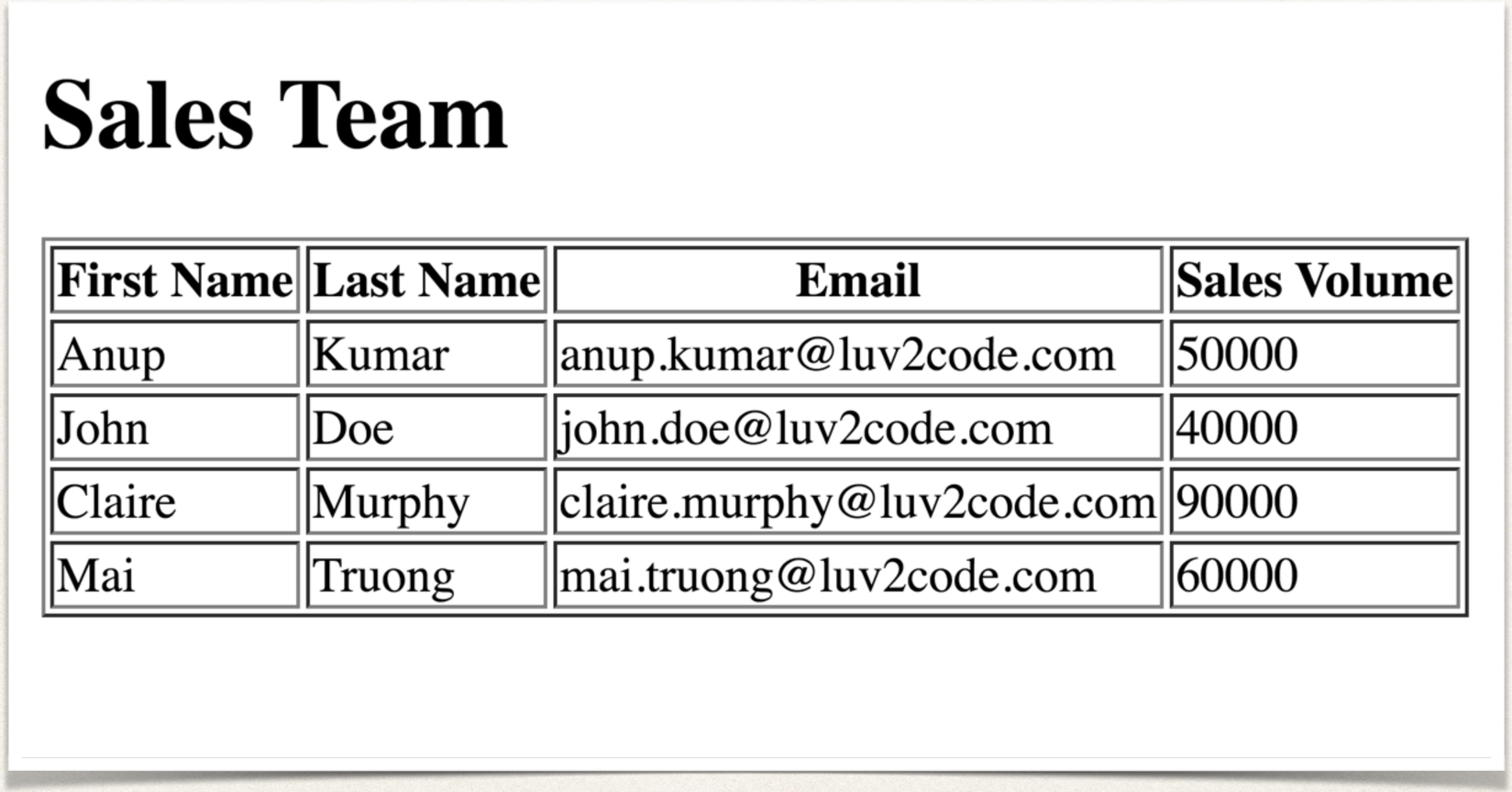
Our Goal

Our Goal

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

Our Goal

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

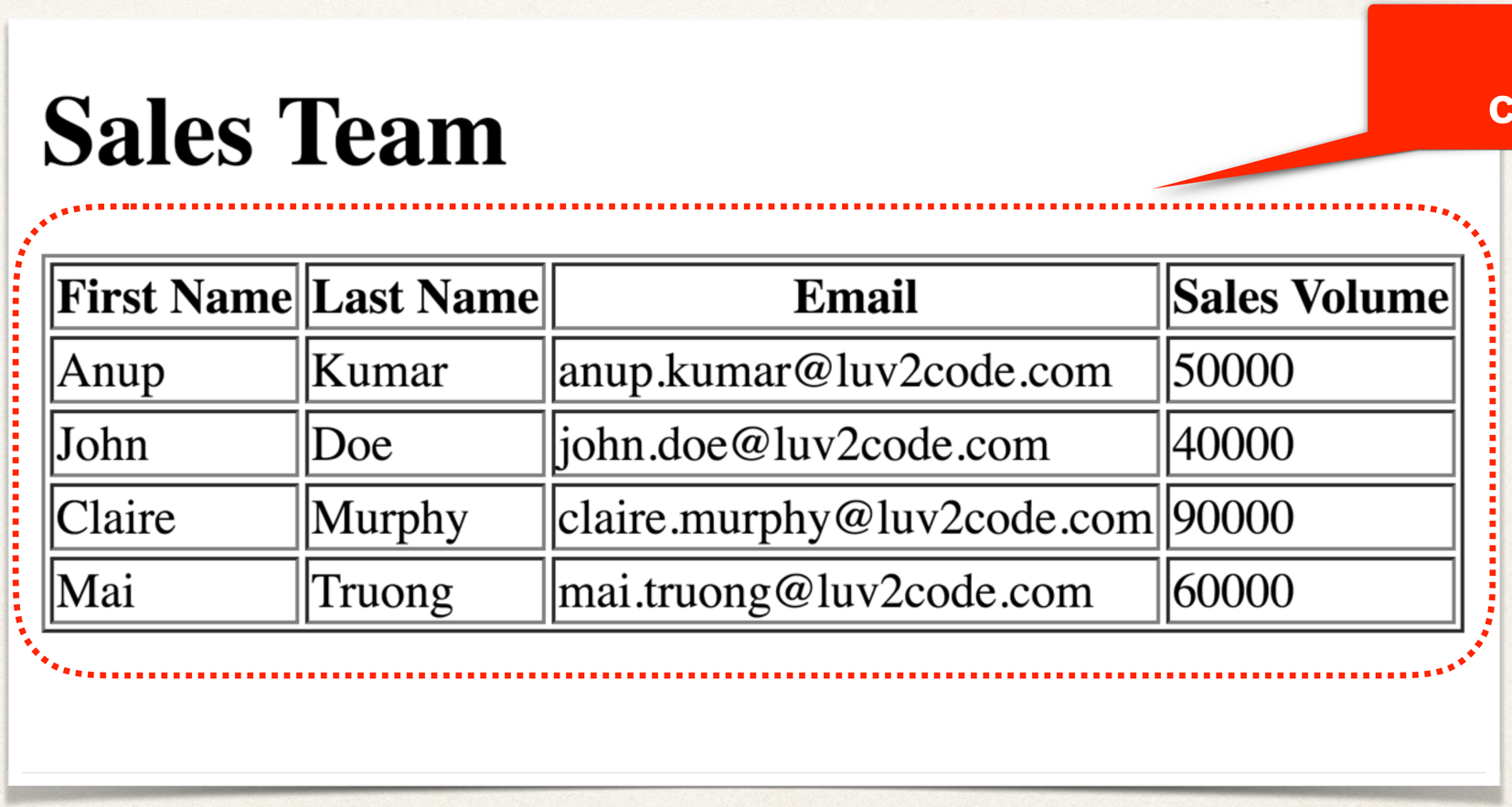


Sales Team

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

Our Goal

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



Development Process

Development Process

1. Create a new project

Development Process

1. Create a new project
2. Update main template page

Development Process

1. Create a new project
2. Update main template page
3. Generate a new component

Development Process

1. Create a new project
2. Update main template page
3. Generate a new component
4. Add new component selector to app template page

Development Process

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

Development Process

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

Development Process

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

Step 1: Create a new project

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

Step 1: Create a new project

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

Step 1: Create a new project

```
> ng new sales-project

> cd sales-project
```

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

Step 2: Update main template page

File: src/app/app.component.html

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

Step 2: Update main template page

File: src/app/app.component.html

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

Remove all of the Angular "placeholder" content

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

Step 3: Generate a new component

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

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

About the Generated Files

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

About the Generated Files

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

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

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

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

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

Main Application Module

```
...

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 { }
```

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

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>
```

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() {
  }

}
```

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

2

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() {
  }

}
```

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

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

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

2

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

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

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() {
  }

}
```

Later on,
we'll add sample data here

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

2

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() {
  }

}
```

Later on,
we'll add sample data here

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

Step 5: Generate a SalesPerson class

Step 5: Generate a SalesPerson class

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

Step 5: Generate a SalesPerson class

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

Creates a basic TypeScript class

Step 5: Generate a SalesPerson class

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

Creates a basic TypeScript class

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

```
export class SalesPerson {
}
```

Step 5: Generate a SalesPerson class

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

Creates a basic TypeScript class

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) {

    }
}
```

Step 5: Generate a SalesPerson class

```
> ng generate
```

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) {

    }
}
```

Step 5: Generate a SalesPerson class

```
> ng generate                                    lesPerson
```

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`

Step 5: Generate a SalesPerson class

```
> ng generate
```

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!

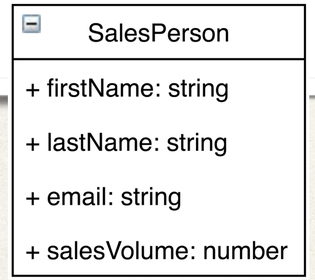
Step 5: Generate a SalesPerson class

```
> ng generate                                    lesPerson
```

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

About "public" properties

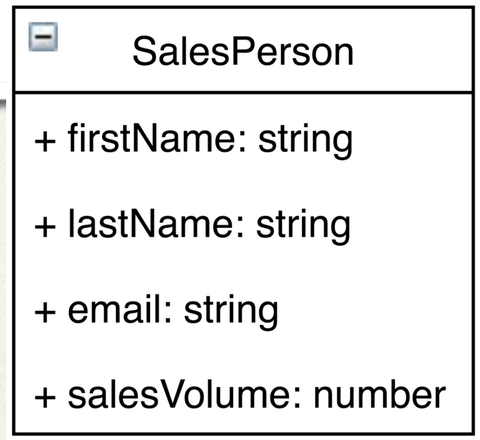
File: src/app/sales-person-list/sales-person.ts

```
export class SalesPerson {

    constructor(public firstName: string,
                public lastName: string,
                public email: string,
                public salesVolume: number) {

    }
}
```

About "public" properties

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

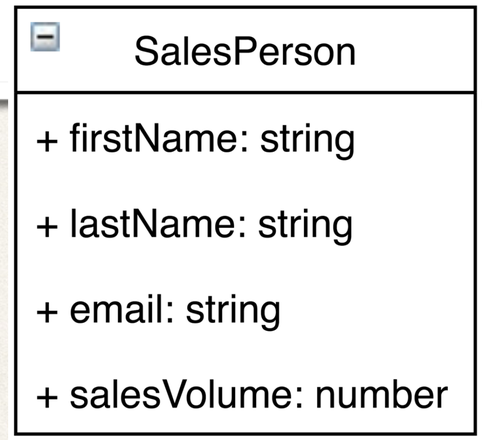
```
export class SalesPerson {

    constructor(public firstName: string,
                public lastName: string,
                public email: string,
                public salesVolume: number) {

    }
}
```



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

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 {









}
```

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 {









}
```

Import our new class

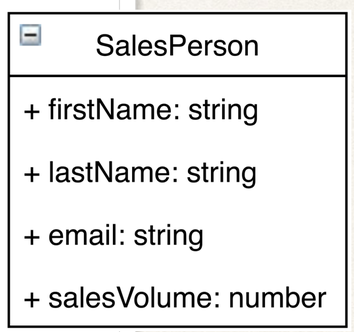
Step 6: In SalesPersonListComponent, create sample data

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



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

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

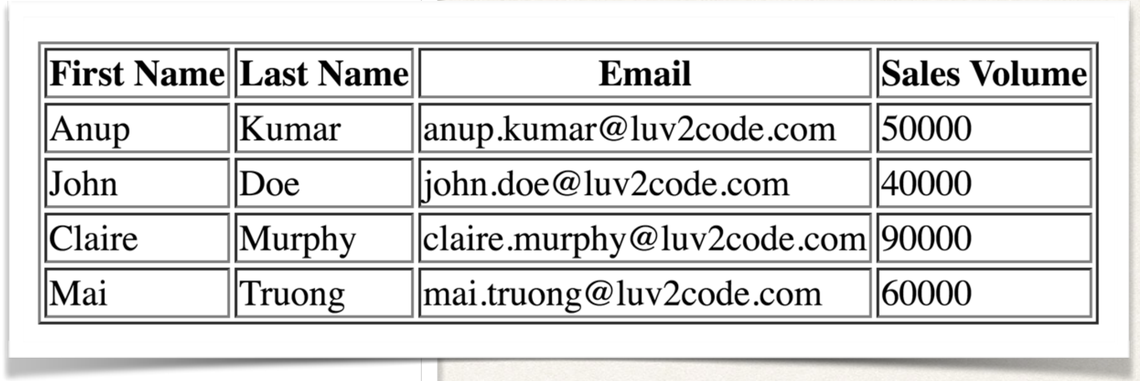
File: src/app/sales-person-list/sales-person-list.component.html

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









</table>
```

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>









</table>
```

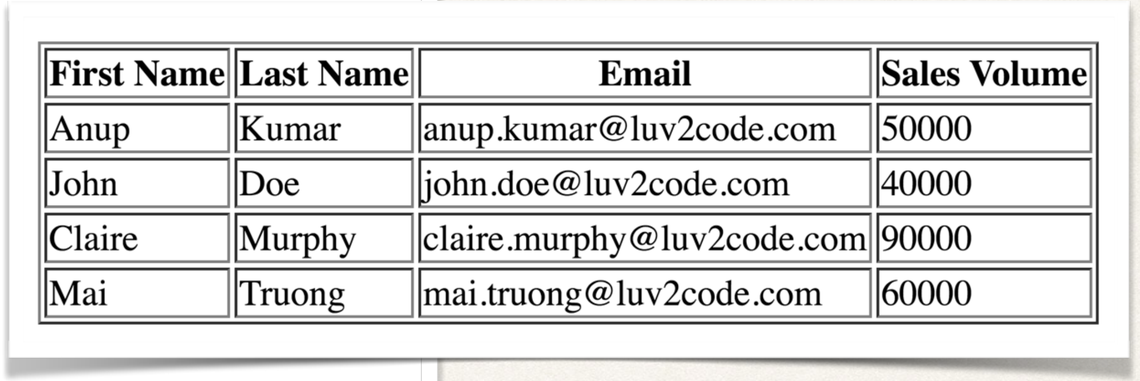


First Name	Last Name	Email	Sales Volume
Anup	Kumar	anup.kumar@luv2code.com	50000
John	Doe	john.doe@luv2code.com	40000
Claire	Murphy	claire.murphy@luv2code.com	90000
Mai	Truong	mai.truong@luv2code.com	60000

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>
```



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

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">

        </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>
```

***ngFor**
Will loop over the array
Create a table row for each array element

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">
        </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>
```

***ngFor**
Will loop over the array
Create a table row for each array element

Access property defined in the related component

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

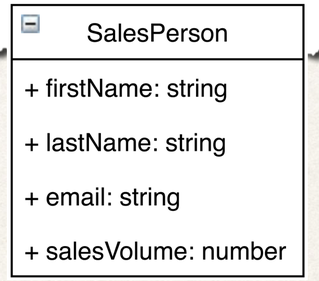
File: src/app/sales-person-list/sales-person-list.component.html

```
<table border="1">
        </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>
```

***ngFor**
Will loop over the array
Create a table row for each array element

Access property defined in the related component

```
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)
  ]
  ...
}
```



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

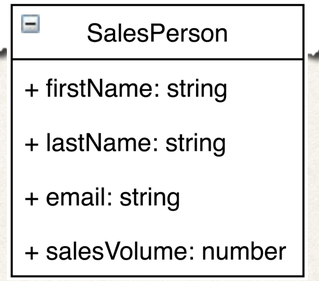
File: src/app/sales-person-list/sales-person-list.component.html

```
<table border="1">
        </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>
```

***ngFor**
Will loop over the array
Create a table row for each array element

Access property defined in the related component

```
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)
  ]
  ...
}
```



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">
        </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>
```

***ngFor**
Will loop over the array
Create a table row for each array element

Access property defined in the related component

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

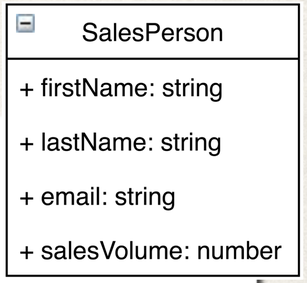
```
import { Component, OnInit } from '@angular/core';
import { SalesPerson } from './sales-person';

@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

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:

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