5.5 Two-way Binding

This section will guide you to:

* Establish communication between parent and child components

This guide has four subsections, namely:

5.5.1 Configuring the Angular application

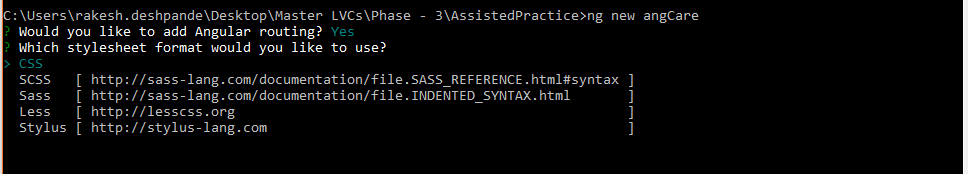
5.5.2 Creating parent and child component

5.5.3 Transferring data from parent to child component and vice versa

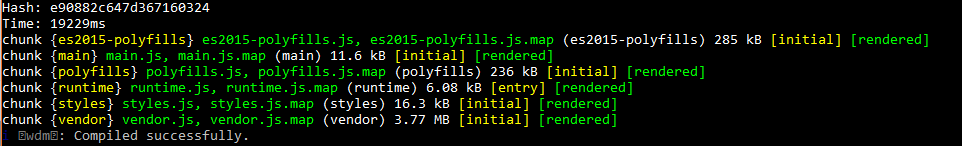
5.5.4 Pushing the code to Github repositories

**Step 5.5.1:** Configuring the Angular application

* Angular CLI is already installed in your practice lab. Refer to **DotNet Lab guide: Phase 2** for more information.
* ***ng new* <your\_app\_name>** (**angCare** is used in this example). Choose the appropriate options according to your requirement.



* ***cd* <your\_app\_name>**
* ***ng serve***



**Step 5.5.2:** Creating parent and child component

* Open Visual Studio Code
* Navigate to your project folder
* Run the below command to create a child component as your app component will be acting as a parent component

**ng g c child**

**Step 5.5.3:** Transferring data from parent to child component and vice versa

* Add below code in **app.module.ts**

import { NgModule } from '@angular/core';

import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { ChildComponent } from './child/child.component';

@NgModule({

  declarations: [

    AppComponent,

    ChildComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export *class* AppModule { }

* Add below code in **app.component.ts**

import { Component } from '@angular/core';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export *class* AppComponent {

public cdata: *string*="";

}

* Add below code in **child.component.ts**

import { Component, EventEmitter } from '@angular/core';

@Component({

  selector: 'app-child',

  templateUrl: './child.component.html',

  styleUrls: ['./child.component.css'],

  inputs: [`pdata`],

  outputs: [`cevent`]

})

export *class* ChildComponent {

  public pdata: *string*="";

  cevent= new EventEmitter<*string*>();

  onChange(*value*:*string*){

    this.cevent.emit(*value*);

  }

}

* Add below code in **app.component.html**

<div>

<h2>Parent Component</h2>

This is Parent Component<br>

Enter Text:

<input type="text" #ptext (keyup)="0"/><br>

The value of Child component is: {{cdata}}

<app-child (cevent)="cdata=$event" [pdata]="ptext.value"></app-child>

</div>

* Add below code in **child.component.html**

<div>

<h2>Child Component</h2>

This is Child Component<br>

Enter Text:

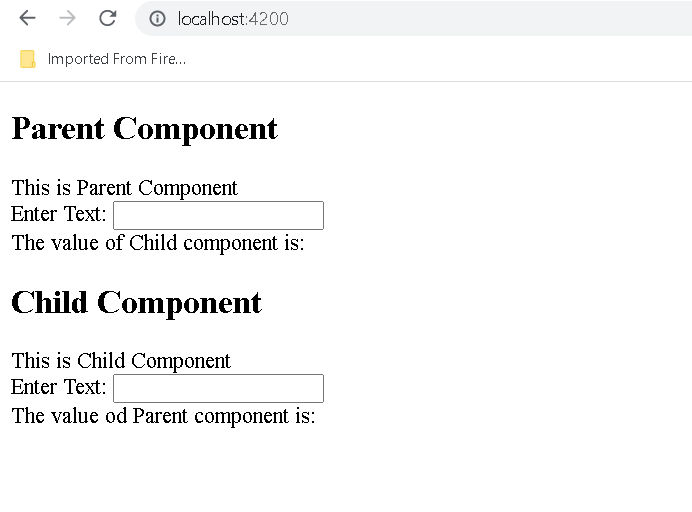
<input type="text" #cdata (keyup)="onChange(cdata.value)"/><br>

The value od Parent component is: {{pdata}}

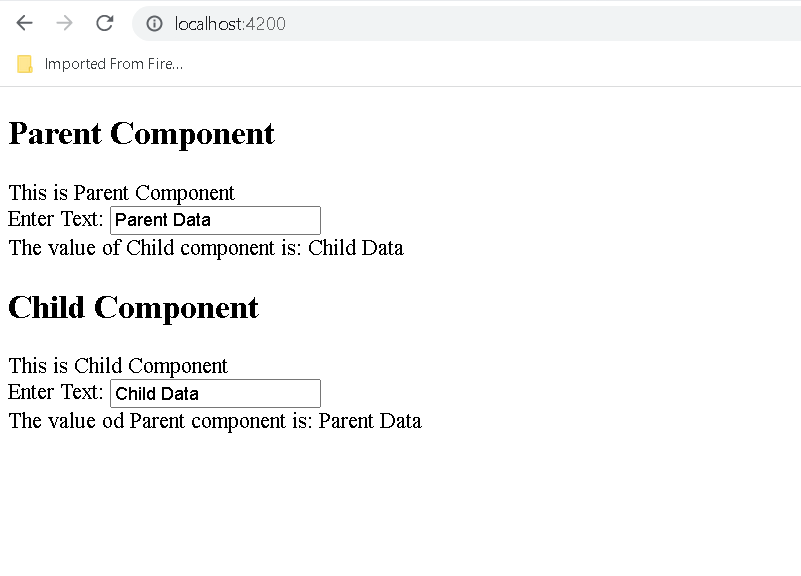
</div>

* Run the application using below command and open **localhost:4200** in your browser:

**ng serve**

****

**After type parent component text and child component text**

****

**Step 5.5.4:** Pushing the code to your GitHub repositories

Open your command prompt and navigate to the folder where you have created your files

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add .

Commit the changes using the following command:

git commit . -m “Changes have been committed.”

Push the files to the folder you initially created using the following command:

git push -u origin master