2.4 Component-Level Interaction

This section will guide you to:

* Establish communication between the parent and child component

This guide has four subsections, namely:

2.4.1 Configuring the Angular application

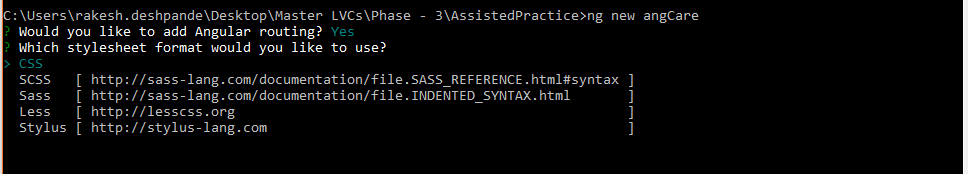
2.4.2 Creating parent and child component

2.4.3 Transferring data from parent to child component and vice versa

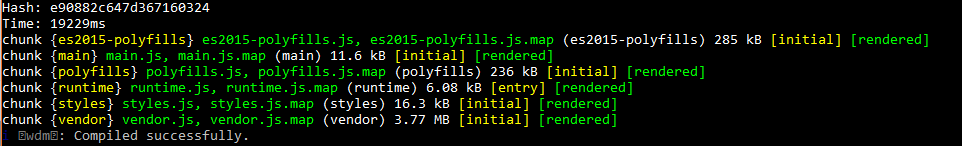
2.4.4 Pushing the code to GitHub repositories

**Step 2.4.1:** Configuring the Angular application

* Node.js and Angular are already installed in your lab. (Refer MEAN: Lab Guide - Phase 2)
* Open the Angular terminal
* Create a folder using **mkdir folder\_name** command. Go to the created folder using cd command
* Create a new project using the command  ***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 2.4.2:** Creating parent and child component

* Open Visual Studio Code
* Open the terminal and navigate to your project folder **->angCare** using cd command
* 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 2.4.3:** Transferring data from parent to child component and vice versa

* Go to /src/app/app.module.ts
* Add below code in **app.module.ts**

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

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

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 { }

* Go to /src/app/app.component.ts
* 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;

}

* Go to /src/app/app.component.html
* Add below code in **app.component.html**

<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>

* Go to /src/app/child/child.component.html
* Add below code in **child.component.html**

<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}}

* Go to /src/app/child/child.component.ts
* Add below code in **child.component.ts**

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

@Component({

selector: 'app-child',

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

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

inputs: [`pdata`],

outputs: [`cevent`]

})

export class ChildComponent implements OnInit {

constructor() { }

ngOnInit() {

}

public pdata: string;

cevent= new EventEmitter<string>();

onChange(value:string){

this.cevent.emit(value);

}

}

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

**ng serve**

**Step 2.4.4:** Pushing the code to 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**