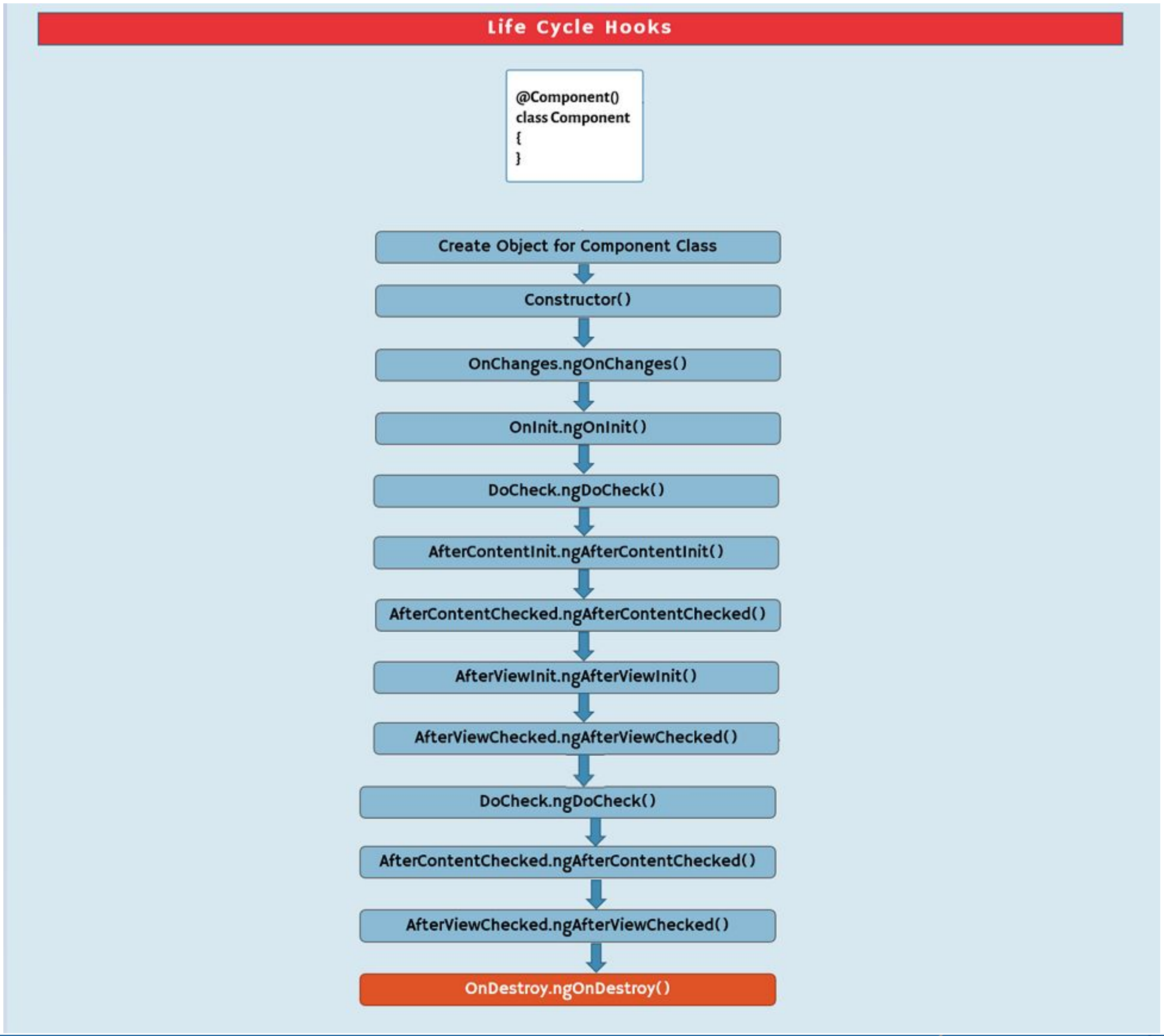
Angular

Anil K

Life Cycle Hooks

- Component has a life cycle, which is managed by Angular.
- Angular creates it, renders it, creates and renders its children, checks it when its properties changed, and destroys it before removing it from the DOM.
- Angular offers lifecycle hooks that provide visibility into these key life moments and the ability to act when they occur.
- The life cycle events will execute automatically at different stages, while executing the component.


Life Cycle Hooks
@Component()
class Component
{
}
Create Object for Component Class
Constructor()
OnChanges.ngOnChanges()
OnInit.ngOnInit()
DoCheck.ngDoCheck()
AfterContentInit.ngAfterContentInit()
AfterContentChecked.ngAfterContentChecked()
AfterViewInit.ngAfterViewInit()
AfterViewChecked.ngAfterViewChecked()
DoCheck.ngDoCheck()
AfterContentChecked.ngAfterContentChecked()
AfterViewChecked.ngAfterViewChecked()
OnDestroy.ngOnDestroy()

First run:

1. Component Object:

First, an object will be created for the component class. That means, the properties and methods of the component class, will be stored in the component object.

2. Constructor:

Next, the “constructor” of component class will be executed. Use the constructor, to set default values to any properties of the component, inject services into the component.

3. OnChanges.ngOnChanges:

Next, “ngOnChanges” method of “OnChanges” interface will be executed. This method executes when a new object is received with the new values of the input properties and just before a moment of assigning those new values into the respective input properties of the component. This method executes, only if the component has input properties.

4. OnInit.ngOnInit:

Next, “ngOnInit” method of “OnInit” interface will be executed. Use this method to call services to get data from database or any other data source.

5. DoCheck.ngDoCheck( ):

Next, “ngDoCheck” method of “DoCheck” interface will execute. This method executes when an event occurs, such as clicking, typing some key in the board etc. Use this method to identify whether the “change detection” process occurs or not.

6. AfterContentInit.ngAfterContentInit( ):

Next, “ngAfterContentInit” method of “AfterContentInit” interface will execute. This method executes after initializing the content of the component, which is passed while invoking the component. Use this method to set the properties of content children.

7. AfterContentChecked.ngAfterContentChecked( ):

Next, “ngAfterContentChecked” method of “AfterContentChecked” interface will execute. This method executes after “change detection” process of the content is completed. Use this method to check any properties of the content children, whether those are having specific values or not.

8. AfterViewInit.ngAfterViewInit( ):

Next, “ngAfterViewInit” method of “AfterViewInit” interface will execute. This method executes after initializing the view (template) of the component. Use this method to set the properties of view children.

9. AfterViewChecked.ngAfterViewChecked( ):

Next, “ngAfterViewChecked” method of “AfterViewChecked” interface will execute. This method executes after “change detection” process of view is completed. Use this method to check any properties of the view children, whether those are having specific values or not.

On an event occurs:

1. DoCheck.ngDoCheck( )

2. AfterContentChecked.ngAfterContentChecked( )

3. AfterViewChecked.ngAfterViewChecked( )

On deleting the component:

1. OnDestroy.ngOnDestroy( ): This method executes when the component is deleted from memory (when we close the web page in the browser).

Steps to handle event:

1. Import the interface:

```
import { interfacename } from “@angular/core”;
```

2. Implement the interface:

```
export  class  componentclassname  implements  interfacename { }
```

3. Create the method:

```
methodname() {     //code here }
```

Life Cycle Hooks - Example

- **Creating Application**
  - Open Command Prompt and enter the following commands:
  - cd c:\angular ng new my-app
  - cd c:\angular\my-app
  - ng g component company
- **c:\angular\my-app\src\app\app.component.html**

```
<div>
      <h4>App</h4>
      <app-company [companyname]=" 'ABC Pvt Ltd' ">
      <h5>one</h5>
      <h5>two</h5>
      <h5>three</h5>
      </app-company>
</div>
```

- **c:\angular\my-app\src\app\app.module.ts**

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from "@angular/forms";
import { AppComponent } from './app.component';
import { CompanyComponent } from './company/company.component';
@NgModule({
declarations: [    AppComponent,     CompanyComponent   ],
imports: [     BrowserModule, FormsModule   ],
providers: [],
bootstrap: [AppComponent]
})
export class AppModule { }
```

- **c:\angular\my-app\src\app\app.component.ts**

```
import { Component } from '@angular/core';
@Component({
selector: 'app-root',
templateUrl: './app.component.html',
styleUrls: ['./app.component.css']
})
export class AppComponent { }
```

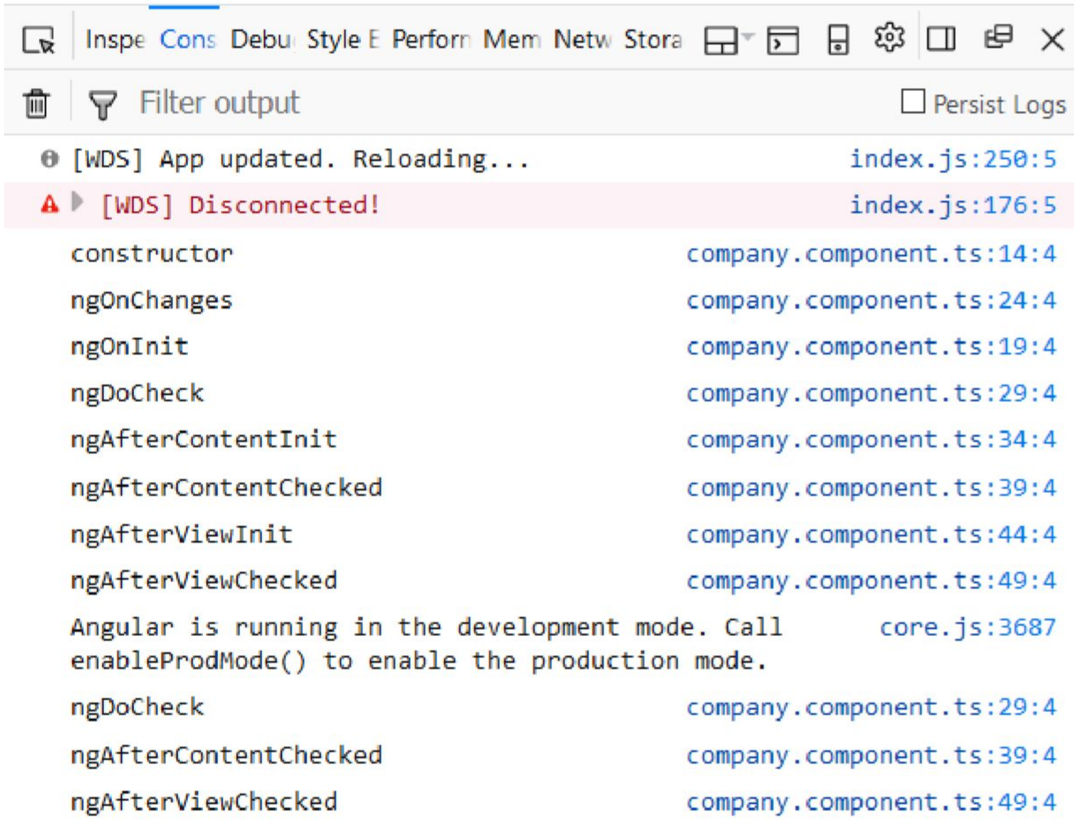
- **c:\angular\my-app\src\app\company\company.component.ts**

```
import { Component, Input, OnChanges, OnInit, DoCheck, AfterContentInit,
AfterContentChecked, AfterViewInit, AfterViewChecked } from '@angular/core';
@Component({
selector: 'app-company',
templateUrl: './company.component.html',
styleUrls: ['./company.component.css’]
})
export class CompanyComponent implements OnChanges, OnInit, DoCheck, AfterContentInit,
AfterContentChecked, AfterViewInit, AfterViewChecked {
@Input() companyname: string;
constructor()   {     console.log("constructor");   }
ngOnInit()   {     console.log("ngOnInit");   }
ngOnChanges()   {     console.log("ngOnChanges");   }
ngDoCheck()   {     console.log("ngDoCheck");   }
ngAfterContentInit()   {     console.log("ngAfterContentInit");   }
ngAfterContentChecked()   {     console.log("ngAfterContentChecked");   }
ngAfterViewInit()   {     console.log("ngAfterViewInit");   }
ngAfterViewChecked()   {     console.log("ngAfterViewChecked");   }
ngOnDestroy()   {     console.log("ngOnDestroy");   } }
```

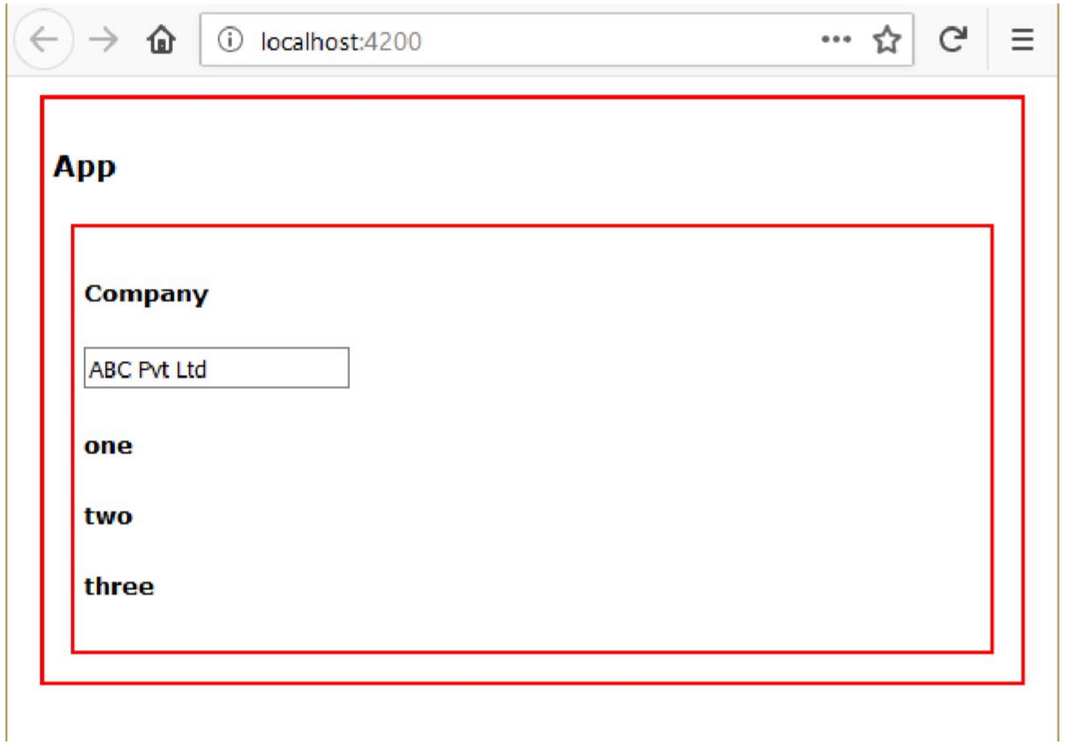
- **c:\angular\my-app\src\app\company\company.component.html**

```
<div>
<h5>Company</h5>
<input type="text" [(ngModel)]="companyname">
<ng-content></ng-content>
</div>
```

- **Executing the application:**
- Open Command Prompt and enter the following commands:
- cd c:\angular\my-app ng serve
- Open the browser and enter the following URL:
- http://localhost:4200



```
[WDS] App updated. Reloading...                         index.js:250:5
[WDS] Disconnected!                                     index.js:176:5
constructor                                  company.component.ts:14:4
ngOnChanges                                  company.component.ts:24:4
ngOnInit                                     company.component.ts:19:4
ngDoCheck                                    company.component.ts:29:4
ngAfterContentInit                           company.component.ts:34:4
ngAfterContentChecked                        company.component.ts:39:4
ngAfterViewInit                              company.component.ts:44:4
ngAfterViewChecked                           company.component.ts:49:4
Angular is running in the development mode. Call          core.js:3687
enableProdMode() to enable the production mode.
ngDoCheck                                    company.component.ts:29:4
ngAfterContentChecked                        company.component.ts:39:4
ngAfterViewChecked                           company.component.ts:49:4
```



App

Company

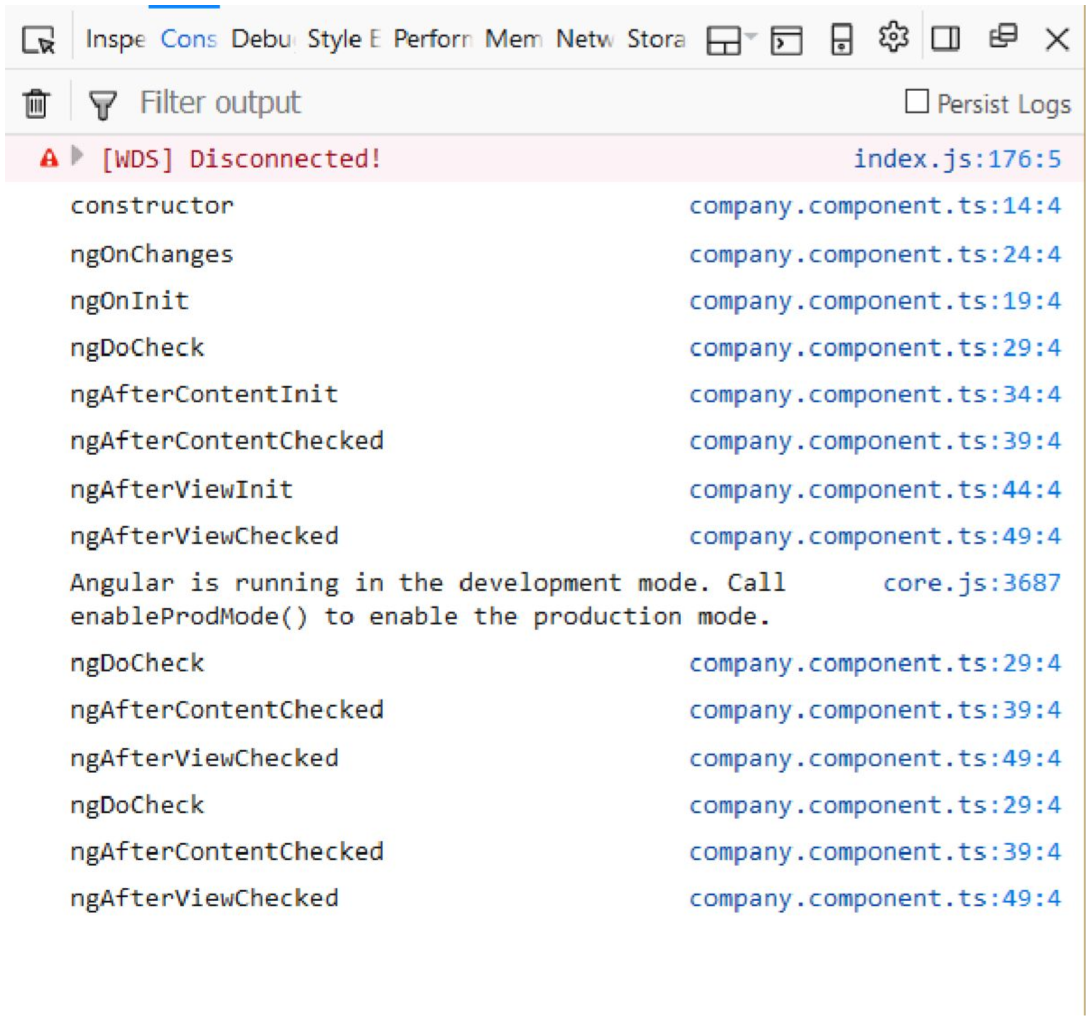
ABC Pvt Ltd

one

two

three

- After typing some character in the textbox:



```
[WDS] Disconnected!                                   index.js:176:5
constructor                               company.component.ts:14:4
ngOnChanges                               company.component.ts:24:4
ngOnInit                                  company.component.ts:19:4
ngDoCheck                                 company.component.ts:29:4
ngAfterContentInit                        company.component.ts:34:4
ngAfterContentChecked                     company.component.ts:39:4
ngAfterViewInit                           company.component.ts:44:4
ngAfterViewChecked                        company.component.ts:49:4
Angular is running in the development mode. Call      core.js:3687
enableProdMode() to enable the production mode.
ngDoCheck                                 company.component.ts:29:4
ngAfterContentChecked                     company.component.ts:39:4
ngAfterViewChecked                        company.component.ts:49:4
ngDoCheck                                 company.component.ts:29:4
ngAfterContentChecked                     company.component.ts:39:4
ngAfterViewChecked                        company.component.ts:49:4
```