

### **App.Component .css**

```
h3 {  
  color: darkblue;  
}
```

### **app.component.html**

```
<div class="container">  
  <div class="row">  
    <div class="col-xs-12">  
      <h3>I'm in the AppComponent!</h3>  
      <hr>  
      <app-servers></app-servers>  
      <!--<div app-servers></div>-->  
      <!--<div class="app-servers"></div>-->  
    </div>  
  </div>  
</div>
```

### **app.component.ts**

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

```
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  // styleUrls: ['./app.component.css']  
  styles: [`  
    h3 {  
      color: dodgerblue;  
    }  
  `]
```

```
`]  
}))  
export class AppComponent {  
}
```

### **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 { ServerComponent } from './server/server.component';  
import { ServersComponent } from './servers/servers.component';  
  
@NgModule({  
  declarations: [  
    AppComponent,  
    ServerComponent,  
    ServersComponent  
  ],  
  imports: [  
    BrowserModule,  
    FormsModule,  
  ],  
  providers: [],  
  bootstrap: [AppComponent]  
})  
export class AppModule { }
```

### **server.component.html**

```
<p
  [ngStyle]="{backgroundColor: getColor()}"
  [ngClass]="{online: serverStatus === 'online'}">
  {{ 'Server' }} with ID {{ serverId }} is {{ getServerStatus() }}
</p>
```

### **server.component.ts**

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

@Component({
  selector: 'app-server',
  templateUrl: './server.component.html',
  styles: [`
    .online {
      color: white;
    }
  `]
})
export class ServerComponent {
  serverId: number = 10;
  serverStatus: string = 'offline';

  constructor() {
    this.serverStatus = Math.random() > 0.5 ? 'online' : 'offline';
  }

  getServerStatus() {
    return this.serverStatus;
  }

  getColor() {
    return this.serverStatus === 'online' ? 'green' : 'red';
  }
}
```

## servers.component.html

```
<label>Server Name</label>
<!--<input-->
  <!--type="text"-->
  <!--class="form-control"-->
  <!--(input)="onUpdateServerName($event)"-->
<input
  type="text"
  class="form-control"
  [(ngModel)]="serverName">
<!--<p>{{ serverName }}</p-->
<button
  class="btn btn-primary"
  [disabled]="!allowNewServer"
  (click)="onCreateServer()">Add Server</button>
<!--<p [innerText]="allowNewServer"></p-->
<!--<p>{{ serverCreationStatus }}</p-->
<p *ngIf="serverCreated">Server was created, server name is {{ serverName }}</p>
<app-server *ngFor="let server of servers"></app-server>
```

## servers.component.ts

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

```
@Component({
  // selector: '[app-servers]',
  // selector: '.app-servers',
  selector: 'app-servers',
  // template: `
  //   <app-server></app-server>
  //   <app-server></app-server>`,
  templateUrl: './servers.component.html',
  styleUrls: ['./servers.component.css']
})
```

```

}))
export class ServersComponent implements OnInit {
  allowNewServer = false;
  serverCreationStatus = 'No server was created!';
  serverName = 'Testserver';
  serverCreated = false;
  servers = ['Testserver', 'Testserver 2'];

  constructor() {
    setTimeout(() => {
      this.allowNewServer = true;
    }, 2000);
  }

  ngOnInit() {
  }

  onCreateServer() {
    this.serverCreated = true;
    this.servers.push(this.serverName);
    this.serverCreationStatus = 'Server was created! Name is ' + this.serverName;
  }

  onUpdateServerName(event: Event) {
    this.serverName = (<HTMLInputElement>event.target).value;
  }
}

```

## **index.html**

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>MyFirstApp</title>

```

```
<base href="/">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<link rel="icon" type="image/x-icon" href="favicon.ico">
```

```
</head>
```

```
<body>
```

```
<app-root></app-root>
```

```
</body>
```

```
</html>
```