

# Building Blog component

Wednesday, 2 January 2019

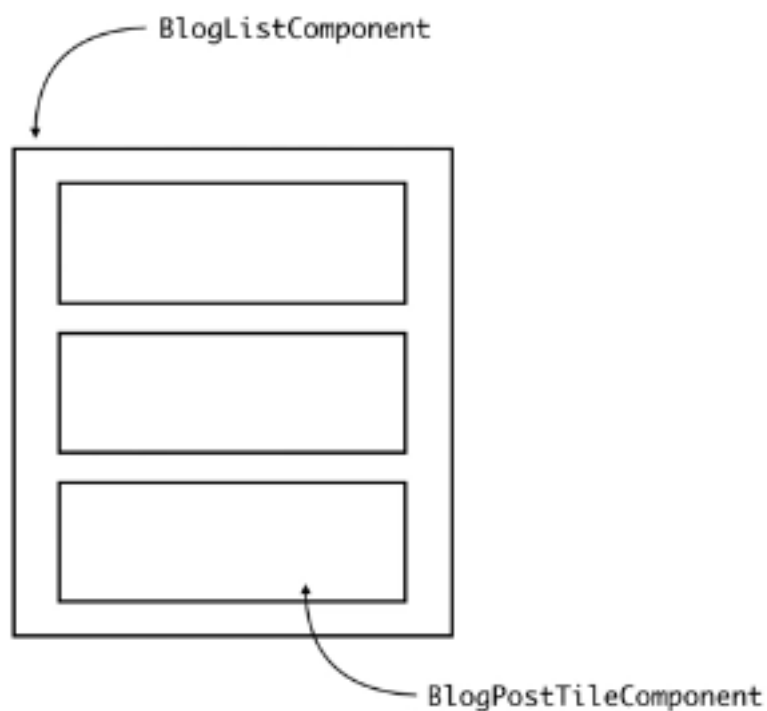
10:56 AM

Objective : creating and using nested components

ng new components-indepth --style=scss

ng generate component blog-post-tile

ng g c blog-list



---

Copy Bootstrap css link in index.html

Bootstrap provides classes that come with predefined styles that you can directly use in your application. Other style elements that come out of the include panels, lists and alerts.

## blog-post-tile.component.html

```
<div class="card">
<div class="card-body">
<h5 class="card-title">{{post.title}}</h5>
<p class="card-text">{{post.summary}}</p>
<a href="#" class="btn btn-primary">Read</a>
</div>
</div>
```

## blog-post-tile.component.ts

```
import { Component, OnInit, Input } from '@angular/core';
import { BlogPost } from '../blog-post';
@Component({
  selector: 'app-blog-post-tile',
  templateUrl: './blog-post-tile.component.html',
  styleUrls: ['./blog-post-tile.component.scss']
})
export class BlogPostTileComponent implements OnInit {
  @Input() post: BlogPost;
  constructor() {}
  ngOnInit() {}
}
```

## blog-list.component.html

```
<!--<div *ngFor="let blog of blogPost">
<app-blog-post-tile [post]="blog"></app-blog-post-tile>
</div>-->
<app-blog-post-tile *ngFor="let item of blogPost" [post]="item">
</app-blog-post-tile>
```

## blog-list.component.ts

```
import { Component, OnInit } from '@angular/core';
import { BlogPost } from '../blog-post';
@Component({
  selector: 'app-blog-list',
  templateUrl: './blog-list.component.html',
  styleUrls: ['./blog-list.component.scss']
})
export class BlogListComponent implements OnInit {
  blogPost: BlogPost[] = [];
  constructor() {}
  ngOnInit() {}
}
```

```
ngOnInit() {  
  this.blogPost.push(new BlogPost("Title - 1","Summary - 1"));  
  this.blogPost.push(new BlogPost("Title - 2","Summary - 2"));  
}  
}
```